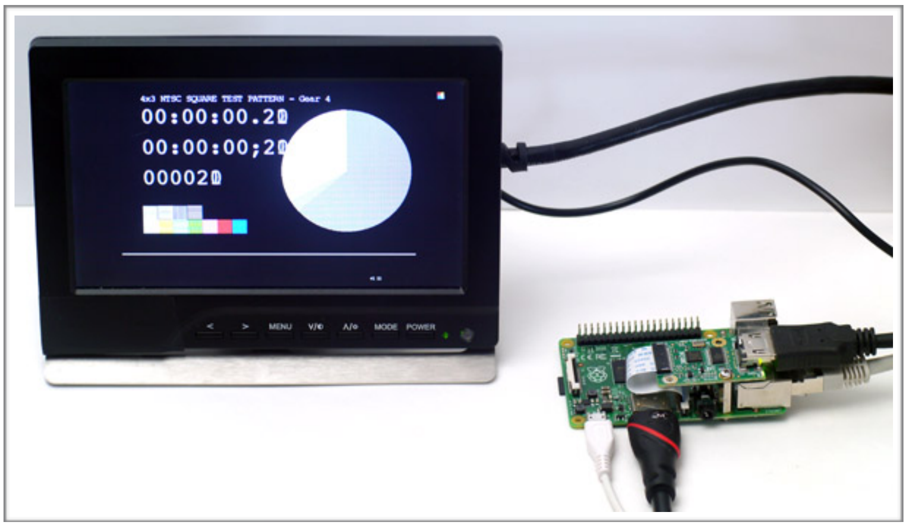
B10x HDMI to CSI-2 bridge

Version 1.4



Preliminary

February 2017

Auvidea GmbH
Kellerberg 3
D-86920 Denklingen

Tel: +49 8243 7714 622
info@auvidea.com
www.auvidea.com

Copyright Notice

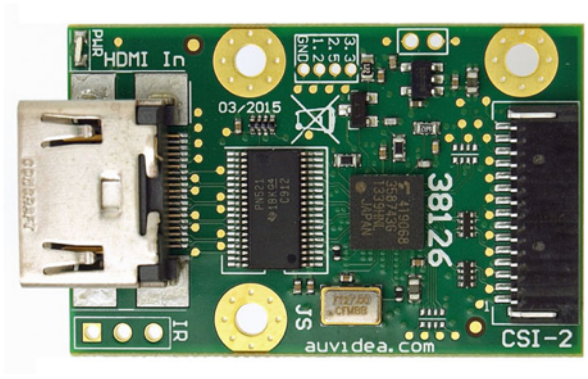
B101

15 pin CSI-2 connector

The 15 pin CSI-2 connector on the right connects to the 15 pin CSI-2 connector on the Raspberry PI or the Hummingboard.

Please use a 15 pin FFC cable with 1mm pitch and contacts on the same side. On the B101 the contacts have to face upwards, when the cable is plugged into the black CSI-2 connector.

On the Raspberry Pi the contacts of the FFC cable should face the opposite side of the locking latch.



Technical details

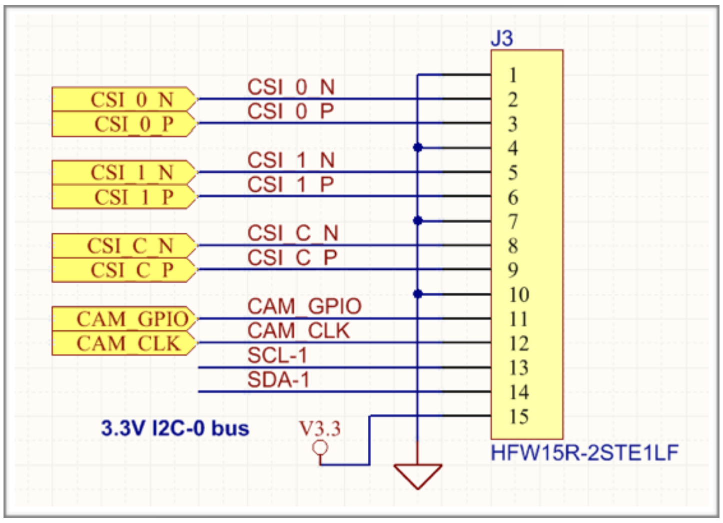
- HDMI input: 1080p25 only on the RPi (software limitation of the RPi at this time)
- HDMI to CSI-2 bridge chip: Toshiba TC358743XBG
- CSI-2: up to 2 lanes plus clock
- CSI-2 connector: 15 pin FPC with 1.0 mm pitch
- compatible to Raspberry Pi and HummingBoard CSI-2 connector - contacts on top
- size: 27 x 42 mm (PCB size)
- mounting: 3x M2.5 mounting holes (do not use M3 screws)
- mount hole spacing: 22mm (h) x 21mm (v)
- power: 3.3V

Compatible systems

- Raspberry Pi models A, B, B+, 3
- HummingBoard (please check availability of driver)
- J100 carrier board for the Nvidia Jetson TX1 (up to 2 modules may be connected) - a very high performance compute platform with H.264/H.265 encoding and graphics processing (CUDA and VisionWorks) for deep learning applications.

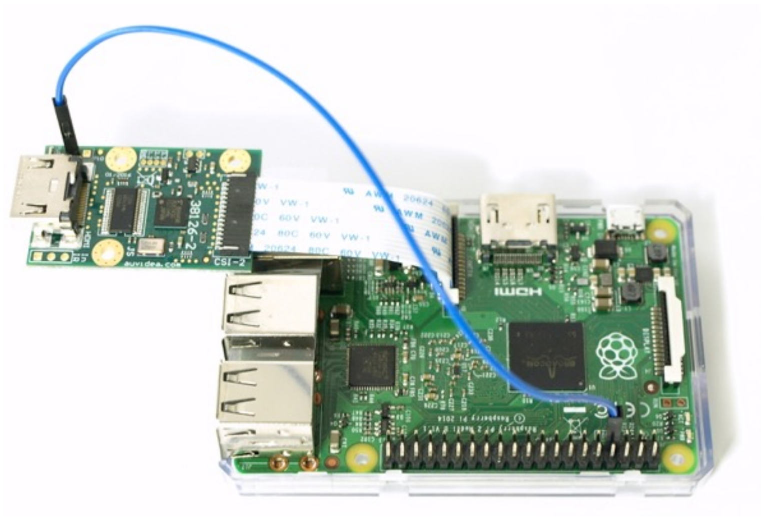
Linux drivers

- Kernel 4 comes with build-in support for the TC358743 chip on the B100, B101 and B102
- for kernel 3 systems this kernel 4 driver could be back ported
- the Raspberry has build-in support for the TC358743 at 1080p25 only (B101)
- drivers for the Jetson TX1 and the TC358743 are provided for a resolution from 640x480 to 1080p (B101 or B102)



B101 rev 2 and rev 3

Starting RASPIVID multiple times



After starting raspivid once the TC358743 chip on the B101 needs to be reset. The reset signal is not connected via the FPC cable. So a jumper cable needs to be connected from the B101 module to a GPIO header pin on the Raspberry Pi. We have chosen GPIO4. Please see the setup in the picture below. When the green LED on the bottom side below the HDMI connector is on, then RESET is active (low).

The GPIO 4 initialization can be executed at startup by putting it into the /etc/rc.local file. Please add the following lines to your rc.local file. You will need sudo rights to do so.

```
echo "4" > /sys/class/gpio/export
echo "out" > /sys/class/gpio/gpio4/direction
```

Now create a script to restart the B101 rev 2 (38126-2) module after each use of raspivid. A „0“ applies RESET and a „1“ removes RESET.

Use `nano restart.sh` to create the file and insert the following 2 lines:

```
echo "0" > /sys/class/gpio/gpio4/value
echo "1" > /sys/class/gpio/gpio4/value
```

After you stopped raspivid, just execute restart.sh and you can run raspivid again.

B101 rev 3.1

B101 rev 3.1 (38126-3)

They support a software RESET to the module. Both Rev 3 and Rev 3.1 have the same marking on the module (38126-3). The difference is that on Rev 3.1 one extra jumper resistor has been added on the bottom side of the module. This resistor (R30) with 0 Ohm connects the RESET input of the TC358743 to the CAM_GPIO line (pin 11) on the 15 pin FFC cable. The Raspberry models connect this CAM_GPIO line to different GPIOs internally.

Starting RASPIVID multiple times

The TC358743 must be reset before raspivid may be started again. For this the GPIO controlling RESET must be temporarily set to 0 (false). When false, the LED below the HDMI connector will be off. The LED is on, when RESET is off (true), and it indicates that the module is operational.

The script below must be run, before raspivid may be started. Without this script there will be no video!

Python script for the Raspberry Pi A/B which uses GPIO 21 for the CAM_GPIO pin.

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(21, GPIO.OUT)
GPIO.output(21, False)
GPIO.output(21, True)
```

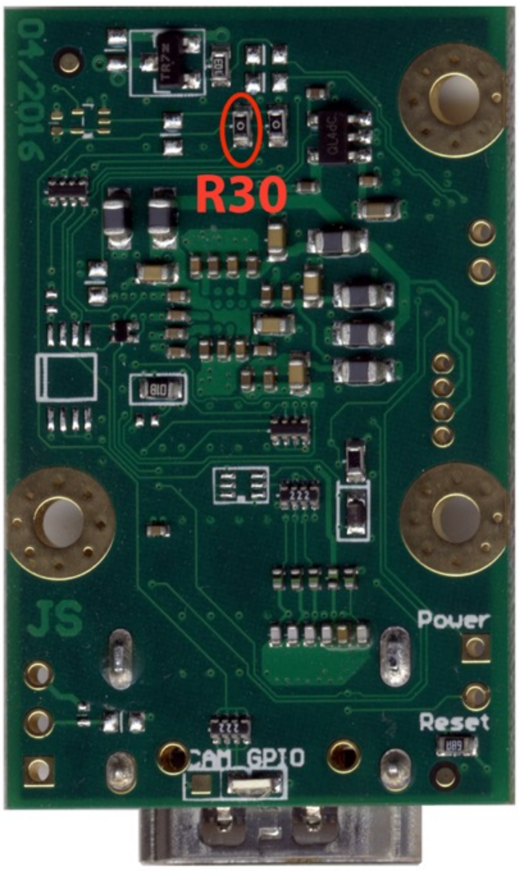
Python script for the Raspberry Pi B+ which uses GPIO 41 for the CAM_GPIO pin. This may be dependent on the Raspbian version. On some it may be GPIO 42.

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(41, GPIO.OUT)
GPIO.output(41, False)
GPIO.output(41, True)
```

Raspberry Pi 3 B: it needs to be determined what GPIO is connected to the CAM_GPIO pin on the 15 pin camera connector. Any info is welcome.

Use `nano restart.py` and insert the 4 lines above for your Raspberry Pi model.

Use `sudo python restart.py` to reset the B101 before you start raspivid again.



The picture shows the bottom side of the 38126-3.

Rev 3: R30 not populated

Rev 3.1 R30 populated with 0 Ohm

Only on rev 3.1 the software reset feature is active. To enable this feature on rev 3 modules R30 needs to be installed. R30 may just be a solder bridge.

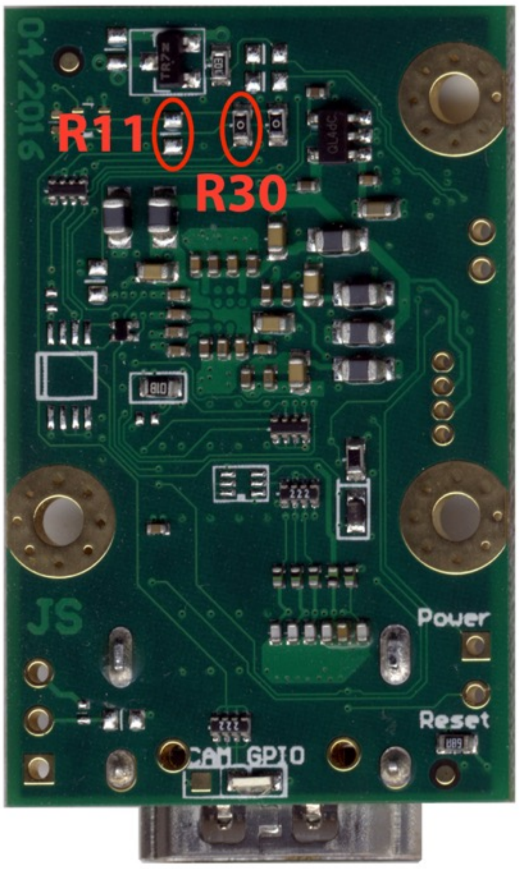
Note: for rev 3.1 to work properly pin 11 on the cable must be driven high or must be floating. A low on this pin puts the module into Reset state. If the application applies a permanent 0 (low level) on this pin, then R30 needs to be removed. Alternatively please order the rev 3 version of the B101. This can be custom build on special request.

Note: for rev 3 the script above may be used to control the GPIO LED.

Hardware modification for cable detect feature

Alternatively the cable detect may be connected to GPIO 21/42 (CAM_GPIO on the 15 pin FFC cable). This modification may be performed on rev 3 and rev 3.1 modules.

1. remove R30 and remove the GPIO LED below the HDMI connector
2. add a solder bridge or 0 Ohm 0603 resistor for R11. This connects the scaled down 5V (reduced to 2.5V) from the HDMI cable to the GPIO pin. In this manner the cable detect status may be read from a GPIO rather then from the TC358743 device, as the I2C bus may be blocked by the driver.
3. wire a cable from the Reset hole (on the bottom right of the photo below) to GPIO 4 on the Raspberry Pi GPIO header. Run the software script as described above to reset the TC358743 chip.

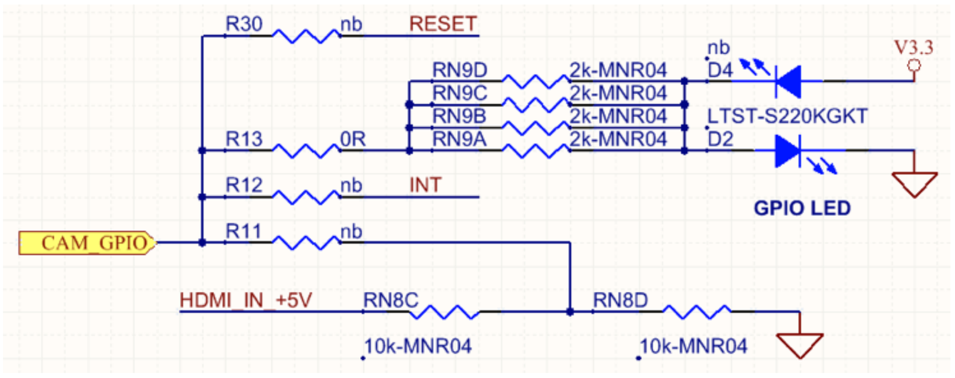


A high (2.5V) indicates that a HDMI cable is plugged in and that the connecting HDMI device is powered up and that it provides 5V on the cable.

A low indicates, that no cable is plugged in or that the HDMI device is not powered up.

Schematics

CAM_GPIO on the 38126-3 may be used for multiple purposes. The default configuration is to control the GPIO LED which is located underneath the HDMI input.



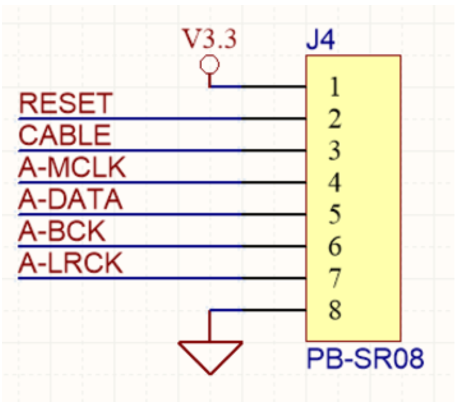
B101 rev 4

The B101 rev 4 has the model number 38126-4. It adds an 8 pin connector for I2S audio, reset and cable detect.

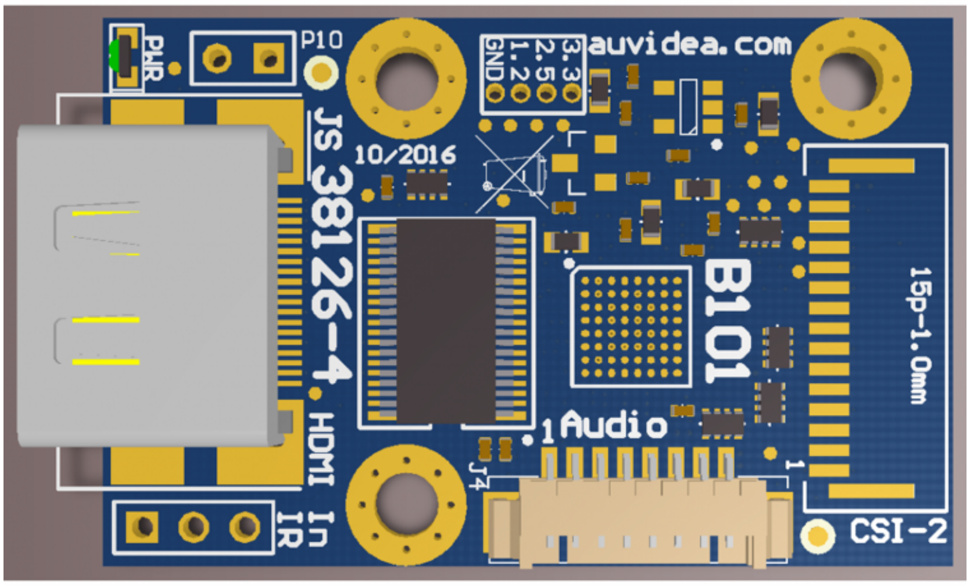
RESET and CABLE may be connected to GPIOs of the GPIO header of the Raspberry Pi to control the hardware reset of the Toshiba TC358743 chip. The B101 rev 4 ships with a cable set for this connector.

Pin 1 is marked on the silkscreen. It is the left pin in the screenshot below.

The FPC cable faces with the contacts up. For The RPi uses an FFC cable with contacts on the same side. Cables in various lengths are available in the Auvidea online store.



Pin	Function	Description
1	V3.3	V3.3 power out (on-board LDO with 300mA max. current)
2	RESET	hardware reset to the TC358743 HDMI to CSI-2 bridge (3.3V level, low active)
3	CABLE	Cable detect: high (3.3V) is HDMI cable is plugged in
4	A-MCLK	I2S audio master clock (3.3V level)
5	A-DATA	I2S audio data out (3.3V level)
6	A-BCK	I2S audio bit clock (3.3V level)
7	A-LRCK	I2S audio word clock (3.3V level)
8	GND	Ground (0V)



B102 rev 1

B102 HDMI to CSI-2 bridge with 4 lanes

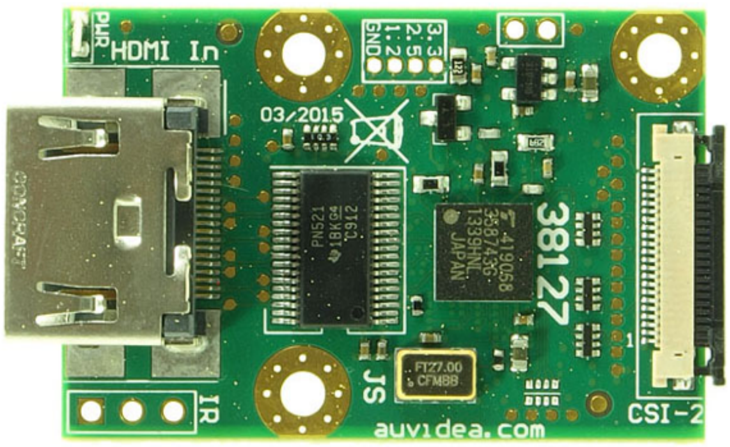
The B102 uses a 22 pin FPC connector with 0.5mm pitch and 4 CSI-2 data lanes. It features the same pinout as the Raspberry Pi 22 pin CSI-2 con

The B102 rev 1 is has the model number 38127.

The FPC cable faces with the contacts down. For The RPi and the Auvidea Jetson carrier boards please use an FFC cable with contacts on the same side. Cables in various lengths are available in the Auvidea online store

The picture on the bottom shows the B102 connected to the J120 Jetson TX1 carrier board.

A GPL driver has been developed by Ridgerun. Ridgerun provides driver customisation services.



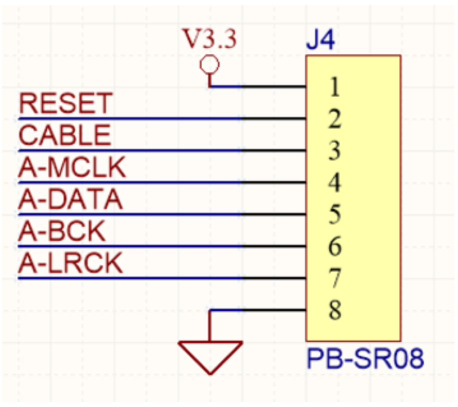
B102 rev 2

The B102 rev 1 is has the model number 38127-2. It adds an 8 pin connector for I2S audio, reset and cable detect.

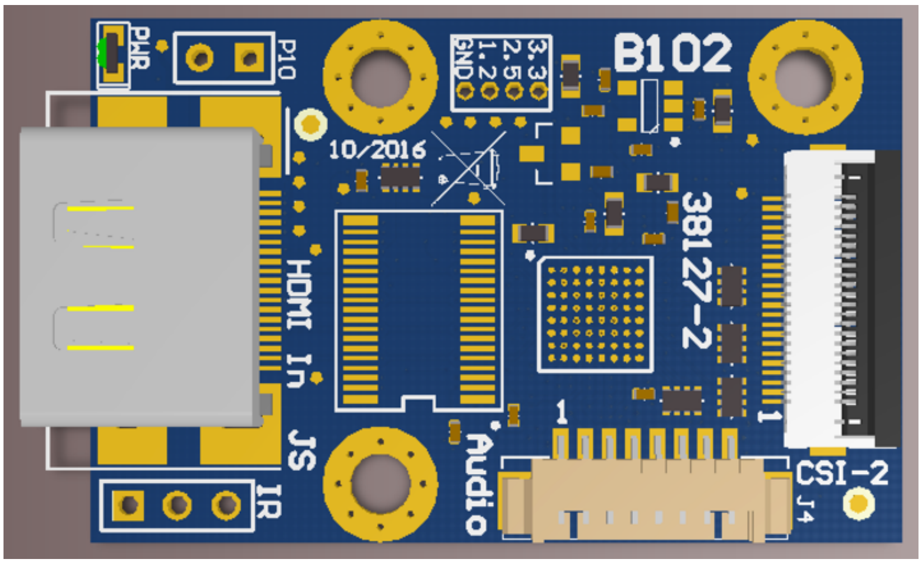
RESET and CABLE may be connected to GPIOs of the GPIO header of the Raspberry Pi to control the hardware reset of the Toshiba TC358743 chip. The B102 rev 2 ships with a cable set for this connector.

Pin 1 is marked on the silkscreen. It is the left pin in the screenshot below.

The FPC cable faces with the contacts down. For The RPi and the Auvidea Jetson carrier boards please use an FFC cable with contacts on the same side. Cables in various lengths are available in the Auvidea online store.



Pin	Function	Description
1	V3.3	V3.3 power out (on-board LDO with 300mA max. current)
2	RESET	hardware reset to the TC358743 HDMI to CSI-2 bridge (3.3V level, low active)
3	CABLE	Cable detect: high (3.3V) is HDMI cable is plugged in
4	A-MCLK	I2S audio master clock (3.3V level)
5	A-DATA	I2S audio data out (3.3V level)
6	A-BCK	I2S audio bit clock (3.3V level)
7	A-LRCK	I2S audio word clock (3.3V level)
8	GND	Ground (0V)

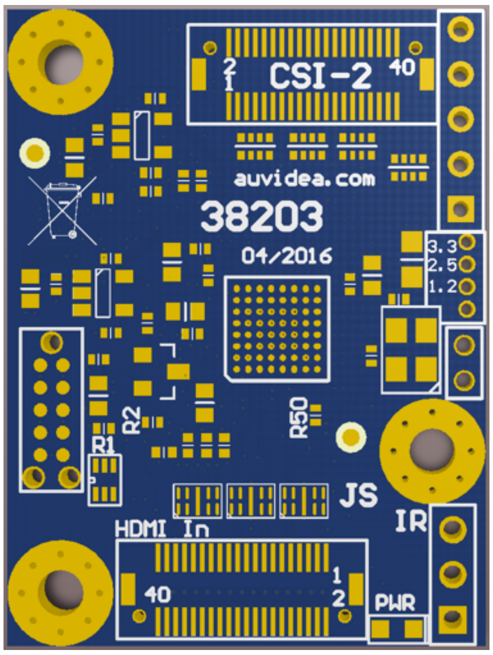


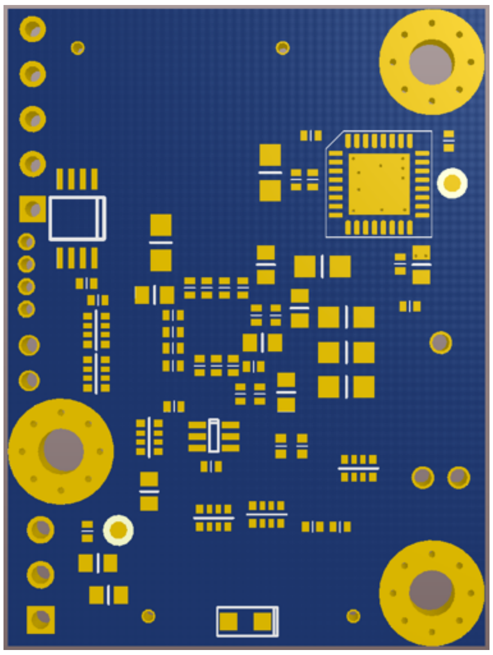
B103

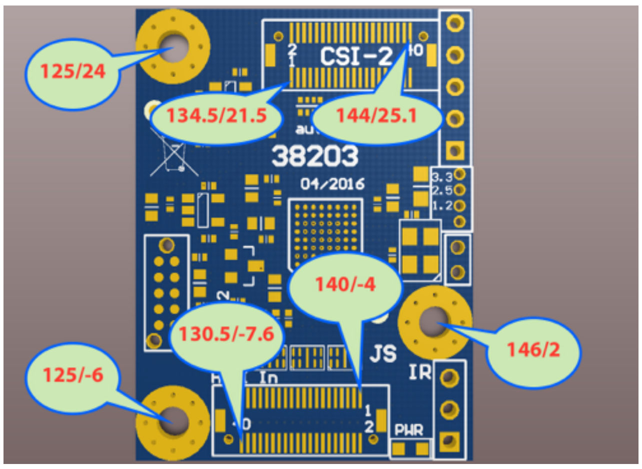
B103 HDMI to CSI-2 bridge for system integration

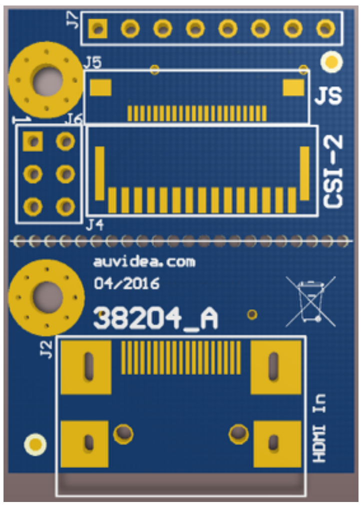
The B103 is a custom version of the B100 where the HDMI input connector got replaced by a 40 pin board to board connector. This allows for easy integration into target systems. To ease the connection and verification two interface modules for the HDMI input and the CSI-2 output may optionally be provided.

size: 27 x 36mm
mounting holes: 4x 2.6mm









Raspberry software install

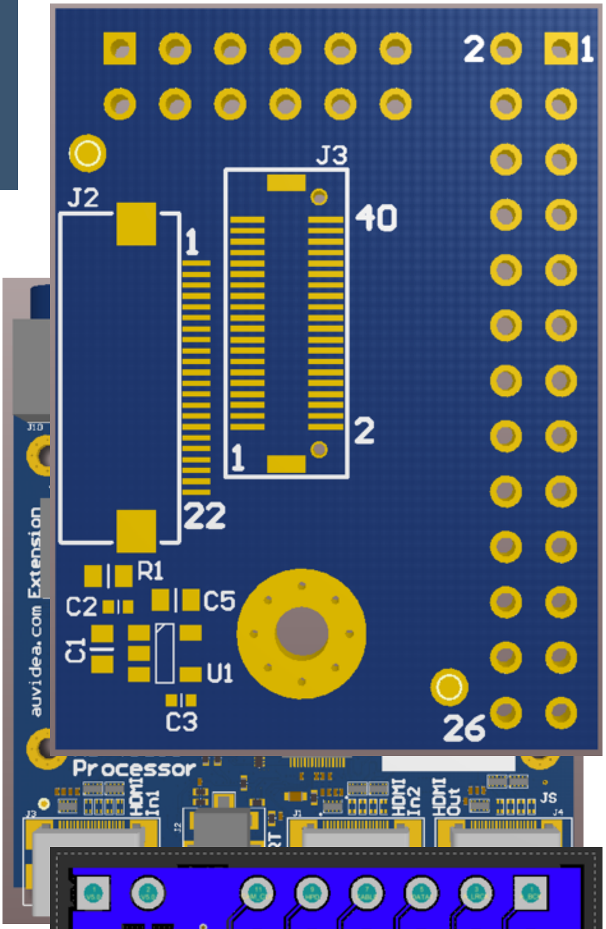
Raspberry software installation

1. we have tested it with the Linux Rapsbian Jessie Lite 4.1.13-v7+ version, which was released on 2015-11-21. First download an official image from https://www.raspberrypi.org/downloads/raspbian/. Alternative distributions are available from third-party vendors. After downloading the .zip file, unzip it to get the .img file for writing to your SD card.
2. write the downloaded image to SD card. Please see appendix B for more information.
3. use raspi-config and set `Enable Camera` to `Enable`. This enables the CSI-2 camera. The B101 emulates this camera. This is the reason that only 1080p25 is supported as input resolution, as this is the only native mode of the PI camera.
4. reboot the Raspberry Pi. After your Raspberry Pi was rebooted and you are successful logged in, you can test your B101 HDMI to CSI-2 Bridge (15 pin FPC), just run following command from terminal:

```
raspivid -t 0
```

Raspivid shows the camera output on the display and optionally saves an H.264 video at the requested bitrate. The parameter `-t` is specified for Time (in ms) to capture for. If not specified, it is set to 5s. Set to zero to disable it.

HD video processor

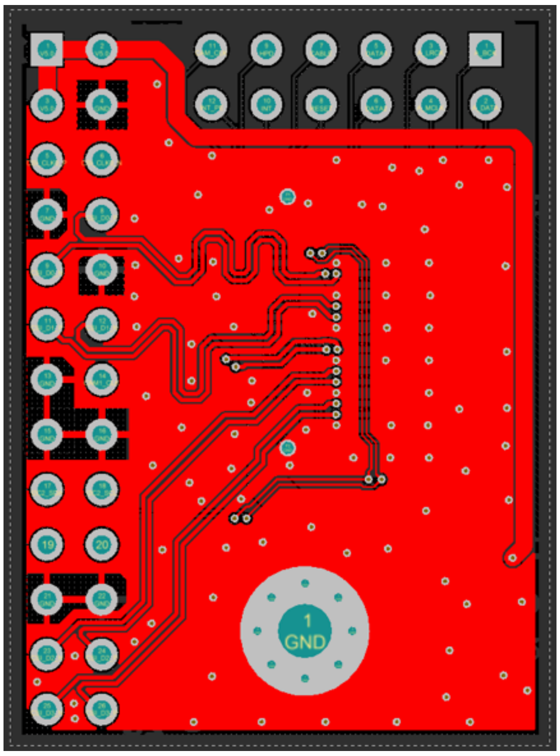


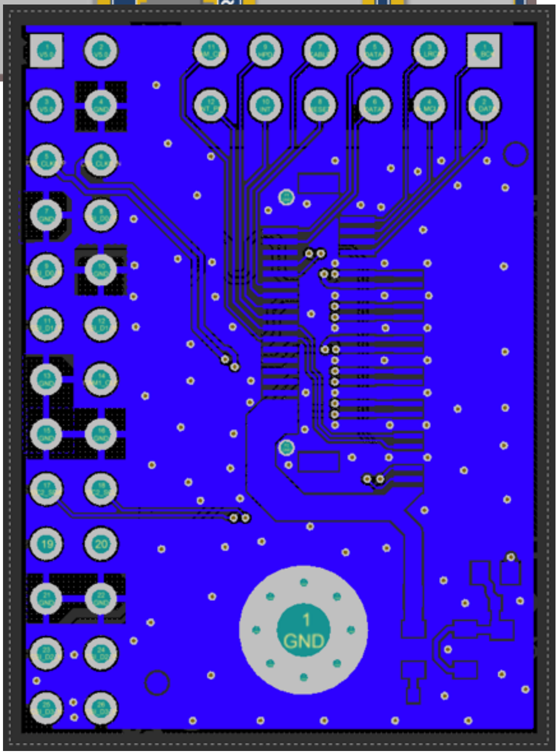
38189-3

The HD video processor could be a very useful companion product to the B101 and the Raspberry Pi. It addresses some short comings of the B101 with the RPi.

Advantages

- output fixed at 1080p25, while any HDMI input at any resolution or frame rate may be connected
- the output is always on - independent whether HDMI inputs are connected
- 1080p60 input support
- the video inputs may be merged
- deinterlacing of 480i to 1080i





Key features

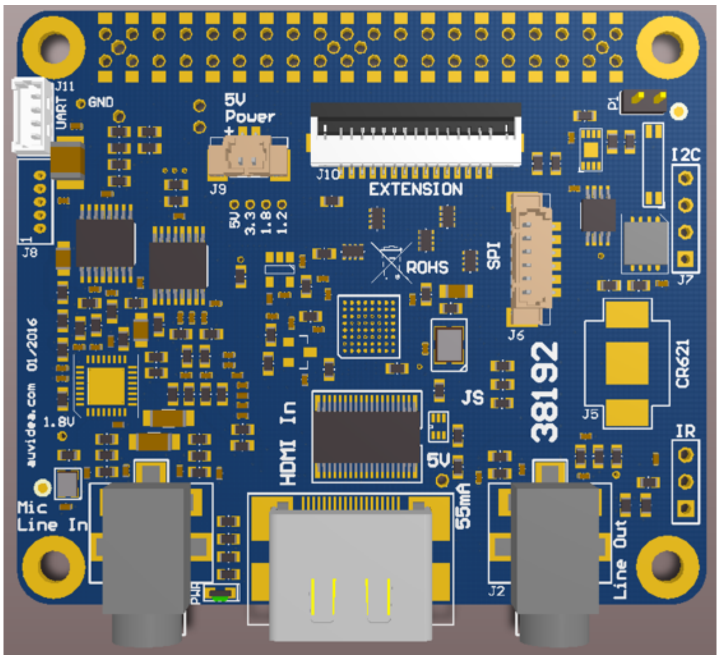
- the HDMI output will keep its preset format, even if the camera (notebook) resolution changes
- one input could be the output of a notebook - may freely be changed (with varying resolution)

- it can down sample 1080p60 to 1080p30, so that encoders can encode it
- 2 HDMI video inputs (up to 1080p60 each)
- deinterlacing, cropping and scaling of each HDMI input
- frame rate conversion
- flexible video positioning in 2 video windows for the output (like 3D view, PIP, PnP, etc)
- 256 color graphic video overlay for logos etc. (BMP file upload)
- size: same size and mounting hole pattern as E1xx - so they could be stacked (79.6 x 79.3 mm)
- on board micro controller (controlled through UART TTL console on J2)
- custom preset configurations selectable via DIP switches
- optional DMX control interface
- UART config interface (in future: possibly config via web interface)
- first prototypes in August 2016

B101 „hat“

38192

This hat is specifically designed for the Raspberry Pi. It integrates a B101, analog audio in/out and audio mixer.



Technical details

- 40 pin GPIO connector
- optional pass through connector on top
- 15 pin CSI-2 connector to RPi on the bottom side
- 15 pin extension connector to the 38189 (to control HD video processor)
- Mic/line input 3.5mm jack
- line out 3.5 mm jack
- on board mixer for analog and digital (HDMI embedded) audio
- 5V power in from 38189 (RPi powered by 12V input of 38189)
- size: 65.1 x 56 mm
- first prototypes: August/September 2016

38186 adapter for B100

38186 general purpose interface adapter for the B100

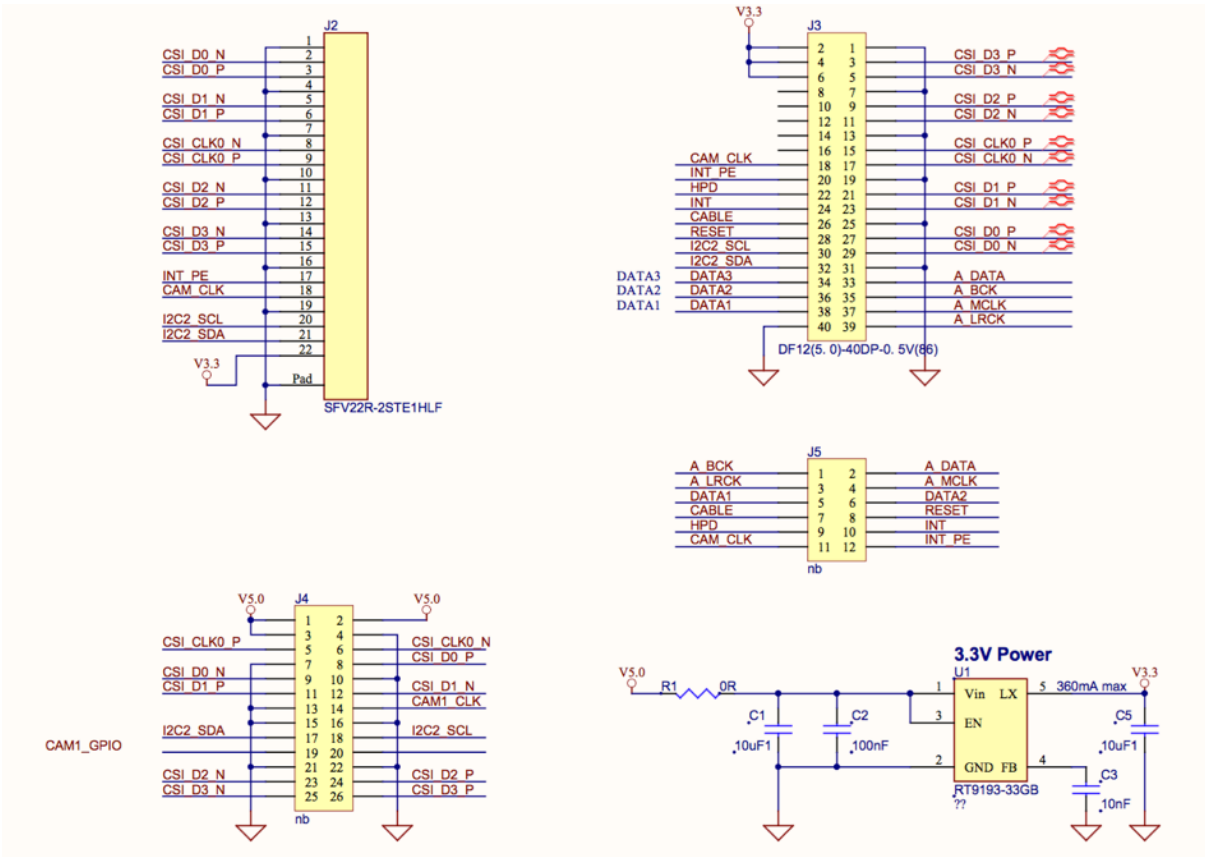
The 38186 was been designed to create a general purpose adopter, so that the B100 may be connected to a larger variety of target systems. For this purpose the CSI-2 signals are connected to a standard 2.54mm header, so that jumper cables may be run to the target system. This allows for quick prototyping. It is not recommended for a production system as the high frequency performance will be rather limited. For production systems a custom adapter board is advised. Auvidea can cost effectively develop such custom adapter boards. Please ask for a quote.

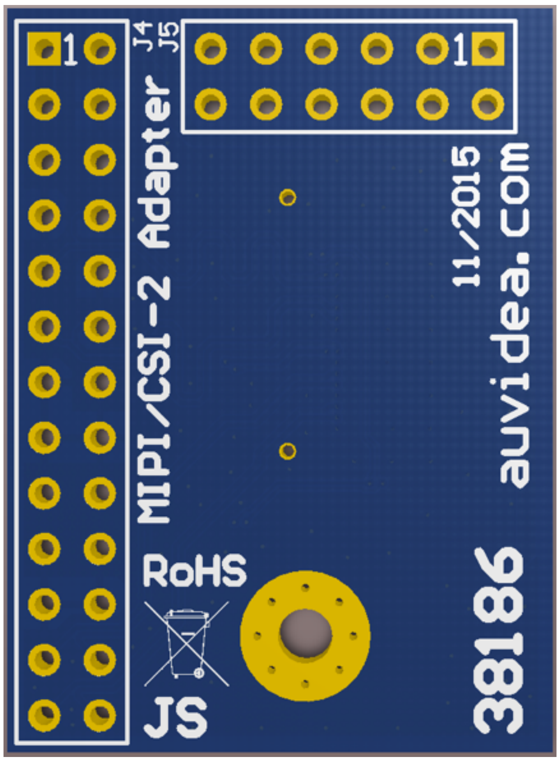
How to connect the B100 to a unsupported target system?

To interface the B10x to other compute platforms 2 things are required:

1. first the B10x has to be interfaced hardware wise. As CSI-2 defines no physical standard interface each manufacturer implements his proprietary interface. There are 2 options. Either a simple custom interface board is developed or you use the general purpose adapter board (38186). Please find details below. In other cases one could run jumper cables from this adapter board to the target hardware. But you carefully need to check the power supply and the I2C voltage level. We use an I2C voltage of 3.3V.

2. the video driver needs to be adopted and ported. Kernel 4 has native drivers for the TC358743. So if your target system uses the Linux Kernel 4 it should be rather simple to get the drivers to work. But most embedded systems still use Kernel 3. Here a bit more work is required and the driver needs to be back ported or adopted from another platform. Example: the driver for the TX1 are based on the Kernel 3.10.



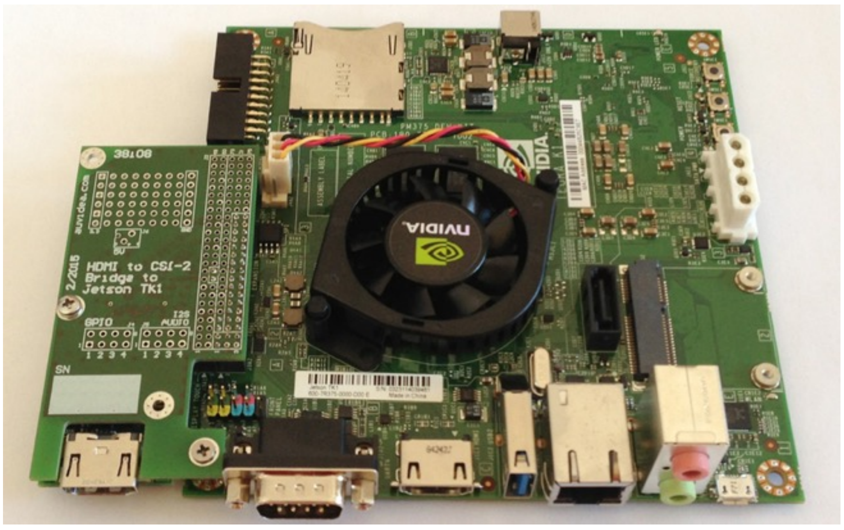

1
J4
J5
1
MIPI/CSI-2 Adapter
11/2015
auvidea.com
RoHS
JS
38186

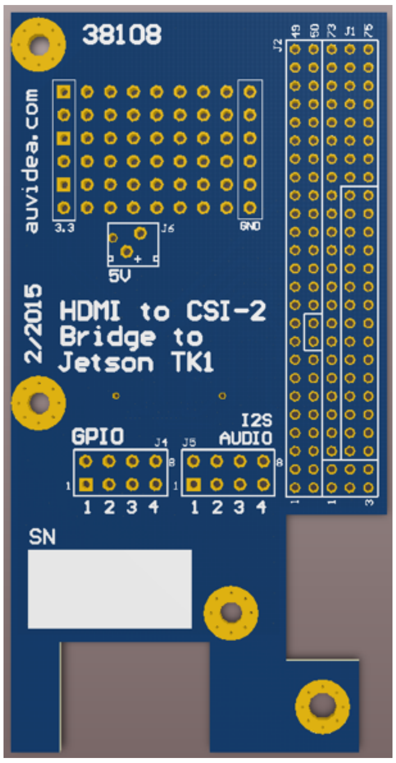
Jetson TK1 adapter

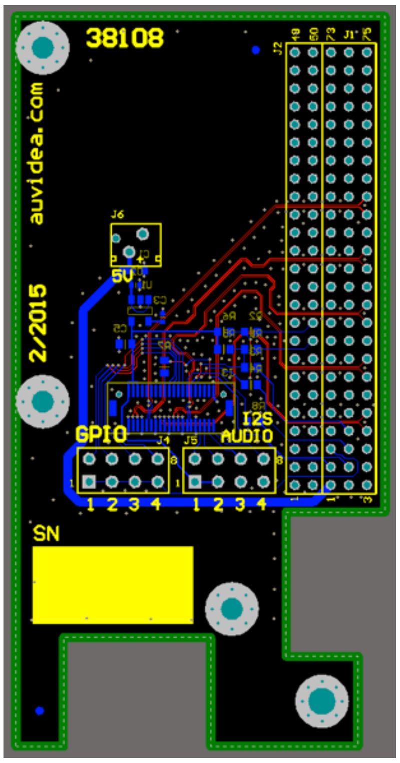
38108-2 B100-TK1 adapter

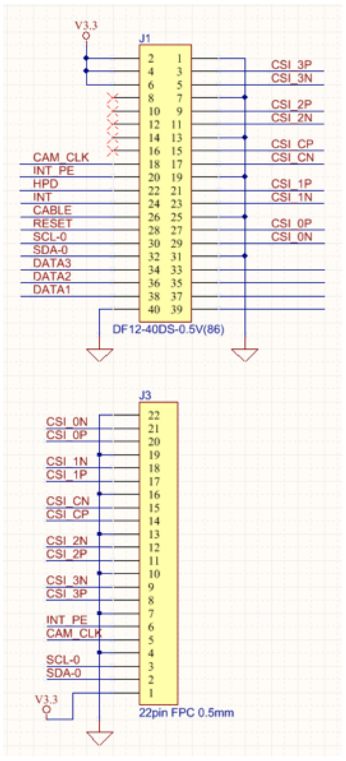
The adapter board is a 2 layer board, which interfaces one B100 to the Jetson TK1. The Tegra K1 processor features two 4 lane CSI-2 ports, but only one is brought out on the extension connector of the Jetson board (pin header with 2mm pitch).

Note: Auvidea does not provide a driver for this setup. For a solution which works out of the box please have a look at the TX1 carrier boards (like the J120).









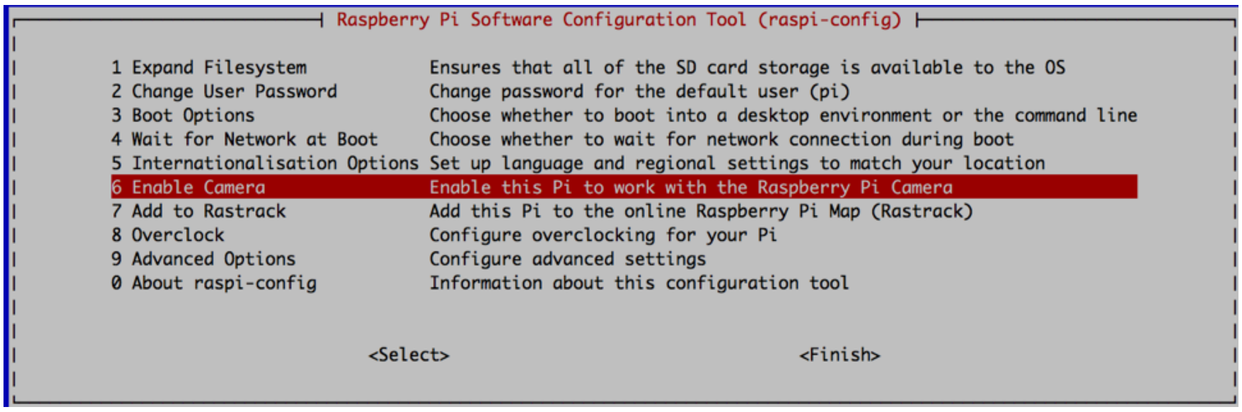
Appendix A

Raspberry Pi configuration tool raspi-config

Run raspi-config to enable the camera interface. If Raspbian just was installed raspi-config to automatically started. To open the configuration tool after this, simply run the following command from the command line:

```
sudo raspi-config
```

`sudo` is required because you will be changing files that you do not own as the `pi` user. You should see a blue screen with options in a grey box in the centre, like so:



```
┤ Raspberry Pi Software Configuration Tool (raspi-config) ├

 1 Expand Filesystem           Ensures that all of the SD card storage is available to the OS
 2 Change User Password        Change password for the default user (pi)
 3 Boot Options                Choose whether to boot into a desktop environment or the command line
 4 Wait for Network at Boot    Choose whether to wait for network connection during boot
 5 Internationalisation Options Set up language and regional settings to match your location
 6 Enable Camera               Enable this Pi to work with the Raspberry Pi Camera
 7 Add to Rastrack             Add this Pi to the online Raspberry Pi Map (Rastrack)
 8 Overclock                   Configure overclocking for your Pi
 9 Advanced Options            Configure advanced settings
 0 About raspi-config          Information about this configuration tool


                 <Select>                             <Finish>
```

Use the up and down arrow keys to move the highlighted selection between the options available. Pressing the right arrow key will jump out of the options menu and take you to the `<Select>` and `<Finish>` buttons. Pressing left will take you back to the options. Alternatively, use the `Tab` key to switch between these.

Generally speaking, `raspi-config` aims to provide the functionality to make the most common configuration changes. This may result in automated edits to `/boot/config.txt` and various standard Linux configuration files. Some options require a reboot to take effect. If you changed any of those, raspi-config will ask if you wish to reboot now when you select the `<Finish>` button.

For the B101 HDMI to CSI-2 bridge, you need to activate the camera module. Select the option and select `Enable`. This will dedicate at least 128MB of RAM to the GPU. Exit `raspi-config` and reboot your Raspberry Pi. Now you can use the camera tools like raspivid.

Appendix B

Writing an image onto a SD card

With the image file of your choice, Raspbian Jessie or Raspbian Jessie Lite, you need to use an image writing tool to write the image on your micro SD card. For every system it's different. We recommend to use 8GB or larger micro SD card for Raspbian image and the Noobs installation which you can download from the official webpage https://www.raspberrypi.org/downloads/noobs/.

Linux

Please note that the use of the dd tool can overwrite any partition of your machine. If you specify the wrong device in the instructions below you could delete your primary Linux partition. Please be careful.

- Run df -h to see what devices are currently mounted.
- If you computer has a slot for SD cards, insert the card. If not, insert the card into an SD card reader, then connect the reader to you computer.
- Run df -h again. The new device that has appear is you SD card. The left column gives the device name of you SD card; it will be listed as something like /dev/mmcblk0pl or /dev/sdd1. The last part (p1 or 1 respectively) is the partition number but you want to write to the whole SD card, no just one partition. You therefore need to remove that part from the name, getting, for example, /dev/mmcblk0 or /dev/sdd as the device name for the whole SD card. Note: that the SD card can show up more than once in the output of df; it will do this if you have previously written Raspberry Pi image to this SD card, because the Raspberry Pi SD images have more than one partition.
- Now that you've noted what the device name is, you need to unmount it so that files can't be read or written to the SD card wile you are copying over the SD card.
- Run unmount /dev/sdd1 , replacing sdd1 with whatever you SD cards device name is (including the partition number).
- If your SD card shows up more than once in the output of df due to having multiple partitions on the SD card, you should unmount all of these partitions.
- In the terminal, write the image to the card with the command below, making sure you replace the input file if= argument with the party to you .img file, and the /dev/sdd in the output file of= argument withe the right device name. This is very important, as you will lose all data on the hard drive if you provide the wrong device name. Make sure the device name is the name of the whole SD card as described above, not just a partition of it; for example, sdd, not sdds1 or sddp1, and mmcblk0, not mmcblk0p1.

```
dd bs=4M if=2016-03-18-raspbian-jessie.img of=/dev/sdd
```

- Please note that block size set to `4M` will work most of the time; if not, please try `1M`, although this will take considerably longer.
- Also note that if you are not logged in as root you will need to prefix this with `sudo`.
- The `dd` command does not give any information of its progress and so may appear to have frozen; it could take more than five minutes to finish writing to the card. If you card reader has an LED is may blink during the write process. To see the progress of the copy operation you can run `pkill -USR1 -n -x dd` in another terminal, prefixed with `sudo` if you are not logged in as root. The progress will be displayed in the original windows and not the windows with he `pkill` command; it may not display immediately, due to buffering.
- Instead of dd you can use dcfldd; it will give a progress report about how much has been written.

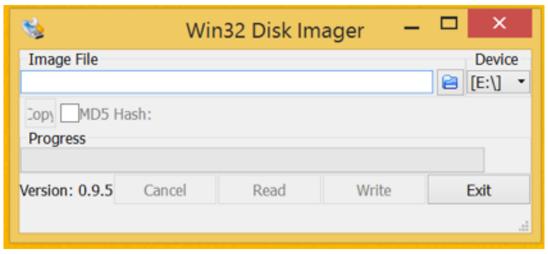
- You can check what's written to the SD card by dd-ing from the card back to another image on you hard disk, truncating the new image to the same size as the original, and then running diff (or md5sum) on those two images.
- The SD card might be bigger than the original image, and dd will make a copy of the whole card. We must therefore truncate the new image to the size of the original image. Make sure you replace the input file if= argument with the right device name. diff should report that the files are identical.

```
        dd bs=4M if=/dev/sdd of=from-sd-card.img
        truncate --reference 2016-03-18-raspbian-jessie-img from-sd-card.img
        diff -s from-sd-card.img 2016-03-18-raspbian-jessie.img
```

- Run `sync`; the will ensure the write cache is flushed and that it is safe to unmount you SD card.
- Remove the SD card from the card reader.

Windows

- Insert the SD card into your SD card reader and check which drive letter was assigned. You can easily see the driver letter, such as `G:`, by looking in the left column of Windows Explorer. You can use the SD card slot if you have one, or cheap SD Adapter in a USB port.
- Download the Win32DiskImager utility form http://sourceforge.net/projects/win32diskimager/ as a zip file; you can run this from a USB drive.
- Extract the executable from the zip file and run the `Win32DiskImager` utility; you may need to run this as administrator. Right-click on the file, and select **Run as administrator**



- Select the image file you extracted earlier.
- Select the drive letter of the SD card in the device box. Be careful to select the correct drive; if you get the wrong you can destroy the data on you computer's hard disk! If you are using an SD card slot in your computer and can't see the drive in the Win32DiskImager window, try using an external SD adapter.
- Click **Write** and wait for the write to complete
- Exit the imager and eject the SD card

Mac OS X

On Mac OS you have the choice of the command line `dd` tool or using the graphical tool ImageWriter to write the image to your SD card.

Graphical Interface

- Connect the SD card reader with the SD card inside. **Note:** that it must be formatted as FAT32
- From the Apple Menu, choose "About This Mac" then click on "More info..."; if you are using Mac OS X 10.8.x Mountain Lion or newer, then click on "System Report"
- Click on "USB" (or "Card Reader" if using a built-in SD card reader) then search for you SD card in the upper-right section of the window. Click on it, then search for the BSD name in the lower-right section; it will look something like `diskn` where `n` is a number (for example, `disk4`). Make sure you take a note of this number.

- Unmount the partition so that you will be allowed to overwrite the disk. To do this, open Disk Utility and unmount it; do not eject it, or you will have to reconnect it. Note: that on Mac OS X 10.8.x Mountain Lion, "Verify Disk" (before unmounting) will display the BSD name as /dev/disk1s1 or similar, allowing you to skip the previous two steps.
- For the terminal, run the following command:

```
sudo dd bs=1m if=path_of_you_image.img of=/dev/rdiskn
```

- Remember to replace `n` with the number of you noted before! If this commands fails, try using `disk` instead of `rdisk`:

Command Line

- If you are comfortable with the command line, you can write the image to a SD card without any additional software. Open a terminal, then run:

```
diskutil list
```

- Identify the disk (not partition) of you SD card e.g `disk4`, not `disk4s1`
- Unmount your SD card by using the disk identifier, to prepare for copying date to it. Disk is your BSD name e.g. `diskutil unmountDisk /dev/disk4`

```
diskutil unmountDisk /dev/disk<disk# form diskutil>
```

- Copy the data to your SD card:

```
sudo dd bs=1m if=image.img of=/dev/rdisk<disk# from diskutil>
```

where disk is your BSD name e.g.

```
sudo dd bs=1m if=2016-03-18-raspbian-jessie.img of=/dev/rdisk4
```

This may result in a dd: invalid number '1m' error if you have GNU coreutils -installed. In that case, you need to use a block size of 1M in the bs= section, as follows:

```
sudo dd bs=1M if=image.img of=/dev/rdisk<disk# from diskutil>
```

This will take a few minutes, depending on the image file size. You can check the progress by sending a SIGINFO signal (press Ctrl+T).

- If this command still fails, try using disk instead of rdisk, for example:

```
sudo dd bs=1m if=2016-03-18-raspbian-jessie.img of=/dev/disk4
or
sudo dd bs=1M if=2016-03-18-raspbian-jessie.img of=/dev/disk4
```

Alternative method

Note: Some users have reported issues with using this method to create SD cards.

These commands and actions need to be performed from an account that has administrator privileges.

- From the terminal run `df -h`
- Connect the SD card reader with the SD card inside
- Run `df -h` again and look for the new device that wasn't listed last time. Record the device name of the filesystem's partition, for example `/dev/disk3s1`
- Unmount the partition so that you will be allowed to overwrite the disk:

```
      sudo diskutil unmount /dev/disk3s1
```

Alternatively, open Disk Utility and unmount the partition of the SD card; do not eject it, or you will have to reconnect it.

- Using the device name of the partition, work out the raw device name for the entire disk by omitting the final `s1` and replacing disk with `rdisk`. This is very important, as you will lose all data on the hard drive if you provide the wrong device name. Make sure the device name is the name of the whole SD card as described above, not just a partition of it - for example, `rdisk3`, not `rdisk3s1`. Similarly, you might have another SD drive name/number like `rdisk2` or `rdisk4`; you can check again by using the `df -h` command both before and after you insert you SD card reader into you Mac. For example, `/dev/disk3s1` becomes `/dev/disk3`.
- In the terminal, write the image to the card with this command, using the raw device name from above. Read the above step carefully to be sure you use the correct `rdisk` number here:

```
      sudo dd bs=1m if=2016-03-18-raspbian-jessie.img of=/dev/rdisk3
```

If the above command reports the error

`dd: bs: illegal numeric value`, please change the block size `bs=1m` to `bs=1M`

If the above command reports the error

```
      dd: /dev/rdisk3: Permission denied
```

It means the partition table of the SD card is being protected against being overwritten by Mac OS. Erase the SD card's partition table using this command:

```
      sudo diskutil partitionDisk /dev/disk3 1 MBR „Free Space‟ „%noformat%‟ 100%
```

That command will also set the permissions on the device to allow writing. Now try the `dd` command again.

Note: that dd will not provide any on-screen information util there is an error or it is finished; when complete, information will be shown and the disk will re-mount. If you wish to view the progress, you can use Ctrl-T; this generates SIGINFO, the status argument of your terminal, and will display information on the process.

- After the dd command finishes, eject the card:

```
sudo diskutil eject /dev/rdisk3
```

Alternatively, open Disk Utility and use this to eject the SD card.

Noobs

Alternatively, NOOBS is available for download on the Raspberry Pi website: https://www.raspberrypi.org/downloads/noobs/. Extract the files from the downloaded zip. We recommend using an SD card with a minimum capacity of 8GB.

- Using the builded-in tools of your operation system format an SD card in FAT32. If you are not sure in your actions, please read this part of the documentation and use offered tools from us.
- Download the SD Formatter 4.0 for either Windows or Mac https://www.sdcard.org/downloads/formatter_4/
- Follow the instructions to install the software.
- Insert you SD card into the computer or laptop's SD card reader and make a not of the driver letter allocated to is, e.g. G:/
- In SD Formatter, select the drive letter for your SD card and format it.
- Once you SD card has been formatted, drag all the files in the extracted NOOBS folder and drop them onto the SD card drive.
- The necessary files will then be transferred to your SD card.
- When this process has finished, safely remove the SD card and insert it into the you Raspberry Pi.
- Plug in your keyboard, mouse and monitor cables.
- Now plug the USB power cable into your Pi.
- Your Raspberry Pi will boot and a window will appear with a list of different operating systems that you can install. Tick the box next to Raspbian and click on Install.
- Raspbian will then run thought its installation process. Note: that this can take a while.
- When the install process has completed, the Raspberry Pi configuration menu (raspi-config) will load. Here you are able to set the time and date for your region, enable a Raspberry Pi camera board, even create users. You can exit this menu by using Tab on your keyboard to move to Finish.

The default login for Raspberry is username pi with the password raspberry.

Note: that you will not see any writing appear when you type the password. This is a security feature in Linux.

To load the graphical user interface, type startx and press Enter.

Appendix C: ffmpeg

FFmpeg

We will need FFmpeg to conversions H264 to MP4 digital multimedia container format and for UDP streaming.

Note: do not install FFmpeg via apt-get, aptitude or yum. For compiling installing FFmpeg read the following part of the documentation. Compiling for the Raspberry Pi takes a little more patience and care. If you want to include support for H.264 video, this needs to be installed before compiling FFmpeg.

Installing H264 support

Run the following commands, one at a time.

```
cd /usr/src
sudo git clone git://git.videolan.org/x264
cd x264
sudo ./configure --host=arm-unknown-linux-gnueabi —enable-static --disable-opencl
sudo make -j4
sudo make install
```

Installation other libraries and formats

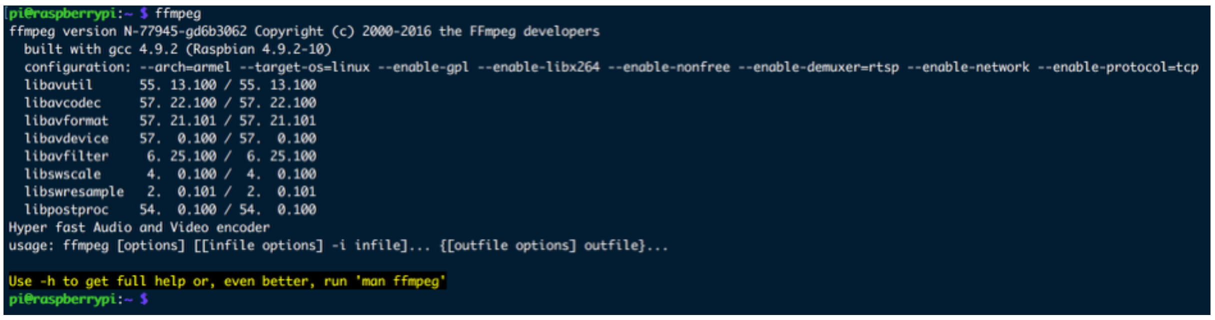
Anything else you would to install should be done now, before compiling FFmpeg. This includes MP3, AAC, etc.

Installation of FFmpeg

Add lines similar to the `--enable-libx264` for anything else installed above.This may take really long time, so be patient.

```
cd /usr/src
sudo git clone git://source.ffmpeg.org/ffmpeg.git
cd ffmpeg
sudo ./configure --arch=armel --target-os=linux --enable-gpl --enable-libx264 --
enable-nonfree
sudo make -j4
sudo make install
```

When the compilation and installation is done, test FFmpeg, just run the command `ffmpeg` in your terminal.



```
pi@raspberrypi:~ $ ffmpeg
ffmpeg version N-77945-gd6b3062 Copyright (c) 2000-2016 the FFmpeg developers
  built with gcc 4.9.2 (Raspbian 4.9.2-10)
  configuration: --arch=armel --target-os=linux --enable-gpl --enable-libx264 --enable-nonfree --enable-demuxer=rtsp --enable-network --enable-protocol=tcp
  libavutil      55. 13.100 / 55. 13.100
  libavcodec     57. 22.100 / 57. 22.100
  libavformat    57. 21.101 / 57. 21.101
  libavdevice    57.  0.100 / 57.  0.100
  libavfilter     6. 25.100 /  6. 25.100
  libswscale      4.  0.100 /  4.  0.100
  libswresample   2.  0.101 /  2.  0.101
  libpostproc    54.  0.100 / 54.  0.100
Hyper fast Audio and Video encoder
usage: ffmpeg [options] [[infile options] -i infile]... {[outfile options] outfile}...

Use -h to get full help or, even better, run 'man ffmpeg'
pi@raspberrypi:~ $
```

UDP Stream and recording

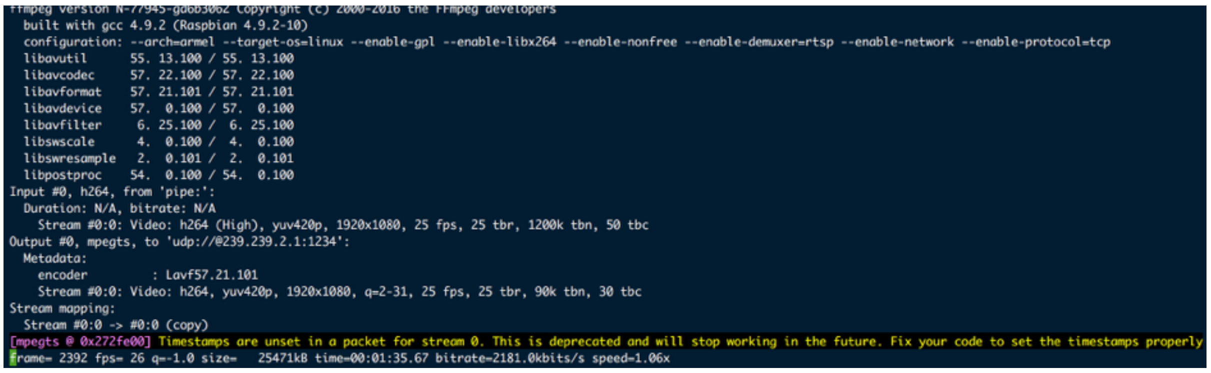
If you have skipped last part of the documentation and not compiling and installed FFmpeg, you will not be able to stream with UDP.

After installing FFmpeg you can start for example permanently UDP stream with command below and watch it in for example in VLC Player, use `udp://239.239.2.1:1234`.

To start the UDP stream, we need to run following command from terminal. Here we use the camera output from raspivid as source for ffmpeg.

```
raspivid -t 0 -fps 25 -hf -b 2000000 -o - | ffmpeg -i - -vcodec copy -an -r 30 -g
30 -bufsize 2000000 -pix_fmt yuv420p -f mpegts udp://@239.239.2.1:1234
```

After UDP stream is successfully started, you will receive information for example like on picture below.



```
ffmpeg version N-77945-gd6b3062 Copyright (c) 2000-2016 the FFmpeg developers
  built with gcc 4.9.2 (Raspbian 4.9.2-10)
  configuration: --arch=armel --target-os=linux --enable-gpl --enable-libx264 --enable-nonfree --enable-demuxer=rtsp --enable-network --enable-protocol=tcp
  libavutil      55. 13.100 / 55. 13.100
  libavcodec     57. 22.100 / 57. 22.100
  libavformat    57. 21.101 / 57. 21.101
  libavdevice    57.  0.100 / 57.  0.100
  libavfilter     6. 25.100 /  6. 25.100
  libswscale      4.  0.100 /  4.  0.100
  libswresample   2.  0.101 /  2.  0.101
  libpostproc    54.  0.100 / 54.  0.100
Input #0, h264, from 'pipe:':
  Duration: N/A, bitrate: N/A
    Stream #0:0: Video: h264 (High), yuv420p, 1920x1080, 25 fps, 25 tbr, 1200k tbn, 50 tbc
Output #0, mpegts, to 'udp://@239.239.2.1:1234':
  Metadata:
    encoder         : Lavf57.21.101
    Stream #0:0: Video: h264, yuv420p, 1920x1080, q=2-31, 25 fps, 25 tbr, 90k tbn, 30 tbc
Stream mapping:
  Stream #0:0 -> #0:0 (copy)
[mpegts @ 0x272fe00] Timestamps are unset in a packet for stream 0. This is deprecated and will stop working in the future. Fix your code to set the timestamps properly
frame= 2392 fps= 26 q=-1.0 size=   25471kB time=00:01:35.67 bitrate=2181.0kbits/s speed=1.06x
```

While UDP stream is started, you can record video. For example to record a .mp4 video:

```
ffmpeg -i udp://@239.239.2.1:1234 -vcodec copy -an -r 25 -t time_in_seconds -
pix_fmt yuv420p -f mp4 path_for_save_video.mp4
```

Remember to edit `time_in_seconds` and `path_for_save_video.mp4` as it is necessary for you.

Note: you can edit the FFmpeg configuration for streaming and recording.

Appendix D: Matterhorn

Opencast Matterhorn (opencast.org)

Matterhorn is a open source lecture recording system. The capture agent captures new content. A Raspberry Pi with the B101 HDMI to CSI-2 bridge may create a simple and low cost capture agent. If other input resolutions than 1080p25 are required, the HD video processor (38189) may be added as a pre-processor to convert any input resolution to a fixed 1080p25. Also 2 input sources may be merged into a single video.

Please follow original documentation on the Matterhorn Opencast installations https://docs.opencast.org/latest/admin/installation/.

Sample installation: below it is described how we have installed Matterhorn and configured the Raspberry based capture agent. It outlines the installation of an all in one Opencast system on Ubuntu 15.10 LTS. First clone the git repository:

```
git clone https://bitbucket.org/opencast-community/matterhorn.git
cd matterhorn
git tag                 <- List all available versions
git checkout TAG        <- Switch to desired version
```

We recommend to use the 2.1.1-rc version, which we tested and is working with this documentation. Next you need to install required components for you server or PC.

```
sudo apt-get install openjdk-7-jdk
or
sudo apt-get install openjdk-8-jdk
```

ffmpeg >= 2.8. Download FFmpeg 2.8.2:

```
wget http://ffmpeg.org/releases/ffmpeg-2.8.2.tar.bz2
```

Extract tarball and compile source of FFmpeg 2.8.2:

```
tar xf ffmpeg-2.8.2.tar.bz2
./configure --disable-yasm
make -j4
```

Install FFmpeg 2.8.2:

```
sudo make install
```

maven >= 3.1:

```
sudo apt-get install maven
```

tesseract >= 3:

```
sudo apt-get install tesseract-ocr
```

hunspell >= 1.2.8 (optional):

```
sudo apt-get install hunspell
```

sox >= 14.4 (optional):

```
sudo apt-get install sox
```

Now you need to build Opencast Matterhorn. Go into folder where is your downloaded matterhorn and type command below in you terminal.

```
mvn clean install
```

This will take very long time be patient. After successful building you can start Matterhorn using command below:

```
/matterhorn/build/opencast-dist-allinone-2.1.1-rc1/bin/start-opencast
```

PyCA - Matterhorn Capture Agent

PyCA is a Python based Matterhorn capture agent designed for the Raspberry Pi and Pi camera. As the B101 emulates the Pi camera, it is also supported. Please refer to the official documentation on git. The following commands show a sample installation, which we have performed and tested.

```
git clone https://github.com/lkiesow/pyCA.git
cd pyCA
sudo apt-get install python-virtualenv python-dev libcurl4-gnutls-dev
virtualenv venv
. ./venv/bin/activate
pip install icalendar python-dateutil pycurl configobj
nano etc/pyca.conf                                     <-- Edit the configuration
./start.sh
```

FAQ

1. raspivid does not show the video

- the output resolution of your HDMI signal must be 1080p25. No other resolution is supported
- the HDMI signal must not be encrypted with HDCP

2. raspivid only starts once

- the B101 module must be reset after each use of raspivid

Disclaimer

Thank you for reading this manual. If you have found any typos or errors in this document, please let us know.

This is the preliminary version of this data sheet. Please treat all specifications with caution as there may be any typos or errors.

The Auvidea Team