

H3A Sample Application Usage Note

This document provides an overview about how to execute H3A sample examples and how to verify statistics generated by H3A engine.

AF Sample Example:

(AF sample application will configure engine with A Law disabled, HMF disabled, GR_BG_BAYER for pixel location and SUM mode for focus value accumulator mode).

1. For hardware setup, Connect XDS 510 USB emulator to the JTAG connector on DM6437 board. Switch on the power supply for board.
2. Open CCS 3.3 setup. Import EVMDM6437_XDS510USB configuration. Click on "Save & Quit" button and exit the setup.
3. Open CCS3.3. This will open "CCStudio: Parallel Debug Manager" window. Connect C64PLUS and open the CCS window.
4. Build the VPFE library with PSP_VIDEO_PATH_ENABLE Macro enabled in psp_vpfe.h.
5. Before running the sample application of Auto Focus run the previewer on the fly application to ensure that sensor is configured correctly and input image is stable.
6. Open
`<root>\packages\ti\sdo\pspdrivers\system\dm6437\bios\evmDM6437\video\sample\build\h3a\psp_bios_af_st_example.pjt.`
7. Compile this project using Project->Build.
8. Load the generated .out file from
`<root>\packages\ti\sdo\pspdrivers\system\dm6437\bios\evmDM6437\video\sample\build\bin\Debug` or
`<root>\packages\ti\sdo\pspdrivers\system\dm6437\bios\evmDM6437\video\sample\build\bin\Release` folder and execute it.
9. After completion of this application execution, "AF Sample Test successful" message will be printed on DSP/BIOS -> Message Log.
10. Output file "af_st_sample.txt" containing statistics generated by Auto Focus engine will be stored in
`<root>\packages\ti\sdo\pspdrivers\system\dm6437\bios\evmDM6437\video\sample\build\bin\Debug` or
`<root>\packages\ti\sdo\pspdrivers\system\dm6437\bios\evmDM6437\video\sample\build\bin\Release` folder.
11. Statistics generated by AF engine for one sample image is as shown below in figure 1.

```

3974    <- Green Sum                                (paxel 0)
122     <- Peak sum from IIR filter 0 (green)      (paxel 0)
1496    <- Peak sum from IIR filter 1 (green)      (paxel 0)
0
1622    <- Red Sum                                  (paxel 0)
62      <- Peak sum from IIR filter 0 (red)         (paxel 0)
831     <- Peak sum from IIR filter 1 (red)         (paxel 0)
0
1532    <- Blue Sum                                 (paxel 0)
115     <- Peak sum from IIR filter 0 (blue)        (paxel 0)
881     <- Peak sum from IIR filter 1 (blue)        (paxel 0)
0
4009    <- Green Sum                                (paxel 1)
163     <- Peak sum from IIR filter 0 (green)      (paxel 1)
372     <- Peak sum from IIR filter 1 (green)      (paxel 1)
0                                              1
1599    <- Red Sum                                  (paxel 1)
69      <- Peak sum from IIR filter 0 (red)         (paxel 1)
202     <- Peak sum from IIR filter 1 (red)         (paxel 1)
0                                              1
1512    <- Blue Sum                                 (paxel 1)
108     <- Peak sum from IIR filter 0 (blue)        (paxel 1)
210     <- Peak sum from IIR filter 1 (blue)        (paxel 1)
0

```

And so on for Paxel 2, 3, etc.

Figure 1. AF output for one sample image

12. After running the sample application of Auto Focus, execute the previewer on the fly application to ensure output image is still stable.

Note:

For Whiter Images (where pixel values are high), with ALAW enabled, sum values difference is noticeable. For other normal images, difference might be negative.

AEW Sample Example:

(AEW sample application will configure engine with A Law disabled and saturation limit as 450)

1. For hardware setup, Connect XDS 510 USB emulator to the JTAG connector on DM6437 board. Switch on the power supply for board.
2. Open CCS 3.3 setup. Import EVMDM6437_XDS510USB configuration. Click on "Save & Quit" button and exit the setup.
3. Open CCS3.3. This will open "CCStudio: Parallel Debug Manager" window. Connect C64PLUS and open the CCS window.
4. Build the VPFE library with PSP_VIDEO_PATH_ENABLE Macro enabled in psp_vpfe.h.
5. Before running the sample application of Auto Exposure/Auto White Balance run the previewer on the fly application to ensure that sensor is configured correctly and input image is stable.
6. Adjust camera so as to focus on "Red" color paper or object.

7. Open
`<root>\packages\ti\sdo\pspdrivers\system\dm6437\bios\evmDM6437\video\sample\build\h3a\psp_bios_aew_st_example.pjt.`
8. Compile this project using Project->Build.
9. Load the generated .out file from
`<root>\packages\ti\sdo\pspdrivers\system\dm6437\bios\evmDM6437\video\sample\build\bin\Debug`
or
`<root>\packages\ti\sdo\pspdrivers\system\dm6437\bios\evmDM6437\video\sample\build\bin\Release` folder and execute it.
10. After completion of this application execution, "AEW Sample Test successful" message will be printed on DSP/BIOS -> Message Log.
11. Output file "aew_st_sample.txt" containing statistics generated by Auto Exposure / Auto White Balance engine will be stored in
`<root>\packages\ti\sdo\pspdrivers\system\dm6437\bios\evmDM6437\video\sample\build\bin\Debug`
or
`<root>\packages\ti\sdo\pspdrivers\system\dm6437\bios\evmDM6437\video\sample\build\bin\Release` folder.
12. Statistics generated by AEW engine for "Red" image is as shown below in figure 2.

```

1496  <- Green sub sample accumulator (Window 0 data)
2057  <- Red   sub sample accumulator (Window 0 data)
1068  <- Blue  sub sample accumulator (Window 0 data)
1468  <- Green sub sample accumulator (Window 0 data)
1496  <- Green saturator accumulator (Window 0 data)
1920  <- Red   saturator accumulator (Window 0 data)
1068  <- Blue  saturator accumulator (Window 0 data)
1468  <- Green saturator accumulator (Window 0 data)
1488  <- Green sub sample accumulator (Window 1 data)
2056  <- Red   sub sample accumulator (Window 1 data)
1064  <- Blue  sub sample accumulator (Window 1 data)
1478  <- Green sub sample accumulator (Window 1 data)
1488  <- Green saturator accumulator (Window 1 data)
1920  <- Red   saturator accumulator (Window 1 data)
1064  <- Blue  saturator accumulator (Window 1 data)
1478  <- Green saturator accumulator (Window 1 data)
1494
2071
1038
1458
1494
1920
1038
1458

```

Figure 2. AEW output for red image

13. After running the sample application of Auto White Exposure/Auto White Balance, execute the previewer on the fly application to ensure output image is still stable.

14. Verify that value of "Red sub sample accumulator" (as shown in figure 2) is higher as compared to "Green" and "Blue" sub sample accumulators for the "Red" input image.
15. For the "Green" image "Green sub sample accumulator" will be higher as compared to "Red" and "Blue" sub sample accumulators.

Note:

For Whiter Images (where pixel values are high), with ALAW enabled, sum values difference is noticeable. For other normal images, difference might be negative.