IEC Type 1 Hardware Setup

- 1.Toolchain
- 2.u-boot
  - Build from source – Bootloader
  - Update the Bootloader
- 3.Creating filesystem
  - Creating Ubuntu filesystem
  - Boot from removable storage
- 4. Compile kernel & application
  - Compile on MACCHIATObin board
    - Preparing
    - Compile kernel
    - compile MUSDK
    - compile dpdk
- 5. Other

Our board is configured to boot from SPI ROM. This can be verified by checking the Boot source DIP switches, and by seeing Active boot device: SPI NOR flash during boot.

http://wiki.macchiatobin.net/tiki-index.php?page=MACCHIATObin+Interface+list

blocked URL

1.Toolchain

http://wiki.macchiatobin.net/tiki-index.php?page=Toolchain+Installation

recommended toolchain version: gcc-linaro-7.3.1-2018.05-x86_64_aarch64-linux-gnu.tar.xz

```
wget https://releases.linaro.org/components/toolchain/binaries/7.3-2018.05/aarch64-linux-gnu/gcc-linaro-7.3.1-2018.05-x86_64_aarch64-linux-gnu.tar.xz

tar xf gcc-linaro-7.3.1-2018.05-x86_64_aarch64-linux-gnu.tar.xz

# set PATH

# export PATH=$PATH:/home/mcbin/toolchain/gcc-linaro-5.3.1-2016.05-x86_64_aarch64-linux-gnu/bin
```

2.u-boot

Build from source – Bootloader

http://wiki.macchiatobin.net/tiki-index.php?page=Build+from+source+-+Bootloader

Compile according to the instructions given by the web page

recommended version: u-boot-2018.03-armada-18.09

Update the Bootloader

http://wiki.macchiatobin.net/tiki-index.php?page=Update+the+Bootloader

This page is about update boot through network, In addition, you can use the USB Flash drive to update boot as following instructions:

```
Marvell>> usb reset

Marvell>> bubt flash-image.bin spi usb

Marvell>> reset

Marvell>> env default -a

Marvell>> env save
```

Make sure that the MACCHIATObin board does not experience power loss during the entire updating process, otherwise it will be bricked due to an unfinished bootloader update.

3.Creating filesystem

Creating Ubuntu filesystem

http://wiki.macchiatobin.net/tiki-index.php?page=Creating+Ubuntu+filesystem&highlight=file+system

Among one of file systems supported by MACCHIATObin is the Ubuntu file system. you can either build the file system manually or download a prebuilt SD card image.

http://macchiatobin.net/software/

This software page provide a compiled kernel image. If want to compile kernel on MACCHIATObin board, You can use this official image temporarily to start the board.

blocked URL

Boot from removable storage

Please see following link as reference about how to setup Marvell board

http://wiki.macchiatobin.net/tiki-index.php?page=Boot+from+removable+storage+-+Ubuntu&highlight=removable

Instructions of the page is suit for 17.10,

U-Boot 18.09:

Using Micro SD card:

```
Marvell>> setenv image_name boot/Image

Marvell>> setenv fdt_name boot/armada-8040-mcbin.dtb

Marvell>> setenv bootcmd 'mmc dev 1; ext4load mmc 1:1 $kernel_addr_r $image_name;ext4load mmc 1:1 $fdt_addr_r $fdt_name;setenv bootargs $console root=/dev/mmcblk1p1 rw rootwait pci=pcie_bus_safe cpuidle.off=1; booti $kernel_addr_r - $fdt_addr_r'

Marvell>>  saveenv

Marvell>>  run bootmmc
```

Using USB Stick:

```
Marvell>> setenv image_name boot/Image

Marvell>> setenv fdt_name boot/armada-8040-mcbin.dtb

Marvell>> setenv bootusb 'usb reset; ext4load usb 0:1 $kernel_addr_r $image_name;ext4load usb 0:1 $fdt_addr_r $fdt_name;setenv bootargs $console root=/dev/sda1 rw rootwait pci=pcie_bus_safe cpuidle.off=1;booti $kernel_addr_r - $fdt_addr_r'

Marvell>>  saveenv

Marvell>>  run bootusb
```

If U-Boot version is 17.10, you should repleace $kernel_addr_r/$fdt_addr_r by $kernel_addr/$fdt_addr

4. Compile kernel & application

Compile on MACCHIATObin board

Preparing

The /etc/apt/sources.list file that comes with the rootfs is very bare. Replace it with this:

```
deb http://ports.ubuntu.com/ubuntu-ports/ xenial main universe restricted multiverse

deb-src http://ports.ubuntu.com/ubuntu-ports/ xenial main universe restricted multiverse


deb http://ports.ubuntu.com/ubuntu-ports/ xenial-updates main universe restricted multiverse

deb-src http://ports.ubuntu.com/ubuntu-ports/ xenial-updates main universe restricted multiverse


deb http://ports.ubuntu.com/ubuntu-ports/ xenial-backports main restricted universe multiverse

deb-src http://ports.ubuntu.com/ubuntu-ports/ xenial-backports main restricted universe multiverse


deb http://ports.ubuntu.com/ubuntu-ports/ xenial-security main restricted universe multiverse

deb-src http://ports.ubuntu.com/ubuntu-ports/ xenial-security main restricted universe multiverse


deb http://ports.ubuntu.com/ubuntu-ports/ xenial-proposed main restricted universe multiverse

deb-src http://ports.ubuntu.com/ubuntu-ports/ xenial-proposed main restricted universe multiverse


deb http://ports.ubuntu.com/ubuntu-ports/ zesty main restricted universe multiverse
```

Make sure to run "apt update" after updating this file.

Install tools:

```
apt-get install git
apt-get install automake
apt-get install libtool
apt-get install pkg-config
apt-get install python dmidecode
apt-get install libelf-dev libdw-dev libunwind-dev libaudit-dev libslang2-dev binutils-dev libiberty-dev
apt-get install pciutils
apt-get install lshw
apt-get install openssh-server
apt-get install ftp
apt-get install libnuma-dev
```

Compile kernel

1) make a new directory

mkdir -p /home/code/kernel/4.14.22

2) download kernel source from git

```
/home/code/kernel/4.14.22# git clone https://github.com/MarvellEmbeddedProcessors/linux-marvell .
/home/code/kernel/4.14.22# git checkout linux-4.14.22-armada-18.09
```

3) Download musdk source

```
/home/code# mkdir musdk

/home/code/musdk#git clone https://github.com/MarvellEmbeddedProcessors/musdk-marvell .
/home/code/musdk#git checkout musdk-armada-18.09
```

4) Patches for the kernel source

```
/home/code/musdk# cd /home/code/kernel/4.14.22/
/home/code/kernel/4.14.22# git am ~/musdk/patches/linux-4.14/*.patch
/home/code/kernel/4.14.22# git am /home/code/patch/kernel/0001-arm64-dts-marvell-mcbin-enable-both-cp110-crypto-eng.patch
```

0001-arm64-dts-marvell-mcbin-enable-both-cp110-crypto-eng.patch be putted in “Cross compile.zip”.

5) Compile kernel

```
export ARCH=arm64
make mrproper
make mvebu_v8_lsp_defconfig
make -j$(($(nproc)+1))
```

6) Copy image&dtb

```
/home/code/kernel/4.14.22# cp ./arch/arm64/boot/Image /boot/

/home/code/kernel/4.14.22# cp ./arch/arm64/boot/dts/marvell/armada-8040-mcbin.dtb  /boot/
```

compile MUSDK

```
mkdir /home/code/musdk -p

cd /home/code/musdk

 /home/code/musdk#  export  KDIR=/home/code/kernel/4.14.22/

/home/code/musdk# sed -i -e 's/O_CREAT/O_CREAT, S_IRUSR | S_IWUSR/' src/lib/file_utils.c

/home/code/musdk# sed -i -e 's/marvell,mv-pp-uio/generic-uio/' modules/pp2/mv_pp_uio.c
```

blocked URL

build kernel modules:

```
cd  modules/cma/

make

cd modules/dmax2/

make

cd modules/neta/

 make

cd modules/pp2/

make

cd .modules/sam/

make
```

Build musdk software:

```
/home/code/musdk# mkdir /home/code/musdk-bin/ -p

/home/code/musdk# ./bootstrap

/home/code/musdk# ./configure --enable-sam

/home/code/musdk# make -j5 install DESTDIR=/home/code/musdk-bin/
```

If you meet following error:

blocked URL

blocked URL

compile dpdk

```
cd /home/code/

git clone https://github.com/DPDK/dpdk.git

cd dpdk

git  checkout v18.11

export LIBMUSDK_PATH=/home/code/musdk-bin/home/code/musdk/usr/local/

export RTE_KERNELDIR=/home/code/kernel/4.14.22/

export RTE_TARGET=arm64-armv8a-linuxapp-gcc

export RTE_SDK=$PWD

make config T=arm64-armv8a-linuxapp-gcc

sed -i "s/MVPP2_PMD=n/MVPP2_PMD=y/" build/.config

sed -i "s/MVSAM_CRYPTO=n/MVSAM_CRYPTO=y/" build/.config

make -j5
```

5. Other

Script is provided to facilitate build of the kernel image, the developer needs to run with root privileges:

https://gerrit.akraino.org/r/gitweb?p=iec.git;a=blob;f=misc/type1/macbin/setup-macbin-kernel.sh;hb=HEAD

Marvell provides guidance on the build toolchain, file system and bootloader, which can be found at the link below:

http://wiki.macchiatobin.net/tiki-index.php?page=Wiki+Home