EM-TF-EVK-AM5728

Linux Development Guide

Version: 1.0 2024-05-23

Disclaimer

 Shenzhen Emtop Co., Ltd. does not provide any type of guarantee for the program source code, software, documents, etc. provided with the product; Regardless of whether it is described or implied, including but not limited to the warranty of fitness for a particular purpose, the entire risk is born by the user.

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

Revision History:

Version	Date	Description
1.0	2024-05-23	First Release

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

Table of Contents

1.	ENVIRONM	ENT SETUP	6
1	.1	BUILD A DEVELOPMENT ENVIRONMENT	6
1	.2	CONFIGURE COMPILATION ENVIRONMENT	6
1	.3	OTHER TOOLS AND SERVICES	7
2.	COMPILE S	OURCE CODE	8
2	.1	U-Воот	8
	2.1.1	GET U-BOOT SOURCE CODE	8
	2.1.2	COMPILE U-BOOT	8
2	.2	KERNEL	8
	2.2.1	GET KERNEL SOURCE CODE	8
	2.2.2	COMPILE KERNEL	9
2	.3	EXTERNAL DRIVE	9
	2.3.1	CONFIGURE ENVIRONMENT VARIABLES	9
	2.3.2	COMPILE EXTERNAL DRIVER	11
	2.3.3	INSTALL EXTERNAL DRIVER	11
3.	CREATE TA	RGET IMAGE	14
3	.1	MAKE IMAGE FILE	14
	3.1.1	FORMAT THE PARTITIONS	14
	3.1.2	COPY FIRMWARE	17
	3.1.3	COPY ROOTFS	17
	3.1.4	INSTALL KERNEL MODULES	18
	3.1.5	INSTALL EXTERNAL DRIVER MODULE	18
3	.2	BURN AND READ IMAGE	19
	3.2.1	BURN IMAGE	19
	3.2.2	READ IMAGE	19

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

4.	TI SDK DE	VELOPMENT	21
4	1.1	SDK INSTALLATION AND CONFIGURATION	21
4	1.2	TOP-LEVEL MAKEFILE USAGE	25
	4.2.1	COMPILE THE ENTIRE SDK	25
	4.2.2	COMPILE AND INSTALL A TARGET SEPARATELY	26
5.	APPLICAT		29
5	5.1	CROSS COMPILE AND RUN ORDINARY C PROGRAMS	29
	5.1.1	WRITE C PROGRAM CODE	29
	5.1.2	COMPILE ON HOST	29
	5.1.3	COMPILE ON ARM BOARD	29
	5.1.4	TRANSMIT TO ARM BOARD AND RUN	30
5	5.2	QT APPLICATION DEVELOPMENT	30
	5.2.1	INSTALL QT CREATOR	30
	5.2.2	CONFIGURE QT CREATOR	37
	5.2.3	CREATE DEMO	41
	5.2.4	RUNNING ON ARM BOARD	47
5	5.3	VIDEO CAPTURE DEMO	50
5	5.4	DUAL DISPLAY DEMO	51
6.	ROOTFS E	BUILD BASED ON YOCTO	53
6	6.1	INSTALL REQUIRED TOOL SOFTWARE	53
6	6.2	CONFIGURE BASH	53
6	6.3	INSTALL COMPILER	54
6	6.4	OBTAIN OE-LAYERTOOL-SETUP.SH	54
6	6.5	BITBAKE BUILD	54
7.	APPENDIX	(56

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

1. ENVIRONMENT SETUP

1.1 BUILD A DEVELOPMENT ENVIRONMENT

Requirements for setting up the development environment:

- Hardware: At least 20GB disk space, 2GB RAM
- Software: Ubuntu 64 bit OS, 14.04 LTS, 16.04LTS or higher LTS version (Ubuntu Desktop or Ubuntu Server)

Note:

If you want to develop Qt, please install Ubuntu version with a GUI interface such as Ubuntu Desktop.

You can also use a virtual machine to run the Ubuntu 64 bit OS. After starting the Ubuntu system, run the following commands to install the software required for development.

- \$ sudo apt-get update
- \$ apt-get install -y openssh-server git kpartx Izop Isb-core libncurses5

1.2 CONFIGURE COMPILATION ENVIRONMENT

Copy gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-gnueabihf.tar.xz from the release folder to the \$HOME directory in the Linux environment, and unzip it:

\$ tar -Jxvf gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-gnueabihf.tar.xz

Set environment variables:

- \$ export CROSS_COMPILE=\$HOME/gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-g nueabihf/bin/arm-linux-gnueabihf-
- \$ export ARCH=arm

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

Note: Environment variables must be configured before compiling u-boot and kernel each

time. For convenience, you can edit a script and then source the script:

\$ cat \$HOME/set_am57_env.sh

#!/bin/bash

export CROSS_COMPILE=\$HOME/gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-gnueabihf /bin/arm-linux-gnueabihf-

export ARCH=arm

\$ source \$HOME/set_am57_env.sh

Note:

- The command strings starting with \$ in this article should run under Ubuntu PC;
- SHOME appearing in the article is not mandatory and should be modified accordingly based on the user's actual directory;
- □ In this article, the command strings starting with "root@arm:~#" should run under the ARM board.

1.3 OTHER TOOLS AND SERVICES

Other tools and services may also be used during the development process, such as:

- ssh login ubuntu/serial port login: PuTTY software
- SAMBA service for transferring files between linux and windows
- NFS
- TFTP

The use of these conventional development tools isn't mentioned in the document. Please search on web for more information.

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

2. COMPILE SOURCE CODE

2.1 U-Boot

2.1.1 GET U-BOOT SOURCE CODE

Copy u-boot*.tar.gz to \$HOME and unzip it:

- \$ cd \$HOME
- \$ tar -xzvf u-boot*.tar.gz

2.1.2 COMPILE U-BOOT

- s cd \$HOME/u-boot
- s make distclean
- smake som_am572x_defconfig
- s make

After compilation is completed, <u>u-boot.img</u> and <u>MLO</u> are generated in the <u>\$HOME/</u> <u>u-boot</u> directory.

2.2 KERNEL

2.2.1 GET KERNEL SOURCE CODE

Copy the Linux kernel source code package linux*.tar.gz to \$HOME and unzip it:

\$ tar -zxvf linux*.tar.gz

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

2.2.2 COMPILE KERNEL

- s cd \$HOME/linux
- \$ make distclean
- \$ make embest_ti_am57xx_defconfig
- \$ make

After compilation is completed, the target file **<u>zImage</u>** and **<u>dtb</u>** are generated:

- \$HOME/linux/arch/arm/boot/zImage
- \$HOME/linux/arch/arm/boot/dts/embest-SOM_AM572x_TM-mode0.dtb
- \$HOME/linux/arch/arm/boot/dts/embest-SOM_AM572x_TM-mode0-LCD.dtb

embest-SOM_AM572x_TM-mode0.dtb sets HDMI as the main display, and

embest-SOM AM572x TM-mode0-LCD.dtb sets LCD as the main display.

2.3 EXTERNAL DRIVE

Since the drivers for some of TI's peripheral modules are released separately, these drivers require additional compilation. These peripheral modules include 2D/3D image acceleration modules, hardware encryption/decryption modules, etc. Copy <u>extra.tar.gz</u> to \$HOME and extract it.

- \$ cd \$HOME
- \$ tar -xzf extra.tar.gz

2.3.1 CONFIGURE ENVIRONMENT VARIABLES

Edit the **<u>Rules.make</u>** file and modify the following variables to the corresponding values:

DESTDIR The path of the root file system, such as the mounting path of the SD card that has been burned with the image under ubuntu, or it can be any other

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

blank directory.

- **CROSS_COMPILE** is the path where the cross-compilation tool chain is located.
- LINUXKERNEL_INSTALL_DIR is the Linux kernel source code path. Before compiling the external driver, make sure that the Linux kernel source code is correctly configured and compiled. Refer to 2.2 KERNEL.
 - \$ cd extra
 - \$ ls

extra-drivers Makefile Rules.make

\$ cat Rules.make

#platform
#platform
PLATFORM=am57xx-evm
#root of the target file system for installing applications
DESTDIR=\$(HOME)/extra/fakeroot
#Cross compiler prefix
export
CROSS_COMPILE=\$(HOME)/gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-gnueabihf/bin/ar
m-linux-gnueabihf-
#The directory that points to the SDK kernel source tree
LINUXKERNEL_INSTALL_DIR=\$(HOME)/linux
CFLAGS= -march=armv7-a -marm -mfpu=neon -mfloat-abi=hard
#Strip modules when installing to conserve disk space
INSTALL_MOD_STRIP=1
export TOOLCHAIN PREFIX=\$(CROSS_COMPILE)

www.emtop-tech.com	https://github.com/EMTOP-TECH					
sales@emtop-tech.com	support@emtop-tech.com					

2.3.2 COMPILE EXTERNAL DRIVER

Before compiling, create a empty directory specified by the **DESTDIR** variable in

Rules.make. Here we create the fakeroot directory:

- \$ cd extra
- \$ mkdir fakeroot
- \$ Is

extra-drivers fakeroot Makefile Rules.make

- \$ make clean
- \$ make

2.3.3 INSTALL EXTERNAL DRIVER

s make install

The compiled ko and related files will be installed under <u>fakeroot</u>. If you only need to update the ko file, you only need to copy the ko files to <u>lib/modules/4.9.28/extra</u> to the corresponding directory of the target board file system.

\$ tree fakeroot

•

fakeroot/
├─── lib
│
│ │ └───jailhouse.bin
│ └─── modules
4.9.28
│
│ │
cmemk.ko
│ │
debugss_kmodule.ko
│ │ ├─── driver
│ │ │ └───jailhouse.ko

www.emtop-tech.com	https://github.com/EMTOP-TECH				
sales@emtop-tech.com	support@emtop-tech.com				

EMBEDDED SOLUTIONS

EM-TF-EVK-AM5728 LDG

12



For development convenience, you can turn off the git version control option in the kernel. The kernel versions compiled in the future will always be linux-4.9.28, so there is no need to repeatedly compile extra-related driver modules.

www.emtop-tech.com	https://github.com/EMTOP-TECH				
sales@emtop-tech.com	support@emtop-tech.com				

EM-TF-EVK-AM5728 LDG

13

Enter make menuconfig in the kernel root directory to open the configuration interf ace, type / in the menuconfig interface, and search CONFIG LOCALVERSION AUT O option, find the relevant option according to the prompts, and set it to [=n]:

					- 6	ماصحد	Con	£;	i a mət	ion F)	an ot or	n -					
Enter	• ((sub))stri	ing	or	rege:	xp t	0	searc	n for	- (i	with a	or	without	"(CONFI	G_")	
CONFI	:G_	LOCA	ALVER	RSIO	DN_A	AUTO												
																		_
						<	Ok	\rightarrow		< He	lp	>						

.config - Linux/arm 4.9.28 Kernel Configuration > Search (CONFIG_LOCALVERSION_AUTO)

- Search Results -Symbol: LOCALVERSION_AUTO [=n] Type : boolean Prompt: Automatically append version information to the version string Location: $(1) \rightarrow$ General setup Defined at init/Kconfig:91 Depends on: !COMPILE_TEST [=n]

During the development phase, CONFIG_MODULE_FORCE_LOAD can be enabled.

Enable loadable module_support

- [×] Forced module loading
- [*] Module unloading
- [*] Forced module unloading [*] Module versioning support
- [*] Source checksum for all modules
- Module signature verification
- Compress modules on installation
- Trim unused exported kernel symbols

www.emtop-tech.com	https://github.com/EMTOP-TECH					
sales@emtop-tech.com	support@emtop-tech.com					

3. CREATE TARGET IMAGE

3.1 MAKE IMAGE FILE

There are prepared image files (such as

<u>EM-TF-EVK-AM5728-TI-ShipmentImage-SDcard-V1.0.3r04.img</u>) in the release folder for users to use the **dd** command (Linux OS) and <u>Win32DiskImager.exe</u> (Windows OS) to write into the SD card or EMMC, please refer to the user manual for details. In this section we will introduce how to make the target files compiled in the previous sections into an image file that can be used for burning. The size of the image file we create should not exceed the capacity of the SD card/EMMC to be burned. Assuming the SD card capacity is 4GB, we can create a 3800MB blank disk file named <u>example.img</u> under Ubuntu:

\$ cd \$HOME

\$ sudo dd if=/dev/zero of=./example.img bs=1M count=3800

3800+0 records in 3800+0 records out 3984588800 bytes (4.0 GB, 3.7 GiB) copied, 9.85035 s, 405 MB/s

3.1.1 FORMAT THE PARTITIONS

Format **example.img**, create two partitions: FAT32 (64MB, storing firmware) and EXT4 (storing rootfs).

- \$ cd \$HOME
- \$ sudo fdisk example.img

```
Welcome to fdisk (util-linux 2.27.1).
Changes will remain in memory only, until you decide to write them.
```

www.emtop-tech.com	https://github.com/EMTOP-TECH					
sales@emtop-tech.com	support@emtop-tech.com					

EMBEDDED SOLUTIONS

EM-TF-EVK-AM5728 LDG

15

Be careful before using the write command. Device does not contain a recognized partition table. Created a new DOS disklabel with disk identifier 0x928aa0d6. Command (m for help): o Created a new DOS disklabel with disk identifier 0x12076151. Command (m for help): n Partition type p primary (0 primary, 0 extended, 4 free) e extended (container for logical partitions) Select (default p): p Partition number (1-4, default 1): 1 First sector (2048-7782399, default 2048): Last sector, +sectors or +size{K,M,G,T,P} (2048-7782399, default 7782399): +64M Created a new partition 1 of type 'Linux' and of size 64 MiB. Command (m for help): t Selected partition 1 Partition type (type L to list all types): c Changed type of partition 'Linux' to 'W95 FAT32 (LBA)'. Command (m for help): a Selected partition 1 The bootable flag on partition 1 is enabled now. Command (m for help): n Partition type p primary (1 primary, 0 extended, 3 free) e extended (container for logical partitions) Select (default p): p Partition number (2-4, default 2): First sector (133120-7782399, default 133120): Last sector, +sectors or +size{K,M,G,T,P} (133120-7782399, default 7782399): Created a new partition 2 of type 'Linux' and of size 3.7 GiB. Command (m for help): p Disk example.img: 3.7 GiB, 3984588800 bytes, 7782400 sectors Units: sectors of 1 * 512 = 512 bytes Sector size (logical/physical): 512 bytes / 512 bytes I/O size (minimum/optimal): 512

www.emtop-tech.com	https://github.com/EMTOP-TECH					
sales@emtop-tech.com	support@emtop-tech.com					

EMBEDDED SOLUTIONS

EM-TF-EVK-AM5728 LDG

16

bytes / 512 bytes							
Disklabel type: do	Disklabel type: dos						
Disk identifier: 0x	12076151						
Device Boo	ot Start	End	Sectors	Size	Id	Туре	
example.img1 *	2048	133119	131072	64M	с	W95 FAT32 (LBA)	
example.img2	133120	7782399	7649280	3.7G	83	Linux	
Command (m for	help): w						
The partition table	e has bee	n altered.					
Syncing disks							

Format the FAT32 partition and set the volume label to boot, format the ext4 partition and set the volume label to rootfs.

\$ sudo losetup /dev/loop0 example.img

\$ sudo kpartx -av /dev/loop0

add map loop0p1 (253:0): 0 131072 linear 7:0 2048 add map loop0p2 (253:1): 0 7649280 linear 7:0 133120

\$ Is /dev/mapper/loop0p*

/dev/mapper/loop0p1 /dev/mapper/loop0p2

\$ sudo mkfs.vfat -F 32 -n "boot" /dev/mapper/loop0p1

```
mkfs.fat 3.0.28 (2015-05-16)
mkfs.fat: warning - lowercase labels might not work properly with DOS or Windows
unable to get drive geometry, using default 255/63
```

\$ sudo mkfs.ext4 -L "rootfs" /dev/mapper/loop0p2

```
mke2fs 1.42.13 (17-May-2015)
Discarding device blocks: done
Creating filesystem with 956160 4k blocks and 239040 inodes
Filesystem UUID: 0820d179-521d-4f91-816f-df13309eee87
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912, 819200, 884736
Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

www.emtop-tech.com	https://github.com/EMTOP-TECH					
sales@emtop-tech.com	support@emtop-tech.com					

EMBEDDED SOLUTIONS

3.1.2 COPY FIRMWARE

Create two temporary directories, respectively for mounting:

- \$ mkdir boot rootfs
- \$ sudo mount /dev/mapper/loop0p1 boot/
- \$ sudo mount /dev/mapper/loop0p2 rootfs/

You can check whether the mount is successful with the Isblk command:

• \$ Isblk

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
loop0	7:0	0	3.7G	0	loop	
├──loop0p2	253:1	0	3.7G	0	part	/home/david/rootfs
└—loop0p1	253:0	0	64M	0	part	/home/david/boot

- \$ sudo cp \$HOME/u-boot/u-boot.img ./boot
- \$ sudo cp \$HOME/u-boot/MLO ./boot
- \$ sudo cp \$HOME/linux/arch/arm/boot/zlmage ./boot
- \$ sudo cp \$HOME/linux/arch/arm/boot/dts/embest-SOM_AM572x_TM-mode0.dtb . /boot
- \$ sudo cp \$HOME/linux/arch/arm/boot/dts/embest-SOM_AM572x_TM-mode0-LCD.
 dtb ./boot

Create uEnv.txt file in directory **boot** and specify the dtb file, for example, use HDMI as the

main display

- \$ sudo touch ./boot/uEnv.txt
- \$ sudo bash -c "echo fdtfile=embest-SOM_AM572x_TM-mode0.dtb > ./boot/uEn v.txt"

3.1.3 COPY ROOTFS

Unzip tisdk-rootfs-image-am57xx-evm.tar.xz to \$HOME/rootfs-arago, and copy all the

www.emtop-tech.com	https://github.com/EMTOP-TECH				
sales@emtop-tech.com	support@emtop-tech.com				

contents of <u>tisdk-rootfs-image-am57xx-evm.tar.xz</u> to the <u>rootfs</u> directory. Note that you need to use **cp** -**ap** option to ensure that the file's attributes remain unchanged.

- \$ mkdir rootfs-arago
- \$ tar -Jxvf tisdk-rootfs-image-am57xx-evm.tar.xz -C rootfs-arago/
- \$ sudo cp -ap rootfs-arago/* rootfs

3.1.4 INSTALL KERNEL MODULES

Installing kernel modules requires root privileges, so you also need to set environment variables for the root account.

- \$ cd \$HOME/linux
- \$ sudo make modules_install INSTALL_MOD_PATH=\$HOME/rootfs ARCH=arm CROSS_COMPILE=\$HOME/gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-gnueabihf/b in/arm-linux-gnueabihf-

3.1.5 INSTALL EXTERNAL DRIVER MODULE

Refer to 2.3.3 INSTALLING EXTERNAL DRIVER, set the DESTDIR variable to

<u>\$HOME/rootfs</u>, and then install it.

٠

If the user has not upgraded the kernel version, the <u>rootfs-arago.tar.gz</u> root file system we provide already contains the kernel module and external driver module, so there is no need to reinstall it.

After the copy operation completes, unmount the two partitions and synchronize the file system:

\$ sudo umount boot rootfs

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

EMBEDDED SOLUTIONS

19

- \$ sudo kpartx -d /dev/loop0
- \$ sudo losetup -d /dev/loop0
- \$ sync

3.2 BURN AND READ IMAGE

3.2.1 BURN IMAGE

Please refer to the user manual.

3.2.2 READ IMAGE

During the development process, it is often necessary to read the image from the SD card

for backup. You can use the following command to obtain the image file from the SD card.

Under Linux OS

Connect the SD card into the card reader and connect it to the computer:

• \$ Isblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT sdb 8:16 1 7.2G 0 disk ├--sdb2 8:18 1 3.8G 0 part └--sdb1 8:17 1 64M 0 part

\$ dd if=/dev/sdb of=./sdcard.img

Under Windows OS

Connect the SD card to the computer and run Win32 Disk Imager

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com



👒 Win32 Disk Imager	
Image File	Device
	[Н: \] 🔻
Copy MD5 Hash:	
Progress	
Version: 0.9.5 Cancel Read Wr:	ite Exit

Select the storage path of the target image file, such as: D:/temp/SDCard.img

👒 Win32 Disk Imager	
Image File	Device
1	EH: \] 🔻
Copy MD5 Hash:	×
Progress	Select Image File
Version: 0.9.5 Cancel Read	Write Exit

Click **Read** button to read the contents of the SD card into the image file:

🎭 Win32 Disk Imager	
Image File	Device
D:/temp/SDCard.img	[G:\] 🔻
Copy MD5 Hash:	
Progress	Read to the Image
L	
Version: 0.9.5 Cancel Read	Write Exit
Read data from 'Device' to 'Image File'	H.

After successful complete, you will get a complete SD image.

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

4. TI SDK DEVELOPMENT

4.1 SDK INSTALLATION AND CONFIGURATION

The TI SDK installation package can be installed on the ssh command line or in the

ubuntu desktop environment. Let's take the desktop environment installation as an example:

Press the key combination Ctrl+Alt+T on the Ubuntu desktop to open the console

- \$ cd \$HOME
- \$ sudo chmod +x ti-processor-sdk-linux-am57xx-evm-04.00.00.04-Linux-x86-Inst all.bin
- \$ sudo ./ti-processor-sdk-linux-am57xx-evm-04.00.00.04-Linux-x86-Install.bin

Pop up a dialog:

😣 🖨 Setup	
	Setup - ti-processor-sdk-linux-am57xx-evm-04.00.00.04
	Welcome to the ti-processor-sdk-linux-am57xx-evm-04.00.00.04 Setup Wizard.
	< <u>Back</u> Cancel

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

EM-TF-EVK-AM5728 LDG

22

According to the guidance, click Next. When the installation path appears, you can use the

default path for installation, or you can customize the installation path.

😣 🖨 Setup	
Choose Destination Location	
Setup will install ti-processor-sdk-linux-am57xx-evm in the follo	wing folder.
To install to this folder, click "Forward". To install to a different fo another folder.	lder, click the browse icon and select
Destination Folder /opt/ti-processor-sdk-linux-am57xx-evm-04	.00.00.04
InstallBuilder	
	< <u>Back</u> <u>N</u> ext > Cancel

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

EMTOP EMBEDDED SOLUTIONS	EM-TF-EVK-AM5728 LDG
	23
😣 🖨 Setup	
Installing	
Please wait while Setup installs ti-processor-sdk-linu:	x-am57xx-evm-04.00.00.04 on your computer.
Insta	lling
Unpacking /opt/ti-pro[]4/processor	-sdk-linux-image-am57xx-evm.tar.xz
InstallPuilder	
instanbunder.	< Back Next > Cancel

After about a few minutes, the installation is complete, click "Finish".



www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

TI's SDK directory is organized as follows:

\$ cd /opt/ti-processor-sdk-linux-am57xx-evm-04.00.00.04/

\$ ls -l

total 542220
drwxr-xr-x 2smbusersmbuser 4096 Aug 10 16:18 bin
drwxr-xr-x 6smbusersmbuser 4096 Jun 29 2017 board-support
drwxr-xr-x 3smbusersmbuser 4096 Jun 29 2017 docs
drwxr-xr-x 19 smbusersmbuser 4096 Jun 29 2017 example-applications
drwxr-xr-x 2smbusersmbuser 4096 Jun 29 2017 filesystem
drwxr-xr-x 3 root root 4096 Aug 10 16:18 linux-devkit
-rwxr-xr-x 1 smbusersmbuser 555147047 Jun 29 2017 linux-devkit.sh
-rwxr-xr-x 1 smbusersmbuser 44597 Jun 29 2017 Makefile
-rwxr-xr-x 1 smbusersmbuser 1324 Aug 10 16:18 Rules.make
-rwxr-xr-x 1 smbusersmbuser 4188 Jun 29 2017 setup.sh

The uses of each directory/file are as follows:

bin	Some tool scripts for making SD cards, setting up TFTP, etc.	
board-support	1. u-boot source code;	
	2. Kernel source code;	
	3. External driver source code;	
	4. Pre-compiled firmware for TI's evaluation board	
docs SDK software list and license		
example-applications Demo program source code		
filesystem Pre-built root file system based on Arago		
linux-devkit	The rootfs system required to compile the entire SDK, which includes	
	the cross-compilation tool chain	
	The compressed package of the linux-devkit folder. This file is needed	
linux-devkit.sn	after moving the installed SDK path.	
	The top-level Makefile can compile the entire SDK, including:	
	1. Linux Kernel	
Makefile	2. U-boot	
	3. External driver	
	4. Demo program	
Rules.make	Some environment variables needed to compile the SDK, such as	
	cross-editing tool chain and target root file system path	
a atum a h	Setting up the development environment actually calls some tool scripts	
setup.sn	in the bin directory.	

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

25

4.2 TOP-LEVEL MAKEFILE USAGE

The top-level Makefile in the SDK contains many targets. Through this Makefile, you can

compile all, clean, and install them all, or you can specify a target individually for

compilation, cleanup, and installation.



4.2.1 COMPILE THE ENTIRE SDK

The default target "all" of the top-level Makefile can compile the entire SDK, so only one

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

command is needed:

\$ sudo make

The compilation process starts from the first target dtbs and Linux. It takes some time. If you need to install the compiled product to the target file system after compilation is completed, you need to modify **DESTDIR** in <u>**Rule.make**</u> to the directory where the actual rootfs system is located.

- \$ cd /opt/ti-processor-sdk-linux-am57xx-evm-04.00.00.04
- \$ sudo make install
- \$ cd ../fakeroot
- \$ Is -I

```
total 12
drwxr-xr-x 2 root root 4096 Aug 10 17:39 boot
drwxr-xr-x 4 root root 4096 Aug 10 17:39 lib
drwxr-xr-x 4 root root 4096 Aug 10 17:39 usr
```

4.2.2 COMPILE AND INSTALL A TARGET SEPARATELY

Take compiling the dual-camera demo program as an example:

- \$ cd /opt/ti-processor-sdk-linux-am57xx-evm-04.00.00.04
- \$ sudo make dual-camera-demo

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

EM-TF-EVK-AM5728 LDG

27

emo-1.0'
echo "manisha" am57xx-evm
manisha am57xx-evm
make[2]: Entering directory
`/opt/ti-processor-sdk-linux-am 57 xx-evm-04.00.00.04/example-applications/dual-camera-d
emo-1.0'
make[2]: Nothing to be done for `first'.
make[2]: Leaving directory
`/opt/ti-processor-sdk-linux-am 57 xx-evm-04.00.00.04/example-applications/dual-camera-d
emo-1.0'
make[1]: Leaving directory
`/opt/ti-processor-sdk-linux-am 57 xx-evm-04.00.00.04/example-applications/dual-camera-d
emo-1.0'

Install

•

\$ sudo make dual-camera-demo_install

Makefile:715: warning: overriding commands for target `ti-gc320-driver'			
Makefile:595: warning: ignoring old commands for target `ti-gc320-driver'			
Makefile:723: warning: overriding commands for target `ti-gc320-driver_clean'			
Makefile:603: warning: ignoring old commands for target `ti-gc320-driver_clean'			
Makefile:731: warning: overriding commands for target `ti-gc320-driver_install'			
Makefile:611: warning: ignoring old commands for target `ti-gc320-driver_install'			
Installing Dual Camera Demo - Release version			
make[1]: Entering directory			
`/opt/ti-processor-sdk-linux-am 57xx-evm-04.00.00.04/example-applications/dual-camera-du			
emo-1.0'			
echo "manisha" am57xx-evm			
manisha am57xx-evm			
make[2]: Entering directory			
`/opt/ti-processor-sdk-linux-am57xx-evm-04.00.00.04/example-applications/dual-camera-d			
emo-1.0'			
make[2]: Nothing to be done for `first'.			
make[2]: Leaving directory			
`/opt/ti-processor-sdk-linux-am57xx-evm-04.00.00.04/example-applications/dual-camera-d			
emo-1.0'			
dual_camera release version installed.			

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

EMBEDDED SOLUTIONS

EM-TF-EVK-AM5728 LDG

28

make[1]: Leaving directory

`/opt/ti-processor-sdk-linux-am 57 xx-evm-04.00.00.04/example-applications/dual-camera-demo-1.0'

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

5. APPLICATION DEVELOPMENT

5.1 CROSS COMPILE AND RUN ORDINARY C PROGRAMS

5.1.1 WRITE C PROGRAM CODE

Let's get a 4G communication program **application.tar.gz** as example, unzip it to \$HOME directory.

5.1.2 COMPILE ON HOST

- \$ cd \$HOME/application/4g_test
- \$ \$HOME/gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-gnueabihf/bin/arm-linux-gnueabihf-gcc 4G_test.c -o 4G_test
- \$ Is

4G_test 4G_test.c readme.md

\$ file 4G_test

4G_test: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked (u ses shared libs), for GNU/Linux 2.6.24, BuildID[sha1]=69a5b54a2de0c56b075f871fff671 0797250a72c, not stripped

5.1.3 COMPILE ON ARM BOARD

The compilation toolchain is already installed in the rootfs system of our release, so the C source code files can be copied to compile the application directly on the board using gcc

```
root@arm:~# gcc 4G_test.c -o 4G_test
```

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

.

EMBEDDED SOLUTIONS

30

root@arm:~# **Is**

4G_test 4G_test.c readme.md

5.1.4 TRANSMIT TO ARM BOARD AND RUN

- Connect the ARM board and computer to the local network and use the scp command to transfer files.
 - root@arm:~# scp <UbuntuUser>@<UbuntulPAddr>:/home/david/application_test_ programs/4g_test/4g_test ./
- Copy using storage media such as USB flash drive.

.....

5.2 QT APPLICATION DEVELOPMENT

5.2.1 INSTALL QT CREATOR

Qt Creator is a graphical designer, so this section of the operation is performed in the

Ubuntu desktop environment. Copy gt-creator-opensource-linux-x86 64-4.1.0.run to

<u>\$HOME</u> and add executable permissions.

\$ sudo chmod +x qt-creator-opensource-linux-x86_64-4.1.0.run

Start to install:

٠

\$./qt-creator-opensource-linux-x86_64-4.1.0.run

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

😣 🗉 Qt Creator 4.1.0 Setup

Qt Account - Your unified login to everything Qt

Qt	Login	Please log in to Qt Account Email Password Forgot password? Need a Qt Account?
Create Once. Deploy Everywhere.	Sign-up	Valid email address Password Confirm Password I accept the service terms.
Settings		< <u>B</u> ack Skip Cancel

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

😣 💷 Qt Creator 4.1.0 Setup

Setup - Qt Creator 4.1.0



www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

EM-TF-EVK-AM5728 LDG

33

😣 🗉 🛛 Qt Creator 4.1.0 Setup

Installation Folder

home/david/qtcreator-4.1.0		B <u>r</u> owse

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

EM-TF-EVK-AM5728 LDG

34

*

Ŧ

🔊 🗊 🛛 Qt Creator 4.1.0 Setup

License Agreement

Please read the following license agreement. You must accept the terms co in this agreement before continuing with the installation.

This is the GNU General Public License version 3, annotated with The Qt Company GPL Exception 1.0:

The Qt Company GPL Exception 1.0

Exception 1:

As a special exception you may create a larger work which contains the output of this application and distribute that work under terms of your choice, so long as the work is not otherwise derived from or based on this application and so long as the work does not in itself generate output that contains the output from this application in its original or modified form.

Exception 2:

As a special exception, you have permission to combine this application

with Plugins licensed under the terms of your choice to produce an

- I have read and agree to the terms contained in the license agreements.
- I do not accept the terms and conditions of the above license agreements.



www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

EMTOP EMBEDDED SOLUTIONS	EM-TF-EVK-AM5728 LDG
	35
QC Creator 4.1.0 Setup	
Ready to Install	
Setup is now ready to begin installing Installation will use 258.04 MiB of dis	g Qt Creator 4.1.0 on your computer. k space.
	< Back Install Cancel
_	

According to the guidance, install Qt creator to **<u>\$HOME/qtcreator-4.1.0</u>**.

After the installation is completed, the following picture appears:

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

👂 💿 🛛 Qt Creator 4.1.0 Setup

Completing the Qt Creator 4.1.0 Wizard



After the last step, please do not start Qt Creator right now. You need to start it from the TI

SDK because there are some necessary environment variables that need to be set.

\$ source /opt/ti-processor-sdk-linux-am57xx-evm-04.00.00.04/linux-devkit/environ ment-setup

Then switch to the Qt Creator installation directory./qtcreator starts

[linux-devkit]:~> cd ~/qtcreator-4.1.0/bin/

•

[linux-devkit]:~/qtcreator-4.1.0/bin> ./qtcreator

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

37

5.2.2 CONFIGURE QT CREATOR

Qt Creator starts as shown below:

80	Qt Creator		
<u>File</u> <u>E</u> dit	<u>Build Debug</u> Analyze Tools W	Vindow Help	
Welcome	Projects	+ New Project 🕞 Open Project	
Edit	Examples	Sessions Recent Projects	
Design	Tutorials	▶ default	
û Debug	New to Qt?		
پ Projects	Learn how to develop your own applications and explore Qt Creator.		
Pelp	Get Started Now		
	Ot Account		
	Online Community		
	Blogs		
Ĩ	Oser Guide		
-			
\nearrow	O Type to locate (Ctrl + K)	1 Incure - 2 Coardo Deculto - 2 Application Output - 4 Compile Output - 5 Debugger Concela - 4	
	P+ Type to locate (CtrI+K)	1 issues 2 Search Results 5 Application Output 4 Compile Output 5 Debugger Console \Rightarrow	

Before using Qt Creator to develop a program that runs on AM5728, you need to configure the cross-compilation tool, QT version, Debuggers version, and Kits.

Configure the cross-compilation tool chain

Click the menu bar Tool->Options->Build&Run->Compilers->Add->GCC, click Browse to select the Compiler path as <u>/opt/ti-processor-sdk-linux-am57xx-evm-04.00.00.0</u> <u>4/linux-devkit/sysroots/x86_64-arago-linux/usr/bin/arm-linux-gnueabihf-g++</u> (gcc p ath of TI SDK), then click Open and click Apply.

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

EMTOP	
-------	--

EMBEDDED SOLUTIONS

EM-TF-EVK-AM5728 LDG

38

3 Dptions	
Filter Build & Run	
Environment General Kits Qt Versions Compilers Debuggers CMake	
Text Editor Name Type	Add 👻
FakeVim GCC (x86 64bit in /usr/bin) GCC	Clone
② Help ▼ Manual	Remove
{} c++	
Qt Quick	
🚯 Build & Run	
Debugger	
Designer	
Analyzer	
Version Control Platform codegen flags:	
Android Platform linker flags:	
°œx QNX ABI: arm-linux ▼ arm ▼ - linux ▼ - generic ▼ - elf ▼ - 32bit ▼	j
Devices	
Code Pasting	
Apply <u>C</u> ance	<u>о</u> к

Configure Qt version

Click "Qt Versions->Add" and select the qmake file path of TI SDK:

/opt/ti-processor-sdk-linux-am57xx-evm-04.00.00.04/linux-devkit/sysroots/x86_64-arago-l

inux/usr/bin/qt5/qmake, then click Apply button.

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

EM-TF-EVK-AM5728 LDG

EMTOP EMBEDDED SOLUTIONS

2	٥
J	J

😣 🗐 Options	
Filter	Build & Run
Environment	General Kits Qt Versions Compilers Debuggers CMake
Text Editor	Name • qmake Location Ty
FakeVim	Auto-deteted ▼ Manual Ot 5.7.1 in PATH (at5) /ont/tijnrosessor.sd/slinux.am57xy.evm.04.00.0 /susropts/x86.64.arago.linux/usr/bin/at5/amake
Help	
{} C++	
Qt Quick	
🕕 Build & Run	
🔍 Debugger	
📡 Designer	
Analyzer	
Version Control	
ndroid	
⊕anx QNX	
Devices	
Code Pasting	Version name: Qt %{Qt:Version} in PATH (qt5)
interview Qbs	qmake location: /opt/ti-processor-sdk-linux-am57xx-evm-04.00.00.04/linux-devkit/sysroots/x86_64-arago-linux/usr/bin/qt5/qmake
	Qt version 5.7.1 for Embedded Linux
	Apply <u>C</u> ancel <u>O</u> K

■ Configure Debuggers version

Click the Debuggers option, click Add, click Browse, and select the GDB compiler in the cross-compiler installation directory, such as

/home/david/ti/gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-gnueabihf/bin/arm-linux-gnuea bihf-gdb. Change the Name option, enter AM5728-Debugger, and click Apply to complete the setting.

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

EM-TF-EVK-AM5728 LDG

40

😣 🗉 Options									
Filter	Build & Ru	ın							
Environment	General K	its Qt Versions	Compilers	Debuggers	CMake				
Text Editor	Name		Location						Add
FakeVim	 Auto-dete System 	cted n GDB at /usr/bin/gd	lb /usr/bin/gdb)					Clone
P Help	Manual AM572	28-Debugger	/home/davio	d/ti/gcc-linaro-6	5.2.1-2016.1	1-x86_64_arm-l	inux-gnueal	bihf/bin/arm-lin	Remove
{} c++									
Qt Quick									
🚺 Build & Run									
🔍 Debugger									
📡 Designer									
Analyzer									
Version Control									
i Android	4							Þ	
9 anx QNX									
Devices	Name: A	M5728-Debugger							
Code Pasting	Path: id	/ti/gcc-linaro-6.2.1-2	2016.11-x86_6	4_arm-linux-gn	ueabihf/bin	/arm-linux-gnue	abihf-gdb	Browse	
Nobs	Type:	ЪВ							
	ABIs:	rm-linux-generic-elf	-32bit						
	Version: 2	016.11.0							
							.		
							Apply	<u>C</u> ancel	<u>о</u> к

Configure Kits

Click "Tool->Options->Build & Run->Kits->Add" in the menu bar, change Name to AM5728, Device type to Generic Linux Device, enter linux-oe-g++ in the Qt mkspec option, and click Apply after configuration, click OK

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

EM-TF-EVK-AM5728 LDG

41

😕 🗈 Options						
Filter	Build & Run					
Environment	General Kits Qt Versions Compilers	Debuggers CMake				
Text Editor FakeVim	Name Auto-detected V Manual AM5728 O Desktop (default)					
C++						
Qt Quick	Name: AM5728 File system name:					
🔍 Debugger	Device type: Generic Linux Device					
📡 Designer	Device:					
Analyzer	Sysroot:					
Version Control	Compiler: GCC (x86 64bit in /usr/b	in)				
in Android	Environment: No changes to apply.					
Banx QNX	Debugger: AM5728-Debugger					
Devices	Qt version: Qt 5.7.1 in PATH (qt5)					
Code Pasting	Qt mkspec: linux-oe-g++					
🎤 Qbs	CMake Generator: CodeLite - Ninja	processor-sak-linux-am5/xx-evm-04.00.00.04/linux-devkit/sysroots/x8b_b4-arago-lii				
	CMake Configuration CMAKE_CXX_COMPILER:S	rRING=%{Compiler:Executable}; QT_QMAKE_EXECUTABLE:STRING=%{Qt:qmakeE>				
		Apply <u>Cancel</u> <u>QK</u>				

5.2.3 CREATE DEMO

In this chapter, let's create a simple widget program hello-world to introduce the

development process of Qt interface.

> Create a new project

Similarly, start Qt Creator from the SDK and store the source code in /home/david/ti.

www.emtop-tech.com	https://github.com/EMTOP-TECH	
sales@emtop-tech.com	support@emtop-tech.com	

EM-TF-EVK-AM5728 LDG

42

or Consolo 🔺 🔥
er Console 🗢

Click the File -> New File or Project menu in the top left corner

8 🗊 New File or Project		
Choose a template:		All Templates 👻
Projects Application Library Other Project Non-Qt Project Import Project Files and Classes C++ Modeling Qt GLSL General Java Python Nim	Qt Widgets Application Qt Console Application Qt Quick Application Qt Quick Controls 2 Application Qt Quick Controls Application Qt Canvas 3D Application Qt Labs Controls Application	Creates a Qt application for the desktop. Includes a Qt Designer-based main window. Preselects a desktop Qt for building the application if available. Supported Platforms: Generic Linux Device
		Cancel Choose

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

EM-TF-EVK-AM5728 LDG

43

8	Qt Widgets	5 Application				
-	Location Kits Details	Introduction and Project Location				
~		This wizard generates a Qt Widgets Application project. The application derives by default from QApplication and includes an empty widget.				
	Summary					
		Name: hello-world				
		Create in: //home/david/nt				
		Use as default project location				
		Next > Cancel				

Select Kit: AM5728

80	Qt Widget	s Application									
	Location	Kit Select	ion								
\$	Kits Details	Qt Creator can use the following kits for project hello-world : Select all kits									
		✓ 📮 AM57	28		Details 🔺						
		✓ Debug	/home/david/qt/build-hello-world-AM5728-Debug		Browse						
		✓ Release	/home/david/qt/build-hello-world-AM5728-Release	Browse							
		✓ Profile	/home/david/qt/build-hello-world-AM5728-Profile		Browse						
		🗌 🖵 Deskt	op		Details 🔻						
				< <u>Back</u> <u>N</u> ext >	Cancel						

Select Base Class: Qwidget

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

EM-TF-EVK-AM5728 LDG

44

80	Qt Widgets	Application									
		Class Information									
	Location Kits	Specify basic inf	ormation about the classes for which you want to generate skeleton source code files.								
\$	Details Summary	<u>C</u> lass name:	Widget								
		<u>B</u> ase class:	QWidget 🔹								
		<u>H</u> eader file:	widget.h								
		Source file:	widget.cpp								
		<u>G</u> enerate form:									
		Form file:	widget.ui								
			< Back Next > Cancel								

Click Finish to enter the new project

<u>F</u> ile <u>E</u> dit <u>B</u> uild <u>D</u> ebug <u>A</u> nalyze <u>T</u> ools <u>W</u> ir	ndow <u>H</u> elp
Projects 🗢 🗧 🕀 🖃	< > 🗈 🎽 widget.ui 🔶 🗧
hello-world	This file can only be edited in Design mode.
Welcome 🕨 📊 Headers	<pre>1 <?xml version="1.0" encoding="UTF-8"?></pre>
Sources	2 ▼ <ui version="4.0"></ui>
📰 🔍 🔻 📝 Forms	<pre>3 <class>Widget</class></pre>
📝 widget.ui	4 ▼ <widget class="QWidget" name="Widget"></widget>
Edit	<property name="geometry"></property>
*	6 ▼ <rect></rect>
	7 <x>0</x>
Design	8 <y>0</y>
	9 <width>400</width>
₩	10 <neight>300</neight>
Debug	11
Jonag	12
رکل 🖌	13 <property name="windowritte"> 14 <pre>string>Widget</pre>/string></property>
	15 c/property>
Projects	16
•	17 <layoutdefault margin="11" spacing="6"></layoutdefault>
	18 <resources></resources>
Help	19 <connections></connections>
	20
	21

Design UI interface

Double-click widget.ui to enter the UI designer

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

EM-TF-EVK-AM5728 LDG

45

<u>F</u> ile	<u>E</u> dit	<u>B</u> ui	ld <u>D</u> ebug	<u>A</u> nalyze	<u>T</u> ools	<u>W</u> indow	N	<u>H</u> elp	•														
		n l	🌶 widget.	ui		\$ >	×	1 20	¥.	5	123,			М	æ		10		L.				
	=	Filter	-				•											: :					
		•		Layouts																			
Welco	ome		Vertical La	yout																			
			Horizontal	Layout																			
Ed	it		Grid Layou	t																			
1	۶.		Form Layo	ut																			
Desi	ign	•		Spacers																			
	- -		Horizontal	Spacer			•															1	•
الا Deb	t .	X	Vertical Sp	acer																			
Den	ug	•		Buttons																			
۶	ن ن	ок	Push Butto	n																			
Proje	ects	2	Tool Buttor	n																			
6		۲	Radio Butt	on																			
Hel	lp	\checkmark	Check Box							• •	• •			• •			• •	• •	• •	 	 		
		Ð	Command	Link Button	1																		
		√x	Dialog But	on Box																			

In the Property window in the lower right corner, modify the windowTitle field and enter

hello-world.

8 🗖 🗊	widge	et.ui - hello-world	- Qt Cr	reator	r									
<u>File</u> Edit	<u>B</u> uild	d <u>D</u> ebug <u>A</u> nalyze	Tools	Winde	ow	<u>H</u> elp								
	🖬 📝	widget.ui*				😼 🖳 🖏 🛛	s, III ≡ I	M Z H H H	ō 🖳					
	Filter				•			• • • • • • • • • •		· · · · •	O	oject Class		
	-	Layouts										- 🍓 Widget 💹 QWi	dget	
Welcome	🗏 V	ertical Layout												
	00 н	lorizontal Layout												
Edit	888 G	irid Lavout			1									
*	88 F	orm Lavout			1									
Design	- aa	Spacers		-										
Design	Real H	lorizontal Spacer			•					•				
- Ŵ	S v	ertical Spacer												
Debug	-	Buttons		-	1									
بو	ок р	ush Button			1									
Projects	<u></u> л	ool Button												
		adio Button												-
V		back Day			1						Fi	lter	+ -	- /~
Help		neck Box			•			-		-	W	idget : QWidget		
		ommand Link Button							_	_	ro	perty	Value	-
	X D	ialog Button Box		_) 👂 👂	xì //-		Filter		ŀ	mouse I racking	NoFocus	_
	-	Item Views (Model-I	Based)	_	Na	me	Used	Text	Shortcut	Ch	e		DefaultCont	
	L	ist View			-							acceptDrops	Deradiceone	
hello-world	T B	ree View									-	windowTitle	hello-world	•
Γ.	Шт	able View										translatable	<	_
Debug	Ш с	olumn View										disambiguation		
Denug	-	Item Widgets (Item-	Based)									comment		_
	Li	ist Widget										windowIcon	Ø }	_
	ST 8	ree Widget										windowOpacity	1.000000	_
	П	able Widget			4					•	-	toolTipDuration	-1	_
	*	Containers			A	Action Editor	Signals &	Slots Editor		,		statusTin	_	
		P. Type to locate (Ct	rl+K)	1	lss	ues 2 Searc	h Results	3 Application (Output 4 Comp	oile Outp	ut	5 Debugger Conso	le 💠	P

Drag a Push Button to the UI in the Buttons window on the left, double-click and change the name to "close".

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

EMBEDDED SOLUTIONS

EM-TF-EVK-AM5728 LDG

46

80	wid	get.ui - hello-world - Qt Cr	eato	r								
File Edit	Bu	ild <u>D</u> ebug <u>A</u> nalyze <u>T</u> ools	Wind	ow	<u>H</u> elp							
		🃝 widget.ui*			🖷 🕰 🔖 🖪	, III ≡ I	4 Z II III					
	Filte	Layouts	A				•			Object Class	Widget	^
weicome		Vertical Layout								puon 🔤 Q	PushButton	
		Horizontal Layout										
Edit		Grid Layout										
	88	Form Layout	_									
Design	TR221	Horizontal Spacer		•								
Ŵ	X	Vertical Spacer					close					
Debug	-	Buttons										
للحر	ок	Push Button										
Projects	2	Tool Button										
•	۲	Radio Button								Filter	+ -	1.
Help		Check Box		-			••••		•••••	Widget : QWidget		
	$\mathbf{\Theta}$	Command Link Button								roperty	Value	-
	YX	Dialog Button Box	_		🌗 🚺 🛛	d /~-		Filter		acceptDrops	DefaultConte	
		Item Views (Model-Based)	_	Nan	me	Used	Text	Shortcut	Ch	windowTitle	hello-world	
hello world		Tree View								translatable	✓	
		Table View								disambiguation	1	
• الح	lm	Column View								windowIcon	0,	
Debug	•	Item Widgets (Item-Based)								windowOpacity	1.000000	
		List Widget								 toolTip 	1	
		Tree Widget								 statusTip 	-1	
*		Table Widget		4					Þ	whatsThis		
>	-	Containers		, Ac	ction Editor	Signals &	Slots Editor			accessibleName		Þ
		P ₊ Type to locate (Ctrl+K)	1	Issu	ues 2 Searc	h Results	3 Application	Output 4 Com	pile Outp	ut 5 Debugger Con	sole 🗢 🧳	• //

Select the close button, right-click and select Go to slot to enter code editing.

😣 🗈 Go to slot		-
Select signal		
clicked()	QAbstractButton	
clicked(bool)	QAbstractButton	
pressed()	QAbstractButton	_
released()	QAbstractButton	
toggled(bool)	QAbstractButton	
destroyed()	QObject	-
	<u>C</u> ancel	<u>о</u> к

Edit code

Edit the response code of the on_pushButton_clicked function. Enter close() here. The function is to close the window.

> Compile

Click the hammer-shaped shortcut button in the lower left corner to compile the pro

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

EM-TF-EVK-AM5728 LDG

47

ject and produce the executable file in directory **<u>\$HOME/qt/build-hello-world-AM57</u>**

28-Debug.

Check the ELF with file command:

\$ file hello-world

hello-world: ELF 32-bit LSB executable, ARM, EABI5 version 1 (GNU/Linux), dynamic ally linked (uses shared libs), for GNU/Linux 2.6.32, BuildID[sha1]=cd9018247cd88be3 3eb2f59fb56fe7af7fee37ea, not stripped



5.2.4 RUNNING ON ARM BOARD

Run under weston desktop

Copy the generated executable file hello-world to the AM5728 board and run ./hello-world.

The display will show the following window:

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

EMDED SOLUTIONS	EM-TF-EVK-AM5728 LDG
	48
	Mon Aug 13, 11:21 AM
hello-wor	ld X
close	R

Click the close button with the mouse (or click on the touch screen) to close the window.

Run without weston desktop

Qt programs can also be run without weston desktop. Close weston first.

- root@arm:~# /etc/init.d/weston stop
- root@arm:~# ./hello-world -platform eglfs

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

EM-TF-EVK-AM5728 LDG

49



Or

•

root@arm:~# ./hello-world -platform linuxfb

close	R	t	

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

50

5.3 VIDEO CAPTURE DEMO

The video collection function includes preview, photo taking, and video recording. Run the Camera application in the UI interface, the icon is shown below.



You can test video preview and capturing.

- Capture: photo shutter, used to take a picture.
- Switch: Swap the main display window and secondary display window on the screen.
- PIP: Close the secondary display window.
- Exit: Terminate the application.

The photos taken can be browsed in Gallery, the icon is



The photo manager interface is as below:

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

EM-TF-EVK-AM5728 LDG



The Delete, Quit, Previous, and Next buttons are to delete the current picture, exit the application, previous picture, and next picture respectively.

Video recording is the video recording function, which is implemented by gstreamer.

Command line reference:

EMTOP

root@arm:~# gst-launch-1.0 -e v4l2src device=/dev/video2 num-buffers=1000 iomode=5 ! 'video/x-raw, \ format=(string)NV12, width=(int)1920, height=(int)1080, framerate=(fraction)30/1' ! ducatimpeg4enc bitrate=12000 ! \ queue !mpeg4vid eoparse !qtmux !filesink location=out.mp4

To play the file recorded above, use the following command (assuming the full path of the generated mp4 file is: <u>/media/out.mp4</u>):

root@arm:~# gst-launch-1.0 playbin uri=file:///media/out.mp4 video-sink=kmssin
 k

5.4 DUAL DISPLAY DEMO

It's a demo program of dual displays that can demonstrate OpenGL, dual cameras, and

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

EM-TF-EVK-AM5728 LDG

EMTOP EMBEDDED SOLUTIONS

52

dual displays working at the same time. This application constructs a rotating cube through OpenGL, obtains dual camera data through the V4L2 video acquisition interface, and then displays it on the LCD and HDMI display screens respectively. It is used to demonstrate the powerful video processing and display capabilities of the platform. The Weston should be terminated before testing.

root@arm:~# /etc/init.d/weston stop

The specific test steps:

- Dual screens display a rotating cube and a camera respectively
 - root@arm:~# kmscube-camera -a
- Dual screens display rotating cube and #0 camera respectively
 - root@arm:~# kmscube-camera -a -i 0
- Dual screens showing rotating cube and #1 camera respectively
 - root@arm:~# kmscube-camera -a -i 1
- Single screen display rotating cube
 - root@arm:~# kmscube-camera -c 32
 - root@arm:~# kmscube-camera -c 36
- Dual screen, one screen displays the rotating cube, and other alternately displays two cameras
 - root@arm:~# kmscube-camera -a -i 2

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

6. ROOTFS BUILD BASED ON YOCTO

Yocto can build a complete embedded system image. This section focuses on building the rootfs system. For more information about Yocto, please visit the official website: https://www.yoctoproject.org/docs/2.5/mega-manual/mega-manual.html

Setting up the Yocot build environment requires a PC with suitable hardware performance, sufficient memory and hardware space, and sufficient network bandwidth. It is recommended that at least 200G free space and 8GB memory be used.

6.1 INSTALL REQUIRED TOOL SOFTWARE

\$ sudo apt-get install git build-essential python diffstattexinfo gawk chrpath d os2unix wget unzip socatdoxygen libc6:i386 libncurses5:i386 libstdc++6:i386 li bz1:i386

6.2 CONFIGURE BASH

\$ sudo dpkg-reconfigure dash

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

The system shell is the default command interpreter for shell scripts.
Using dash as the system shell will improve the system's overall performance. It does not alter the shell presented to interactive users.
Use dash as the default system shell (/bin/sh)?
<yes></yes>

Select "No".

6.3 INSTALL COMPILER

If the cross-compilation tool chain has been installed before, please skip the step.

- \$ wget https://releases.linaro.org/components/toolchain/binaries/6.2-2016.11/armlinux-gnueabihf/gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-gnueabihf.tar.xz
- \$ tar -Jxvf gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-gnueabihf.tar.xz -C \$HOM
 E

6.4 OBTAIN oe-layertool-setup.sh

- \$ git clone git://arago-project.org/git/projects/oe-layersetup.git tisdk
- \$ cd tisdk
- \$./oe-layertool-setup.sh -f configs/processor-sdk/processor-sdk-04.00.00.04-con fig.txt

6.5 BITBAKE BUILD

- \$ cd ..
- \$ cd build

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

.

55

- \$. conf/setenv
- \$ export PATH=\$HOME/gcc-linaro-6.2.1-2016.11-x86_64_arm-linux-gnueabihf/bin:
 \$PATH
- \$ MACHINE=am57xx-evm bitbake arago-core-tisdk-image

The initial build takes a long time, ranging from a few hours to dozens of hours, and the

specific time varies depending on PC performance and network bandwidth.

After the build is complete, the target file system is

tisdk/build/arago-tmp-external-linaro-toolchain/deploy/images/am57xx-evm/tisdk-r

ootfs-image-am57xx-evm.tar.xz.

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com

7. APPENDIX

www.emtop-tech.com	https://github.com/EMTOP-TECH
sales@emtop-tech.com	support@emtop-tech.com