Android 7.1 BSP Source Release Notes

v1.00

TABLE OF CONTENTS

Supported Products

This release supports the following products:

- PICO-IMX6 / PICO-IMX7 + DWARF/NYMPH/HOBBIT/PI
- EDM1-IMX6 / EDM1-CF-IMX6 + EDM1-FAIRY
- TC-0700 / TC-0710
- TC-1010
- TEK3-IMX6
- TEP1010-IMX6
- TEP1560-IMX6
- TEP0500-IMX7
- TEP0700-IMX7

Obtaining the Android 7.1 BSP source code

You can download the Android 7.1 source code for TechNexion products.

The easiest way is to download it directly from TechNexion FTP at

ftp://ftp.technexion.net/development_resources/development_tools/NXP/android/7.1/

Preparing a build system

The recommended system requirements are as belows:

- 16 GB RAM
- 300 GB hard disk space available

Using a virtual machine

For the impatient, TechNexion hosts a virtual machine image where the build environment is already setup. This virtual machine image can be downloaded from TechNexion FTP at ftp://ftp.technexion.net/development_resources/development_tools/vm/

Setting up a linux host

An android build can be very "picky" about software versions on the build machine.

Google maintains setup instructions for official Android releases at http://source.android.com/source/initializing.html

TechNexion (and NXP/Freescale) Android source code requires some additional packages on top of the Google requirements.

On a Ubuntu 16.04LTS based system, these additional packages can be installed by:

```
$ cd /opt

$ sudo tar -Jxvf gcc-5.1-2015.08-x86_64_arm-linux-gnueabihf.tar.xz

$ sudo apt-get update

$ sudo apt-get install bc make build-essential python

$ sudo apt-get install zip unzip libxml2 flex bison m4 ccache

$ sudo apt-get install libc6:i386 libncurses5:i386 libstdc++6:i386

$ sudo apt-get install uuid uuid-dev

$ sudo apt-get install zlib1g-dev liblz-dev

$ sudo apt-get install liblzo2-2 liblzo2-dev

$ sudo apt-get install lzop

$ sudo apt-get install git-core curl

$ sudo apt-get install u-boot-tools

$ sudo apt-get install mtd-utils
$ sudo apt-get install android-tools-fsutils

$ sudo apt-get install openjdk-8-jdk

$ sudo apt-get install g++-multilib

$ sudo apt-get install gcc-multilib

$ sudo apt-get install lib32ncurses5-dev

$ sudo apt-get install lib32readline6-dev

$ sudo apt-get install lib32z1-dev

$ sudo apt-get install libxml2-utils
```

Building Android

Building Android is done in two steps. The first step configures the target product, and the second step does the build itself.

Building commands overview

The general commands for Android builds are

```
source {file}
```

to set the environment for a specific target (specified by *{file}*),

```
cook
```

for a full clean build ,

```
heat
```

for an incremental build, and

```
throw
```

for cleaning the build files. The '`cook`' and '`heat`' commands can accept a parameter '`-j N`' to build on *N* processors. It is recommended to use a value of *N* matching the number of CPUs in the computer. This can greatly decrease the build time.

An example build

To configure the build target, one 'sources' the necessary environment variables from the cookers/ folder. For example, to build Android for EDM1-IMX6P module on ad EDM1-Fairy with HDMI display, one can issue the commands:

```
source cookers/env.bash.imx6.edm1cf-nand.fairy.hdmi

cook -j4
```

See the cookers/ folder for additional system configurations.

Changing build modes

The default Android compile setting is to build in "engineering mode". This is for creating a development image with debug tools and looser permissions.

To change to "user mode" (sometimes the preferred production build) can be done by modifying some of the build scripts.

Edit `cookers/env.bash`, and within the `heat()` and `cook()` functions change the lunch command lines (one per function) from

```
lunch "$TARGET_DEVICE"-eng
```

to

```
lunch "$TARGET_DEVICE"-user
```

Deploying Android into the board

This chapter describes how to deploy a completed build into your board.

There are two ways to easily install an Android system into your board. The first method uses an OTG or USB Type C cable (depending on your product), and writes the Android to the onboard eMMC from the build computer.

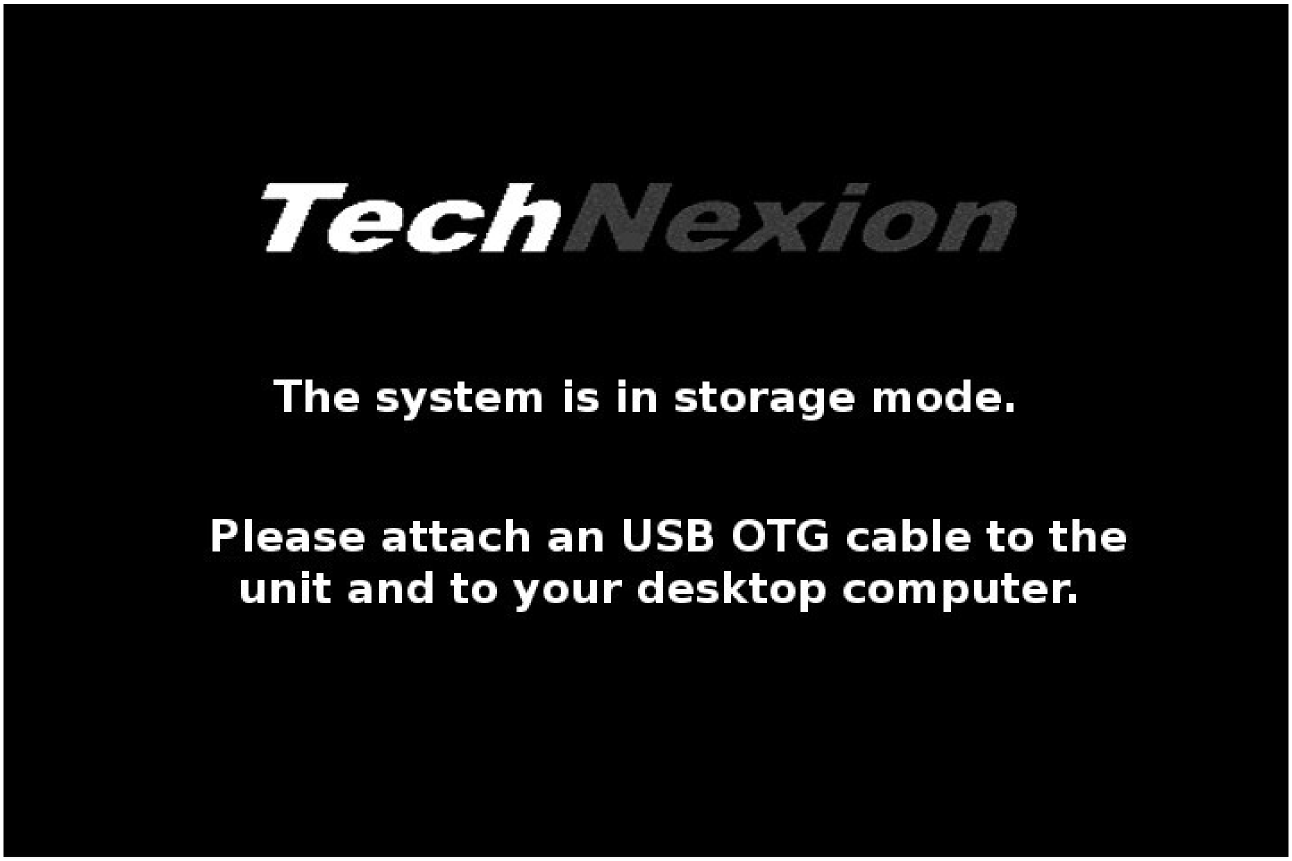
The second method creates a bootable SD card that installs Android when booted. This second method is only outlined here.

Direct deployment into eMMC

Both methods to deploy Android uses a bootable SD card to boot the board. This SD card can then either be used to

1. Download "generic installer" for your product from TechNexion ftp: ftp://ftp.technexion.net/development_resources/development_tools/installer/
2. Make an SD card of the installer image (see Appendix A for instructions).
3. Setup your boot mode to SD card boot (see Appendix B, or your board documentation).
4. Connect your board to your PC using an OTG (or if not applicable, use a USB Type-C) cable.
5. Insert the SD card with the generic installer in your board.
6. Boot up your board.

7. Wait until the system indicates it is in storage mode. A USB mass storage device might be indicated on the build computer.



TechNexion

The system is in storage mode.

Please attach an USB OTG cable to the unit and to your desktop computer.

8. Run the commands

```
    sudo umount /dev/sdX*

    flashcard /dev/sdX
```

where sdX is the block device for your SD card. This can be different on different computers.

NOTE: be certain you use the correct block device, using the wrong device can result in data loss on other storage devices attached to your computer, including harddrives.

Examples of Android installers (to try or reference) can be downloaded from ftp://ftp.technexion.net/demo_software/

Making an installer SD card

This documents how to make a self-installing SD card for TechNexion products.

First, make a generic installer SD card as described in steps 1 and 2 in the preceding section.

The generic installer SD card has two partitions. The first one, is a FAT partition containing boot files, and more importantly the image to be installed. The second partition is the installer itself, and users should not have to touch the second partition.

The quickest way to make an installer is:

1. Prepare an image file to be installed. This can be done by cloning the content of a unit that works acceptably:

   a) Follow steps 1 through 7 of the procedure in the previous section

   b) Instead of using the flashcard script, read the content of the eMMC with

   ```
   dd if=/dev/sdX of=android bs=1M
   ```

   c) Compress the resulting file with xz

   ```
   xz -9 android
   ```

2. Place the installation file in images/android.xz on the FAT partition. This way, the installer will install it into eMMC when booted.

HINT: It can be benficial not to 'dd' out the entire eMMC content. Only the space until the end of the last partition is required.

HINT: Clear the unused areas of the eMMC. An eMMC (or SD card) that has been in active use can contain 'garbage' (like remnants of old, deleted files) in the 'free' space. One way to clean this up is by filling up all the free diskspace in each partition with a large file with zeroes, and the removing that file. Example:

```
dd if=/dev/zero of={file} bs=64k

rm -f {file}
```

The benefits of cleaning up the empty area are for instance: faster installs, less diskspace needed for installer, and avoiding information 'leakage'.

Troubleshooting

Thjs section describes answers to common questions.

General

Read the board documentation. Also TechNexion kernel releases have software a troubleshooting guide.

WiFi Problems

The firmware for some WiFi chips (AP6212 and AP6335) are not distributed with this source code package. Contact `support@technexion.com` to receive these firmwares and instructions on how to deploy them.

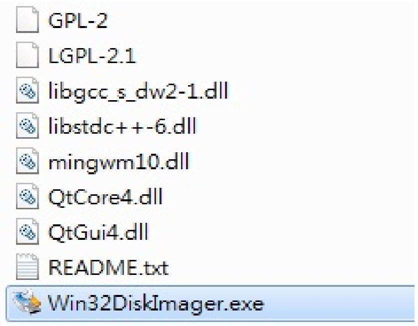
Appendix A: Making a bootable SD card from an image file

This chapter describes how to make a bootable card from an SD card image file.

NOTE: All of these will overwrite the SD card content.

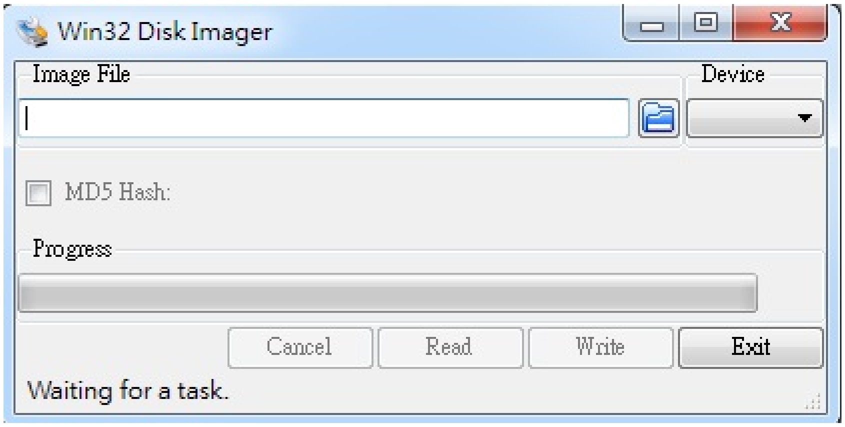
Using a Windows PC

1. Download and install the free WinDiskImager32 software. This software is also included in many TechNexion demo images.
2. Insert an empty SD card into your computer's card reader
3. Execute `windiskimager32.exe`



GPL-2
LGPL-2.1
libgcc_s_dw2-1.dll
libstdc++-6.dll
mingwm10.dll
QtCore4.dll
QtGui4.dll
README.txt
Win32DiskImager.exe

4. You are presented with a window similar to



Waiting for a task.

5. Select your SD card device under "Device", and an image using the folder icon

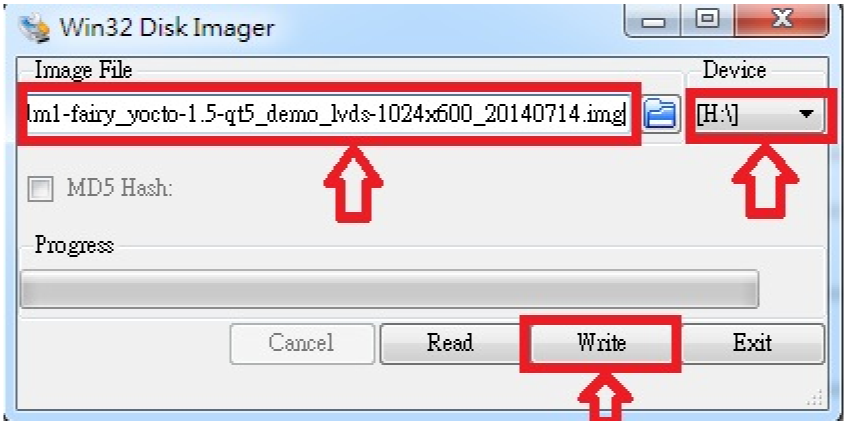


Image File: lm1-fairy_yocto-1.5-qt5_demo_lvds-1024x600_20140714.img

Device: [H:\]

☐ MD5 Hash:

6. Press the write button (as outlined above).

Using a Linux PC

This outlines how to make an SD card using a Ubuntu Linux. There are two ways described here.

Command line

The steps are:

1. Insert an SD card into your card reader
2. Use the `dd` command to write your SD card:

```
sudo dd if={file.img} of=/dev/sdX bs=1M oflag=dsync
```

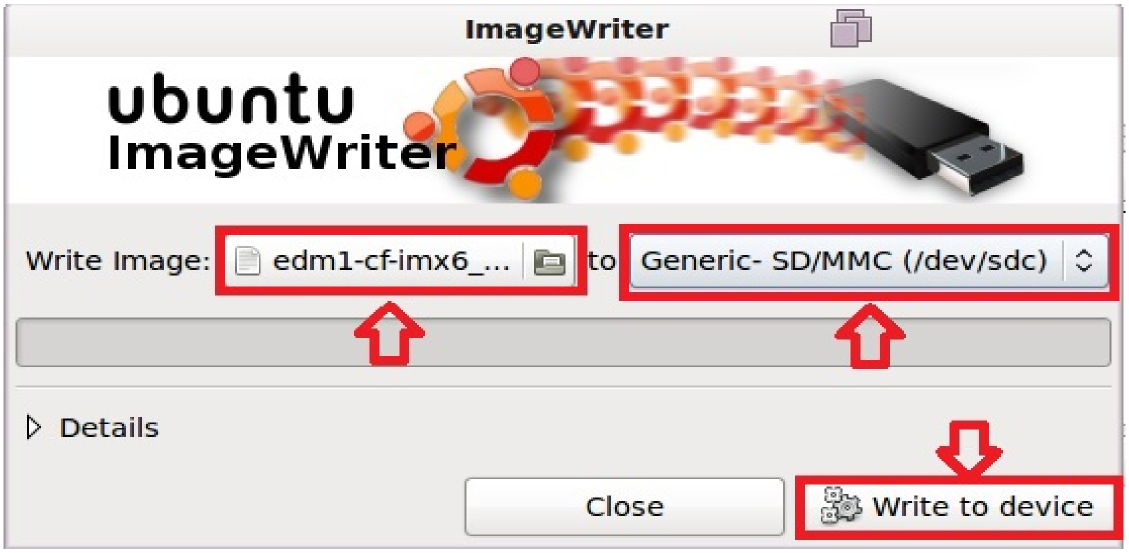
USB Imagewriter

Alternatively, it is also possible to use a tool named USB imagewriter: https://apps.ubuntu.com/cat/applications/precise/usb-imagewriter/

If using Ubuntu distribution, USB imagewriter can be installed and exectued with

```
sudo apt-get install usb-imagewriter

sudo imagewriter
```



Choose your image file, your SD card device and then click "Write to device".

Appendix B: Changing board boot mode

This chapter describes how to change the boot device on your board. Typically this is used to switch between booting from SD card and booting from onboard storage (like an eMMC flash chip).

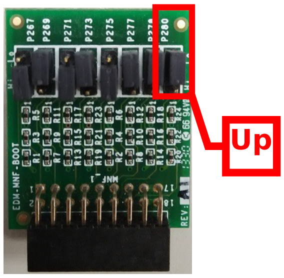
This appendix is provided for completeness, please refer to your board documentation for further info on boot mode selections.

EDM-Series: EDM1-FAIRY / EDM1-GOBLIN / EDM2-ELF

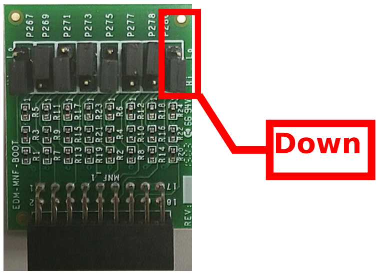
Many EDM1 baseboards uses a boot PCB to select boot mode. For simplicity, the instructions below will refer to the EDM1-Fairy baseboard, but the instructions apply to EDM1-Goblin and EDM2-Elf boards as well.

Set the boot jumpers on your **EDM-MNF-BOOT PCB** (delivered with your baseboard) as below.

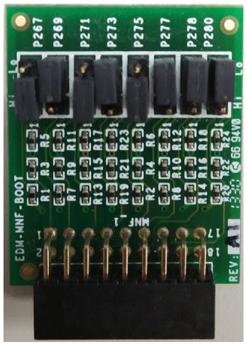
For EDM1-CF-IMX6 / EDM1-IMX6P

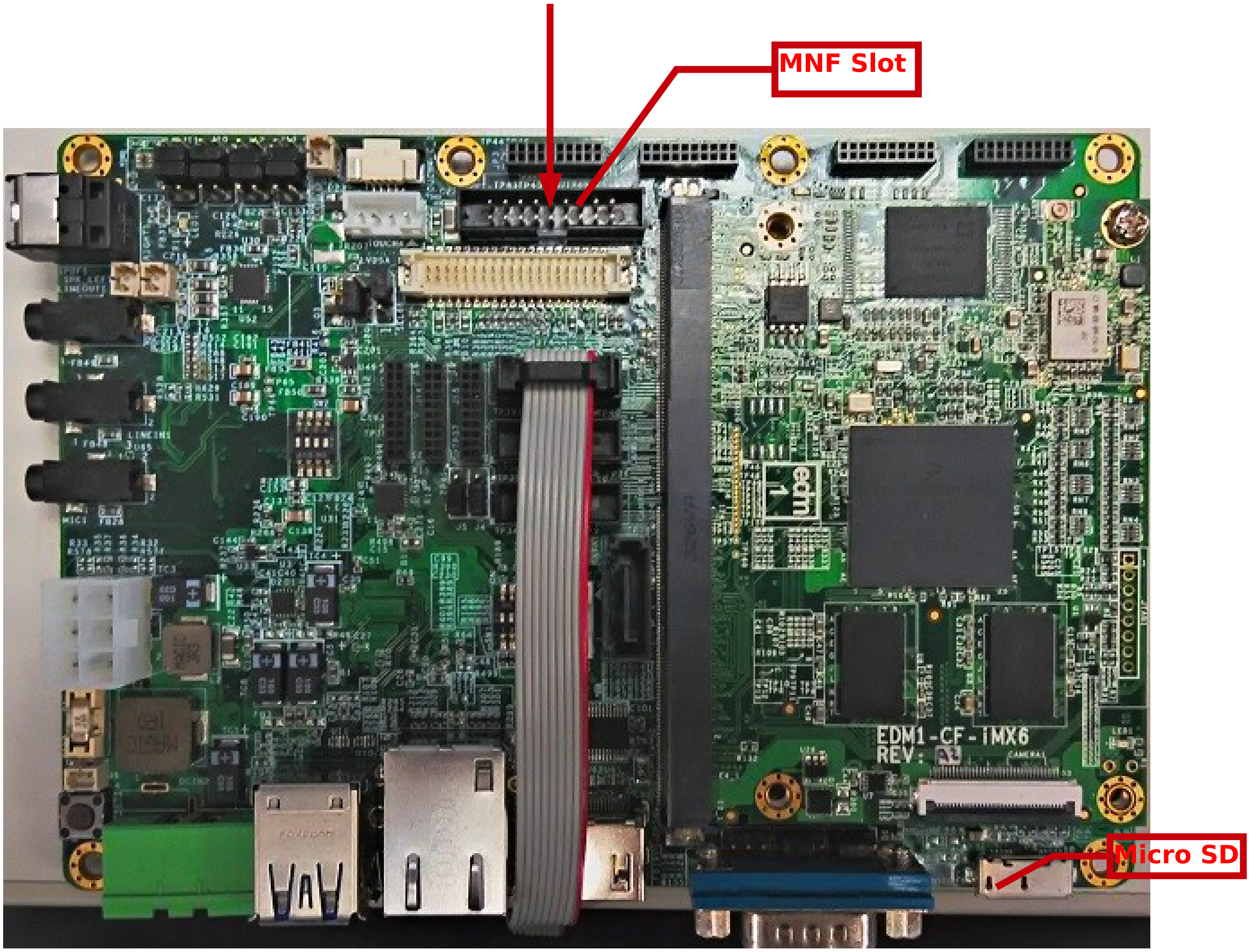


For EDM1-CF-IMX6SX



Next, attach the MNF-BOOT PCB into the boot PCB slot on the EDM1-Fairy baseboard (see picture below). This will cause the module to boot from the external microSD card slot instead of the onboard eMMC.





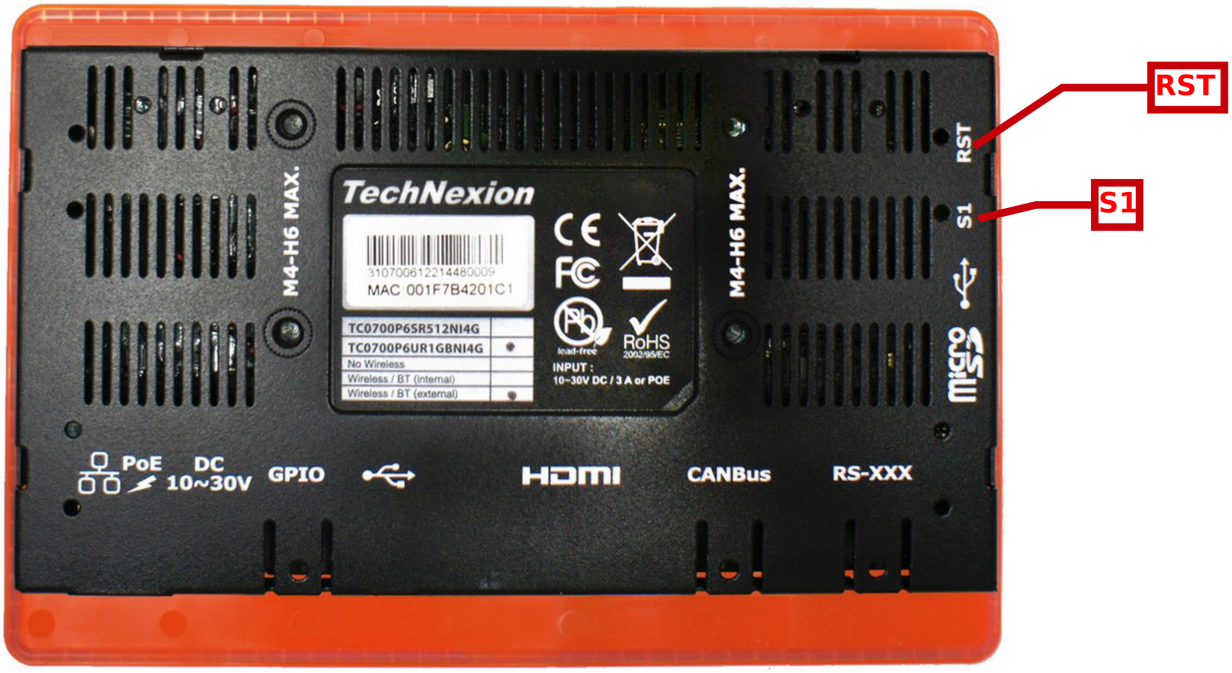
Then, insert MicroSD card with Android installer image inside into EDM1-Fairy baseboard.

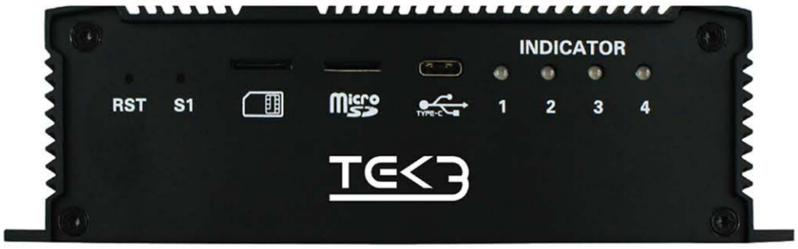
Systems (TC/Toucan, TEK, TEP)

The boot mode of TechNexion systems are usually controlled by the "S1" button on the side of the unit.

Hold down the S1 button and then press the RST button (next to the S1). Keep the S1 button down for a short time after releasing the RST button. This way the unit boots from the SD card slot instead of the onboard storage.

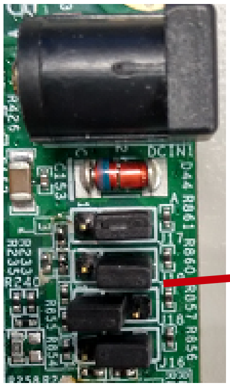
TC-07x0 ("Toucan"):





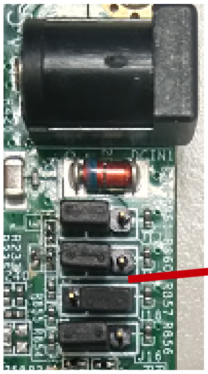
PICO-Series: PICO-IMX6 / PICO-IMX7 Module and PICO-Dwarf Baseboard

The PICO series boot is controlled using jumpers on the baseboard. Many PICO-Series modules come in two versions: with on-board eMMC (or NAND) or with an SD card slot instead of on-board storage. Set your boot jumpers accordingly.



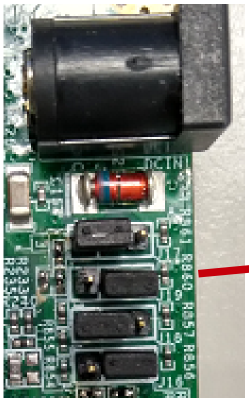
Boot from SD of baseboard

Install jumpers as below, and board will boot from eMMC card of CPU module:



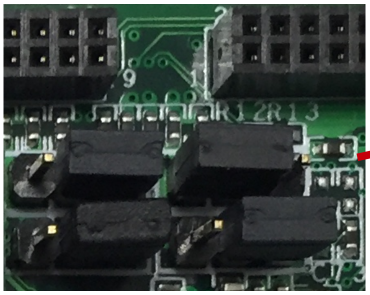
Boot from eMMC of CPU module

Install jumpers as below, and board will boot from SD card of CPU module:



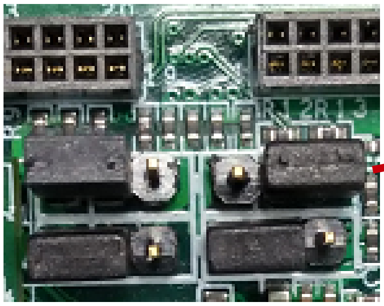
Boot from SD card of CPU module

PICO-Series: PICO-IMX6 / PICO-IMX7 Module and PICO-Hobbit Baseboard
Install jumpers as below, and board will boot from SD card of baseboard:



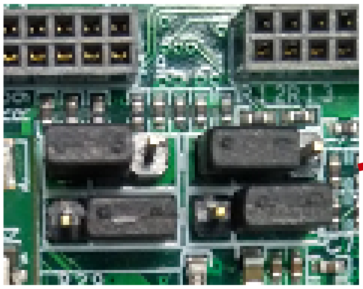
Boot from SD of baseboard

Install jumpers as below, and board will boot from eMMC card of CPU module:



Boot from eMMC of baseboard

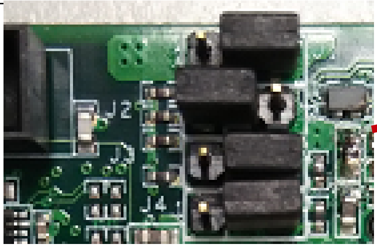
Install jumpers as below, and board will boot from SD card of CPU module:



Boot from SD of CPU module

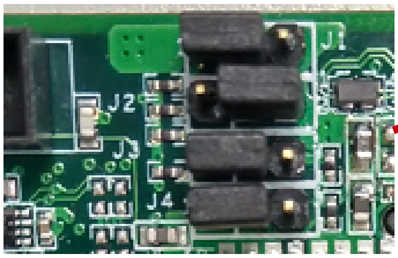
PICO-Series: PICO-IMX6 / PICO-IMX7 Module and PICO-Nymph Baseboard

Install jumpers as below, and board will boot from SD card of baseboard:



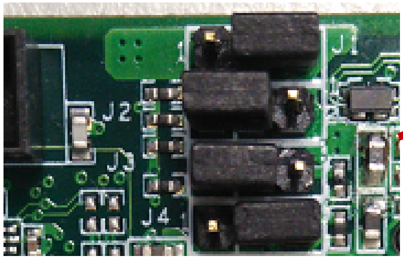
Boot from SD of baseboard

Install jumpers as below, and board will boot from eMMC card of CPU module:



Boot from eMMC of CPU module

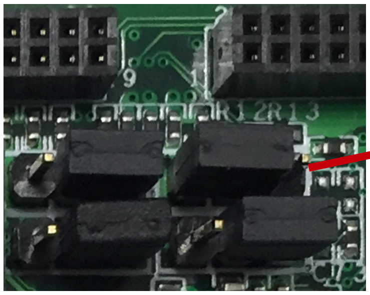
Install jumpers as below, and board will boot from SD card of CPU module:



Boot from SD of CPU module

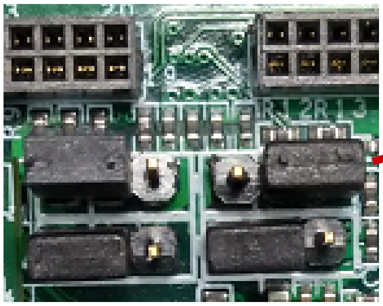
PICO-Series: PICO-IMX6 / PICO-IMX7 Module and PICO-Pi Baseboard

Install jumpers as below, and board will boot from SD card of baseboard:



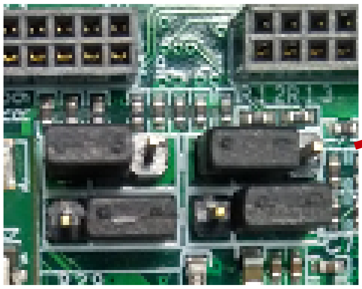
Boot from SD of baseboard

Install jumpers as below, and board will boot from eMMC card of CPU module:



Boot from eMMC of CPU module

Install jumpers as below, and board will boot from SD card of CPU module:



Boot from SD of CPU module