Horizon 2020 PROGRAMME *ICT-01-2014: Smart Cyber-Physical Systems*

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No 643924

D3.5

Test sets and results with Factor Form Board

1. DOCUMENT INFORMATION

Deliverable Number	D3.5
Deliverable Name	Test sets and results with Factor Form Board
Authors	J. Parra (UCLM), N. Vallez (UCLM), J. M. Rico (UCLM), J. L. Espinosa-Aranda (UCLM), A. Dehghani (MOVIDIUS), S. Krauss (DFKI), R. Reiser (DFKI), A. Pagani (DFKI)
Responsible Author	Oscar Deniz (UCLM) e-mail:Oscar.Deniz@uclm.es phone: +34 926295300 Ext.6286
Keywords	EoT Firmware, Tests, EoT Factor form board
WP	WP3
Nature	R
Dissemination Level	PU
Planned Date	01.06.2016
Final Version Date	25.09.2017
Reviewed by	O. Deniz (UCLM)
Verified by	C. Fedorczak (THALES)

2. DOCUMENT HISTORY

Person	Date	Comment	Version
O. Deniz	30.05.2017	Initial	0.1
J. Parra-Patiño	14.07.2017		0.2
N. Vallez	04.09.2017		0.3
J.L. Espinosa-Aranda	25.09.2017		0.4
O. Deniz	25.09.2017	Final	0.5

3. ABSTRACT

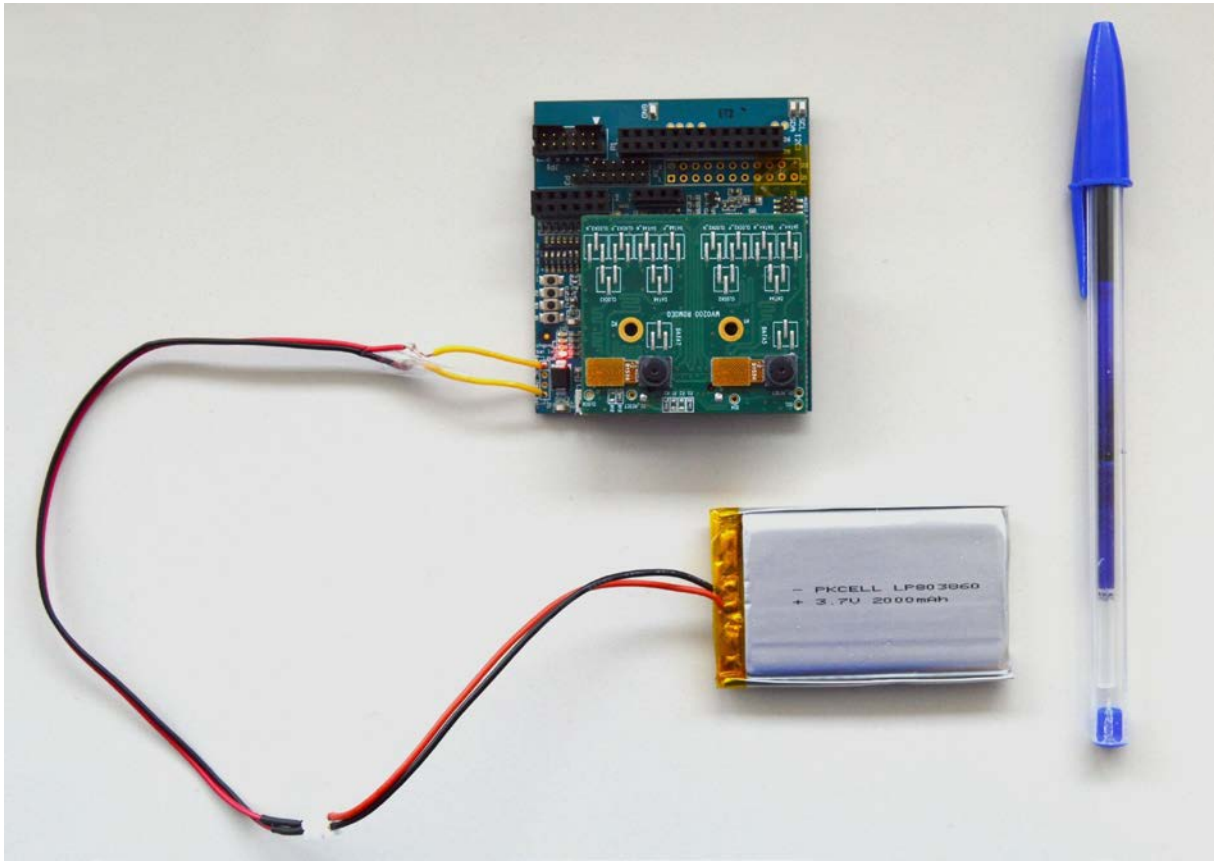
In this deliverable, we describe all software tests which we performed in the EoT factor form board (FFboard) so far. The tests were all conducted with revision R1 board. A R2 revision introducing minor changes and corrections is being manufactured at the time of writing. These tests are divided into: 1) Firmware tests, i.e. unit tests for the different modules developed as part of EoT's firmware and 2) Integration tests, i.e. non-unit tests that check that major modules can work together. All the tests passed satisfactorily.

4. TABLE OF CONTENTS

5. UNIT TESTS

Introduction

In this section of the document, we describe the set of unit tests applied to the FFboard ('Form-Factor board'), see Figure below. Most of the tests were provided by Movidius and these check that low-level functionalities work as in the previously-used MV0182 and Rev1 board prototypes. All of the tests described in this deliverable passed on the FFBoard R1.



SimpleRTEMS_sdCard

5.2.1. Software description

The example initializes the SDIO Driver in Posix_Init and then creates and starts a thread that mounts a filesystem, creates a file, writes 5 MB data to the file, synchronizes, reads back 5 MB, verifies data, and finally unmounts the filesystem.

5.2.2. Hardware requirements

SDHC Cards - V2.0 onwards (High Speed supported). Cards must be formatted FAT32, maximum 32K Clusters for maximum performance.

5.2.3. Expected output

The results consist in several printf seen using the debugger.

```
UART:
UART: RTEMS POSIX Started
UART:
UART: osBoard0231Initialise 0
UART:
UART: Sdio driver initialising
UART:
UART: OsDrvSdioInit sc RTEMS_SUCCESSFUL
UART:
UART: rtems_bdpart_register_from_disk sc RTEMS_SUCCESSFUL
UART:
UART: Mounting File System RTEMS_SUCCESSFUL
UART:
UART: Thread 1 created
UART:
UART: Creating file /mnt/sdcard/myfile
UART:
UART: Writing 5242880 bytes to file
UART:
UART: Perform fsync
UART:
UART: Closing file
UART:
UART: Opening file /mnt/sdcard/myfile
UART:
UART: Read 5242880 characters
UART:
UART: Verifying data...
UART:
UART: Card successfully unmounted
UART:
```

[redacted]IMU

5.3.1. Software description

It tests IMU functionality. By rotating the board in X and Y direction, the IMU values printed on screen will be changed.

This example reads the IMU measurements (accelerometer, gyroscope, magnetometer) and outputs them to the serial terminal. Note that these are raw, uncorrected IMU measurements

5.3.2. Expected output

The results consist in several printf seen using the debugger.

```
UART: I2C test result=0 => IMU I2C SUCCES!
```

```
UART:
UART: Thread camera created
UART: -------- ACC ---------
UART: X angle in degrees: 108.000000
UART: Y angle in degrees: 88.000000
UART: -------- GYRO ---------
UART: X angle in degrees: 27.000000
UART: Y angle in degrees: 60.000000
UART: -------- MAG ---------
UART: X angle in degrees: 37.000000
UART: Y angle in degrees: 39.000000
UART: -------- ACC ---------
UART: X angle in degrees: 108.000000
UART: Y angle in degrees: 89.000000
UART: -------- GYRO ---------
UART: X angle in degrees: 23.000000
UART: Y angle in degrees: 29.000000
UART: -------- MAG ---------
UART: X angle in degrees: 30.000000
UART: Y angle in degrees: 32.000000
```

LedGpioPmu

5.4.1. Software description

It tests basic functionality of the RC5T619 PMU.

The test consists of three parts:

1. Set 'chging' and 'bat lvl' LEDs as follows: all LEDs off, LED 1 Hz flicker, LED 4 Hz flicker, all LEDs on, all LEDs off.
2. Repeatedly read the values of DIP switches 1 and 2.
3. Read all and set some of the DCDC and LDO output voltages of the PMU.

5.4.2. Expected output

The debug console should show the following messages:

```
UART: Clocks started.
UART: About to init board...
UART: In boardI2CInit
UART: Initialising PMU
UART: Initialising GPIO
UART: Board: MV0231  Revision: R1M0E0
UART: Board Mv0231 initialized, revision = R1M0E0
UART: Begin Tests
UART:
UART:
UART:
UART: ---------- LED and GPIO setting tests ----------
UART:
UART: All LEDs off
UART:
UART: LED 1 Hz flicker
```

```
UART:
UART: LED 4 Hz flicker
UART:
UART: LED on
UART:
UART: All LEDs off
UART:
UART:
UART:
UART: ---------- GPIO reading tests ----------
UART:
UART: GPIO 2: 0
UART: GPIO 4: 0
UART:
UART: GPIO 2: 0
UART: GPIO 4: 0
UART:
UART: GPIO 2: 0
UART: GPIO 4: 1       ---- (DIP 1 toggled)
UART:
UART: GPIO 2: 0
UART: GPIO 4: 0
UART:
UART: GPIO 2: 0
UART: GPIO 4: 0
UART:
UART:
UART:
UART: ---------- DCDC and LDO getting and setting tests ----------
UART:
UART: DCDC1 voltage is 0.900000
UART: DCDC2 voltage is 0.900000
UART: DCDC3 voltage is 3.300000
UART: DCDC4 voltage is 1.850000
UART: DCDC5 voltage is 1.200000
UART: DCDC3 voltage is 3.300000
UART: LDO1 voltage is 3.300000
UART: LDO2 voltage is 2.800000
UART: LDO3 voltage is 3.300000
UART: LDO4 voltage is 3.300000
UART: LDO5 voltage is 1.200000
UART: LDO6 voltage is 1.800000
UART: LDO7 voltage is 1.800000
UART: LDO8 voltage is 1.200000
UART: LDO9 voltage is 1.800000
UART: LDO10 voltage is 1.800000
```

UsbVideo208

5.5.1. Software description

This test streams data from the imx208 camera through the USB bus.

This application uses the Leon code and a SIPP pipeline running on shaves. Basically, these are the few steps made during the application:

1. Start the USB DataPump on LeonOS.
2. General configurations of the board (internal clocks, external clock generator for sensors, GPIOs, ...).
3. Configure the camera sensors and the in chip datapath (MIPI, SIPP).

5.5.2. Hardware requirements

The FFboard and an USB cable connection from it to a PC host. The application works with both USB 3 and USB 2 hosts.

5.5.3. Expected output

The results consist in several printf seen using the debugger and a video stream that can be seen in a PC.

```
UART: In boardI2CInit
UART: Initialising PMU
UART: Initialising GPIO
UART: Board: MV0231  Revision: R1M0E0
UART: Board MV0231 initialized
UART: Configuring the SIPP pipeline
UART: Configuring imx208 camera and datapath
UART:
UART:
UART: UsbPump_Rtems_DataPump_Startup()!
UART:
UART:
UART: Camera initialized in HighSpeed mode
```

The displayed video will have a resolution of 1920 x 1080.
When the application starts, the PC host should identify the camera and the USB device speed, displaying one of the following message:

```
UART: Camera initialized in SuperSpeed mode for SuperSpeed
UART: Camera initialized in HighSpeed mode for HighSpeed
```

Depending on the USB speed the framerate of the video is different: 60fps for SuperSpeed and around 11fps for High Speed.

The video streaming can be captured using a stream capture application. Example of stream viewers:

- Windows: amcap, vlc, skype
- Linux: cheese, guvcview, xawtv, skype, vlc

EoT_Video_Streaming_Test

5.6.1. Software description

This application uses the Leon code and a SIPP pipeline running on shaves. Basically, these are the few steps made during the application:

1. Configure the WiFi module in AP mode.
2. General configurations of the board (internal clocks, external clock generator for sensors, GPIOs, …).
3. Configure the camera sensors and the in chip datapath (CIF, SIPP).

5.6.2. Hardware requirements

The EoT FFboard and a WiFi enabled device running an RTSP player.

5.6.3. Expected output

The results consist in several printf seen using the debugger and video stream that can be seen in every device with an RTSP player.

```
UART: In boardI2CInit
UART: Initialising PMU
UART: Initialising GPIO
UART: Board: MV0231  Revision: R1M0E0
UART: Board MV0231 initialized, revision = 256
UART:
UART:
 Getting started with WLAN access-point application - Version 1.2.0
************************************************************************
***********
 Device is configured in default state
 Device started as Access Point
 Waiting for clients to connect...!
 Client connected to the device
 Pinging...!
 Device and the station are successfully connected
Configuring camera and datapath
UART:     Start     VLC     in     client     and     connect     to
rtsp://192.168.1.1:8554/mjpeg/1
UART: Message Received:
UART:  OPTIONS rtsp://192.168.1.1:8554/mjpeg/1 RTSP/1.0
UART: CSeq: 1
UART: User-Agent: Lavf55.44.100
UART:
UART:
UART:
UART: Message Send:
UART:  RTSP/1.0 200 OK
UART: CSeq: 1
UART: Public: DESCRIBE, SETUP, TEARDOWN, PLAY, PAUSE
```

[redacted]WiFi_Get_Time_test_TI_CC31XX

5.7.1. Software description

This test connects the WiFi module to a local WiFi access point and acquires the current time from a time server over HTTP.

The WiFi access point SSID and password must be specified by the user (common/wifi_defaults.h).

This application does, only using the Leon code, the following tasks:

1. The Clock Power reset module is initialized.
2. The DDR is also initialized.
3. The L2 Cache must be invalidated.
4. On this step, the setup is complete. The desired message can now be printed.
5. Time is acquired from remote time server and printed to the standard output.
6. Programme repeats.

5.7.2. Expected output

The results consist in several printf seen using the debugger.

```
UART:
UART: --------- Initialise Board ----------
UART:
UART: In boardI2CInit
UART: Initialising PMU
UART: Initialising GPIO
UART: Board: MV0231  Revision: R1M0E0
UART: Board Mv0231 initialized, revision = R1M0E0
UART:
UART: --------- End of Board Init ----------
UART:
UART:
UART: BEGIN
UART:
UART:
 Get time application - Version 1.2.0
*************************************************************************
***********
 Configuring simplelink to default state
UART:  In AP mode, waiting...
UART:  About to switch to station mode
UART:  sl wlan set mode
UART:  Get device version info
UART:  All profiles removed
UART:  Device is configured in default state
 Device started as STATION
 Connection established w/ AP and IP is acquired
UART:
UART:  Server 0.in.pool.ntp.org has responded with time information
UART:
UART:  Mon Apr 4 2017 16:26:48
UART:
```

GpioHeaders

5.8.1. Software description

It tests GPIO pins. GPIO pins are available on headers P1 and P2.

GPIOs on P1 =
{54,68,75,67,01,66,02,65,03,58,45,46,13,12};
Pins on header P1 =
{04,06,08,10,11,12,13,14,15,16,21,23,24,26};

GPIOs on P2 =
{33,74,04,47,05,48,06,49,35,50,36,51,37,52,38,43,39,42,40,41,56,53};
Pins on header P2 =
{01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,17,18,19,20,21,22};

5.8.2. Expected output

Pin values should be checked with a multimeter (expect 1.8 I/O voltage when pin is toggled on).

```
Clocks started.
UART: ------ Board Init------
UART: In boardI2CInit
UART: Initialising PMU
UART: Initialising GPIO
UART: Board: MV0231  Revision: R1M0E0
UART: Board Mv0231 initialized, revision = R1M0E0
UART: ------ Board Init Complete------
UART: ------ Board Edge Header GPIO Test ------
UART:
UART:
UART:  P1
UART:
UART: toggling header P1 pin 4 (GPIO_54)...
UART: toggling header P1 pin 6 (GPIO_68)...
UART: toggling header P1 pin 8 (GPIO_75)...
UART: toggling header P1 pin 10 (GPIO_67)...
UART: toggling header P1 pin 11 (GPIO_1)...
UART: toggling header P1 pin 12 (GPIO_66)...
UART: toggling header P1 pin 13 (GPIO_2)...
UART: toggling header P1 pin 14 (GPIO_65)...
UART: toggling header P1 pin 15 (GPIO_3)...
UART: toggling header P1 pin 16 (GPIO_58)...
UART: toggling header P1 pin 21 (GPIO_45)...
UART: toggling header P1 pin 23 (GPIO_46)...
UART: toggling header P1 pin 24 (GPIO_13)...
UART: toggling header P1 pin 26 (GPIO_12)...
UART:
UART:
UART:  P2
UART:
```

```
UART: toggling header P2 pin 1 (GPIO_33)...
UART: toggling header P2 pin 2 (GPIO_74)...
UART: toggling header P2 pin 3 (GPIO_4)...
UART: toggling header P2 pin 4 (GPIO_47)...
UART: toggling header P2 pin 5 (GPIO_5)...
UART: toggling header P2 pin 6 (GPIO_48)...
UART: toggling header P2 pin 7 (GPIO_6)...
UART: toggling header P2 pin 8 (GPIO_49)...
UART: toggling header P2 pin 9 (GPIO_35)...
UART: toggling header P2 pin 10 (GPIO_50)...
UART: toggling header P2 pin 11 (GPIO_36)...
UART: toggling header P2 pin 12 (GPIO_51)...
UART: toggling header P2 pin 13 (GPIO_37)...
UART: toggling header P2 pin 14 (GPIO_52)...
UART: toggling header P2 pin 15 (GPIO_38)...
UART: toggling header P2 pin 16 (GPIO_43)...
UART: toggling header P2 pin 17 (GPIO_39)...
UART: toggling header P2 pin 18 (GPIO_42)...
UART: toggling header P2 pin 19 (GPIO_40)...
UART: toggling header P2 pin 20 (GPIO_41)...
UART: toggling header P2 pin 21 (GPIO_56)...
UART: toggling header P2 pin 22 (GPIO_53)...
UART: ------ End of Test ------
```

[redacted] SPI_flash_test

5.9.1. Software description

This application flashes a program (PM LEDs blink), runs it and then erases it. In particular, it causes that “chging” and “bat lvl” LEDs starts blinking at 1 Hz.

5.9.2. Expected output

During the flash process, you will see these messages:

```
UART: Flash Utility Version:02.10
UART: Flash programmer for Myriad2 ... using SPI1_SS77
UART:   <spiJEDEC_RDID> MFGID=0x20 MTYPE=0xBB MCAPACITY=0x17
UART: Initializing N25Q_SingleDie SPI Flash chip
UART: Analyzing image to write...0xFD20 0x7832414D
UART:  format=MVCMD len=65056 (0xFE20)
UART: Partially erasing chip...
UART: Done partially erasing chip. (0x00000000 len 65536 bytes)
UART: Image seems ok; 254 pages to program; target_addr=0x000000
UART:
WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART:
WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART:
WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART:
WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
```

```
UART:
WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART:
WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART:
WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART:   WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART: Image CRC32 (64800 bytes): 0x30E64C2D
UART: Readback entire image from Flash as final check...
UART:
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
UART:
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
UART:
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
UART:
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
UART:
UART: Readback Image CRC Matches Expected  (64800 bytes): 0x30E64C2D
UART: Image programmed successfully..... Exiting.
```

Running the app:
After the flash, if you power off and then on the board, "PmicLedBlink" application is run automatically (LEDs blink one after another).

Erasing the Flash:
You will see the following messages:

```
UART: Flash Utility Version:02.10
UART: Flash programmer for Myriad2 ... using SPI1_SS77
UART:   <spiJEDEC_RDID> MFGID=0x20 MTYPE=0xBB MCAPACITY=0x17
UART: Initializing N25Q_SingleDie SPI Flash chip
UART: Analyzing image to write...0x0 0x69A608
UART: Just full erase chip ...
UART: Erasing full chip...
UART: Done erasing full chip.
```

[redacted] Motor_Control_GPIO-EN-PWN-I2C_test

5.10.1. Software description

Simulation of motor control by an Arduino board.

This application performs tests on the GPIO and I2C pins used for robot control. The Arduino board simulates the presence of a motor and processes the data from the Myriad level-shifted interface (5V in this case) and shows the results on a serial port.

At the end of the GPIO and I2C tests, the user will be informed if the tests passed or failed.

5.10.2. Hardware requirements

Arduino Uno board and a serial terminal (e.g. Arduino IDE).

5.10.3. Expected output

The testing will be repeated indefinitely until the user exits the application.

```
UART:  ------ Motor control test app ------
UART:  Please connect the arduino board via USB and open a serial terminal
UART:  with the following settings: 9600 baud, 8 bits, parity: none, 1 stop
UART:
UART: START: GPIO TEST
UART:   Test status should be verified on the Arduino board
UART:   Toggling pin status ...
UART:   Toggling pin status ...
UART:   Toggling pin status ...
UART: END: GPIO TEST
UART:
UART: START: I2C TEST
UART:   Writing on the Arduino board ...
UART:   DONE!
UART:   Reading from the Arduino board ...
UART:   DONE!
UART:   ********* PASSED *********
UART: END: I2C TEST
UART:
```

Expected output on the Arduino

==========

The Arduino will wait for any board to be connected and start the testing indefinitely.

```
-------------- Motor control test app ---------------
START: GPIO TEST
  *************** PASSED ***************
END: GPIO TEST
START: I2C TEST
Sent OK to master
END: I2C TEST!
```

[redacted]NanEyeUsbVideo

5.11.1. Software description

It sends the stream data from the NanEye2D camera over USB bus. The NanEye2D sensor is interfaced to the Myriad2 via CIF.

This application uses the Leon code and a SIPP pipeline running on shaves. Basically, these are the few steps made during the application:

1. Start the USB DataPump on LeonOS.

2. General configurations of the board (internal clocks, external clock generator for sensors, GPIOs, …).
3. Configure the camera sensors and the in chip datapath (CIF, SIPP).

5.11.2. Hardware requirements

An USB cable connection from the FFboard to a PC host (the application works with both USB 3 and USB 2 hosts) and a NanEye2D image sensor.

5.11.3. Expected output

```
UART: In boardI2CInit
UART: Initialising PMU
UART: Initialising GPIO
UART: Board: MV0231  Revision: R1M0E0
UART: Board MV0231 initialized, revision = 256
UART: Configuring the SIPP pipeline
UART: Configuring NanEye camera and datapath
UART: Result of MachRead(): 0
UART: MachXO3 firmware revision: 0x7
UART: Mach CTRL0 register setting: 0x8
UART: Result of MachRead(): 0
UART: Result of MachRead(): 0
UART:
UART:
UART: UsbPump_Rtems_DataPump_Startup()!
UART:
UART:
UART: Camera initialized in HighSpeed mode
UART:
UART:  USB config complete.
UART: #frames: 46
UART: #frames: 96
UART: #frames: 146
UART: #frames: 196
UART: #frames: 246
UART: #frames: 296
```

The framerate of the video is ~22fps.

The video streaming can be captured using a stream capture application.
Example of stream viewers:

- Windows: amcap, vlc, skype
- Linux: cheese, guvcview, xawtv, skype, vlc

EoT_Audio

5.12.1. Software description

This test verifies operation of the audio codec, as well as the 3.5mm headphone jack interface (stereo audio out, mono microphone in).

A 4-pin headphone jack, wired in the CTIA format, should be attached to the audio header.

The test plays 30s of a stereo wav file (which must be copied from assets/audioExample.wav and written to the SD card).

Next, the test records 10s of audio from the mic input and writes it to the SD card. Finally, the recording is played back in mono format over the headphone jack audio out.

5.12.2. Expected output

```
UART:
UART: osBoard0231Initialise 0
UART: AudioPlay() returning 1
UART: playing...
UART: audio stopped at position 2.000000, moving on...
UART: recording...
UART: playing...
```

6. INTEGRATION TESTS

Introduction

In this section, we describe the integration tests, i.e. non-unit tests that ensure that major modules can work together. The core software module in EoT is 'Control mode' (also known as 'Pulga'), which includes access to Wifi, camera, SD card and Flash memory. These capabilities have been developed by different partners. Some of the hardware components share a common hardware component, so the tests also allow to ensure that no conflict exists or, if it exists, it can be solved. As an example, WiFi and Flash did not work together in the MV0182 board-based prototype that was used at the start of the project, which only had a single SPI interface available. This was later corrected in Rev1 board (also known as 'DevBoard'). All of the integration tests required intervention from the tester. MDK version 17.04.5 was used.

Integration Tests on the FFboard

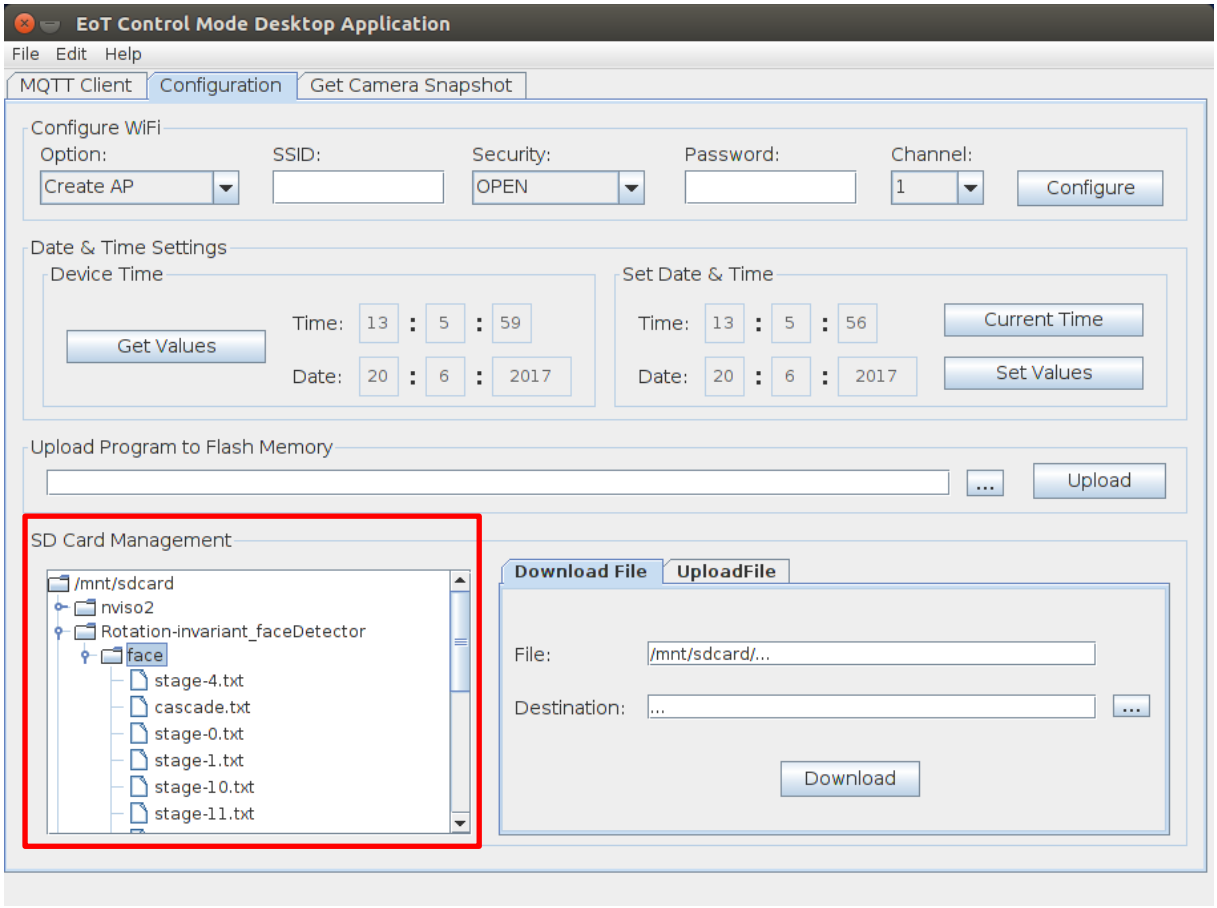
6.2.1. Control mode + SD

6.2.1.1. Software description

This test starts Pulga and allows the user to access the SD card filesystem through an MQTT client such as the EoT Android or desktop clients. The application mounts the SD card filesystem, lists directories, removes files and folders and uploads new files to the SD card.

6.2.1.2. Expected output

If the EoT Control Mode desktop application is used, the content of the SD card is shown in the Configuration tab as follows:



The output of the debug console is:

```
runw
UART: Starting mv0231 board initialisation.
UART: mv0231 board initialisation from cfg file
UART: register I2C0 as master
UART:
UART: Registering i2c devices
UART: Registering i2c device rc5t619
UART: Polling address 0x32 on bus 2
UART: i2c_drv_minor: 10290
UART: i2c device rc5t619 registered at bus 2 with address 0x32
UART: ----------------------------------------
UART: Registering i2c devices
UART: Registering i2c device imx208_right
UART: Polling address 0x36 on bus 0
UART: i2c_drv_minor: 16438
UART: i2c device imx208_right registered at bus 0 with address 0x36
UART: ----------------------------------------
UART: Registering i2c device imx208_left
UART: Polling address 0x37 on bus 0
UART: i2c_drv_minor: 24631
UART: i2c device imx208_left registered at bus 0 with address 0x37
UART: ----------------------------------------
UART: Registering i2c device bmx055_accel
UART: Polling address 0x18 on bus 1
```

```
UART: i2c_drv_minor: -27
UART: Polling address 0x19 on bus 1
UART: i2c_drv_minor: 33817
UART: i2c device bmx055_accel registered at bus 1 with address 0x19
UART: ----------------------------------------
UART: Registering i2c device bmx055_gyro
UART: Polling address 0x68 on bus 1
UART: i2c_drv_minor: 42088
UART: i2c device bmx055_gyro registered at bus 1 with address 0x68
UART: ----------------------------------------
UART: Registering i2c device bmx055_magnet
UART: Polling address 0x10 on bus 1
UART: i2c_drv_minor: -27
UART: Polling address 0x11 on bus 1
UART: i2c_drv_minor: 50193
UART: i2c device bmx055_magnet registered at bus 1 with address 0x11
UART: ----------------------------------------
UART: Board version R1M0E0
UART: PMU switches configured.
UART:
UART: LED2 configured.
UART:
UART: LED3 configured.
UART:
UART: LED4 configured.
UART:
UART: Revision specific board configuration.
UART: LED2 set.
UART: DCDC set to 3.3V.
UART: WiFi nReset set.
UART: MV0231 board initialized successfully
UART:
UART: LeonRT Started.
UART:
UART: Configuring camera and datapath
UART: Initialising I2C0 in bare metal mode...
UART: Bare metal I2C initialisation complete
UART: Camera initialised!
UART: Camera started!
UART:
UART: Streaming ...
```

6.2.2. Control mode + Camera

As explained in previous deliverable D3.3, the Camera module operates using the I2C buses for the communication with sensors. This module needs to configure the I2C buses and the interrupt service routine (ISR) to periodically update the new buffer address where the image is stored.

These aspects should be taken into account when using this module together with Pulga. Pulga is based on the WifiFunctions library. This library uses the I2C buses to send the nHIB signal to the WiFi chip and, therefore, there might be some conflicts while using WiFi and Camera together. In this respect, camera an

WiFi initializations remain in the same order as required in the previous Rev1 board.

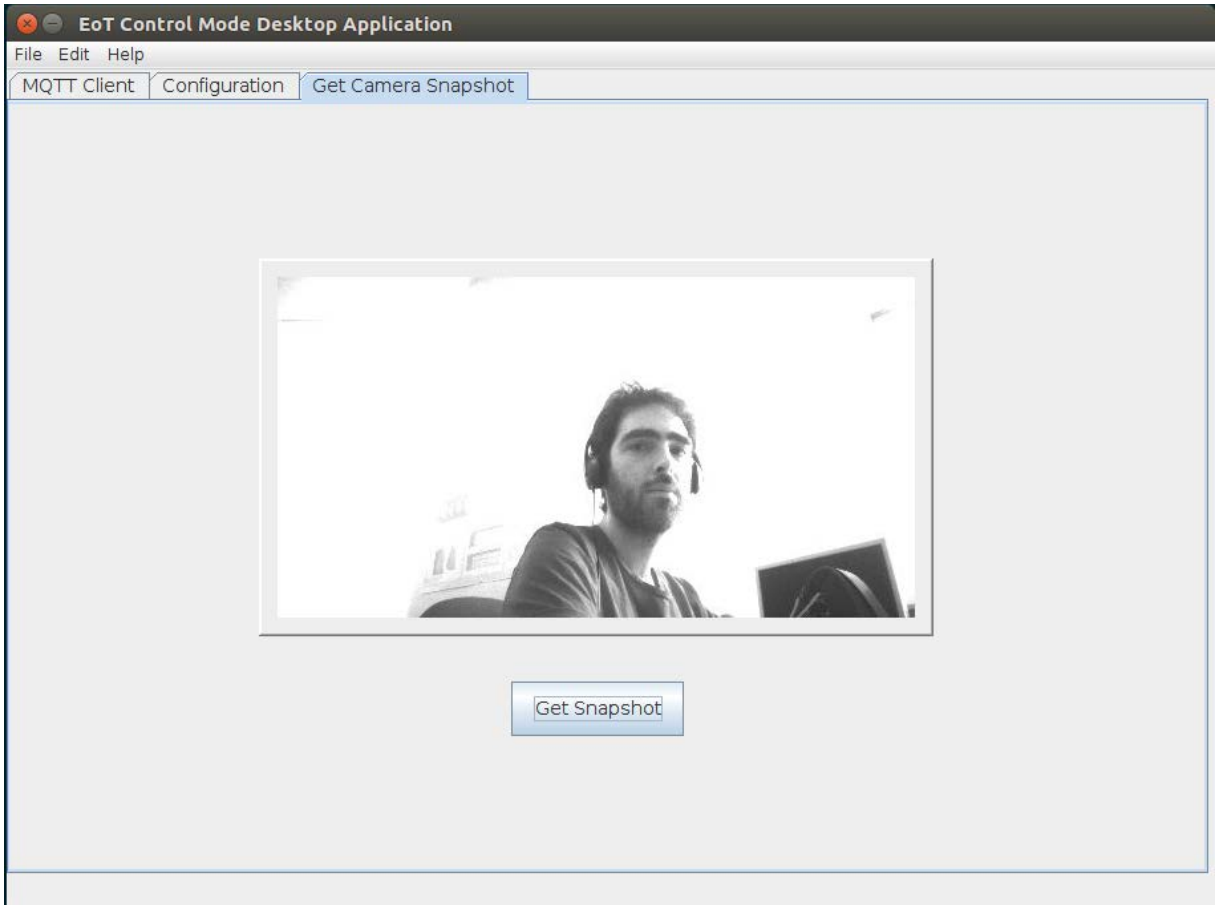
In addition, Camera library has been adapted to the new MDK paradigm. Camera code runs now on Leon RT to achieve better performance.

6.2.2.1. Software description

This test starts Pulga and allows the user to access and retrieve a camera frame through EoT Android or Desktop clients. The application makes all the initializations, accesses the camera buffers using the Camera module and sends the image encapsulated into several MQTT messages.

6.2.2.2. Expected output

The output in the EoT Control Mode desktop application should be an image like the following one:



The output of the debug console is:

```
UART: Starting mv0231 board initialisation.
UART: mv0231 board initialisation from cfg file
UART: register I2C0 as master
```

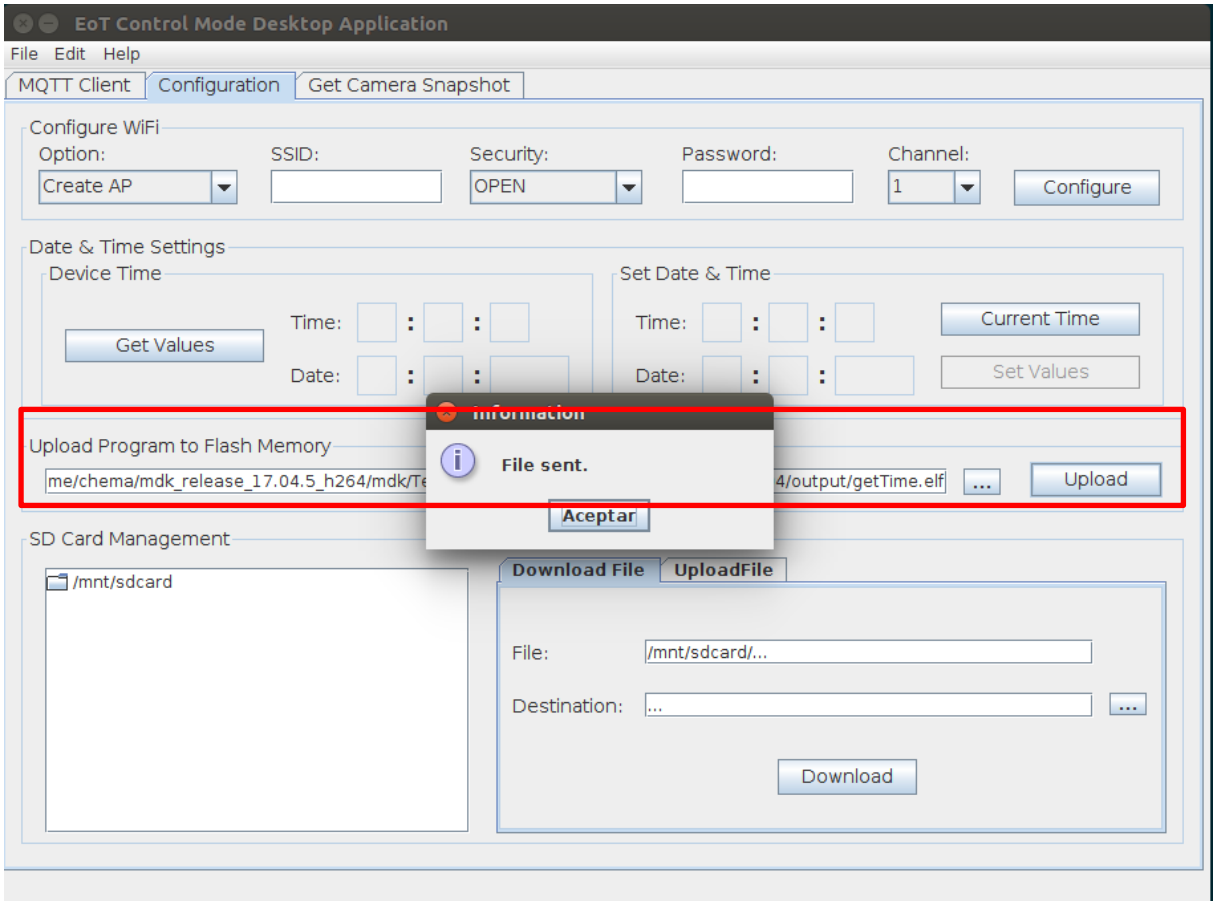
```
UART:
UART: Registering i2c devices
UART: Registering i2c device rc5t619
UART: Polling address 0x32 on bus 2
UART: i2c_drv_minor: 10290
UART: i2c device rc5t619 registered at bus 2 with address 0x32
UART: ----------------------------------------
UART: Registering i2c devices
UART: Registering i2c device imx208_right
UART: Polling address 0x36 on bus 0
UART: i2c_drv_minor: 16438
UART: i2c device imx208_right registered at bus 0 with address 0x36
UART: ----------------------------------------
UART: Registering i2c device imx208_left
UART: Polling address 0x37 on bus 0
UART: i2c_drv_minor: 24631
UART: i2c device imx208_left registered at bus 0 with address 0x37
UART: ----------------------------------------
UART: Registering i2c device bmx055_accel
UART: Polling address 0x18 on bus 1
UART: i2c_drv_minor: -27
UART: Polling address 0x19 on bus 1
UART: i2c_drv_minor: 33817
UART: i2c device bmx055_accel registered at bus 1 with address 0x19
UART: ----------------------------------------
UART: Registering i2c device bmx055_gyro
UART: Polling address 0x68 on bus 1
UART: i2c_drv_minor: 42088
UART: i2c device bmx055_gyro registered at bus 1 with address 0x68
UART: ----------------------------------------
UART: Registering i2c device bmx055_magnet
UART: Polling address 0x10 on bus 1
UART: i2c_drv_minor: -27
UART: Polling address 0x11 on bus 1
UART: i2c_drv_minor: 50193
UART: i2c device bmx055_magnet registered at bus 1 with address 0x11
UART: ----------------------------------------
UART: Board version R1M0E0
UART: PMU switches configured.
UART:
UART: LED2 configured.
UART:
UART: LED3 configured.
UART:
UART: LED4 configured.
UART:
UART: Revision specific board configuration.
UART: LED2 set.
UART: DCDC set to 3.3V.
UART: WiFi nReset set.
UART: MV0231 board initialized successfully
UART:
UART: LeonRT Started.
UART:
UART: Configuring camera and datapath
```

```
UART: Initialising I2C0 in bare metal mode...
UART: Bare metal I2C initialisation complete
UART: Camera initialised!
UART: Camera started!
UART:
UART: Streaming ...
UART:
UART: snapshot
```

6.2.3. Control mode + Flash

This test performs some reads and writes in the flash memory while Pulga is running to ensure that WiFi and flash can be accessed simultaneously.

If this test is executed and a MQTT client is connected to Pulga, the expected output of this test is:



Board initialization:

```
UART: Starting mv0231 board initialisation.
UART: mv0231 board initialisation from cfg file
UART: register I2C0 as master
UART:
UART: Registering i2c devices
UART: Registering i2c device rc5t619
UART: Polling address 0x32 on bus 2
```

```
UART: i2c_drv_minor: 10290
UART: i2c device rc5t619 registered at bus 2 with address 0x32
UART: ----------------------------------------
UART: Registering i2c devices
UART: Registering i2c device imx208_right
UART: Polling address 0x36 on bus 0
UART: i2c_drv_minor: 16438
UART: i2c device imx208_right registered at bus 0 with address 0x36
UART: ----------------------------------------
UART: Registering i2c device imx208_left
UART: Polling address 0x37 on bus 0
UART: i2c_drv_minor: 24631
UART: i2c device imx208_left registered at bus 0 with address 0x37
UART: ----------------------------------------
UART: Registering i2c device bmx055_accel
UART: Polling address 0x18 on bus 1
UART: i2c_drv_minor: -27
UART: Polling address 0x19 on bus 1
UART: i2c_drv_minor: 33817
UART: i2c device bmx055_accel registered at bus 1 with address 0x19
UART: ----------------------------------------
UART: Registering i2c device bmx055_gyro
UART: Polling address 0x68 on bus 1
UART: i2c_drv_minor: 42088
UART: i2c device bmx055_gyro registered at bus 1 with address 0x68
UART: ----------------------------------------
UART: Registering i2c device bmx055_magnet
UART: Polling address 0x10 on bus 1
UART: i2c_drv_minor: -27
UART: Polling address 0x11 on bus 1
UART: i2c_drv_minor: 50193
UART: i2c device bmx055_magnet registered at bus 1 with address 0x11
UART: ----------------------------------------
UART: Board version R1M0E0
UART: PMU switches configured.
UART:
UART: LED2 configured.
UART:
UART: LED3 configured.
UART:
UART: LED4 configured.
UART:
UART: Revision specific board configuration.
UART: LED2 set.
UART: DCDC set to 3.3V.
UART: WiFi nReset set.
UART: MV0231 board initialized successfully
UART:
UART: LeonRT Started.
UART:
UART: Configuring camera and datapath
UART: Initialising I2C0 in bare metal mode...
UART: Bare metal I2C initialisation complete
UART: Camera initialised!
UART: Camera started!
```

```
UART:
UART: Streaming ...
UART:
UART: number_of_packets 22540
UART: actual_packet 1000
UART: actual_packet 2000
UART: actual_packet 3000
UART: actual_packet 4000
UART: actual_packet 5000
UART: actual_packet 6000
UART: actual_packet 7000
UART: actual_packet 8000
UART: actual_packet 9000
UART: actual_packet 10000
UART: actual_packet 11000
UART: actual_packet 12000
UART: actual_packet 13000
UART: actual_packet 14000
UART: actual_packet 15000
UART: actual_packet 16000
UART: actual_packet 17000
UART: actual_packet 18000
UART: actual_packet 19000
UART: actual_packet 20000
UART: actual_packet 21000
UART: actual_packet 22000
UART: Current packet received: 0 Current frc: 3 Current actual
packet: 0 1030
UART: Current packet received: 1000 Current frc: 3 Current actual
packet: 1000 1030
UART: Current packet received: 2000 Current frc: 3 Current actual
packet: 2000 1030
UART: Current packet received: 3000 Current frc: 3 Current actual
packet: 3000 1030
UART: Current packet received: 4000 Current frc: 3 Current actual
packet: 4000 1030
UART: Current packet received: 5000 Current frc: 3 Current actual
packet: 5000 1030
UART: Current packet received: 6000 Current frc: 3 Current actual
packet: 6000 1030
UART: Current packet received: 7000 Current frc: 3 Current actual
packet: 7000 1030
UART: Current packet received: 8000 Current frc: 3 Current actual
packet: 8000 1030
UART: Current packet received: 9000 Current frc: 3 Current actual
packet: 9000 1030
UART: Current packet received: 10000 Current frc: 3 Current actual
packet: 10000 1030
UART: Current packet received: 11000 Current frc: 3 Current actual
packet: 11000 1030
UART: Current packet received: 12000 Current frc: 3 Current actual
packet: 12000 1030
UART: Current packet received: 13000 Current frc: 3 Current actual
packet: 13000 1030
```

```
UART: Current packet received: 14000 Current frc: 3 Current actual
packet: 14000 1030
UART: Current packet received: 15000 Current frc: 3 Current actual
packet: 15000 1030
UART: Current packet received: 16000 Current frc: 3 Current actual
packet: 16000 1030
UART: Current packet received: 17000 Current frc: 3 Current actual
packet: 17000 1030
UART: Current packet received: 18000 Current frc: 3 Current actual
packet: 18000 1030
UART: Current packet received: 19000 Current frc: 3 Current actual
packet: 19000 1030
UART: Current packet received: 20000 Current frc: 3 Current actual
packet: 20000 1030
UART: Current packet received: 21000 Current frc: 3 Current actual
packet: 21000 1030
UART: Current packet received: 22000 Current frc: 3 Current actual
packet: 22000 1030
```

6.2.4. Bootloader + Control mode + Flash

6.2.4.1. Software description

This test has been performed to test the Bootloader application as the basis of the EoT FFBoard configuration. The Bootloader is used either to boot into 'control mode' or to load an app, which the user has previously installed on the EoT device. The choice what to boot is made considering the position of a DIP switch on the device.

When and application is started via JTAG, the program checks if a common used resource (e.g: ddr-memory, or i2c0) will be used and enable it automatically. However, using the bootloader-app and the FFboard, the implicit enabling of resources must be done by it. Therefore, the Bootloader application has been fixed to enable all common used resources before starting the "real" application.

6.2.4.2. Expected output

Flashing the bootloader:

The first step is to flash the Bootloader and Control applications:

```
Movidius Debugger (moviDebug2) v00.83.2 Build 121447
Loading Tcl library...
Loading debug library...
Connecting to 127.0.0.1:30001
Connection         : OK [127.0.0.1:30001 | Olimex ARM-USB-TINY-H @
3000kHz]
Target Voltage  : OK
JTAG IDCODE     : ma2x5x                  [0x1C450661]
Starting TCF agent...
breset
Leon halt status: ok
Loading section <.source_cmx> start = 0x70000000 length = 4 bytes.
```

```
Section loaded OK.
Loading section <.source_ddr> start = 0x80000000 length = 4 bytes.
Section loaded OK.
Loading section <.readback_ddr> start = 0x84000000 length = 4 bytes.
Section loaded OK.
Loading section <.text> start = 0x70180000 length = 55520 bytes.
Section loaded OK.
Loading section <.ctors> start = 0x7018D8E0 length = 4 bytes.
Section loaded OK.
Loading section <.rodata> start = 0x7018D8E8 length = 9760 bytes.
Section loaded OK.
Loading section <.init> start = 0x7018FF08 length = 60 bytes.
Section loaded OK.
Loading section <.fini> start = 0x7018FF44 length = 44 bytes.
Section loaded OK.
Loading section <.data> start = 0x7018FF70 length = 1264 bytes.
Section loaded OK.
Loading section <.heapSection> start = 0x70190460 length = 6144
bytes.
Section loaded OK.
Total bytes loaded = 72808.
UART: Flash Utility Version:02.10
UART: Flash programmer for Myriad2 ... using SPI1_SS77
UART:   <spiJEDEC_RDID> MFGID=0x00 MTYPE=0x00 MCAPACITY=0x00
UART:   not a known chip... Trying to wake up a known chip from deep
power-down mode...
UART:   <spiJEDEC_RDID> MFGID=0x20 MTYPE=0xBB MCAPACITY=0x17
UART: Initializing N25Q_SingleDie SPI Flash chip
UART: Analyzing image to write...0x0 0xF03ED6E
UART: Just full erase chip ...
UART: Erasing full chip...
UART: Done erasing full chip.
LOS:  LeonOS  (P0:ALOS)  suspended  at  0x7018032C  (Application
terminated successfully)
moviDebug2 exiting
Shutting down TCF agent...
TCF agent shutdown complete.
moviDebug2 exited
# bootloader
Movidius Debugger (moviDebug2) v00.83.2 Build 121447
Loading Tcl library...
Loading debug library...
Connecting to 127.0.0.1:30001
Connection        : OK [127.0.0.1:30001 | Olimex ARM-USB-TINY-H @
3000kHz]
Target Voltage  : OK
JTAG IDCODE     : ma2x5x                [0x1C450661]
Starting TCF agent...
breset
Leon halt status: ok
Loading section <.source_cmx> start = 0x70000000 length = 4 bytes.
Section loaded OK.
Loading section <.source_ddr> start = 0x70000004 length = 4 bytes.
Section loaded OK.
Loading section <.readback_ddr> start = 0x70000008 length = 4 bytes.
```

```
Section loaded OK.
Loading section <.text> start = 0x70190000 length = 61436 bytes.
Section loaded OK.
Loading section <.text.eh_frame> start = 0x7019F000 length = 128
bytes.
Section loaded OK.
Loading section <.ctors> start = 0x7019F080 length = 4 bytes.
Section loaded OK.
Loading section <.dtors> start = 0x7019F084 length = 4 bytes.
Section loaded OK.
Loading section <.rodata> start = 0x7019F088 length = 6233 bytes.
Section loaded OK.
Loading section <.init> start = 0x701A08F0 length = 92 bytes.
Section loaded OK.
Loading section <.fini> start = 0x701A094C length = 68 bytes.
Section loaded OK.
Loading section <.data> start = 0x701A0990 length = 2548 bytes.
Section loaded OK.
Loading section <.heapSection> start = 0x701A1390 length = 6144
bytes.
Section loaded OK.
Total bytes loaded = 76669.
UART: Flash Utility Version:02.10
UART: Flash programmer for Myriad2 ... using SPI1_SS77
UART:   <spiJEDEC_RDID> MFGID=0x20 MTYPE=0xBB MCAPACITY=0x17
UART: Initializing N25Q SPI Flash chip
UART:
UART:
UART: SOURCE_ADDR_CMX
UART:
UART: Analyzing image to write...
UART: isControlApp 0x0
UART: Size 81268
UART: Value 0x7832414D
UART: Image seems ok; 318 pages to program; target_addr=0x000000
UART:
WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART:
WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART:
WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART:
WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART:
WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART:
WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART:
WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART:
WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART:
WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART:   WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART: Image CRC32 (81268 bytes): 0x2B121B72
```

```
UART: Readback entire image from Flash as final check...
UART:
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
UART:
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
UART:
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
UART:
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
UART:
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
UART:
UART: Readback Image CRC Matches Expected  (81268 bytes): 0x2B121B72
UART: Image programmed successfully..... Exiting.
LOS:  LeonOS  (P0:ALOS)  suspended  at  0x7019032C  (Application
terminated successfully)
moviDebug2 exiting
Shutting down TCF agent...
TCF agent shutdown complete.
moviDebug2 exited
Reading back from DDR unchanged [OK]
# control app
Movidius Debugger (moviDebug2) v00.83.2 Build 121447
Loading Tcl library...
Loading debug library...
Connecting to 127.0.0.1:30001
Connection         : OK [127.0.0.1:30001 | Olimex ARM-USB-TINY-H @
3000kHz]
Target Voltage  : OK
JTAG IDCODE     : ma2x5x                 [0x1C450661]
Starting TCF agent...
breset
Leon halt status: ok
Loading section <.source_cmx> start = 0x70000000 length = 4 bytes.
Section loaded OK.
Loading section <.source_ddr> start = 0x70000004 length = 4 bytes.
Section loaded OK.
Loading section <.readback_ddr> start = 0x70000008 length = 4 bytes.
Section loaded OK.
Loading section <.text> start = 0x70190000 length = 61436 bytes.
Section loaded OK.
Loading  section  <.text.eh_frame>  start  =  0x7019F000  length  =  128
bytes.
Section loaded OK.
Loading section <.ctors> start = 0x7019F080 length = 4 bytes.
Section loaded OK.
Loading section <.dtors> start = 0x7019F084 length = 4 bytes.
Section loaded OK.
Loading section <.rodata> start = 0x7019F088 length = 6233 bytes.
Section loaded OK.
Loading section <.init> start = 0x701A08F0 length = 92 bytes.
Section loaded OK.
Loading section <.fini> start = 0x701A094C length = 68 bytes.
Section loaded OK.
Loading section <.data> start = 0x701A0990 length = 2548 bytes.
```

```
Section loaded OK.
Loading section <.heapSection> start = 0x701A1390 length = 6144
bytes.
Section loaded OK.
Total bytes loaded = 76669.
UART: Flash Utility Version:02.10
UART: Flash programmer for Myriad2 ... using SPI1_SS77
UART:   <spiJEDEC_RDID> MFGID=0x20 MTYPE=0xBB MCAPACITY=0x17
UART: Initializing N25Q SPI Flash chip
UART:
UART:
UART: SOURCE_ADDR_CMX
UART:
UART: Analyzing image to write...
UART: isControlApp 0x1
UART: Size 693612
UART: Value 0x464C457F
UART: Image seems ok; 2710 pages to program; target_addr=0x080000
UART:
WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART:
WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART:
WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART:
WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART:
WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART:
WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART:
```

…………………

```
WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART:
WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART:
WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART:
WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART:
WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART:
WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART:   WvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWvWv
UART: Image seems ok; 1 pages to program; target_addr=0x7FF000
UART: Wv
UART: Image CRC32 (693612 bytes): 0x3217FF76
UART: Readback entire image from Flash as final check...
UART:
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
UART:
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
```

```
UART:
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
UART:
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
UART:
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
UART:
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR

…………………

UART:
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
UART:
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
UART:
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
UART:
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
UART:
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
UART:   RRRRRRRRRRRRRRRRRRRRRR
UART:
UART:  Readback  Image  CRC  Matches  Expected   (693612  bytes):
0x3217FF76
UART: Image programmed successfully..... Exiting.
LOS:  LeonOS  (P0:ALOS)  suspended  at  0x7019032C  (Application
terminated successfully)
moviDebug2 exiting
Shutting down TCF agent...
TCF agent shutdown complete.
moviDebug2 exited
Reading back from DDR unchanged [OK]
```

<u>Running the bootloader (DIP switch 1 in Control Mode position):</u>

If there is no flashed application or the DIP switch 1 position is configured as ON, the Bootloader will run the Control Application. The following output shows this procedure including the process of flashing an app:

```
Movidius Debugger (moviDebug2) v00.83.2 Build 121447
Loading Tcl library...
Loading debug library...
Connecting to 127.0.0.1:30001
Connection        : OK [127.0.0.1:30001 | Olimex ARM-USB-TINY-H @
3000kHz]
Target Voltage  : OK
JTAG IDCODE     : ma2x5x                [0x1C450661]
Starting TCF agent...
breset
Leon halt status: ok
startupcore LOS
target LOS
loadfile ./output/bootloader.elf
```

```
Loading section <.text> start = 0x701E0000 length = 70068 bytes.
Loading section <.text.eh_frame> start = 0x701F11C0 length = 128
bytes.
Loading section <.dtors> start = 0x701F1240 length = 4 bytes.
Loading section <.rodata> start = 0x701F1248 length = 7560 bytes.
Loading section <.init> start = 0x701F2FD0 length = 56 bytes.
Loading section <.fini> start = 0x701F3008 length = 40 bytes.
Loading section <.data> start = 0x701F3030 length = 1364 bytes.
Loading section <.heapSection> start = 0x701F3590 length = 1024
bytes.
Total bytes loaded = 80244.
runw
UART: Starting mv0231 board initialisation.
UART: mv0231 board initialisation from cfg file
UART: register I2C0 as master
UART:
UART: Registering i2c devices
UART: Registering i2c device rc5t619
UART: Polling address 0x32 on bus 2
UART: i2c_drv_minor: 10290
UART: i2c device rc5t619 registered at bus 2 with address 0x32
UART: ----------------------------------------
UART: Registering i2c devices
UART: Registering i2c device imx208_right
UART: Polling address 0x36 on bus 0
UART: i2c_drv_minor: 16438
UART: i2c device imx208_right registered at bus 0 with address 0x36
UART: ----------------------------------------
UART: Registering i2c device imx208_left
UART: Polling address 0x37 on bus 0
UART: i2c_drv_minor: 24631
UART: i2c device imx208_left registered at bus 0 with address 0x37
UART: ----------------------------------------
UART: Registering i2c device bmx055_accel
UART: Polling address 0x18 on bus 1
UART: i2c_drv_minor: -27
UART: Polling address 0x19 on bus 1
UART: i2c_drv_minor: 33817
UART: i2c device bmx055_accel registered at bus 1 with address 0x19
UART: ----------------------------------------
UART: Registering i2c device bmx055_gyro
UART: Polling address 0x68 on bus 1
UART: i2c_drv_minor: 42088
UART: i2c device bmx055_gyro registered at bus 1 with address 0x68
UART: ----------------------------------------
UART: Registering i2c device bmx055_magnet
UART: Polling address 0x10 on bus 1
UART: i2c_drv_minor: -27
UART: Polling address 0x11 on bus 1
UART: i2c_drv_minor: 50193
UART: i2c device bmx055_magnet registered at bus 1 with address 0x11
UART: ----------------------------------------
UART: Board version R1M0E0
UART: PMU switches configured.
UART:
```

```
UART: LED2 configured.
UART:
UART: LED3 configured.
UART:
UART: LED4 configured.
UART:
UART: Revision specific board configuration.
UART: LED2 set.
UART: DCDC set to 3.3V.
UART: WiFi nReset set.
UART: MV0231 board initialized successfully
UART: Failed to get a profile
UART: No profile found on index 0
UART:
UART: LeonRT Started.
UART:
UART: Configuring camera and datapath
UART: Initialising I2C0 in bare metal mode...
UART: Bare metal I2C initialisation complete
UART: Camera initialised!
UART: Camera started!
UART:
UART: Start
UART: Received 0
UART:
UART: Flash Handler: 8022D360
UART:
UART: Flash write : 1024
UART:
UART: Flash write : 1024

.....

UART: Flash write : 1024
UART:
UART: Flash write : 1024
UART:
UART: Flash write : 1024
UART:
UART: Flash write : 1024
UART:
UART: Flash write : 1024
UART:
UART: Flash write : 1024
UART:
UART: Flash write : 1024
UART:
UART: Flash write : 60
UART:
UART: Closing file
```

Running the bootloader (DIP switch 1 in Application position):

After flashing an application using the Control Application, it is possible to execute it by changing the DIP switch 1 to OFF position. The following output show this process after flashing the RTSP application as an example:

```
Movidius Debugger (moviDebug2) v00.83.2 Build 121447
Loading Tcl library...
Loading debug library...
Connecting to 127.0.0.1:30001
Connection         : OK [127.0.0.1:30001 | Olimex ARM-USB-TINY-H @
3000kHz]
Target Voltage  : OK
JTAG IDCODE     : ma2x5x                  [0x1C450661]
Starting TCF agent...
breset
Leon halt status: ok
startupcore LOS
target LOS
loadfile ./output/bootloader.elf
Loading section <.text> start = 0x701E0000 length = 70068 bytes.
Loading section <.text.eh_frame> start = 0x701F11C0 length = 128
bytes.
Loading section <.dtors> start = 0x701F1240 length = 4 bytes.
Loading section <.rodata> start = 0x701F1248 length = 7560 bytes.
Loading section <.init> start = 0x701F2FD0 length = 56 bytes.
Loading section <.fini> start = 0x701F3008 length = 40 bytes.
Loading section <.data> start = 0x701F3030 length = 1364 bytes.
Loading section <.heapSection> start = 0x701F3590 length = 1024
bytes.
Total bytes loaded = 80244.
runw
UART: Starting mv0231 board initialisation.
UART: mv0231 board initialisation from cfg file
UART: register I2C0 as master
UART:
UART: Registering i2c devices
UART: Registering i2c device rc5t619
UART: Polling address 0x32 on bus 2
UART: i2c_drv_minor: 10290
UART: i2c device rc5t619 registered at bus 2 with address 0x32
UART: ----------------------------------------
UART: Registering i2c devices
UART: Registering i2c device imx208_right
UART: Polling address 0x36 on bus 0
UART: i2c_drv_minor: 16438
UART: i2c device imx208_right registered at bus 0 with address 0x36
UART: ----------------------------------------
UART: Registering i2c device imx208_left
UART: Polling address 0x37 on bus 0
UART: i2c_drv_minor: 24631
UART: i2c device imx208_left registered at bus 0 with address 0x37
UART: ----------------------------------------
UART: Registering i2c device bmx055_accel
UART: Polling address 0x18 on bus 1
UART: i2c_drv_minor: -27
UART: Polling address 0x19 on bus 1
UART: i2c_drv_minor: 33817
UART: i2c device bmx055_accel registered at bus 1 with address 0x19
UART: ----------------------------------------
UART: Registering i2c device bmx055_gyro
```

```
UART: Polling address 0x68 on bus 1
UART: i2c_drv_minor: 42088
UART: i2c device bmx055_gyro registered at bus 1 with address 0x68
UART: ----------------------------------------
UART: Registering i2c device bmx055_magnet
UART: Polling address 0x10 on bus 1
UART: i2c_drv_minor: -27
UART: Polling address 0x11 on bus 1
UART: i2c_drv_minor: 50193
UART: i2c device bmx055_magnet registered at bus 1 with address 0x11
UART: ----------------------------------------
UART: Board version R1M0E0
UART: PMU switches configured.
UART:
UART: LED2 configured.
UART:
UART: LED3 configured.
UART:
UART: LED4 configured.
UART:
UART: Revision specific board configuration.
UART: LED2 set.
UART: DCDC set to 3.3V.
UART: WiFi nReset set.
UART: MV0231 board initialized successfully
UART:
UART: Generating the AP...
UART: Failed to get a profile
UART: No profile found on index 0
UART: AP generated
UART: Waiting AP connections
UART: Client connected to the AP
UART:
UART: LeonRT Started.
UART:
UART: Configuring camera and datapath
UART: Initialising I2C0 in bare metal mode...
UART: Bare metal I2C initialisation complete
UART: Camera initialised!
UART: Camera started!
UART:
UART: Streaming ...
UART:
UART:
UART: Thread camera created
UART: RTSPServer is running
UART: Message Received:
UART:  OPTIONS rtsp://192.168.1.1:8554/mjpej/2 RTSP/1.0
UART: CSeq: 2
UART: User-Agent: LibVLC/2.2.2 (LIVE555 Streaming Media v2016.02.09)
UART:
UART:
UART:
UART: Message Send:
UART:  RTSP/1.0 200 OK
```

```
UART: CSeq: 2
UART: Public: DESCRIBE, SETUP, TEARDOWN, PLAY, PAUSE
UART:
UART:
UART: Message Received:
UART:  DESCRIBE rtsp://192.168.1.1:8554/mjpej/2 RTSP/1.0
UART: CSeq: 3
UART: User-Agent: LibVLC/2.2.2 (LIVE555 Streaming Media v2016.02.09)
UART: Accept: application/sdp
UART:
UART:
UART:
UART: Message Send:
UART:  RTSP/1.0 200 OK
UART: CSeq: 3
UART: Content-Base: rtsp://192.168.1.1:8554
UART: Content-Type: application/sdp
UART: Content-Length: 93
UART:
UART: v=0
UART: o=- 1481765933 1 IN IP4 192.168.1.1
UART: s=
UART: t=0 0
UART: m=video 0 RTP/AVP 26
UART: c=IN IP4 0.0.0.0
UART:
UART: Message Send:
UART:  RTSP/1.0 200 OK
UART: CSeq: 4
UART: Transport: RTP/AVP/TCP;unicast;interleaved=0-1
UART: Session: -805308858
UART:
UART:
UART: Message Send:
UART:  RTSP/1.0 200 OK
UART: CSeq: 5
UART: Range: npt=0.000-
UART: Session: -805308858
UART: RTP-Info: url=rtsp://127.0.0.1:8554/mjpeg/1/track1
UART:
UART:
UART: Message Received:
UART:  TEARDOWN rtsp://192.168.1.1:8554 RTSP/1.0
UART: CSeq: 6
UART: User-Agent: LibVLC/2.2.2 (LIVE555 Streaming Media v2016.02.09)
UART: Session: -805308858
```

7. CONCLUSIONS

This deliverable describes software tests performed in the EoT factor-form board. The tests were divided into: 1) Firmware tests, i.e. unit tests for the different modules developed as part of EoT's firmware and 2) Integration tests, i.e. non-unit tests that check that major modules can work together. The results of these tests were all satisfactory.

8. GLOSSARY

AON	Always ON
AP	Access Point
API	Application Programming Interface
BRISK	Binary Robust Invariant Scalable Keypoints
BSD	Berkeley Software Distribution
CISC	Complex Instruction Set Computing
CNN	Convolutional Neural Network
CSS	CPU Subs-System
DDR	Double Data Rate
DIP	Dual in-line Package
DMA	Direct Memory Access
DSS	DDR Sub-System
DoW	Description of Work
GPIO	General Purpose Input/Output
GPL	General Public License
HD	Hard Disk
HOG	Histogram of Oriented Gradients
HTTP	Hypertext Transfer Protocol
HW	Hardware
I2C	Inter-Integrated Circuit
I2S	Integrated Interchip Sound
IO	Input/Output
ISR	Interrupt Service Routine
JPEG	Joint Photographic Experts Group
LED	Light-Emitting Diode
LK	Lucas-Kanade
MAC	Media Access Control
MDK	Movidius Development Kit
MIPI	Mobile Industry Processor Interface
MQTT	Message Query Telemetry Transport
MSS	Media Sub-System
MvCv	Movidius Computer Vision
OS	Operating System
PC	Personal Computer
PDF	Portable Document Format
PMB	Processor Memory Block
PNG	Portable Network Graphics
POSIX	Portable Operating System Interface
QR	Quick Response
REST	Representational State Transfer
RISC	Reduced Instruction Set Computing
RTEMS	Real-Time Executive for Multiprocessor Systems
RTP	Real-Time Transport Protocol
RTSP	Real-Time Streaming Protocol
SD	Secure Digital
SIFT	Scale-Invariant Feature Transform

SoC	System on Chip
SPI	Serial Peripheral Interface
SVM	Support Vector Machines
SW	Software
TCP/IP	Transmission Control Protocol / Internet Protocol
TI	Texas Instruments
VPU	Vision Processing Unit
WAV	Waveform Audio File
WEP	Wired Equivalent Privacy
WPA	WiFi Protected Access

- End of document -