BQ

BQ e-readers developers program

por
BQ

publicado el
10 Mayo, 2018 (10 Mayo, 2018)

The target of the BQ e-readers developers program is to allow to any user with the required knowledge the possibility to modify our ereaders firmware, create his own firmware versions and share them with the community. In order to do that we provide to you both kernel and application source code.

Our e-readers **Developers edition firmware** is the gate to our device and turns your e-reader in an open device for your free development. This firmware grants full access to the linux system that runs on the device and allow you to modify the firmware of the device and to create new applications. Even though you install this firmware in your device the default warranty service kept.

This Developers Edition firmware is compatible with the following e-reader models:

- bq Cervantes Touch / Fnac Touch (2012-2013)
- bq Cervantes Touch Light / Fnac Touch Plus (2012-2013)
- bq Cervantes 2013 / Fnac Touch Light (2013)
- BQ Cervantes 3 / Fnac Touch Light 2 (2016)
- BQ Cervantes 4 (2017)

Additional features of developers firmware

Telnet access to device

From "*Settings*" -> "*Development*" options you will be able to enable telnet to connect over wifi and usb networking.

Default **root** password is empty, so we strongly recommend you to setup a new password (if you plan to use telnet over Wi-Fi we also recommend you also to install **ssh).**

USB Networking

You can configure the USB port as a network interface from "*Settings" -> "Development options*". If you enable this option the device will set up a network on **usb0** interface with 192.168.4.1 as IP address. Set your computer **usb0** interface with another ip address in that range and you would be able to telnet it through USB interface

Boot of an external system using MicroSD

The developers firmware allows to boot the system included in a inserted microSD instead of the one in the internal memory. This is the recommended method to use for new development, since you can keep the original internal firmware untouched for normal ereader usage and try your stuff without any risk of bricking your device. To boot from SD follow these instructions:

1. Format the first partition of your microSD as ext3.

2. Copy a working root filesystem on that partition of the, you can copy the entire rootfs partition from the device internal memory (*mmcblk0p1* for Fnac Touch/Cervantes Touch/Fnac Touch Plus/Cervantes Touch Light y *mmcblk0p05* all the later models)

3. Copy a kernel image with path **/boot/uImage** (base kernel is included in the base **rootfs** included in the link of previous step).

4. Insert the microSD on the device.

5. Power on while keeping pressed **Power** Key + **Home** Key until boot process is over (movingsquares dissapear from screen). If base **rootfs** proposed is used it is possible to confirm that system in microSD have been used thanks to an additional SD icon on booting image.

Warning: It is really important to keep both keys pressed on step 5, if only Home Key is kept pressed during the initial boot an upgrade will be attempted instead of SD boot.

Useful commands

In order to to run your own applications without collision with main reading application you ca use the following commands:

- **stopapp.sh.** This script kills the reading application
- **restart.sh.** This script restarts the reading application

Limited features

Developers edition firmware has a limited functionality because of security and licensing reasons:

- All purchased ebooks from the associated shop are removed from the device per editorial requirements, however they are still available in any other of your linked devices or after restoring original stock firmware. Every file stored in the internal memory or microSD will be kept.
- Dictionary features are disabled.
- Automatic version update warning disabled

Installing Developers Edition firmware

Requirements

You just need to have an account for the shop associated to your device, *nubico* for the BQ devices. You can create your user using the device or at **www.nubico.es**.

Installation process

You can install the developers and there is no need you contact us to do so, however we let you know that the update to hackers will be registered in our database in order to provide a better support if any maintenance issue is necessary later. Anyway, **device original warranty is kept regardless you install developers firmware.**

These are the steps to follow for the upgrade:

- Download the firmware developers installation package:

  Cervantes Touch/Fnac Touch and Cervantes Touch Light/Fnac Touch Plus. (Version 5.2.0)

  Cervantes 2013/Fnac Touch Light (Version 5.2.0)

  BQ Cervantes 3 / Fnac Touch Light 2 (Version 6.0.3)

  BQ Cervantes 4 (Version 7.0.2)

- Copy to a microSD card **(bqupdate** folder directly on microSD card root).

- Create a file with name **I_WANT_TO_BE_A_HACKER** and write on it the Serial Number of your device with the following format: SN-SERIALNUM (e.g: for S/N E606722CB03884 you must write SN-E606722CB03884 on the file). No extension must be set to the file, just plain text.

- Copy **I_WANT_TO_BE_A_HACKER** file to the root of the microSD card.

- Insert the microSD card in the device and a dialog must appear to proceed with device registration on the developers program. You will see a confirmation of the successful authorization in the device.

- After successful activation, power off the device.

- Press power on while keeping **Home** key pressed to perform installation.

Note: Do not pay attention to first warning if telling loss of warranty because of installation, it's a dialog error to be solved. **Device warranty is kept.**

Restoring original firmware

If you want to restore the default firmware (full functionality and access restriction) you just need to install the restoration firmware package:

- Restore upgrade package for Cervantes Touch/Fnac Touch and Cervantes Touch Light/Fnac Touch Plus (Version 4.2.0)

- Restore upgrade package for Cervantes 2013/Fnac Touch Light (Version 5.0.2)

- Cervantes 3/Fnac Touch Light 2 (Version 6.0.3)

- Cervantes 4 (Version 7.0.2)

The procedure to do so it's the usual one, copy the bqupdate on the root of your microSD, insert it in your device and keep the **Home** key pressed while switching the device on. Keep only **Home** Key pressed and make sure you release **Power** Key as soon as green led switches on, otherwise microSD boot would be tried instead of upgrade.

Notice that restoring process takes much longer than usual firmware update, around 40 minutes. It's a very critical operation and it's really important to wait until finishing without shutting down, green led at the bottom of the device will be blinking during the process. Make sure your device is connected to a charger to ensure battery does not get empty during the process.

Warning: The whole internal memory is erased when restoring standard firmware, all the files stored will be lost and the device will be reset to factory settings. Therefore, you may need to do a backup of your personal files before restoration.

Warning: You need to keep the original rescue partition (mmcblk0p2) untouched in order to use the standard restore procedure. If you have removed rescuefs partition you would need to do a manual restoration procedure:

1. Boot the device using a bootable system in your microSD (see how to do it below, at microSD boot instructions).
2. Extract manually the **restore** binary and **original.img** from the restore image file using the keyring.
3. Execute **./restore original.img**.

Re-installing developers firmware

Once you have installed the firmware for developers you can install easily the same firmware or later versions. Upgrade packages are available to be installed following the procedure described in previous section.

- Developers upgrade package for Cervantes Touch/Fnac Touch and Cervantes Touch Light/Fnac Touch Plus and Cervantes 2013/Fnac Touch Light:
  Version 4.2.0

  Version 5.0.0

  Version 5.0.2

  Version 5.2.0

  Versión 6.0.3

  Versión 7.0.2

Development Info

e-reader application

You can find all the code of our application in the following repo:

https://github.com/bq/cervantes

You can find all the info about how to build and emulate the application in the README included. To do so you need our **pre-generated SDK**.

Base system

The base system is a Debian Wheezy distribution for ARM, so you can easily install new software using **apt-get.**

We strongly recommend you to keep the internal firmware "as is" and do all your changes and modifications on a bootable external SD card having an copy of the original developers firmware. If you still plan to install software with **apt** on the internal firmware you may need to expand the **root** partition of mount **/var** as a separate partition with enough space for **apt** data.

System connectivity is provided by **connman**. If you don't want to use the stock reader application to connect to a wifi network you can use the **connman** command line tools or the **dbus-interface.**

Kernel

- **Cervantes Touch/Fnac Touch and Cervantes Touch Light/Fnac Touch Plus**

You can find the source code of our kernel in **https://github.com/bq/linux-e60622**.

The correct code to produce a working kernel for the devices is contained in the **topic/fixes** branch. For simple uses, like compiling a kernel, this branch can be checked out normally, but to access all the features of this topic branch the use of topgit is recommended. The full tree of branches in use looks like:

topic/fixes – Fixes on top of everything

- topic/backports – Backports of mainline features (> 2.6.35)

- e60622_2.6.35 – eReader base branch

- imx_2.6.35 – Freescale base-branch for i.MX5 SOCs

A working .config file is contained in the base directory of the kernel sources. Therefore the standard cross-compilation-commands should be sufficient to produce a kernel image:

```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi-uImage
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi-modules
```

The kernel needs to be in **uImage** format for **u-boot** to recognize it.

To let **u-boot** start this new kernel, copy it to **/boot** on the first partition of your boot medium (for example the external card), overwriting the **uImage** file present in this location.

- **Cervantes 2013/Fnac Touch Light**

You can find the source code of our kernel in **https://github.com/bq/linux-e60q22**.

The correct code to produce a working kernel for the devices is contained in the **topic/features** branch. For simple uses, like compiling a kernel, this branch can be checked out normally, but to access all the features of this topic branch the use of topgit is recommended. The full tree of branches in use looks like:

topic/fixes – Fixes on top of everything

- e60q22 – eReader base branch

- master – Freescale base-branch for i.MX6 SOCs

A working **.config** file is contained in the base directory of the kernel sources. Therefore the standard cross-compilation-commands should be sufficient to produce a kernel image:

```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi-uImage
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi-modules
```

The kernel needs to be in **uImage** format for **u-boot** to recognize it.

To let **u-boot** start this new kernel, copy it to **/boot/uImage.e60q22** on the first partition of your boot medium (for example the external card), overwriting the uImage file present in this location. The kernel image needs to be renamed from uImage to **uImage.e60q22** to be recognized by the bootloader.

- **Cervantes 3/Fnac Touch Light 2**

You can find the source code of our kernel in **https://github.com/bq/linux-e60qh2**.

The correct code to produce a working kernel for the devices is contained in the **devel/features** branch. For simple uses, like compiling a kernel, this branch can be checked out normally, but to access all the features of this topic branch the use of topgit is recommended. The full tree of branches in use looks like:

devel/features – Feature-development on top of everything

- e60qh2 – eReader base branch

- master – Freescale base-branch for i.MX6 SOCs

A working **.config** file is contained in the base directory of the kernel sources. Therefore the standard cross-compilation-commands should be sufficient to produce a kernel image:

```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- uImage
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- modules
```

The kernel needs to be in **uImage** format for **u-boot** to recognize it.

To let **u-boot** start this new kernel, copy it to **/boot/uImage.e60qh2** on the first partition of your boot medium (for example the external card), overwriting the uImage file present in this location. The kernel image needs to be renamed from uImage to **uImage.e60qh2** to be recognized by the bootloader.

- **Cervantes 4**

You can find the source code of our kernel in **https://github.com/bq/linux-e60qp2**.

El código correcto para crear un kernel funcional para el dispositivo está contenido en la rama topic/features. Para un uso simple, como compilar el kernel, puedes hacer checkout de la rama de manera normal, pero para acceder a todas las funcionalidades de esta rama es recomendable el uso de topgit. El árbol completo de las ramas es el siguiente:

devel/features – Fixes sobre todo lo demás

- e60qp2 – Rama base de la plataforma

- master – Rama vase Freescale para i.MX6 SOCs

Puedes encontrar un fichero .config funcional en el directorio base de las fuentes del kernel. Por lo tanto, los comandos de compilación cruzada estándar deberían ser suficientes para construir una nueva imagen de kernel:

```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi-uImage
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi-modules
```

El kernel debe tener formato uImage para que el u-boot lo pueda reconocer.

Para que el u-boot arranque el nuevo kernel, cópialo a la ruta /boot/uImage.e60qp2 de la primera partición de tu sistema (por ejemplo en la uSD),sobrescribiendo el fichero uImage.e60qp2 que se encuentre allí.

Memory layout on your device and precautions

First 10MB of the internal memory are reserved for critical low level device data directly related to specific hardware of your device. Data such as following is stored there:

> uboot: The binary data required to boot the system, **corruption of this info makes impossible to boot the device and needs a complex repair**.
>
> Serial number.
>
> Hardware info.
>
> Individual data associated to screen properties: **corruption of this info may lead to partial or total malfunctioning of eink screen**. Sometimes impossible to recover 100%.

Environment

Installing a Cross-Compiler

As the hardware-architecture of the device is ARM-based, for both kernel and application development a cross-compiler is needed that can create arm-executables on x86 hardware.

On **Ubuntu-based** systems a suitable cross-compiler can be found in the universe part of the Ubuntu-archives, so a simple sudo **apt-get install g++-arm-linux-gnueabi gcc-arm-linux-gnueabi** should suffice.

On **Debian** systems the needed cross-compiler is currently not part of the base distribution, but part of the Emdebian subproject. So, to install the necessary packages just follow the instructions on the official wiki page: **http://wiki.debian.org/EmdebianToolchain#Get_the_binaries.**

On all distributions you can also use a generic toolchain from Linaro: **https://launchpad.net/linaro-toolchain-binaries/trunk/2012.04/+download/gcc-linaro-arm-linux-gnueabi-2012.04-20120426_linux.tar.bz2**

Dependencies

To compile **Qt** applications targeting the device you need to set up this env variables so that **qmake** uses the cross-compiler and link against the arm **Qt** libraries:

```
export ROOTFS=/path/to/your/rootfs-sdk/

export QTDIR=$ROOTFS/usr

export QMAKESPEC=$QTDIR/mkspecs/linux-arm-gnueabi-g++/
```

Before running you Qt application you need to set up some env variables for Qt and *tsllib* (the library handling touch events):

```
export QWS_DISPLAY=einkfb
export QWS_MOUSE_PROTO=tslib:/dev/input/event1
export QT_QWS_FONTDIR=/usr/lib/fonts
export QWS_KEYBOARD=eb600keypad:/dev/input/event0
export TSLIB_CONSOLEDEVICE=none
export TSLIB_FBDEVICE=/dev/fb0
export TSLIB_TSDEVICE=/dev/input/event1
export TSLIB_CALIBFILE=/usr/etc/pointercal
export TSLIB_CONFFILE=/etc/ts.conf
export TSLIB_PLUGINDIR=/usr/lib/arm-linux-gnueabi/ts0/
export POINTERCAL_FILE=/usr/etc/pointercal
```

Other applications

For writing your own applications you have different options:

- Use **Qt** with currently provided plugins. This is the easy and recommended way for starters.
- Install **Xorg** and do custom **X** development. You would need to hack the *driver* support for *framebuffer*
- Other options (direct **fb, gtk** with **fb, wayland/weston**...)

To link your applications you may need the target libraries and headers. You can manually generate this using debian multistrap tool or copy it from the device root filesystem. However if you don't want to mess with multistrap stuff or copying it, the easiest way is to download the **pre-generated SDK** . This SDK also includes some utility libraries to make application development easier.

Creating custom firmware

If you are a pro-developer and you create nice applications you can package them as a firmware image to share it with other developers and users.

To create this image:

1. Package the **root** filesystem on a tar.gz.
2. Use **gpg** to encrypt and sign this file using the hackers-keyring-host.tar.gz keyring.

3. Place encrypted image and the signature inside a folder called **bqupdate** with names **update.img** and **update.asc** respectively.
4. Zip the folder.
5. Share the folder with the community

BQ

Somos un equipo formado por ingenieros, periodistas y apasionados de la tecnología que trabajan día a día en BQ. Con nuestros artículos buscamos acercar la tecnología de una forma didáctica, facilitar el uso de nuestros dispositivos y compartir con vosotros las novedades de la marca.

Ayudamos a las personas a entender la tecnología, les animamos a utilizarla e inspiramos a desarrollarla.

ACTUALIDAD

APLICACIONES

FOTOGRAFÍA

TRUCOS BQ

CULTURA TECNOLÓGICA

SUBIR ↑

Newsletter

Suscríbete a nuestra newsletter para recibir las últimas novedades de BQ.

Escribe aquí tu correo electrónico

SUSCRIBIRME