Introduction

The E412 is a compact H.264 encoder. It is a second generation encoder of the E12/E110/E112 encoders. The main differences are:

- fast boot up: only 2 seconds
- smaller and lower power consumption
- flexible: add your host system for stream formatting
  - a compact host system in E412 form factor is planned (it will install below E412)

Key features

- video input: standard size HDMI connector
- video resolution up to 1080p30 or 1080p60 (depending on the firmware loaded into the E412)
- video codec: H.264 (base, main and high profile)
- only progressive video formats are supported (no interlaced video input)
- audio: PCM or compressed audio
- micro USB connector to connect to host system (with UVC driver)
- alternatively 2x2 2mm pin header to host system (on the bottom side)
- host system receives the compressed video stream and formats it with GStreamer or similar
- any protocol support through GStreamer (RTSP, RTMP, HLS, UDP or more)
- control of encoding parameters and firmware upgrade through MXCAM utility
- power consumption: 0.5 watt (with 1080p30 firmware) to 1 watt (with 1080p60 firmware)
- size: 22 x 64 mm

Host software

Auvidea provides the binary versions of the software below. Please find a list of host systems supported in the list below. If your host system is not on the list, please contact us for the source code of the software and its licensing details.

- optimised UVC driver for best performance
- MXCAM: control interface to the E412 module (including encoder firmware upgrade)
- QUERYDUMP: get detailed encoding parameters and statistics

Host systems

- x86 system with Ubuntu 14.04 LTS
- Raspberry Pi
Micro Controller Firmware

Release 1.0.2 (November 2016)

- firmware upgrade via UART2 interface not supported yet
- no I2C1 interface between Max64380 and micro controller (get status via UART2)
Encoder Firmware

Release 4.3.27 (December 2016)

- support for audio (audio=external)
- 8, 16, 48 kHz audio sampling rate supported

Please let us know, if you do require other audio sample rates (such as 44.1kHz).

Firmware Images

The are 3 firmware images provided. fw-64380.img is the default image. The firmware may be changed and/or upgraded with the MXCAM app. Please choose the default firmware unless you need to support 1080p50 or 1080p60 encoding. Please note, that the other firmware sets a higher clock frequency and doubles the power dissipation. This may require active cooling of the module.

Please use the QUERYDUMP app to monitor the clock frequencies set.

<table>
<thead>
<tr>
<th>firmware</th>
<th>MEM CLK</th>
<th>AVC CLK</th>
<th>max. resolution</th>
<th>typ. power</th>
</tr>
</thead>
<tbody>
<tr>
<td>fw-64380</td>
<td>200MHz</td>
<td>150MHz</td>
<td>1080p30</td>
<td>0.5 watts</td>
</tr>
<tr>
<td>fw-64380-fast</td>
<td>333MHz</td>
<td>200MHz</td>
<td>1080p60</td>
<td>1 watt</td>
</tr>
<tr>
<td>fw-64380-400MHz</td>
<td>400MHz</td>
<td>225MHz</td>
<td>1080p60</td>
<td>1 watt</td>
</tr>
</tbody>
</table>

 Below is a partial printout of QUERYDUMP with the default (200MHz) firmware image loaded.

$ sudo querydump

. . .

Clock rates

XIN : 240000000
PLL0 : 3000000000
PLL1 : 240000000
PLL2 : 24576000
PLL3 : 399500000
PLL4 : 0
VS : 149812500
VPU : 0
QMM : 150000000
AVC : 150000000
PME : 0
MPEG2 : 0
JPEG : 100000000
AUD0 : 12288000
AUD1 : 8192000
V0PixClk : 240000000
V1PixClk : 0
V2PixClk : 240000000
MemClk : 199750000

. . .
The E412 supports various input timings (video resolutions):

active: the visible resolution of the video (horizontal, vertical)
total: the total resolution of the video (including front and back porch)
e.g. 1650 x 750 x 60 = 74,250,000 Hz = pixel frequency

<table>
<thead>
<tr>
<th>input timing</th>
<th>resolution</th>
<th>vic</th>
<th>active</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>576p50</td>
<td>720x576 p (50 Hz)</td>
<td>17</td>
<td>[720,576]</td>
<td>[864,625]</td>
</tr>
<tr>
<td>720p50</td>
<td>1280x720 p (50 Hz)</td>
<td>19</td>
<td>[1280,720]</td>
<td>[1980,750]</td>
</tr>
<tr>
<td>720p60</td>
<td>1280x720 p (60 Hz)</td>
<td>4</td>
<td>[1280,720]</td>
<td>[1650,750]</td>
</tr>
<tr>
<td>1080p24</td>
<td>1920x1080 p (24 Hz)</td>
<td>32</td>
<td>[1920,1080]</td>
<td>[2750,1125]</td>
</tr>
<tr>
<td>1080p25</td>
<td>1920x1080 p (25 Hz)</td>
<td>33</td>
<td>[1920,1080]</td>
<td>[2640,1125]</td>
</tr>
<tr>
<td>1080p30</td>
<td>1920x1080 p (30 Hz)</td>
<td>34</td>
<td>[1920,1080]</td>
<td>[2200,1125]</td>
</tr>
<tr>
<td>1080p50</td>
<td>1920x1080 p (50 Hz)</td>
<td>31</td>
<td>[1920,1080]</td>
<td>[2640,1125]</td>
</tr>
<tr>
<td>1080p60</td>
<td>1920x1080 p (60 Hz)</td>
<td>16</td>
<td>[1920,1080]</td>
<td>[2200,1125]</td>
</tr>
</tbody>
</table>

AUVIDEA GMBH
TECHNICAL REFERENCE MANUAL 7
Custom UVC driver

The E412 requires a custom UVC driver to be installed to get the best performance (frame rate).

Basic functionality is provided by the standard UVC driver, but the frame rate may be limited. Below are the DMESG messages which shows the activation of the standard UVC driver.

```bash
$ dmesg
[ 5.871950] uvcvideo: Found UVC 1.10 device MAX64380 (0b6a:4d52)
[ 5.884253] input: MAX64380 as /devices/platform/soc/3f980000.usb/usb1/1-1/1-1.4/1-1.4:1.0/input/input3
[ 5.884849] uvcvideo: Found UVC 1.10 device MAX64380 (0b6a:4d52)
[ 5.896172] uvcvideo: Unable to create debugfs 1-6 directory.
[ 5.896998] input: MAX64380 as /devices/platform/soc/3f980000.usb/usb1/1-1/1-1.4/1-1.4:1.2/input/input4
[ 5.896172] uvcvideo: Unable to create debugfs 1-6 directory.
[ 5.896998] input: MAX64380 as /devices/platform/soc/3f980000.usb/usb1/1-1/1-1.4/1-1.4:1.2/input/input4
[ 5.897476] usbcore: registered new interface driver uvcvideo
[ 5.897489] USB Video Class driver (1.1.1)
```
Control and configuration of the encoder

MXCAM communicates with the Max64380 encoder firmware on the E412.

Complete list of MXCAM subcommands

```
sudo ./mxcam
Usage: mxcam <subcommand> [options] [arguments]
Type 'mxcam help <subcommand>' for help on a specific subcommand

Available subcommands:
  list      : lists all compatible devices
  boot      : boot the camera with specified images
  flash     : flash the specified image(s) on camera
  bootmode  : display or change the existing bootmode
  getkey    : read the value stored on camera for the given key
  getccr    : read the ccr details from camera
  setkey    : write the key and value on camera
  removekey : remove the key from config area
  erase     : erase the config area on camera
  info      : prints the information about the camera from usb boot/snor
  reset     : reset the camera
  gpiowr    : gpio write
  gpiord    : gpio read
  help      : describe the usage of mxcam or subcommands
  seteptype : set the end point of camera to the specified type
  i2cwrite  : program i2c device on camera
  i2cread   : read i2c device from camera
  tcw       : read/write spi device's timing control word value of camera
  isp       : read/write ISP register
  version   : display the version of mxcam
  usbstest  : set USB test mode
  qcc       : read/write QCC registers
  whoami    : informations about maxim camera and supported commands
  memtest   : DDR memory test
  pwm       : read/configure pwm

Global options:
  --poll    : wait until ANY maxim device is connected
  --device  : select device with device-id
  --bus     : select device on bus X with addr Y (--addr required)
  --addr    : select device on bus X with addr Y (--bus required)
  --verbose : display verbose messages
  --oldcam  : communicate with old generation Maxim cameras
```
<table>
<thead>
<tr>
<th>CCR</th>
<th>Key</th>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BOOTMODE</td>
<td>usb, snor</td>
<td>boot mode of camera</td>
</tr>
<tr>
<td>2</td>
<td>BOARD</td>
<td>raptorbub</td>
<td>Board type</td>
</tr>
<tr>
<td></td>
<td></td>
<td>raptorcam1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>raptorcam2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>raptorcam4</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>NVPP_SCALE</td>
<td>0, 1</td>
<td>Selects whether to use single LVPP + 2 NVPPs in datapath</td>
</tr>
<tr>
<td>4</td>
<td>NVPP_VID_PASSTHROUGH</td>
<td>0, 1</td>
<td>Selects whether to set NVPP video to pass through mode</td>
</tr>
<tr>
<td>5</td>
<td>USB_AUDIO_SYNC</td>
<td>0, 1</td>
<td>ST/Link-2 programming interface of the micro controller (data)</td>
</tr>
<tr>
<td>6</td>
<td>BF</td>
<td>0, 1, 2</td>
<td>Type of beamformer: 0: Disable beamformer</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1: Beamformer template</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2: 4mic</td>
</tr>
<tr>
<td>7</td>
<td>beamformer</td>
<td>0 to 10 (3)</td>
<td>Thirdparty beamformer etc.</td>
</tr>
<tr>
<td>8</td>
<td>NRL</td>
<td>0</td>
<td>Value of Noise reduction level [-MAXINT to +MAXINT]</td>
</tr>
<tr>
<td>9</td>
<td>NPA</td>
<td>0 to 10 (0)</td>
<td>Type of Noise processing algorithm</td>
</tr>
<tr>
<td>10</td>
<td>BW</td>
<td>70</td>
<td>Value of Beam width [-MAXINT to +MAXINT]</td>
</tr>
<tr>
<td>11</td>
<td>MS</td>
<td>70</td>
<td>Value of Mic spacing [-MAXINT to +MAXINT]</td>
</tr>
<tr>
<td>12</td>
<td>CUS1</td>
<td>1</td>
<td>Custom 1 [-MAXINT to +MAXINT]</td>
</tr>
<tr>
<td>13</td>
<td>CUS2</td>
<td>7</td>
<td>Custom 2 [-MAXINT to +MAXINT]</td>
</tr>
<tr>
<td>14</td>
<td>CUS3</td>
<td>0</td>
<td>Custom 3 [-MAXINT to +MAXINT]</td>
</tr>
<tr>
<td>15</td>
<td>AGC</td>
<td>0, 1</td>
<td>Automatic gain control enable</td>
</tr>
<tr>
<td>16</td>
<td>PT</td>
<td>3</td>
<td>Peak threshold in dB [0, MAXINT]</td>
</tr>
<tr>
<td>17</td>
<td>AT</td>
<td>22</td>
<td>Average target power in dB [0, MAXINT]</td>
</tr>
<tr>
<td>18</td>
<td>AEC</td>
<td>0, 1</td>
<td>Acoustic echo cancellation enable</td>
</tr>
<tr>
<td>19</td>
<td>NR</td>
<td>0, 1</td>
<td>Enable noise reduction</td>
</tr>
<tr>
<td>20</td>
<td>NS</td>
<td>20</td>
<td>Noise reduction suppression level in dB [1 to 30]</td>
</tr>
<tr>
<td>21</td>
<td>MAXRES_INPUT</td>
<td>1080p60</td>
<td>Maximum support resolution for video input</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[qvga</td>
<td>wqvga</td>
</tr>
<tr>
<td>22</td>
<td>MAXRES_MAIN</td>
<td>1080p60</td>
<td>Maximum supported resolution for main channel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[qvga</td>
<td>wqvga</td>
</tr>
<tr>
<td>23</td>
<td>MAXRES_PV</td>
<td>auto</td>
<td>Maximum supported resolution for preview channel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[qvga</td>
<td>wqvga</td>
</tr>
<tr>
<td>24</td>
<td>MAXRES_LVPP</td>
<td>1080p60</td>
<td>Maximum supported resolution for LVPP (live input module)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[qvga</td>
<td>wqvga</td>
</tr>
<tr>
<td>25</td>
<td>VRES_VPP</td>
<td>auto</td>
<td>Resolution for NVPP channels, in descending order</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[qvga</td>
<td>wqvga</td>
</tr>
<tr>
<td>26</td>
<td>VRES_AVC</td>
<td>auto</td>
<td>Resolution for AVC encoders</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[qvga</td>
<td>wqvga</td>
</tr>
<tr>
<td>27</td>
<td>VRES_MJPEG</td>
<td>auto</td>
<td>Resolution for MJPEG encoders</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[qvga</td>
<td>wqvga</td>
</tr>
<tr>
<td>Key</td>
<td>Options</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>VRES_RAW</td>
<td>auto</td>
<td>Resolution for YUV/RAW encoder. Only one allowed. {none</td>
<td>qvga</td>
</tr>
<tr>
<td>LENSSHADING</td>
<td>internal</td>
<td>Select where lens shading correction will be performed. {off</td>
<td>internal</td>
</tr>
<tr>
<td>SENSOR</td>
<td>auto</td>
<td>Sensor type {auto</td>
<td>none}</td>
</tr>
<tr>
<td>SENSORVOLTAGE</td>
<td>2.8V</td>
<td>Indicates the AVDD of sensor input. {2.8v</td>
<td>3.3v}</td>
</tr>
<tr>
<td>SENSOREXP</td>
<td>default</td>
<td>Initial sensor exposure setting</td>
<td></td>
</tr>
<tr>
<td>SENSORMAXAGAIN</td>
<td>auto</td>
<td>Set maximum sensor analog gain</td>
<td></td>
</tr>
<tr>
<td>SENSORFLIP</td>
<td>0, 1, 2, 3</td>
<td>Set the sensor horizontal/vertical flip. (bit 1: vflip, bit 0: hflip)</td>
<td></td>
</tr>
<tr>
<td>AWB</td>
<td>weighted</td>
<td>AWB algorithm select {gray</td>
<td>weighted</td>
</tr>
<tr>
<td>LENS</td>
<td>default</td>
<td>Lens type for a corresponding sensor</td>
<td></td>
</tr>
<tr>
<td>RADSHPADNG</td>
<td>auto</td>
<td>Radial shading parameters. [&quot;#,hiTemp,loTemp,data,…&quot;]</td>
<td></td>
</tr>
<tr>
<td>HISTOEQ</td>
<td>0, 1</td>
<td>Histogram equalization enable</td>
<td></td>
</tr>
<tr>
<td>AWBPROBPEAK</td>
<td>auto</td>
<td>Specifies a point to create a probability peak {gr,gb}</td>
<td></td>
</tr>
<tr>
<td>SENSORPWR DWNMODE</td>
<td>0, 1</td>
<td>GPIO pin polarity to power down sensor</td>
<td></td>
</tr>
<tr>
<td>NIGHTGPIO</td>
<td>-1</td>
<td>GPIO pin and polarity to indicate night mode (BytePolarity</td>
<td>GPIOn)</td>
</tr>
<tr>
<td>ADAPTCCM</td>
<td>auto</td>
<td>Adaptive CCM. [&quot;#,hiTemp,loTemp,data,…&quot;]</td>
<td></td>
</tr>
<tr>
<td>POWERFREQ</td>
<td>50, 60</td>
<td>Power line frequency</td>
<td></td>
</tr>
<tr>
<td>ADC</td>
<td>auto</td>
<td>Specifies ADC module. [WM8750</td>
<td>WM8737</td>
</tr>
<tr>
<td>AUDCLK</td>
<td>external</td>
<td>I2S audio mode must be set to external for E412 (I2S slave mode)</td>
<td></td>
</tr>
<tr>
<td>ADC_MASTER</td>
<td>auto</td>
<td>Choose which ADC port to be master. {0</td>
<td>1</td>
</tr>
<tr>
<td>ADCMAXVOL</td>
<td>auto</td>
<td>Specifies the max volume setting in ADC (in dB)</td>
<td></td>
</tr>
<tr>
<td>AIN0_INPUT</td>
<td>0, 1</td>
<td>the audio input port to which Max64380 audio input 0 has to connect</td>
<td></td>
</tr>
<tr>
<td>AIN1_INPUT</td>
<td>0, 1</td>
<td>the audio input port to which Max64380 audio input 1 has to connect</td>
<td></td>
</tr>
<tr>
<td>AEC_DELAY</td>
<td>4</td>
<td>specifies the window length within which the echo cancellation is done. Should be power of 2. Window length for echo cancellation is = (AEC_DELAY) x (no:of samples per frame)</td>
<td></td>
</tr>
<tr>
<td>MAPPEN</td>
<td>0, 1</td>
<td>Control mobiapp support</td>
<td></td>
</tr>
<tr>
<td>RAWDUMP_RES</td>
<td>auto</td>
<td>Set raw dump resolution {width,height}</td>
<td></td>
</tr>
<tr>
<td>WATCHDOG</td>
<td>0, 1</td>
<td>HW Watchdog Enable</td>
<td></td>
</tr>
<tr>
<td>CAMERA_MODE</td>
<td>SKYPE_BULK</td>
<td>camera mode {IPCAM</td>
<td>SECS</td>
</tr>
<tr>
<td>3D_MODE</td>
<td>0, 1</td>
<td>3D mode</td>
<td></td>
</tr>
<tr>
<td>AUDIO_INTERVAL</td>
<td>1, 2, 3, 4</td>
<td>Polling interval for audio endpoint</td>
<td></td>
</tr>
</tbody>
</table>
$sudo ./mxcam getccr
-------------------------------
MAXIM AREA CCR LIST
key, default, description
-------------------------------
key count   : 55

List the keys configured

$sudo ./mxcam getkey
BOOTMODE=snor
BF=1
CUS1=1
CUS2=7
MS=70
BW=70
BOARD=raptorcam2
CAMERA_MODE=SKYPE_BULK
VIDEO_INPUT=YUV1_16
MAXRES_INPUT=1080p60
MAXRES_LVPP=1080p60
MAXRES_MAIN=1080p60
HSTART=190
VSTART=36
HSEL=1
LSEL=0
VIDEOCONTROL=80380828
NVPP_VID_PASSTHROUGH=1
AUDCLK=external

key count   : 19
config size : 262/1024 bytes
Get statistics of the encoder

$sudo ./querydump
Error opening the file for Query Dump....
Outputting to the Console
Core: Retrieved 0x405c0bcc as GPB

vinh is attached type = 2
lvpp0 is attached type = 3
lvpp1 is attached type = 3
avcenc0 is attached type = 5
avcenc1 is attached type = 5
jpegenc1 is attached type = 13
audfil0 is attached type = 16
audfil1 is attached type = 16
audfil2 is attached type = 16
audenc0 is attached type = 11

System Control : ID 0
Heartbeat : 319724
Dropped Events: 0
Event Handler State:
  Event Queue Pointers: Read 4 / Write 5
  Event Pending: 0
  Event Handler Load Average : 0% ints/s 0 events/s 0
Codec load: Enc HW 42% (MB 42%) Enc SW 5% Dec HW 0% Dec SW 0%
Audio load: Enc 0% Dec 0%
Mux load: 0%
USB load: 0%
Allocated pool memory : 10858960/127918080
Heap 0 Utilization : 932480/2097152
Heap 1 Utilization : 568440/2097152

VIN0:
  INT: now:0us max:0us n:18736
  ISR: min:307us avg:567us max:1334us hwload:72.61% swload:3.32% dbg:0
  Interleave: twsync:0x0000 syncerrors:0 nullframes:0
  Input status: 0x1 (top vlines=1 hwidth=0 )

VIN1:
  INT: now:3999us max:230213us n:113350
  ISR: min:865us avg:996us max:1561us hwload:0.00% swload:0.00% dbg:0
  Interleave: twsync:0x0000 syncerrors:0 nullframes:0
  Input status: 0x0 (top vlines=0 hwidth=0 )

AIN0:
  INT: now:10034us max:10069us n:2411
  ISR: min:80us avg:302us max:886us dbg:4
  framesCaptured:2410 captureFailures:2 captureOverflow:0
  SamplesPerFrame:480 RampTime:10us

AIN1:
  INT: now:0us max:0us n:0
  ISR: min:0us avg:0us max:0us dbg:0
  framesCaptured:0 captureFailures:0 captureOverflow:0
  SamplesPerFrame:0 RampTime:0us

Memory bandwidth (bytes/sec) over the last 5040ms:

<table>
<thead>
<tr>
<th></th>
<th>sys qmm read: 2696238</th>
<th>sys qmm write: 2732745</th>
<th>vid qmm read: 3437740</th>
<th>vid qmm write: 299570</th>
<th>vid qmm read: 1041898</th>
<th>vid qmm write: 4351502</th>
</tr>
</thead>
<tbody>
<tr>
<td>ahb per read:</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ahb per write:</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vpp0 read :</td>
<td>4325070</td>
<td>vpp0 write : 4359095</td>
<td>fbr0 : 4359095</td>
<td>fbr1 : 0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>fbw :</td>
<td>0</td>
<td>avc reade 0 : 4922596</td>
<td>avc addd 1 : 2772273</td>
<td>avc db : 2668856</td>
<td></td>
<td></td>
</tr>
<tr>
<td>all 0 :</td>
<td>776677</td>
<td>all 1 : 0</td>
<td>jpeg read: 0</td>
<td>jpeg write : 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sbc read :</td>
<td>0</td>
<td>sbc write : 0</td>
<td>spi read : 0</td>
<td>spi write : 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sbc read :</td>
<td>0</td>
<td>sbc write : 0</td>
<td>spi read : 0</td>
<td>spi write : 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bsr read :</td>
<td>0</td>
<td>bsr write : 0</td>
<td>avc bits : 165062</td>
<td>avc stat : 5070463</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total: 43978880 bytes/sec

Clock rates

| XIN          | 240000000 |
| PLL0         | 3000000000 |
| PLL1         | 2400000000 |
| PLL2         | 2457600000 |
| PLL3         | 3995000000 |
| PLL4         | 0         |
| VS           | 1498125000 |
| VPU          | 0         |
| QMM          | 1500000000 |
| AVC          | 1500000000 |
| PHE          | 0         |
VERSION 1.1
E412

MPEG2: 0
JPEG: 100000000
AUD0: 12288000
AUD1: 8192000
V0PixClk: 24000000
V1PixClk: 0
V2PixClk: 24000000
MemClk: 199750000

UVC statistics
- Audio frames - queued: 2000
- Audio frames - transferred: 37
- Audio frames - transfer error: 0
- Audio frames - dropped: 1946
- Audio frames - max time between queueing (us): 11731
- Audio frames - min time between queueing (us): 8273
- Audio frames - max time between transfer (us): 12916
- Audio frames - min time between transfer (us): 7988

LVPP: ID 1 "vinh"
- Frames Captured: 13350
- Capture Failures: 0
- Capture Overflow: 0
- A/V Synch Drops: 0
- A/V Synch Repeats: 0
- Input Drops: 0
- Output PTS: 0x0013188cc
- Destination Size: 1920x1088
- Tamper frame number: 0
- Tamper Defocus Comp:Thd: 0
- Tamper FieldofView Comp:Thd: 0
- Tamper Block Comp:Thd: 0
- Temporal Filter Strength: 0
- Horizontal Filter Level: 0

NVPP: ID 2 "lvpp0"
- Frame transit time min:14302 avg:14350 max:15043
- Frames Captured: 5387
- Capture Failures: 0
- Capture Overflow: 0
- A/V Synch Drops: 0
- A/V Synch Repeats: 0
- Input Drops: 0
- Output PTS: 0x0038c5d24
- Destination Size: 1280x720

AVC Encoder: ID 4 "avcenc0"
- Frames Encoded: 5363
- Performance (IPB): 18575/20431/0
- PerformanceJitter (IPB): 1/5362/0
- Latency: 662130644 us
- Buffer Fullness: 1931 / 6500000 (max 1080156)
- Base QP: 12
- State: 0
- Frames Input: 5387
- Input frame drops: 0
- Rate control drops: 23
- Last RC drop cause: 4
- Last RC drop pic type: 0
- Last RC drop pic size: 73945
- Last RC drop pic QP: 14
- Last RC drop pic Edge: 23432
- Telecine field drops: 0
- Malus exceed limit: 0
- Complexity RC override: 0
- Last block ready addr: 0x0
- Last block ready size: 0x0
- Last block done addr: 0x0
- Last block done size: 0x0
- Current PTS: 0x0038c5d24
- Y SNR: 45.12
- Ref Pool Used: 2
- Recon Preview Drops: 0
- Frame output delta E: 16152/161202 us (Min/Max)
- Frame output delta S: -1/0 us (Min/Max)
- Frame signature: 62856826
- Current bitrate: 378.570862 kb/s
- Frame Rate: 25.00

AVC Encoder: ID 5 "avcenc1"
- Frames Encoded: 0
- Performance (IPB): 0/0/0
PerformanceJitter (IPB) : 0/0/0
Latency : 0 us
Buffer Fullness : 0 / 0 (max 0)
Base QP : 0
State : Idle
Frames Input : 0
Input frame drops : 0
Rate control drops : 0
Last RC drop cause : 0
Last RC drop pic type : 0
Last RC drop pic size : 0
Last RC drop pic QP : 0
Last RC drop pic Edge : 0
Telecine field drops : 0
Nalus exceed limit : 0
Complexity RC override: 0
Last block ready addr : 0x0
Last block ready size : 0x0
Last block done addr : 0x0
Last block done size : 0x0
Current PTS : 0x0000000000000000
Y SNR: 50.00
Ref Pool Used : 0
Recon Preview Drops : 0
Frame output delta E : 0/0 us (Min/Max)
Frame output delta S : 0/0 us (Min/Max)
Frame signature : 0
Current bitrate : 0.000000 kb/s
Frame Rate : 0.00

JPEGENC : ID 6 "jpegenc1"
Encoder index : 0
Encoder state : Idle(0)
In DATA tokens dequeued : 0
Frames encoded : 0
Last quality factor : 0
Min quality factor allowed : 0
Max quality factor allowed : 0
Last frame size (bytes) : 0 [min: 0 max: 0]
Total encoded frames size (bytes) : 0
Frames encoded since last reset: 0
Encoded frames size since last reset: 0
Current bitrate (bits/s): 0
Current PTS : 0x0000000000000000
Token standby time (us) : 0
Software setup time (us) : 0
Hardware encode time (us) : 0
Input data que occupancy : 0/0
Input data que overflows : 0
Output buffer fullness (bytes) : 0/0
Downstream overflows (current)/(total) : 0/0
Encoder status: Notified.
Mastser ISR status: Notified.

AUDFLTR : ID 7 "audfil0"
Filter stream ID : 0
Filter state : Idle (0)
Last received command/interrupt : 18
Input ports : 0
Output ports : 1
Input frames received in current round : 0
Rounds of input frames received from src channels : 0
Output frames sent in current round : 0
Rounds of output frames sent to output ports : 0
Output buffer overflow count (current) : 0
Output buffer overflow count (total) : 0
Input buffer occupancy: (0/0)
Output buffer occupancy: (0/0)
Audio Sync: Delta 0 Resample Ratio 0x0 (0 ppm)
Computation latency: 0 us

AUDFLTR : ID 8 "audfil1"
Filter stream ID : 1
Filter state : Waiting for input frames (1)
Last received command/interrupt : 3
Input ports : 1
Output ports : 1
Input frames received in current round : 0
Rounds of input frames received from src channels : 30649
Output frames sent in current round : 1
Rounds of output frames sent to output ports : 30649
Output buffer overflow count (current) : 0
Output buffer overflow count (total) : 0
Input buffer occupancy: (0/128)
Output buffer occupancy: (0/32)
Audio Sync: Delta 0 Resample Ratio 0x0 (0 ppm)
Computation latency: 1620 us

AUDFLTR : ID 9 "audfil2"
Filter stream ID : 2
Filter state : Idle (0)
Last received command/interrupt : 18
Input ports : 0
Output ports : 1
Input frames received in current round : 0
Rounds of input frames received from src channels : 0
Output frames sent in current round : 0
Rounds of output frames sent to output ports : 0
Output buffer overflow count (current) : 0
Output buffer overflow count (total) : 0
Input buffer occupancy: (0/0)
Output buffer occupancy: (0/0)
Audio Sync: Delta 0 Resample Ratio 0x0 (0 ppm)
Computation latency: 0 us
Audio Encoder : ID 10 "audenc1"
Frames Encoded : 0
State : Idle
Latency : 0 us
Buffer Fullness : 0 / 0
Frames Input : 0
Input frame drops : 0
Current PTS: : 0x00000000
Frame output delta E : 0/0 us (Min/Max)
Frame output delta S : 0/0 us (Min/Max)
Current bitrate : 0.000000 kb/s

usb mem interface World End
**Connectors**

**HDMI input (J1)**

Please connect an non HDCP video source with progressive timing. Interlaced video is not supported. At startup the onboard micro controller inits the HDMI input chip and programs the EDID. Once this has completed the HDP pin (19) is raised high, to flag to the HDMI source connected, that the receiver is ready to receive a signal and that the EDID may now be read.

**Micro USB 2.0 (J4)**

This USB port puts out the H.264 encoded video stream and allows control and setup of the encoder and the HDMI input chip via tunnelled I2C. As the USB port does not transfer any raw uncompressed data but only compressed H.264 video data, the bandwidth of USB 2.0 (480Mb/s max.) is more than sufficient. J6 is an optional connector to bridge USB 2.0 from board to board.

<table>
<thead>
<tr>
<th>Pin</th>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5V</td>
<td>5V power input of the E412 module (100mA typical)</td>
</tr>
<tr>
<td>2</td>
<td>USB-D-</td>
<td>USB 2.0 data</td>
</tr>
<tr>
<td>3</td>
<td>USB-D+</td>
<td>USB 2.0 data</td>
</tr>
<tr>
<td>4</td>
<td>USB_ID</td>
<td>not connected</td>
</tr>
<tr>
<td>5</td>
<td>GND</td>
<td>Ground</td>
</tr>
</tbody>
</table>
**UART2 (J5)**

This UART2 port is the console interface to the on-board micro controllers. The commands are described in a section below. Commands allow to setup the system and to retrieve the status like the video properties of the HDMI video source connected. The baud rate settings of UART2 are 115200 8/N/1. Please use a USB to TTL UART converter. Swap RxD and TxD when connecting this interface.

<table>
<thead>
<tr>
<th>Pin</th>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.3V</td>
<td>3.3V reference voltage (do not apply power from an external source to this pin)</td>
</tr>
<tr>
<td>2</td>
<td>UART2_TxD</td>
<td>TTL UART transmit data of the micro controller (3.3V)</td>
</tr>
<tr>
<td>3</td>
<td>UART2_RxD</td>
<td>TTL UART receive data of the micro controller (3.3V)</td>
</tr>
<tr>
<td>4</td>
<td>SWCLK</td>
<td>ST/Link-2 programming interface of the micro controller (clock)</td>
</tr>
<tr>
<td>5</td>
<td>SWDIO</td>
<td>ST/Link-2 programming interface of the micro controller (data)</td>
</tr>
<tr>
<td>6</td>
<td>GND</td>
<td>Ground</td>
</tr>
</tbody>
</table>

**Encoder Debug UART (J3)**

This optional UART port shows the debug and boot up messages of the Max64380 encoder chip. Baud rate: 115200 8/N/1.

<table>
<thead>
<tr>
<th>Pin</th>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.3V</td>
<td>3.3V reference voltage (do not apply power from an external source to this pin)</td>
</tr>
<tr>
<td>2</td>
<td>DBG_TxD</td>
<td>TTL UART transmit data of the Max64380 (3.3V)</td>
</tr>
<tr>
<td>3</td>
<td>DBG_RxD</td>
<td>TTL UART receive data of the Max64380 (3.3V)</td>
</tr>
<tr>
<td>4</td>
<td>GND</td>
<td>Ground</td>
</tr>
</tbody>
</table>

**I2C port (J2)**

This optional I2C port interconnects the Max64380, the on-board micro controller and the HDMI in receiver chip. Normally the micro controller controls this bus as master. Please use the console interface on UART2, to manually read and write registers of the HDMI receiver chip for debug purposes. It is not recommended to use the MXCAM I2C tunnel to let the Max64380 take control of this I2C bus, as this access could collide with the micro controller driving this bus.

<table>
<thead>
<tr>
<th>Pin</th>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.3V</td>
<td>3.3V reference voltage (do not apply power from an external source to this pin)</td>
</tr>
<tr>
<td>2</td>
<td>I2C0_CLK</td>
<td>I2C bus 0 clock (3.3V)</td>
</tr>
<tr>
<td>3</td>
<td>I2C0_DAT</td>
<td>I2C bus 0 data (3.3V)</td>
</tr>
<tr>
<td>4</td>
<td>GND</td>
<td>Ground</td>
</tr>
</tbody>
</table>
Console control

UART2

Please connect a terminal emulation program with a USB to TTL UART converter, to connect to the console interface of the on-board micro controller. This provides a command line interface to setup and control the E412 module.

h

Print out a list of commands supported.

>h
  v: print version number
  t: print timing information
  r: read i2c register
  w: write i2c register
  l: list i2c devices
  avi: print avi infoframe
  col: set color mode
  edid: print edid
  bootloader: prepare for firmware upgrade

v

Print out the version of the firmware.

>v
  38156_v1.0.2

t

Print the timing details for the HDMI video source connected. The example below shows the 1080p25 timing with an actual resolution of 1920x1080 an a total size of 2640x1125. 0 is progressive. “528,148,44” is the horizontal timing (front porch, back porch, pulse width). “4,36,5” is the vertical timing (front porch, back porch, pulse width).

horizontal: 1920 + 528 + 148 + 44 = 2640
vertical: 1080 + 4 + 36 + 5 = 1125
“01” is the status.

>t
  1920,1080,2640,1125,0,528,148,44,4,36,5,01

r <i2c device> <register>

Read I2C register. “0x3C” reads the HDMI receiver chip.

> r 0x3C 0x1
  reg: 0x01 data: 0x00

w <i2c device> <register> <data>

Writes I2C register. “0x3C” reads the HDMI receiver chip.

> w 0x3c 0x40 0x2f
List all I2C devices. On the E412 this is just a single I2C device at address “0x3C”.

Print out the avinfoframe received by the HDMI receiver chip.

Read and write the color mode of the HDMI receiver chip. “auto” means that the micro controller reads and avinfoframe and determines the color model by the HDMI input. This is then programmed into the HDMI receiver chip, so that the color model is correct.

Print out the 256 byte EDID of the HDMI receiver chip. This EDID may be also read by the HDMI source connected via the DDC interface on the HDMI cable.
bootloader

Prepare the micro controller for firmware upgrade via the UART2 interface. This feature is not supported by the firmware yet. Please use the SWD interface and a ST/LINK-2 programmer to perform a firmware upgrade of the E412.

>bootloader
ready for firmware upgrade
please terminate your terminal emulation program now
and start the STM Flash Loader Demonstrator
Raspberry to E412 UART2 connection

You may connect the UART of the Raspberry Pi to the UART2 on the E412 with 3 wires (J5 of the E412). The UART pins on the GPIO header of the Raspberry Pi have the right voltage (3.3V).

- **GND**: pin 14 of the RPi to pin 6 of J5 (E412)
- **RxD**: pin 10 of the RPi to pin 2 (TxD) of J5 (E412)
- **TxD**: pin 8 of the RPi to pin 3 (RxD) of J5 (E412)
Terminal emulation

Tera Term (PC)
We have tested this terminal application (Tera Term 4.92) on Windows 7 and 10. Please use an USB to UART TTL with pin header. Then connect three wires: GND, RxD and TxD of UART2 to this pin header. Please make sure to swap RxD and TxD.

1. 
   Please install the FTDI 232RL driver. Windows 10 can do this automatically. Please check the device manager that this hardware was installed successfully.

2. 
   The FTDI 232RL driver creates a virtual COM port. The COM number may vary from system to system. Please go to setup - serial port and configure COM port number and baud rate (115200).

3. 
   Now you should be able to communicate with the E412. Test the „v“ command.

   v — print the version

minicom (Raspberry Pi)
We have tested this terminal application (minicom) on the Raspberry Pi. Please configure it for 115200 baud and 8/N/1.
Raspberry Pi video out

For testing purposes the Raspberry Pi may be used as a programmable video output generator. It supports 59 video modes (CEA) and 86 computer modes (DMT). Therefore the output resolution may be very flexibly adjusted. Up to date information may be retrieved from:

https://www.raspberrypi.org/documentation/configuration/config-txt.md

1. log into the Raspberry Pi with mouse, keyboard and monitor or with ssh remote access.

   ssh pi@<ip address> - connect via ssh
   password: raspberry

2. edit config.txt to modify the HDMI output settings

   sudo nano /boot/config.txt - edit config file

3. you may want to disable the EDID, to force the Pi into a specific output mode and therefore ignoring any EDID it may get from the E412 or another system it is connected to. Search for the line and edit it.

   hdmi_ignore_edid=0xa5000080 - ignore EDID

4. next select the hdmi_group 1, to specify CEA output modes

   hdmi_group=1

5. on the web site you find the complete list of CEA codes. Also they are listed in the timing table on page 6 of this document. Examples:

   hdmi_mode=16 - 1080p60
   hdmi_mode=31 - 1080p50
   hdmi_mode=32 - 1080p24
   hdmi_mode=33 - 1080p25
   hdmi_mode=34 - 1080p30

6. now save and close the config.txt file

   <ctrl> o - save the file
   <ctrl> x - close the file and editor
   <ctrl> c - close without saving

7. reboot the Raspberry Pi so that the changes can take effect

8. the Raspberry may also be used to read the EDID of the E412, if the HDMI out of the Raspberry Pi is connected to the HDMI input of the E412. Please make sure that the EDID is enabled in the config.txt file.

   #hdmi_ignore_edid=0xa5000080 - edit the config.txt file, to read EDID

   tvservice -d test.edid - save the EDID of the E412 to a file
   hexdump test.edid - dump the EDID hex fil
The firmware of the E412 may be upgraded via the UART connection. A UART connection to a Windows PC is required as the STM flash software only supports Windows at this time.

2. connect E412 via USB-to-UART cable to your Windows PC
3. start terminal emulation software and connect with your E412 (FT232R USB UART) (USB Serial COM Port)
4. check firmware version with the API command „v“
   example: 38156_v1.0.2
5. start bootloader with the API command „bootloader“
6. terminate terminal emulation program. We have tested this with TeraTerm. Some other terminal emulation programs may not terminate correctly (e.g. Putty).
7. start STM Flash Loader Demonstrator and program the micro controller. First select the port name (COM12 in our example). If “next” flags an error, then the terminal emulation program did not release the com port.
   - select the target device: STM32F0_4x_32k (STM32F042)
   - select the firmware xxx.elf.hex file
   - flash the device (do not interrupt or power down until the flashing has finished)
8. restart E412
9. verify that new firmware has been programmed with the API command „v“

Note: if the flashing has failed and the E412 does not boot anymore, it can be rescued by connecting an ST-LINK programmer SWD interface on connector J5.
FAQ

1.
Disclaimer

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

The Auvidea Team