



DEVKIT ADDON CAM-TS-A01

Haikou-Video-Demo for the **HAIKOU CB-MINI-ITX**

featuring a 5" 720p MIPI-DSI touchscreen and 5 MP MIPI-CSI camera

USER MANUAL

Contents

1	Introduction	1
1.1	Package Contents	1
1.2	Mounting Instructions	2
1.3	RINGNECK SOM-PX30-uQ7 Kernel Configuration	8
1.4	PUMA SOM-RK3399-Q7 Kernel Configuration	8
2	Camera	9
2.1	Libcamera installation on Debian	9
2.2	Camera examples using GStreamer	10
2.3	Known issues	11
3	Contact	12
4	Revision History	13

1 Introduction

DEVKIT ADDON CAM-TS-A01 is meant to be used in conjunction with HAIKOU CB-MINI-ITX.

The display is a 5-inch color active matrix thin film transistor (TFT) liquid crystal display (LCD) with 720X1280 pixel resolution and 5-point capacitive touch.

The camera has a 1/5-inch, 5 megapixel (2592x1944) at 30 fps in 10-bit resolution sensor.



Fig. 1.1: Complete setup

Note: In favor of readability in this document DEVKIT ADDON CAM-TS-A01 will be referenced by its code name Haikou Video Demo and the HAIKOU CB-MINI-ITX as Haikou carrier board.

1.1 Package Contents

1. 1 x Haikou-Video-Demo jig with display, camera and nuts preassembled:
 - a) 1 x Haikou Video Demo jig
 - b) 1 x Display LTK050H3146W-08QC (V3)
 - c) 1 x Camera SF2V5675AY10
 - d) 2 x M3 nut
2. 1 x Haikou-Video-Demo adapter PCB

3. 3 x hexagonal M3 metal standoffs, 4.5cm each (Würth 971450471 or similar)
4. 6 x M3x5mm screws
5. 4 x M3 nut

1.2 Mounting Instructions

Step 1: Lay the pre-assembled Haikou Video Demo jig with display and camera face-down on a flat surface as shown below:

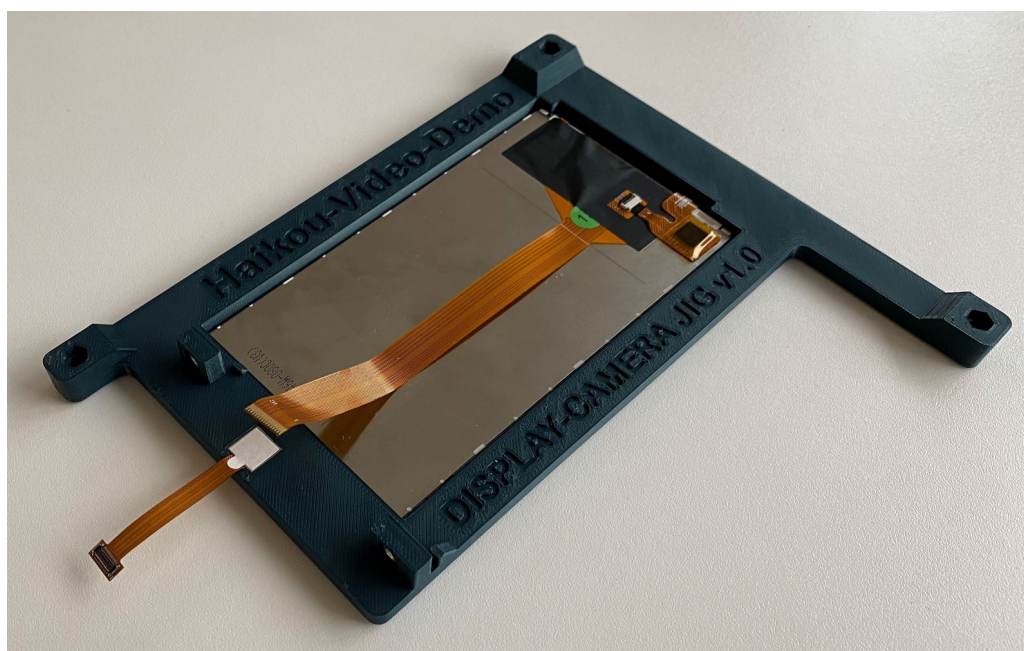


Fig. 1.2: Mounting the OV5675 Camera

Step 2: Rotate the display and connect flat-flex cable to the Haikou Video Demo adapter PCB. On the PCB the display connector has a small cover, open the cover, insert the flat-flex cable and close the cover.



Fig. 1.3: Connecting flat-flex cable to the Haikou Video Demo PCB

Step 3: Place the Haikou Video Demo adapter PCB on the PCB holder as shown below:

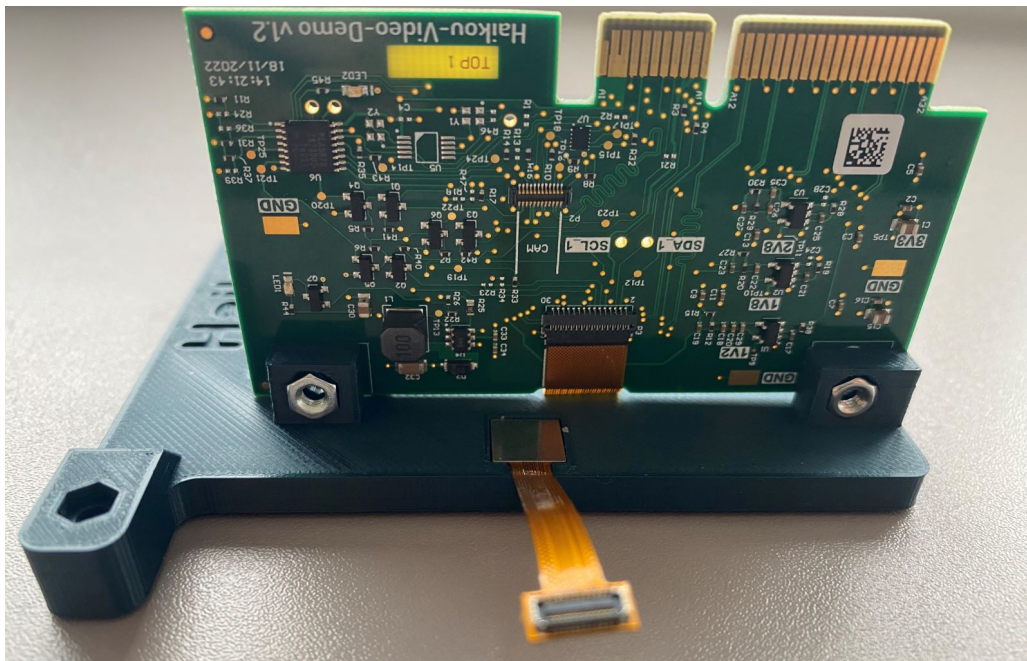


Fig. 1.4: Mounting Haikou Video Demo PCB to the display jig

Step 4: Screw two M3 screws through the PCB into the nuts in the jig. The board is now secured.

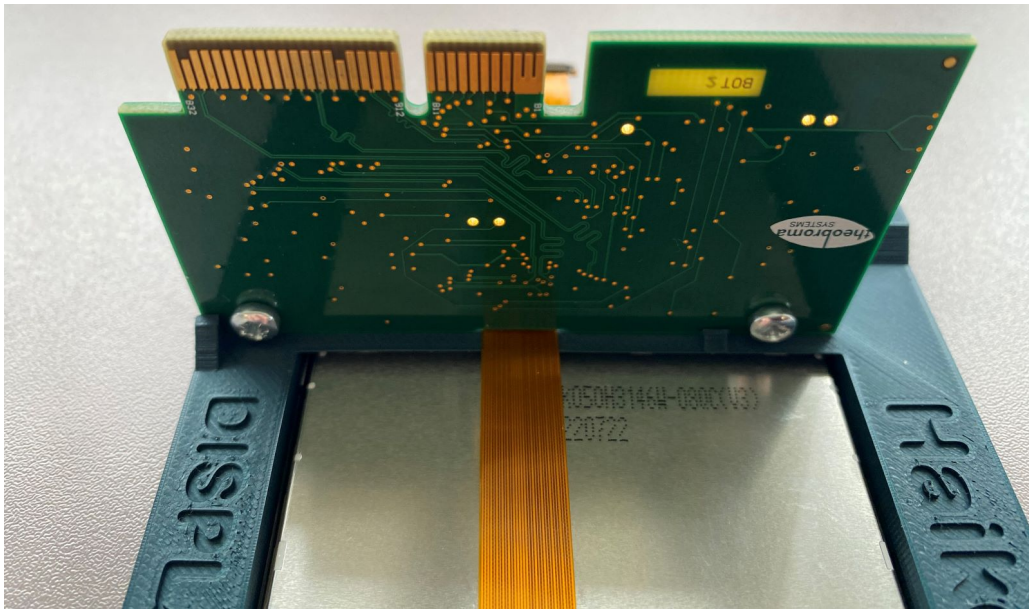


Fig. 1.5: Screwing Haikou Video Demo PCB to the display jig

Step 5: Connect the OV5675 Camera connector to the PCB connector as shown below:



Fig. 1.6: Connecting camera to the Haikou Video Demo PCB

Step 6: Mount the three hexagonal metal standoffs to the display jig. The display jig has three M3 threads sticking out. Screw the standoffs onto the threads with the M3 screws.

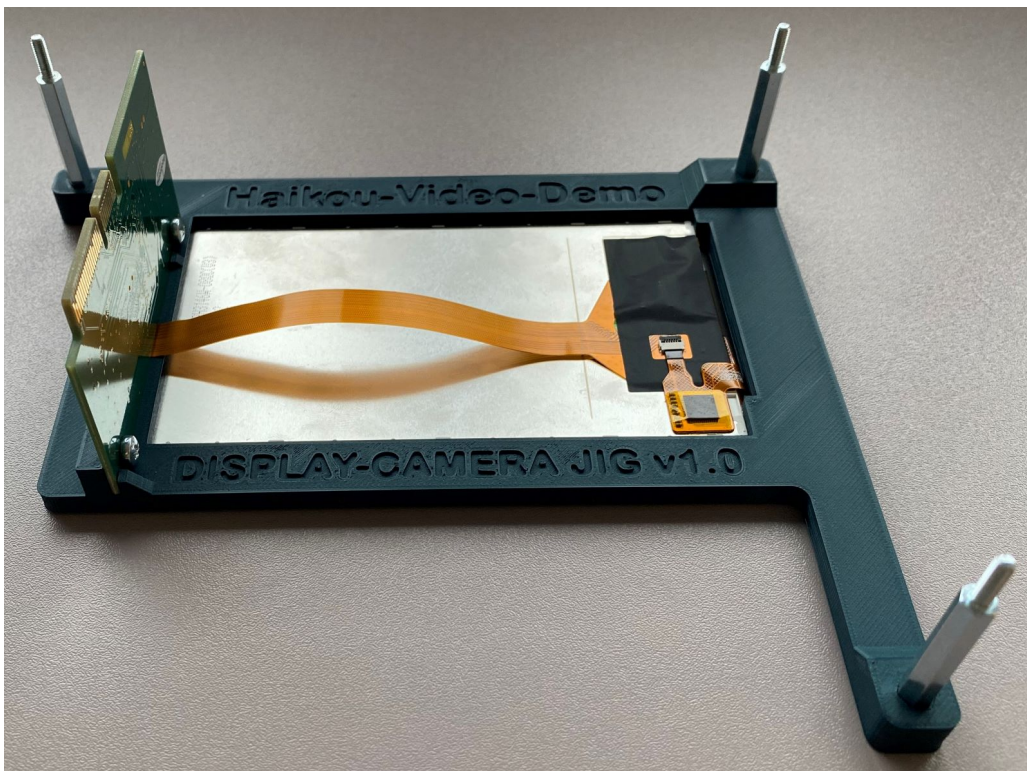


Fig. 1.7: Mounting three hexagonal metal standoffs



Fig. 1.8: Screwing standoffs to the jig

Step 7: Flip the display assembly face-up and hover it above the Haikou carrier board. Plug the Haikou Video Demo adapter PCB into the slot marked “Video connector” on the Haikou Development Kit. Slide the threads of the three metal standoffs into the mounting holes of the Haikou carrier board.



Fig. 1.9: Plug in the Haikou Video Demo on the Haikou carrier board

Step 8: Flip the whole setup (display assembly and Haikou carrier board) around to get access to the threads sticking through the mounting holes. Stick one M3 screw through the unoccupied hole. Now screw on the four M3 nuts that will serve as feet.

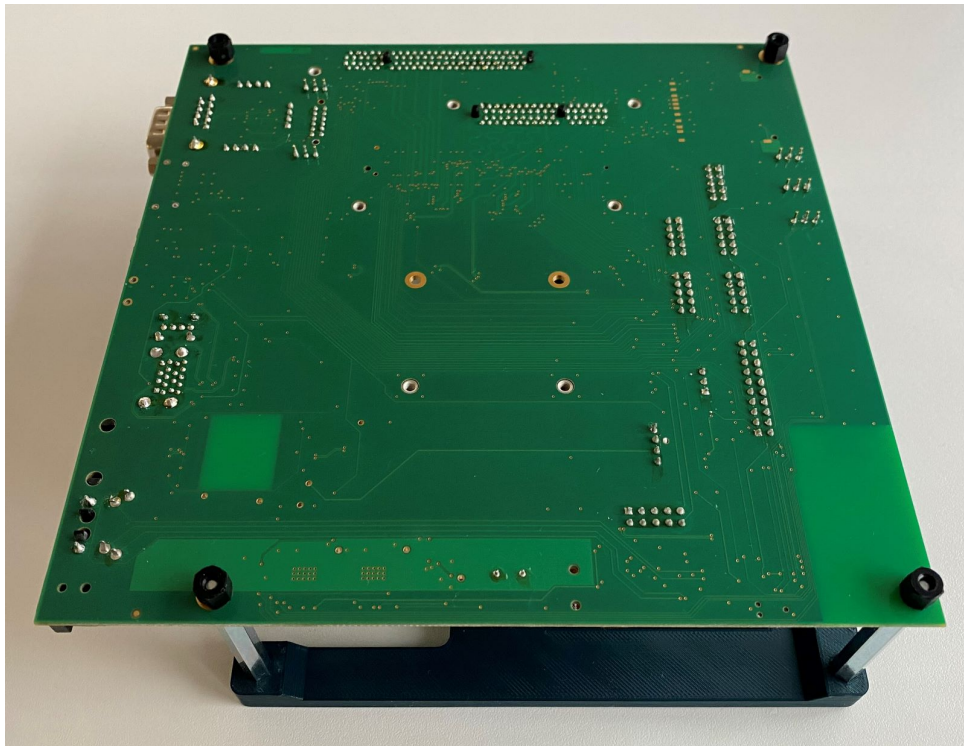


Fig. 1.10: M3 nuts that serve as feet for the Haikou carrier board

Step 9: Flip the setup around again. The result should look as seen in the photo.



Fig. 1.11: Complete setup

1.3 RINGNECK SOM-PX30-uQ7 Kernel Configuration

To enable display output to the Haikou Video Demo board, the correct Device Tree file must be used by Linux. The used file can be configured through the file `/boot/extlinux/extlinux.conf`.

For the RINGNECK SOM-PX30-uQ7 module, open the file with an editor (for example, nano):

```
nano /boot/extlinux/extlinux.conf
```

Replace the line:

```
FDT /boot/px30-ringneck-haikou.dtb
```

with the line:

```
FDT /boot/px30-ringneck-haikou-video-demo.dtb
```

Save the file and exit the editor.

Reboot the RINGNECK SOM-PX30-uQ7 module using:

```
reboot
```

Once the kernel has initialized the display controller (about 10 seconds into the boot), you should see the kernel boot messages on the display.

1.4 PUMA SOM-RK3399-Q7 Kernel Configuration

Note: Only our Linux kernel images based on v6.0.2 or later have support for Haikou Video Demo adapter.

To enable display output to the Haikou Video Demo board, the correct Device Tree file must be used by Linux. The used file can be configured through the file `/boot/extlinux/extlinux.conf`.

For the PUMA SOM-RK3399-Q7 module, open the file with an editor (for example, nano):

```
nano /boot/extlinux/extlinux.conf
```

Replace the line:

```
FDT /boot/rk3399-puma-haikou.dtb
```

with the line:

```
FDT /boot/rk3399-puma-haikou-video-demo.dtb
```

Save the file and exit the editor.

Reboot the PUMA SOM-RK3399-Q7 module using:

```
reboot
```

Once the kernel has initialized the display controller (about 10 seconds into the boot), you should see the kernel boot messages on the display.

2 Camera

The camera sensor is connected to the Image Signal Processor (ISP) block of the SoC. ISP is responsible for two main tasks:

- image correction and enhancement
- image statistics measurement

Based on the image statistics, 3A algorithms (auto exposure, auto focus, auto white balance) can control the camera sensor settings (e.g., gain and exposure) to provide the best image quality for the current lighting and scene.

To make use of the ISP and achieve this image control loop, a proper camera software stack is required. In this example, the libcamera stack will be used.

2.1 Libcamera installation on Debian

Libcamera package is not available in Debian Bullseye, so it needs to be compiled from source. Instructions how to build libcamera camera stack from source are available below.

1. Install build dependencies:

```
apt update
apt install -y git g++ cmake meson ninja-build pkg-config libyaml-dev \
python3-yaml python3-ply python3-jinja2 libssl-dev libdw-dev libudev-dev \
libevent-dev libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev
```

2. Install GStreamer and basic plugins:

```
apt-get install -y gstreamer1.0-tools gstreamer1.0-plugins-base \
gstreamer1.0-plugins-good gstreamer1.0-plugins-bad gstreamer1.0-gl
```

3. Download sources and build libcamera:

```
git clone -b v0.0.4 https://git.libcamera.org/libcamera/libcamera.git
cd libcamera
meson setup build -D prefix=/usr/ --libdir=lib/aarch64-linux-gnu \
-D pipelines=rkisp1 -D cam=enabled -D gstreamer=enabled
ninja -C build -j 6 install
```

4. Run the following command to confirm the libcamera stack was correctly installed on your system:

```
cam -l
```

The above command should list available cameras. The output should be similar to this:

```
root@puma-debos:~# cam -l
[0:18:12.403466748] [7679] INFO Camera camera_manager.cpp:299 libcamera v0.0.4
Available cameras:
1: Internal front camera (/base/i2c@ff110000/camera@36)
```

For more details about libcamera dependencies and build process, you can visit [libcamera Getting Started webpage](#).

2.2 Camera examples using GStreamer

Libcamera provides a GStreamer plugin (libcamerasrc) that can be used as a video source in the GStreamer pipeline.

2.2.1 Camera preview on display

The following command can be used to show a camera preview on the Haikou-Video-Demo display:

```
gst-launch-1.0 libcamerasrc ! video/x-raw, colorimetry=bt709, format=NV12, \
  width=960, height=720, framerate=29/1 ! queue max-size-buffers=16 ! \
  glimagesink
```

2.2.2 Camera preview via network stream

The following command streams the camera images via RTP. Replace 10.11.12.13 with the actual IP address of your PC.

```
gst-launch-1.0 libcamerasrc ! \
  video/x-raw,format=NV12,width=640,height=480,framerate=25/1 ! \
  jpegenc ! rtpjpegpay ! udpsink host=10.11.12.13 port=5202
```

On your PC, receive and display the network stream using the following command:

```
gst-launch-1.0 udpsrc port=5202 ! \
  application/x-rtp,encoding-name=JPEG,payload=26 ! \
  rtpjpegdepay ! jpegdec ! autovideosink
```

2.2.3 Still image capture

GStreamer can be used to capture a raw image from camera and convert it to a more convenient format.

Firstly, capture a raw frame in raw NV12 format. Libcamera algorithms usually need 1-2s of input data to calculate optimum values and stabilize the image. Because of that, we will skip the first few seconds of the stream and save only the last frame:

```
timeout -s INT 5s gst-launch-1.0 libcamerasrc ! video/x-raw, colorimetry=bt709, \
  format=NV12, width=1920, height=1440, framerate=29/1 ! \
  multifilesink location=/tmp/frame.nv12 max-files=1
```

The output raw frame will be saved to the /tmp/frame.nv12.

The raw NV12 frame can be converted to PNG using the following command:

```
gst-launch-1.0 filesrc location=/tmp/frame.nv12 ! videoparse format=nv12 \
  width=1920 height=1440 framerate=0/1 ! videoconvert ! pngenc ! \
  filesink location=frame.png
```

The output PNG file will be saved to frame.png in the current directory.

2.3 Known issues

1. There is a bug in libcamera that limits the maximum output resolution to 1920x1920. There is already available patch with the fix, but still in the libcamera review process. Bug report: https://bugs.libcamera.org/show_bug.cgi?id=180.
2. Auto-focus support is currently missing in the libcamera, but there are works in progress to include it. Patch series with auto-focus support: <https://lists.libcamera.org/pipermail/libcamera-devel/2023-March/037250.html>.

3 Contact

Cherry Embedded Solutions GmbH
Seestadtstraße 27
1220 Vienna
Austria

Inquiries: sales-es@cherry.de
Technical Support: support-es@cherry.de

4 Revision History

Date	Revision	Changes
Jan 17, 2023	v0.0.1	First internal release
Jan 27, 2023	v0.0.2	Change document name, add preassembled part
Feb 03, 2023	v0.0.3	Some rewording, typo fixing and support for RK3399-Q7 Puma module
Mar 23, 2023	v1.0.0	Add camera section with libcamera build instructions and gstreamer examples
Mar 23, 2023	v1.1.0	Add network stream instructions
Dec 15, 2023	v1.1.1	recalled version
Feb 16, 2024	v1.1.2	add contact information
Mar 01, 2024	v1.2.0	rename to new product name DEVKIT ADDON CAM-TS-A01 change document type from appnote to user manual
Apr 02, 2024	v1.3.0	Theobroma Systems is now CHERRY Embedded Systems