MJPEG Video Streaming on Avnet MiniZed

Quick Start Guide

27 Sept 2017
Version 1.1

Introduction

This document is provided to help user to bring up the MJPEG Video Streaming demo on Avnet MiniZed board. The bring-up procedures here is targeted to Linux environment but the basic workflow should be similar in Windows platform.

Platform Setup

Software

- Ubuntu Linux v14.04.LTS 64-bit OS
  - Chrome browser installed
  - Xilinx Software command-Line Tools 2017.1 or later
  - FTDI USB/UART-JTAG driver

Hardware

- Ubuntu PC with 4G RAM
- Avnet MiniZed
- Avnet TDnext 1.26Mpixel Pmod Camera Kit
- 2xMicro-USB cable
- USB Thumb drive with 1GB FAT partition

Prerequisites

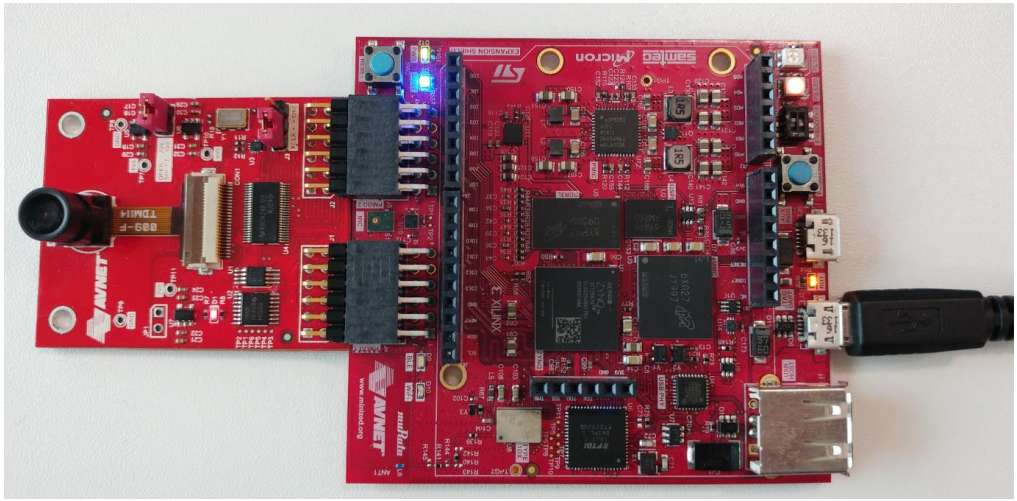
- Knowledge on general Linux operation
- Knowledge on using VI editor

© 2017 Avnet. All rights reserved. All trademarks and registered trademarks are the property of their respective owners. All specifications are subject to change without notice.

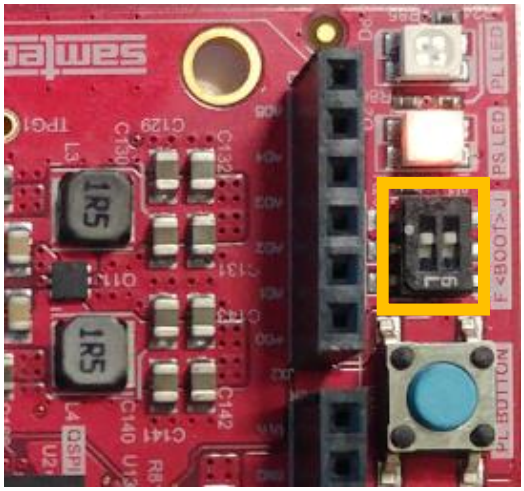
NOTICE OF DISCLAIMER: Avnet is providing this design, code, or information "as is." By providing the design, code, or information as one possible implementation of this feature, application, or standard, Avnet makes no representation that this implementation is free from any claims of infringement. You are responsible for obtaining any rights you may require for your implementation. Avnet expressly disclaims any warranty whatsoever with respect to the adequacy of the implementation, including but not limited to any warranties or representations that this implementation is free from claims of infringement and any implied warranties of merchantability or fitness for a particular purpose.

Procedures

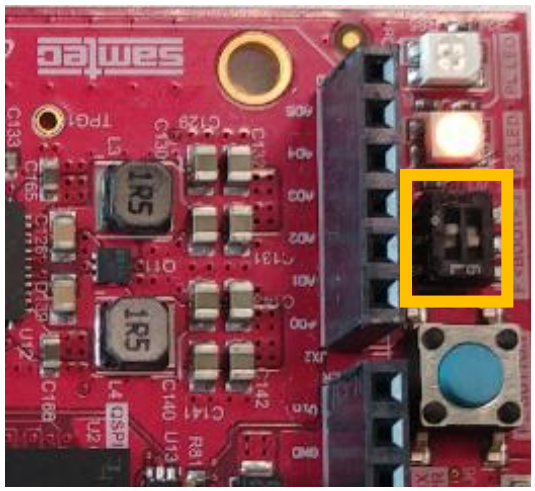
1. Assemble the hardware. Connect the TDnext Pmod Camera and the USB JTAG cable as shown below.



- Set JTAG boot mode



JTAG Boot Mode



QSPI Boot Mode

2. Copy and extract reference design archive (MiniZed_MJPEG_xxxxxxxx) to Ubuntu file system home directory. The folder structure is depicted as follow

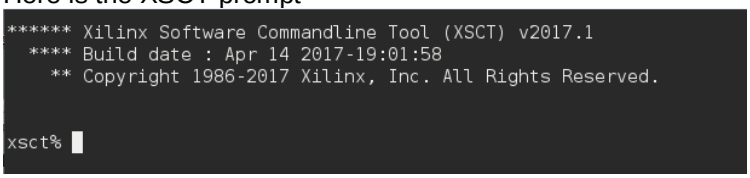
```
MiniZed_MJPEG_xxxxxxxx
    |--- boot
            |--- boot_QSPI.bin
            |--- smallboot.bin
    |--- image.ub
    |--- init.sh
    |--- mjpeg_demo
            |--- app
            |--- bcm_wifi
```

- File **image.ub** is the prebuilt of kernel image and root file system
- File **init.sh** is an auto-run script to start the demo after Linux boot up
- Folder **boot** contains boot code which is supposed to be flashed into QSPI through SDK + JTAG
- Folder **mjpeg_demo** contains application binary, WIFI device driver module and scripts

3. Flash the boot image **boot_QSPI.bin** to QSPI. This is a minimal boot image contains all necessary binaries to boot up a Linux system on MiniZed. We need this environment to transfer code to eMMC in later steps

- Set jumper to JTAG boot mode
- Launch Xilinx Software Command-line Tool (XSCT)
  # cd /opt/Xilinx/SDK/<version>/bin
  # ./xsct

  Here is the XSCT prompt



  Remark:
  - <version> is the tool version. In this tutorial, we're using XSCT 2017.1 and so <version> = 2017.1
  - Require XSCT version 2017.1 or later to work well with MiniZed JTAG chip

- Navigate to the flash image path **$HOME/ MiniZed_MJPEG_xxxxxxxx/boot**
  xsct% cd ~/ MiniZed_MJPEG_xxxxxxxx/boot

- Program the QSPI flash
  xsct% exec program_flash -f boot_QSPI.bin -offset 0 -flash_type qspi_single -verify -cable type xilinx_tcf url TCP:127.0.0.1:3121

  Remark:
  - The text should all go in one line and not have a line break in it. If you want to copy and paste the text into Ubuntu command window, please use the below command in one line

```
exec program_flash -f boot_QSPI.bin -offset 0 -flash_type qspi_single -verify -cable type xilinx_tcf url TCP:127.0.0.1:3121
```

Here is the prompt messages after successful programming

```
  **** SW Build 1846317 on Fri Apr 14 18:54:47 MDT 2017
    ** Copyright 1986-2017 Xilinx, Inc. All Rights Reserved.

Connecting to hw_server @ TCP:127.0.0.1:3121

WARNING: Failed to connect to hw_server at TCP:127.0.0.1:3121
Attempting to launch hw_server at TCP:127.0.0.1:3121

Connected to hw_server @ TCP:127.0.0.1:3121
Available targets and devices:
Target 0 : jsn-openjtag2-1234-oj1A
        Device 0: jsn-openjtag2-1234-oj1A-4ba00477-0

Retrieving Flash info...

Initialization done, programming the memory
BOOT_MODE REG = 0x00000000
f probe 0 0 0
Performing Erase Operation...
Erase Operation successful.
INFO: [Xicom 50-44] Elapsed time = 5 sec.
Performing Program Operation...
0%...10%...100%
Program Operation successful.
INFO: [Xicom 50-44] Elapsed time = 15 sec.
Performing Verify Operation...
0%...10%...50%...60%...100%
INFO: [Xicom 50-44] Elapsed time = 21 sec.
Verify Operation successful.

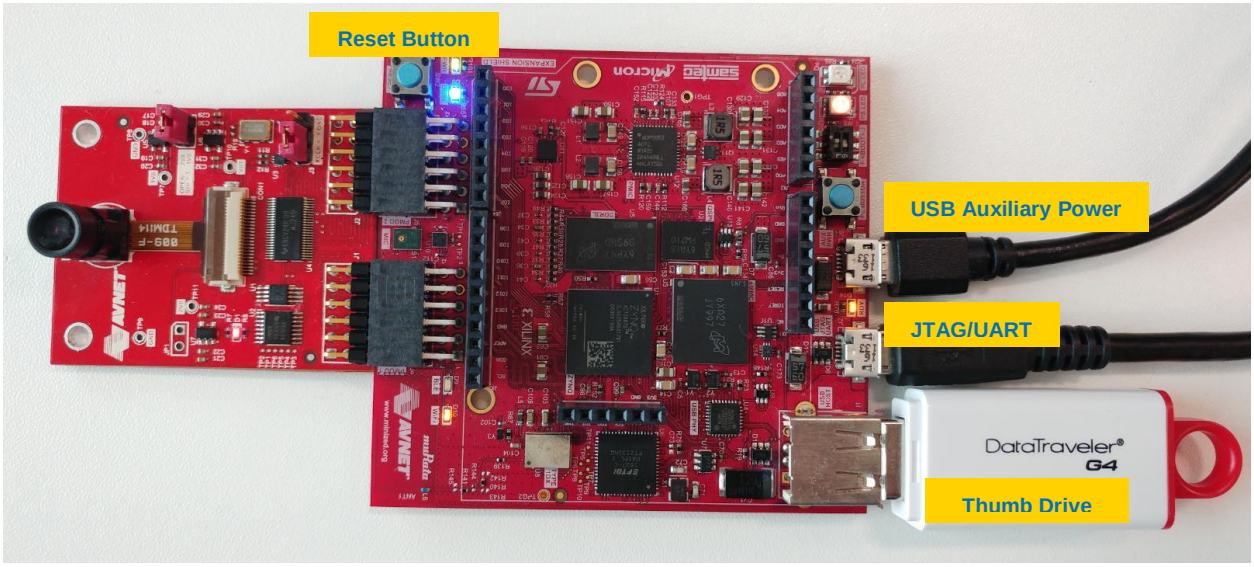
Flash Operation Successful
xsct%
```

- At this point, MiniZed is ready to boot from QSPI
- Set jumper to QSPI boot mode
- Cycle the power and verify that you can see Linux boot up message in uart console.
- If everything all right, we can proceed to next step.

4. System can boot up properly and we can access the eMMC on MiniZed. It's the time to put all necessary binaries for this demo into the USB thumb drive. We can then transfer them from the thumb drive to the eMMC card.

   - Copy the follow files and folder to the root of the USB thumb drive
     - image.ub
     - init.sh
     - mjpeg_demo

   - Connect an extra USB cable for auxiliary power input and plug in the USB thumb drive to MiniZed. Press the Reset button



- Waiting for the Linux boot up end. Login with the below credentials
  Login: **root**
  Password: **root**

- Transfer files from USB thumb drive to eMMC

  The mount point of USB thumb drive is
  /run/media/sda1

  The mount point of eMMC is
  /run/media/mmcblk1p1

  Copying with the below commands

```
# cd /run/media/sda1
# cp image.ub /run/media/mmcblk1p1
# cp init.sh /run/media/mmcblk1p1
# cp -rf mjpeg_demo /run/media/mmcblk1p1
# sync
```

  Remark: You can optionally backup the original binaries in the eMMC card

- Shutdown system properly

```
# shutdown -h now
```

  At this point, eMMC contains the kernel image, file system, app binary and driver module for our demo. We are almost done. The next step is to update boot code so that system can boot up with the latest binaries.

5. Flash the QSPI with the actual demo boot image
   - Remove the USB thumb drive
   - Set jumper to JTAG boot mode
   - Keep the auxiliary USB power cable
   - Keep USB JTAG cable
   - Reset MiniZed

- Flash the QSPI again but this time we use another image
  $HOME/ MiniZed_MJPEG_xxxxxxxx/boot/smallboot.bin

- Launch Xilinx Software Command-line Tool (XSCT)
  # cd /opt/Xilinx/SDK/<version>/bin
  # ./xsct

- Navigate to the flash image path **$HOME/ MiniZed_MJPEG_xxxxxxxx/boot**
  xsct% cd ~/ MiniZed_MJPEG_xxxxxxxx/boot
  xsct% exec program_flash -f smallboot.bin -offset 0 -flash_type qspi_single -verify -cable type xilinx_tcf url TCP:127.0.0.1:3121

  Remark:
  - The text should all go in one line and not have a line break in it. If you want copy and paste the text into Ubuntu command window, please use the below command in one line

```
exec program_flash -f smallboot.bin -offset 0 -flash_type qspi_single -verify -cable type xilinx_tcf url TCP:127.0.0.1:3121
```

- At this point, we completed hardware setup. Hardware has been loaded with all necessary binaries to start the demo
- Set jumper to QSPI boot mode
- We still have the last step - configure the WIFI Access Point

6. Configure the WIFI network parameters. There are two ways to go. You could either modify the AP setting of your WIFI network or the default setting on MiniZed

   [Modify the AP setting]
   - MiniZed is hardcoded to connect to AP with the below configuration. Please modify your AP setting accordingly

     SSID : Level7
     Security : disable
     DHCP : enable

   [Modify the default setting on MiniZed]

   The configuration file is stored in MiniZed file system. It's located at
   /run/media/mmcblk1p1/mjpeg_demo/bcm_wifi/wpa_supplicant.conf

   - Cycle the power of MiniZed
   - Waiting for the Linux boot up end. Login with the below credentials
     Login: **root**
     Password: **root**

   - Navigate to the configuration file location
     # cd /run/media/mmcblk1p1/mjpeg_demo/bcm_wifi

- Use vi editor to open and modify the file **wpa_supplicant.conf**

  Example: with security enabled

  ```
  network={
          key_mgmt=WPA-PSK
          ssid="Level7"
          psk="0026556824"
  }
  ```

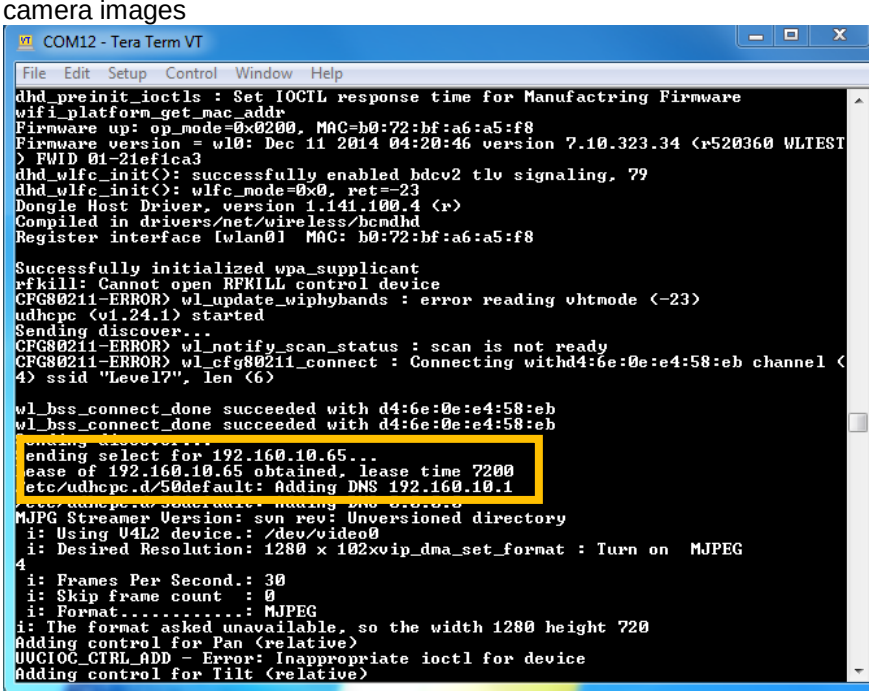
  Example: with security disabled

  ```
  network={
          key_mgmt=NONE
          ssid="Level7"
  }
  ```

  For demo purpose, it's recommended to disable WIFI security. The video streaming performance is better
- Save the file
- Make sure new modification is written to eMMC
  # **sync**

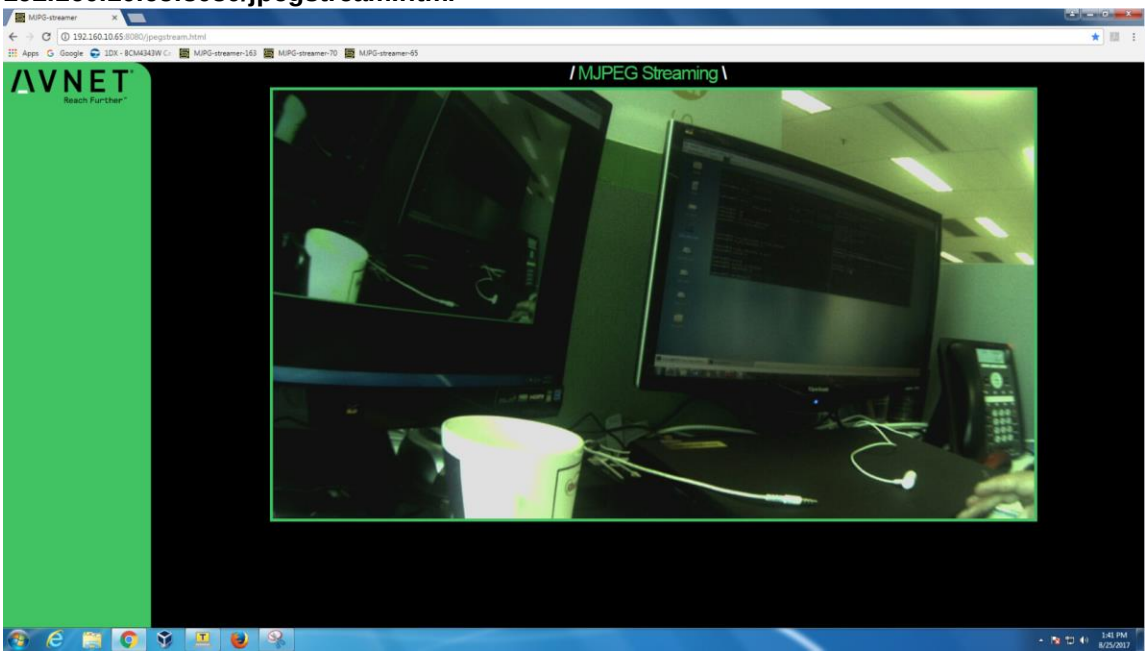
7. Kick off the demo
   - Cycle the power of MiniZed
   - Waiting for the Linux boot up end. Login with the below credentials
     Login: **root**
     Password: **root**
   - Navigate to demo application location and start
     # cd /run/media/mmcblk1p1/mjpeg_demo
     # ./run.sh

- Waiting wireless lan up and grab the IP address assigned to MiniZed from the console message Here we find **192.160.10.65**. This is the IP address of the webserver which streams out the camera images



- In PC side, launch Chrome browser and put the webserver IP address

192.160.10.65:8080/jpegstream.html



Remark:

- This reference design is developed and tested on Chrome browser. It's highly recommend to kick off the demo with the same browser. It may work on others but confirmed not on Internet Explorer.

- Enjoy!

Revision History

Date	Version	Revision
4-Sep-17	1.0	Initial Release
27-Sep-17	1.1	Corrected typos on Page 6 & 7 Added remarks for the QSPI program command line Amended remark on supported web browser