

PICO-PI-IMX7

This is a page about the Freescale based i.MX7; PICO-PI-IMX7.

- Availability
- Vendor Documentation
- Basic Requirements
- imx_usb_loader
- ARM Cross Compiler: GCC
- Bootloader: U-Boot
- Bootloader: imx_usb_loader
- Linux Kernel
- Root File System
 - Debian 9
 - Ubuntu 18.04 LTS
- Setup microSD card
- Install Kernel and Root File System
 - Copy Root File System
 - Setup extlinux.conf
 - Copy Kernel Image
 - Copy Kernel Device Tree Binaries
 - Copy Kernel Modules
 - File Systems Table (/etc/fstab)
 - Remove microSD/SD card
- Comments

Availability

Boards:

1406-0013-ND at Digi-Key

Vendor Documentation

- TechNexion: <https://www.technexion.com/>
 - PICO-PI-IMX7-DEV: <https://www.technexion.com/products/evaluation-kits/detail/PICO-PI-IMX7-DEV>

Basic Requirements

- Running a recent release of Debian, Fedora or Ubuntu; without OS Virtualization Software.
- ARM Cross Compiler – Linaro: <https://www.linaro.org>
 - Linaro Toolchain Binaries: <https://www.linaro.org/downloads/>
- Bootloader
 - Das U-Boot – the Universal Boot Loader: <http://www.denx.de/wiki/U-Boot>
 - Source: <https://github.com/u-boot/u-boot/>
- Linux Kernel
 - Linus's Mainline tree: <https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git>
- ARM based rootfs
 - Debian: <https://www.debian.org>
 - Ubuntu: <https://www.ubuntu.com>

imx_usb_loader

imx_usb_loader – USB & UART loader for i.MX5/6/7/8 series:

Source: https://github.com/boundarydevices/imx_usb_loader

Download/build:

```
~/
```

```
git clone https://github.com/boundarydevices/imx_usb_loader
cd imx_usb_loader/
make
```

ARM Cross Compiler: GCC

This is a pre-built (64bit) version of Linaro GCC that runs on generic linux, sorry (32bit) x86 users, it's time to upgrade...
Download/Extract:

```
~/
```

```
wget -c https://releases.linaro.org/components/toolchain/binaries/6.4-2018.05/arm-linux-gnueabi/gcc-linaro-6.4.1-2018.05-x86_64_arm-linux-gnueabi.tar.xz
tar xf gcc-linaro-6.4.1-2018.05-x86_64_arm-linux-gnueabi.tar.xz
export CC=`pwd`/gcc-linaro-6.4.1-2018.05-x86_64_arm-linux-gnueabi/bin/arm-linux-gnueabi-
```

Test Cross Compiler:

```
~/
```

```
`${CC}gcc --version
arm-linux-gnueabi-gcc (Linaro GCC 6.4-2018.05) 6.4.1 20180425 [linaro-6.4-2018.05 revision 7b15d0869c096fe39603ad63dc19ab7cf035eb70]
Copyright (C) 2017 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Bootloader: U-Boot

Das U-Boot – the Universal Boot Loader: <http://www.denx.de/wiki/U-Boot>
Download:

```
~/
```

```
git clone https://github.com/u-boot/u-boot
cd u-boot/
git checkout v2019.01 -b tmp
```

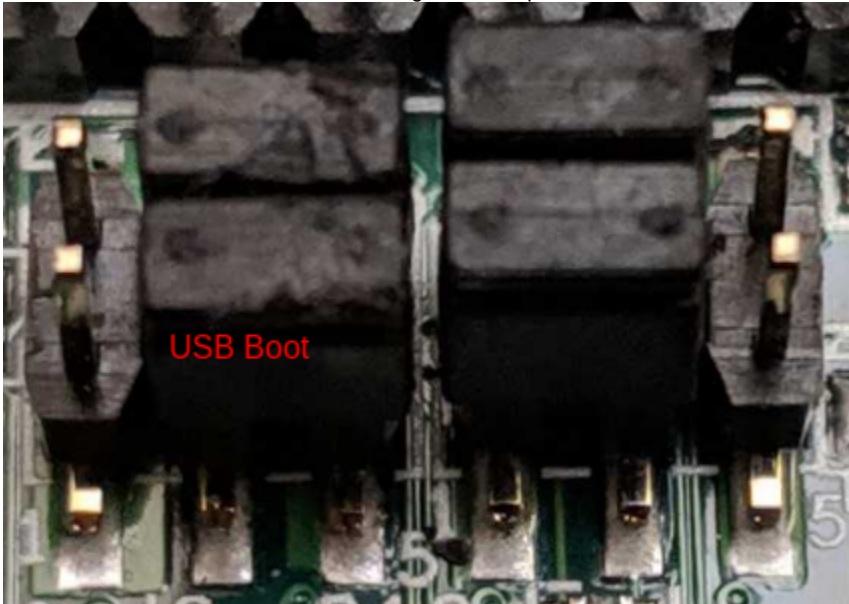
Configure and Build:

```
~/u-boot
```

```
make ARCH=arm CROSS_COMPILE=${CC} distclean  
make ARCH=arm CROSS_COMPILE=${CC} pico-pi-imx7d_defconfig  
make ARCH=arm CROSS_COMPILE=${CC}
```

Bootloader: imx_usb_loader

Place the board into USB boot mode, setting these Jumpers:



Power Board and Check lsusb:

```
~/
```

```
lsusb | grep Freescale  
Bus 00x Device 00y: ID 15a2:0076 Freescale Semiconductor, Inc
```

Load the SPL binary via USB:

```
~/u-boot
```

```
sudo ~/imx_usb_loader/imx_usb SPL
```

Load the u-boot.img binary via USB:

```
~/u-boot
```

```
sudo ~/imx_usb_loader/imx_usb u-boot.img
```

Using any key, break into U-Boot Console:

```
SERIAL_CONSOLE_U_BOOT
```

```
Loading Environment from MMC... OK
In:    serial
Out:   serial
Err:   serial
Net:   FEC0
Hit any key to stop autoboot:  0
=>
```

Use the default environment variables:

```
SERIAL_CONSOLE_U_BOOT
```

```
=> env default -f -a
=> saveenv
```

Configure mmc 0 to boot from the data partition, and disable access to boot partitions:

```
SERIAL_CONSOLE_U_BOOT
```

```
=> mmc partconf 0 0 7 0
```

Run the DFU agent so we can flash the new images using dfu-util tool:

```
SERIAL_CONSOLE_U_BOOT
```

```
=> dfu 0 mmc 0
```

Load the SPL binary via dfu-util:

```
~/u-boot
```

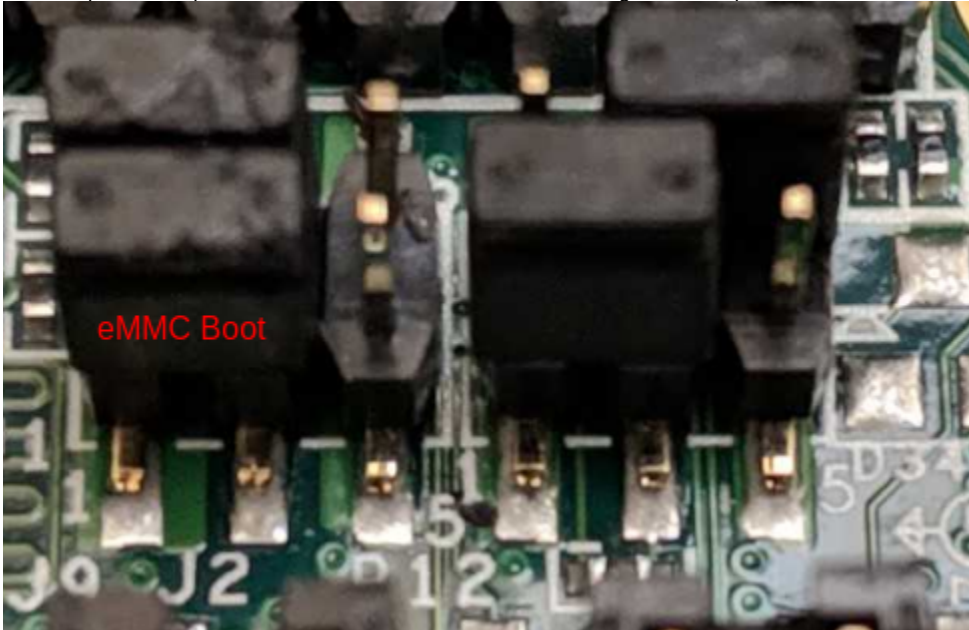
```
sudo dfu-util -D SPL -a spl
```

Load the SPL binary via dfu-util:

```
~/u-boot
```

```
sudo dfu-util -R -D u-boot.img -a u-boot
```

Remove power and place the board into eMMC boot mode, setting these Jumpers:



Power Board and enable U-Boot USB Mass Storage mode:

```
SERIAL_CONSOLE_U_BOOT
```

```
Loading Environment from MMC... OK
In:    serial
Out:   serial
Err:   serial
Net:   FEC0
Hit any key to stop autoboot:  0
=> ums 0 mmc 0
```

Verify Linux loaded the USB Mass Storage device, this will be the SD Card used thru out this wiki:

```
~/
```

```
lsusb | grep Netchip
Bus 00x Device 00y: ID 0525:a4a5 Netchip Technology, Inc. Pocketbook Pro
903
```

Linux Kernel

This script will build the kernel, modules, device tree binaries and copy them to the deploy directory.

Download:

```
~/  
  
git clone https://github.com/RobertCNelson/armv7-lpae-multiplatform  
cd armv7-lpae-multiplatform/
```

For v4.20.x (Stable):

```
~/armv7-lpae-multiplatform/  
  
git checkout origin/v4.20.x -b tmp
```

Build:

```
~/armv7-lpae-multiplatform/  
  
./build_kernel.sh
```

Root File System

Debian 9

User	Password
debian	tempwd
root	root

Download:

```
~/  
  
wget -c https://rcn-ee.com/rootfs/eewiki/minfs/debian-9.6-minimal-armhf-  
2018-11-26.tar.xz
```

Verify:

```
~/  
  
sha256sum debian-9.6-minimal-armhf-2018-11-26.tar.xz  
c9d9b2623131829eeefa153274219d89887a1984b8dde72cfbfe29f29c7a10d  debian-  
9.6-minimal-armhf-2018-11-26.tar.xz
```

Extract:

```
~/  
tar xf debian-9.6-minimal-armhf-2018-11-26.tar.xz
```

Ubuntu 18.04 LTS

User	Password
ubuntu	tempwd

Download:

```
~/  
wget -c https://rcn-ee.com/rootfs/eewiki/minifs/ubuntu-18.04.1-minimal-armhf-2018-11-26.tar.xz
```

Verify:

```
~/  
sha256sum ubuntu-18.04.1-minimal-armhf-2018-11-26.tar.xz  
d8190bc858bab9db83bec03707b2636dc23646fca2a0420fb49adbedf7d492c5  ubuntu-18.04.1-minimal-armhf-2018-11-26.tar.xz
```

Extract:

```
~/  
tar xf ubuntu-18.04.1-minimal-armhf-2018-11-26.tar.xz
```

Setup microSD card

We need to access the External Drive to be utilized by the target device. Run lsblk to help figure out what linux device has been reserved for your External Drive.

Example: for DISK=/dev/sdX

```
lsblk
NAME MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda   8:0    0 465.8G  0 disk
sda1  8:1    0   512M  0 part /boot/efi
sda2  8:2    0 465.3G  0 part /
Root Partition
sdb   8:16   1   962M  0 disk
Device
sdb1  8:17   1   961M  0 part
Partition
```

<- Development Machine
<- microSD/USB Storage
<- microSD/USB Storage

Thus you would use:

```
export DISK=/dev/sdb
```

Example: for DISK=/dev/mmcblkX

```
lsblk
NAME MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda   8:0    0 465.8G  0 disk
sda1  8:1    0   512M  0 part /boot/efi
sda2  8:2    0 465.3G  0 part /
Machine Root Partition
mmcblk0 179:0   0   962M  0 disk
Storage Device
mmcblk0p1 179:1  0   961M  0 part
Storage Partition
```

<- Development
<- microSD/USB
<- microSD/USB

Thus you would use:

```
export DISK=/dev/mmcblk0
```

Erase partition table/labels on microSD card:

```
sudo dd if=/dev/zero of=${DISK} bs=1M count=10
```

Install Bootloader:


```
~/
```

```
sudo dd if=./u-boot/SPL of=${DISK} seek=1 bs=1k  
sudo dd if=./u-boot/u-boot.img of=${DISK} seek=69 bs=1k
```

Create Partition Layout:

With util-linux v2.26, sfdisk was rewritten and is now based on libfdisk.

```
sudo sfdisk --version  
sfdisk from util-linux 2.27.1
```

```
sfdisk >= 2.26.x
```

```
sudo sfdisk ${DISK} <<-__EOF__  
1M,,L,*  
__EOF__
```

```
sfdisk <= 2.25.x
```

```
sudo sfdisk --unit M ${DISK} <<-__EOF__  
1,,L,*  
__EOF__
```

Format Partition:

```
for: DISK=/dev/mmcblkX  
sudo mkfs.ext4 -L rootfs ${DISK}p1  
  
for: DISK=/dev/sdX  
sudo mkfs.ext4 -L rootfs ${DISK}1
```

Mount Partition:

On most systems these partitions may will be auto-mounted...

```
sudo mkdir -p /media/rootfs/

for: DISK=/dev/mmcblkX
sudo mount ${DISK}p1 /media/rootfs/

for: DISK=/dev/sdX
sudo mount ${DISK}1 /media/rootfs/
```

Install Kernel and Root File System

To help new users, since the kernel version can change on a daily basis. The kernel building scripts listed on this page will now give you a hint of what kernel version was built.

```
-----
Script Complete
eewiki.net: [user@localhost:~$ export kernel_version=4.X.Y-Z]
-----
```

Copy and paste that "export kernel_version=4.X.Y-Z" exactly as shown in your own build/desktop environment and hit enter to create an environment variable to be used later.

```
export kernel_version=4.X.Y-Z
```

Copy Root File System

```
~/

sudo tar xfvp ./*-*-*-armhf-*/armhf-rootfs-*.tar -C /media/rootfs/
sync
sudo chown root:root /media/rootfs/
sudo chmod 755 /media/rootfs/
```

Setup extlinux.conf

```
~/
```

```
sudo mkdir -p /media/rootfs/boot/extlinux/  
sudo sh -c "echo 'label Linux ${kernel_version}' > /media/rootfs/boot  
/extlinux/extlinux.conf"  
sudo sh -c "echo '    kernel /boot/vmlinuz-${kernel_version}' >> /media  
/rootfs/boot/extlinux/extlinux.conf"  
sudo sh -c "echo '    append root=/dev/mmcblk2p1 ro rootfstype=ext4  
rootwait quiet' >> /media/rootfs/boot/extlinux/extlinux.conf"  
sudo sh -c "echo '    fdtfile /boot/dtbs/${kernel_version}/' >> /media  
/rootfs/boot/extlinux/extlinux.conf"
```

Copy Kernel Image

Kernel Image:

```
~/
```

```
sudo cp -v ./armv7-lpae-multiplatform/deploy/${kernel_version}.zImage  
/media/rootfs/boot/vmlinuz-${kernel_version}
```

Copy Kernel Device Tree Binaries

```
~/
```

```
sudo mkdir -p /media/rootfs/boot/dtbs/${kernel_version}/  
sudo tar xfv ./armv7-lpae-multiplatform/deploy/${kernel_version}-dtbs.tar.  
gz -C /media/rootfs/boot/dtbs/${kernel_version}/
```

Copy Kernel Modules

```
~/
```

```
sudo tar xfv ./armv7-lpae-multiplatform/deploy/${kernel_version}-modules.  
tar.gz -C /media/rootfs/
```

File Systems Table (/etc/fstab)

```
sudo sh -c "echo '/dev/mmcblk2p1 / auto errors=remount-ro 0 1' >>
/media/rootfs/etc/fstab"
```

Remove microSD/SD card

```
sync
sudo umount /media/rootfs
```

Stop U-Boot USB Mass Storage mode, with CTRL-C, and reset the board:

SERIAL_CONSOLE_U_BOOT

```
Loading Environment from MMC... OK
In:    serial
Out:   serial
Err:   serial
Net:   FEC0
Hit any key to stop autoboot:  0
=> ums 0 mmc 0
UMS: LUN 0, dev0, hwpart 0, section 0x0, count 0x720000
Ctrl+C to exit ...
=> reset
```

Comments

Comments, feedback, and questions can be sent to: ewiki@digkey.com
Please use the Digi-Key's TechForum: [TechForum](#)