MCV Frame Buffer Device Quick Start Documentation

Release 3

Wolfgang Grandegger

November 13, 2019

CONTENTS

CHAPTER

ONE

ABOUT THIS MANUAL

1.1 Imprint

Adress:

ARIES Embedded GmbH

Schöngeisinger Str. 84

D-82256 Fürstenfeldbruck

Germany

Phone:

+49 (0) 8141/36 367-0

Fax:

+49 (0) 8141/36 367-67

1.2 Disclaimer

ARIES Embedded does not guarantee that the information in this document is up-to-date, correct, complete or of good quality. Liability claims against ARIES Embedded, referring to material or non-material related damages caused, due to usage or non-usage of the information given in this document, or due to usage of erroneous or incomplete information, are exempted, as long as there is no proven intentional or negligent fault of ARIES Embedded. ARIES Embedded explicitly reserves the rights to change or add to the contents of this Preliminary User's Manual or parts of it without notification.

1.3 Copyright

1.4 Registered Trademarks

1.5 Care and Maintenance

- Keep the device dry. Precipitation, humidity, and all types of liquids or moisture can contain minerals that will corrode electronic circuits. If your device does get wet, allow it to dry completely.
- Do not use or store the device in dusty, dirty areas. Its moving parts and electronic components can be damaged.
- Do not store the device in hot areas. High temperatures can shorten the life of electronic devices, damage batteries, and warp or melt certain plastics.
- Do not store the device in cold areas. When the device returns to its normal temperature, moisture can form inside the device and damage electronic circuit boards.
- Do not attempt to open the device.
- Do not drop, knock, or shake the device. Rough handling can break internal circuit boards and fine mechanics.
- Do not use harsh chemicals, cleaning solvents, or strong detergents to clean the device.
- Do not paint the device. Paint can clog the moving parts and prevent proper operation.
- Unauthorized modifications or attachments could damage the device and may violate regulations governing radio devices.

CHAPTER

TWO

IP CORE "VIP FRAME BUFFER II" FOR THE ARIES EMBEDDED MCVEVP REV.2 BOARD

Aries Embedded offers a solution to realize a frame buffer device in the FPGA of the MCV [1] module using Intel's IP core "VIP Frame Buffer II". This application note describes how to get it working on a MxxDK [2] 7 inch TFT display with a capacitive touch sensor mounted on the MCVEVP rev. 2 board.

This guide assumes that

- MCV is connected via serial console to your host PC
- MCV is accessible via network and that you can logon to the device using i.e. ssh
- the USB port of your MCVEVP is connected to a USB port of your host PC
- the MxxDK 7" TFT display with capacitive touch is properly installed on your MCVEVP

2.1 Download the Source Code and Configure Options

The Yocto layer "meta-aries" [3] already has support for the frame buffer. Just download and setup the BSP as described in "meta-aries". But before you type *bitbake core-image-full-cmdline-aries*, you may want to include some QT5 example programs by adding the following lines to the end of *conf/local.conf*:

```
$ cat conf/local.conf
...
DISTRO_FEATURES_remove = "x11 wayland opengl pulseaudio opengles egl"

PACKAGECONFIG_DISTRO_append_pn-qtbase = " no-opengl linuxfb icu libinput fontconfig
mtdev examples"

IMAGE_INSTALL_append = " \
     dbus \
     openssl \
     icu \
     libudev \
     libinput \
     fontconfig \
     freetype \
     ttf-dejavu-sans \
     ttf-dejavu-sans-mono \
     ttf-dejavu-serif \
     ttf-dejavu-serif-condensed \
```

[1] https://www.aries-embedded.com/system-on-module/fpga/cyclone-v-intel-fpga-mcv-som-hps-altera-soc-pcie-transceiver
[2] https://www.aries-embedded.com/evaluation-kit/display/MxxDK-7-800x480px
[3] https://github.com/ARIES-Embedded/meta-aries

```
    ttf-dejavu-mathtexgyre \
    qtbase \
    qtbase-plugins \
    qtbase-examples \
"
```

2.2 Build and program the Image

Then build the image as usual:

```
$ bitbake core-image-full-cmdline-aries
```

Start your MCV and interrupt the boot process in U-Boot. Type

```
=> ums 0 mmc 0
```

to expose the eMMC memory of MCV as a device on your host PC by using the U-Boot USB Mass Storage gadget. If needed unmount the partitions related to the eMMC memory of MCV on your host PC.

Then, copy Yocto image onto the eMMC of MCV by using i.e.

```
$ sudo dd if=/<yourworkingdirectory>/core-image-full-cmdline-aries-
mcvevk-20191110174826.rootfs.wic of=/dev/sdb bs=512
```

! WARNING ! Using the wrong partition in this command may destroy the data on your host PC. Please adjust /dev/sdb accoringly for your machine.

Boot linux and login.

Copy the IP-file which contains the time-bomb IP binary to the /boot directory of your MCV SoM, i.e. using scp:

```
$ scp frmbuf_c2-2019-10-11.rbf root@192.168.0.10:/boot
```

! NOTE ! Please note, that for each FPGA size a dedicated rbf-file has to be used. This guide assumes that

- MCV-2DB and MCV-X2DB use the frmbuf_c2-2019-10-11.rbf file
- MCV-6DB and MCV-X6DB use the frmbuf_c6-2019-10-11.rbf file

Load the time-bomb IP binary *frmbuf_c2-2019-10-11.rbf* and configure U-Boot to boot Linux with the device tree blob *socfpga_cyclone5_mcvevk_fb.dtb*. This can be achieved with the following U-Boot environment setup:

```
=> setenv fpgafile frmbuf_c2-2019-10-11.rbf
=> setenv dtb_file socfpga_cyclone5_mcvevp_rev2_fb.dtb
=> setenv fpgaload 'mmc rescan; load mmc 0:2 ${loadaddr} ${fpgafile};
fpga load 0 ${loadaddr} ${filesize}'
=> setenv fbenable 'bridge enable; mw.l 0xffc25080 0x00003fff'
=> setenv mmc_mmc_fb 'run fpgaload fbenable mmcload mmcargs addargs;
bootm ${kernel_addr_r}:kernel@1 - ${kernel_addr_r}:fdt@${dtb_file}'
=> setenv bootcmd run mmc_mmc_fb
=> saveenv
```

Next boot the system with:

```
=> boot
```

2.3 Run a first Qt Application

And finally you can run QT5 example programs, e.g. the *fingerpaint* program demonstrates nicely the multi-touch functionality:

```
# export QT_QPA_PLATFORM=linuxfb
# cd /usr/share/examples/touch/fingerpaint
# ./fingerpaint &
```

Feel free to play with the other example programs as well.