

VAR-SOM-MX6 - Debian Stretch 9.5 with imx_4.9.88_2.0.0_ga-var01 Linux release

Contents [\[hide\]](#)

1. Overview
2. Create build environment
 - 2.1. Installing required packages
 - 2.2. Deploy source
3. Make Debian
 - 3.1. Build all
 - 3.2. Build by parts
 - 3.2.1. Build bootloader
 - 3.2.2. Build kernel, dtb files and kernel modules
 - 3.2.3. Build rootfs
 - 3.2.4. Pack rootfs
4. Create boot SD card
 - 4.1. Create a boot SD card image using a loop device
5. Boot the board with a bootable SD card
 - 5.1. Setting the Boot Mode
 - 5.1.1. MX6CustomBoard
 - 5.1.2. SoloCustomBoard
 - 5.1.3. DT6CustomBoard
 - 5.2. Automatic device tree selection in U-Boot
 - 5.2.1. Enable/Disable Automatic Device Tree selection
6. Build Results
7. Linux console access
8. Flash images to eMMC
9. How-to: Test and use an interface
10. How-to: Modify the kernel configuration
11. Build a sample C "Hello, world!" program

1. Overview

This page describes how to build and install Debian distribution (Stretch) on Variscite boards with VAR-SOM-MX6.

These instructions were tested on **Ubuntu 16.04/18.04 x64 host PCs**. When using other distributions, there may be issues.

Please note that the build script is based on *debootstrap*. As described in the following instructions, it's kindly suggested to **create the build folder on the main Ubuntu volume on your host PC** rather than an external media.

Using an external media, although automatically mounted, will cause *debootstrap* to complain about access rights (even when running the script with `sudo`).

2. Create build environment

2.1. Installing required packages

On Ubuntu building machine:

```
$ sudo apt-get install binfmt-support qemu qemu-user-static debootstrap kpartx \
lvm2 dosfstools gpart binutils bison git lib32ncurses5-dev libssl-dev python-m2crypto gawk wget \
git-core diffstat unzip texinfo gcc-multilib build-essential chrpath socat libstd1.2-dev \
autoconf libtool libglib2.0-dev libarchive-dev python-git xterm sed cvs subversion \
kmod coreutils texi2html bc docbook-utils python-pysqlite2 help2man make gcc g++ \
desktop-file-utils libgl1-mesa-dev libglu1-mesa-dev mercurial automake groff curl \
lzop asciidoc u-boot-tools mtd-utils device-tree-compiler flex cmake
```

2.2. Deploy source

Download archive containing the build script and support files for building Debian Stretch for this board:

```
$ cd ~
$ git clone https://github.com/varigit/debian-var.git -b debian_stretch_mx6_var02 debian_var-som-mx6
```

Create environment (*Internet connection should be available*):

```
$ cd ~/debian_var-som-mx6
$ ./make_var_som_mx6_debian.sh -c deploy
```

This environment prepared to build.

3. Make Debian

3.1. Build all

Internet connection should be available

```
$ cd ~/debian_var-som-mx6
$ sudo ./make_var_som_mx6_debian.sh -c all |& tee build.log
```

3.2. Build by parts

3.2.1. Build bootloader

```
$ cd ~/debian_var-som-mx6
$ sudo ./make_var_som_mx6_debian.sh -c bootloader
```

3.2.2. Build kernel, dtb files and kernel modules

```
$ cd ~/debian_var-som-mx6
$ sudo ./make_var_som_mx6_debian.sh -c kernel
$ sudo ./make_var_som_mx6_debian.sh -c modules
```

3.2.3. Build rootfs

Internet connection should be available

```
$ cd ~/debian_var-som-mx6
$ sudo ./make_var_som_mx6_debian.sh -c rootfs
```

3.2.4. Pack rootfs

To create the root file system archive (rootfs.tar.gz), run the following commands:

```
$ cd ~/debian_var-som-mx6
$ sudo ./make_var_som_mx6_debian.sh -c rtar
```

4. Create boot SD card

1. Follow the above steps for make rootfs, kernel, bootloader;
2. Insert the SD card to card reader connected to a host system;
3. Run the following commands (Caution! All data on the card will be destroyed):

```
$ cd ~/debian_var-som-mx6
$ sudo ./make_var_som_mx6_debian.sh -c sdcard -d /dev/sdX
```

where '/dev/sdX' path to the block SD device in your system.

4.1. Create a boot SD card image using a loop device

It is also possible to use the "./make_var_som_mx6_debian.sh" script to create a boot SD card image, while using a loop device instead of attaching a real SD card.

Create an empty file using the following command:

```
$ dd if=/dev/zero of=var-som-mx6-debian-sd.img bs=1M count=3720
```

The above command creates a 3700MiB file representing the SD card.

Attach the first available loop device to this file:

```
$ sudo losetup -Pf var-som-mx6-debian-sd.img
```

To find the actual loop device being used, run: \$ losetup -a | grep var-som-mx6-debian-sd.img

To find the default loop device being used, run `losetup -a` (grep for som-mx6-debian.img)

Write the content to the loop device to generate the SD card image:

```
$ sudo ./make_var_som_mx6_debian.sh <options> /dev/loopX
```

(Replace `/dev/loopX` with your actual loop device, e.g. `/dev/loop0`)

Detach the loop device from the file:

```
$ sudo losetup -d /dev/loopX
```

To compress the SD card image file use the following command:

```
$ gzip -9 var-som-mx6-debian-sd.img
```

To write the SD card image to a real SD card device use the following command:

```
$ zcat var-som-mx6-debian-sd.img.gz | sudo dd of=/dev/sdX bs=1M && sync
```

(Replace `/dev/sdX` with your actual SD device, e.g. `/dev/sdb`)

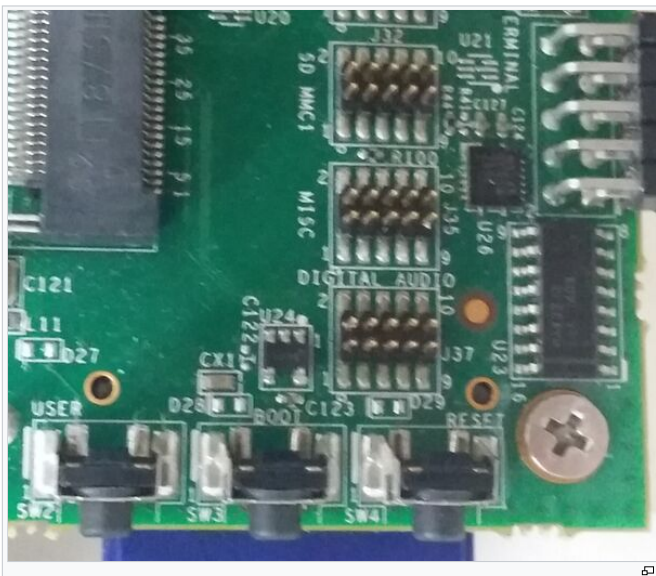
5. Boot the board with a bootable SD card

5.1. Setting the Boot Mode

Follow the instruction below according to the appropriate carrier board type:

5.1.1. MX6CustomBoard

Booting your MX6CustomBoard system from SD card requires pushing the middle button while powering up the system. See picture below.

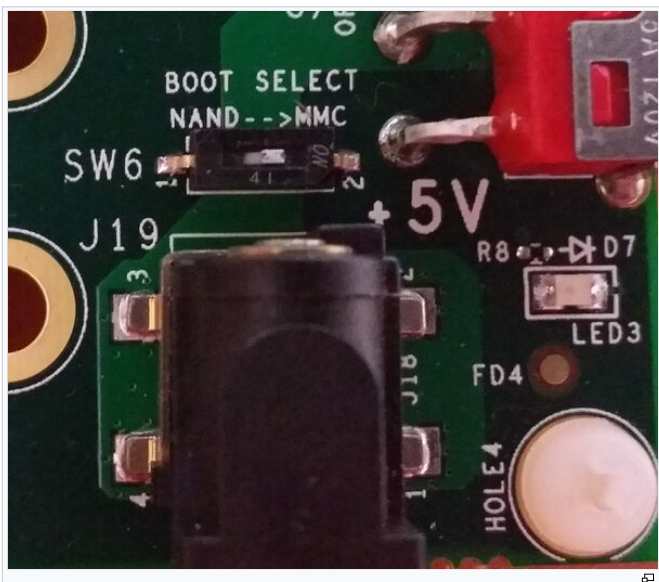


To boot a board using an SD card, follow the steps below:

- Power-off the board.
- Insert the SD card into the SD/MMC slot of the carrier board (DVK)
- Push the middle button (Boot Select) and hold
- Power-up the board
- Release the middle button (Boot Select) after system starts to boot.
- The board will automatically boot into Linux from the SD card

5.1.2. SoloCustomBoard

Booting your system requires switching the relevant DIP switch to "Boot from MMC". See picture below.

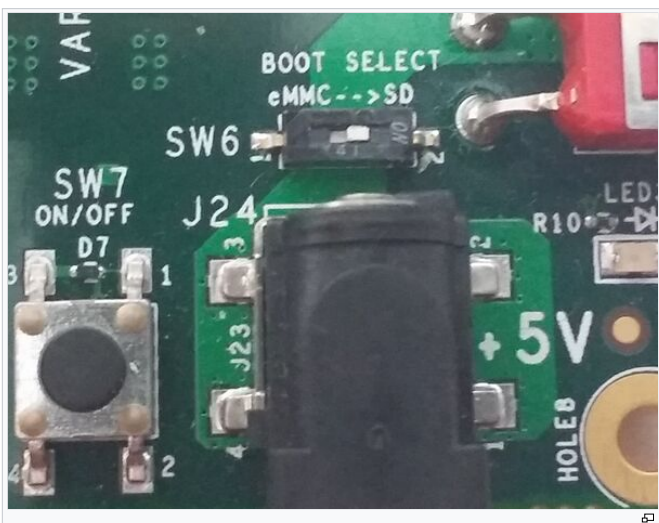


To boot board with SD card, Follow the steps below:

- Power-off the board.
- Insert the SD card into the SD/MMC slot of the carrier board (DVK)
- Switch the relevant DIP switch to "Boot from MMC"
- Power-up board
- The board will automatically boot into Linux from SD card

5.1.3. DT6CustomBoard

Booting your system requires switching the relevant DIP switch to "Boot from SD card". See picture below.



To boot board with SD card, Follow the steps below:

- Power-off the board.
- Insert the SD card into the SD/MMC slot of the carrier board (DVK)
- Switch the relevant DIP switch to "Boot from SD card"
- Power-up board
- The board will automatically boot into Linux from SD card

5.2. Automatic device tree selection in U-Boot

As shown in the [Build Results](#) table above, we have different kernel device trees, corresponding to our different H/W configurations (sometimes they are renamed without the "ulmage-" prefix).

We implemented a script in U-Boot's environment, which sets the fdt_file environment variable based on the detected hardware.

5.2.1. Enable/Disable Automatic Device Tree selection

To enable the automatic device tree selection in U-Boot (already enabled by default):

```
$ setenv fdt_file undefined
$ saveenv
```

To disable the automatic device tree selection in U-Boot, set the device tree file manually:

```
$ setenv fdt_file YOUR_DTB_FILE
$ saveenv
```

6. Build Results

The resulted images are located in `~/debian_var-som-mx6/output/`.

Image name	How to use
rootfs.tar.gz	Root filesystem tarball used for installation on SD card and eMMC
uImage	Linux kernel image
SPL.nand	SPL built for NAND. The SPL is pre-U-Boot SW component, required for DDR initialization
SPL.emmc	SPL built for SD card and eMMC boot. The SPL is pre-U-Boot SW component, required for DDR initialization
u-boot.img.nand	U-Boot built for NAND flash
u-boot.img.mmc	U-Boot built for SD card or eMMC boot
Device Tree name	Details
imx6dl-var-som-solo-cap.dtb	VAR-SOM-SOLO with iMX6S/DL on MX6CustomBoard with capacitive touch
imx6dl-var-som-solo-res.dtb	VAR-SOM-SOLO with iMX6S/DL on MX6CustomBoard with resistive touch
imx6dl-var-som-solo-vsc.dtb	VAR-SOM-SOLO with iMX6S/DL on SOLOCustomBoard with capacitive touch
imx6dl-var-som-cap.dtb	VAR-SOM-MX6 with iMX6S/DL on MX6CustomBoard with capacitive touch
imx6dl-var-som-res.dtb	VAR-SOM-MX6 with iMX6S/DL on MX6CustomBoard with resistive touch
imx6dl-var-som-vsc.dtb	VAR-SOM-MX6 with iMX6S/DL on SOLOCustomBoard with capacitive touch
imx6q-var-som-cap.dtb	VAR-SOM-MX6 with iMX6D/Q on MX6CustomBoard with capacitive touch
imx6q-var-som-res.dtb	VAR-SOM-MX6 with iMX6D/Q on MX6CustomBoard with resistive touch
imx6q-var-som-vsc.dtb	VAR-SOM-MX6 with iMX6D/Q on SOLOCustomBoard with capacitive touch
imx6qp-var-som-cap.dtb	VAR-SOM-MX6 with iMX6DP/QP on MX6CustomBoard with capacitive touch
imx6qp-var-som-res.dtb	VAR-SOM-MX6 with iMX6DP/QP on MX6CustomBoard with resistive touch
imx6qp-var-som-vsc.dtb	VAR-SOM-MX6 with iMX6DP/QP on SOLOCustomBoard with capacitive touch
imx6q-var-dart.dtb	DART-MX6 with iMX6D/Q on DARTCustomBoard with capacitive touch

7. Linux console access

User name	User password	User descriptor
root	root	system administrator
user	user	local user
x_user		used for X session access

8. Flash images to eMMC

To install Debian to the on-SOM eMMC, run the following command **as root**:

```
# debian-install.sh -b <mx6cb|scb|dart> -t <cap|res>
```

where the "-b" option provide the carrier board used (MX6CustomBoard/SOLOCustomBoard/DART-MX6) and the "-t" option provide the touch type when using MX6CustomBoard (ignored otherwise). The above script is located in `/usr/sbin` in the rootfs of the SD card used to boot Debian.

9. How-to: Test and use an interface

Please see this section in the [Yocto developer guide page](#). It is the same for Debian.

10. How-to: Modify the kernel configuration

To modify the kernel configuration (add/remove features and drivers) please follow the steps below:

```
1. $ cd ~/debian_var-som-mx6/src/kernel
2. $ sudo make ARCH=arm mrproper
3. $ sudo make ARCH=arm imx_v7_var_defconfig
4. $ sudo make ARCH=arm menuconfig
```

```
7. $ sudo make ARCH=arm menuconfig
```

```
5. Navigate the menu and select the desired kernel functionality
```

```
6. Exit the menu and answer "Yes" when asked "Do you wish to save your new configuration?"
```

```
7. $ sudo make ARCH=arm savedefconfig
```

```
8. $ sudo cp arch/arm/configs/imx_v7_var_defconfig arch/arm/configs/imx_v7_var_defconfig.orig
```

```
9. $ sudo cp defconfig arch/arm/configs/imx_v7_var_defconfig
```

```
10. Follow the instructions above to rebuild kernel and modules, repack rootfs images and recreate SD card
```

11. Build a sample C "Hello, world!" program

Create a file called myhello.c with the following content:

```
#include <stdio.h>

int main() {
    printf("Hello, World!\n");
    return 0;
}
```

Export the C (cross-)compiler path:

```
$ export CC=/debian_var-som-mx6/toolchain/gcc-linaro-6.3.1-2017.05-x86_64_arm-linux-gnueabihf/bin/arm-linux-gnueabihf-gcc
```

Compile:

```
$ $CC myhello.c -o myhello
```

Now you should have an app called myhello, that can be run on your target board.

You can add it to your rootfs image or copy it directly to the rootfs on the board (using scp, for example).

Categories: [VAR-SOM-MX6](#) | [Debian](#)

This page was last edited on 16 September 2021, at 20:02.

[Privacy policy](#) [About Variscite Wiki](#) [Disclaimers](#)

© [2021] Variscite. All rights reserved