Linux STM32F4

BSP (Board Support Package) Guide for the STMicroelectronics STM3240G-EVAL Board

Release 1.12.0

Table of Contents

1. Overview

This document is a Linux STM32F4 BSP (Board Support Package) Guide for the STMicroelectronics STM3240G-EVAL board, Release 1.12.0.

The BSP provides a software development environment for evaluation and development of Linux on the Cortex-M4 processor core of the STMicroelectronics STM32F4 microcontroller using the STMicroelectronics STM3240G-EVAL board in harness with the Emcraft STM-MEM add-on board as a hardware platform.

This BSP is provided as part of the STMicroelectronics Linux STM32F4 Evaluation Kit. The evaluation kit provides a hardware platform and Linux software development environment for the STMicroelectronics STM32F4 microcontroller.

2. Product Contents

This product includes the following components.

2.1. Shippable Hardware Items

The following hardware items are shipped to customers of this product:

1. STM-MEM add-on board;
2. STM3240G-EVAL board - NOT INCLUDED. Please purchase the STM3240G-EVAL board from STMicroelectronics or its distributors.

2.2. Downloadable Hardware Materials

The following hardware materials are available for download from Emcraft's web site to customers of this product:

1. `STM-MEM-2_0-schem.pdf` - STM-MEM schematics in PDF format;
2. `STM-MEM-2_0-bom.xls` - STM-MEM Bill-Of-Materials (BOM) in Excel format.

2.3. Downloadable Software Materials

The following software materials are available for download from Emcraft's web site to customers of this product:

1. `u-boot.bin` - prebuilt U-Boot file in the format suitable for installation into embedded Flash of Cortex-M4 on the STM3240G-EVAL board;
2. `networking.uImage` - prebuilt Linux image ready to be loaded to the STM3240G-EVAL board;
3. `linux-STM32F4-EVAL-1.12.0.tar.bz2` - Linux STM32F4 software development environment, including:
   a) U-Boot firmware;
   b) Linux kernel;
   c) `busybox` and other target components;
   d) Linux-hosted cross-development environment;
   e) Framework for developing multiple projects (embedded applications) from a single installation, including sample projects allowing to kick-start software development for Linux STM32F4.

2.4. Downloadable Documentation Materials

The following documentation materials are available for download from Emcraft's web site to customers of this product:

1. `linux-cortexm-um-1.12.0.pdf` - Linux Cortex-M User's Manual;
2. `linux-STM3240G-EVAL-bspg-1.12.0.pdf` - Linux STM32F4 BSP (Board Support Package) Guide for the STMicroelectronics STM3240G-EVAL Board (this document).

3. Software Functionality

3.1. Supported Features

The following list summarizes the features and capabilities of Linux STM32F4, Release 1.12.0:

- U-Boot firmware:
  - U-Boot v2010.03;
  - Target initialization from power-on / reset;
  - Runs from the internal eNVM and internal SRAM (no external memory required for standalone operation);
  - Serial console;
  - Ethernet driver for loading images to the target;
  - Serial driver for loading images to the target;
  - Device driver for built-in Flash (eNVM) and self-upgrade capability;
  - Device driver for storing environment and Linux images in external Flash;
  - Autoboot feature, allowing boot of OS images from Flash or other storage with no operator intervention;
  - Persistent environment in Flash for customization of target operation;
  - Sophisticated command interface for maintenance and development of the target.
- Linux:
  - uClinux kernel v2.6.33;
  - Boot from compressed and uncompressed images;
  - Ability to run critical kernel code from integrated Flash of STM32F4;
  - Serial device driver and Linux console;
  - Ethernet device driver and networking (`ping`, NFS, Telnet, FTP, `ntpd`, etc.);
  - `busybox` v1.17;
  - POSIX pthreads;
  - Process-to-kernel and process-to-process protection using the Memory Protection Unit (MPU) of the STM32F4 core;
  - Hardened exception handling; an exception triggered by a process affects only the offending process;
  - Loadable kernel modules;
  - Support for the hardware FPU;
  - Secure shell (`ssh`) daemon;
  - Web server;
  - MTD-based Flash partitioning and persistent JFFS2 Flash file system for external Flash;

  - Device driver for the DMA interface;
  - SD Card device driver;
  - I^{2}C device driver;
  - RTC device driver;
  - SPI controller master-mode device driver;
  - GPIO device driver.
- Development tools:
  - ARMv7-optimized GNU toolchain from CodeSourcery (`2010q1`) is used for development of U-Boot, Linux and user-space applications (toolchain must be downloaded separately from the CodeSourcery web site);
  - Cross GDB for debugging user-space applications;
  - `mkimage` tool used by the Linux kernel build process to create a Linux image bootable by U-Boot.
- Development environment:
  - Linux-hosted cross-development environment;
  - Development of multiple projects (embedded applications) from a single installation;
  - `hello` sample project ("Hello, world!" single-process configuration);
  - `networking` sample project (basic shell, networking and Flash management tools demonstration);
  - `developer` sample project (template project that can be used to jump-start development of custom user-space applications and loadable kernel modules).

3.2. New and Changed Features

This section lists new and changed features of this release:

- None.

3.3. Known Problems & Limitations

This section lists known problems and limitations of this release:

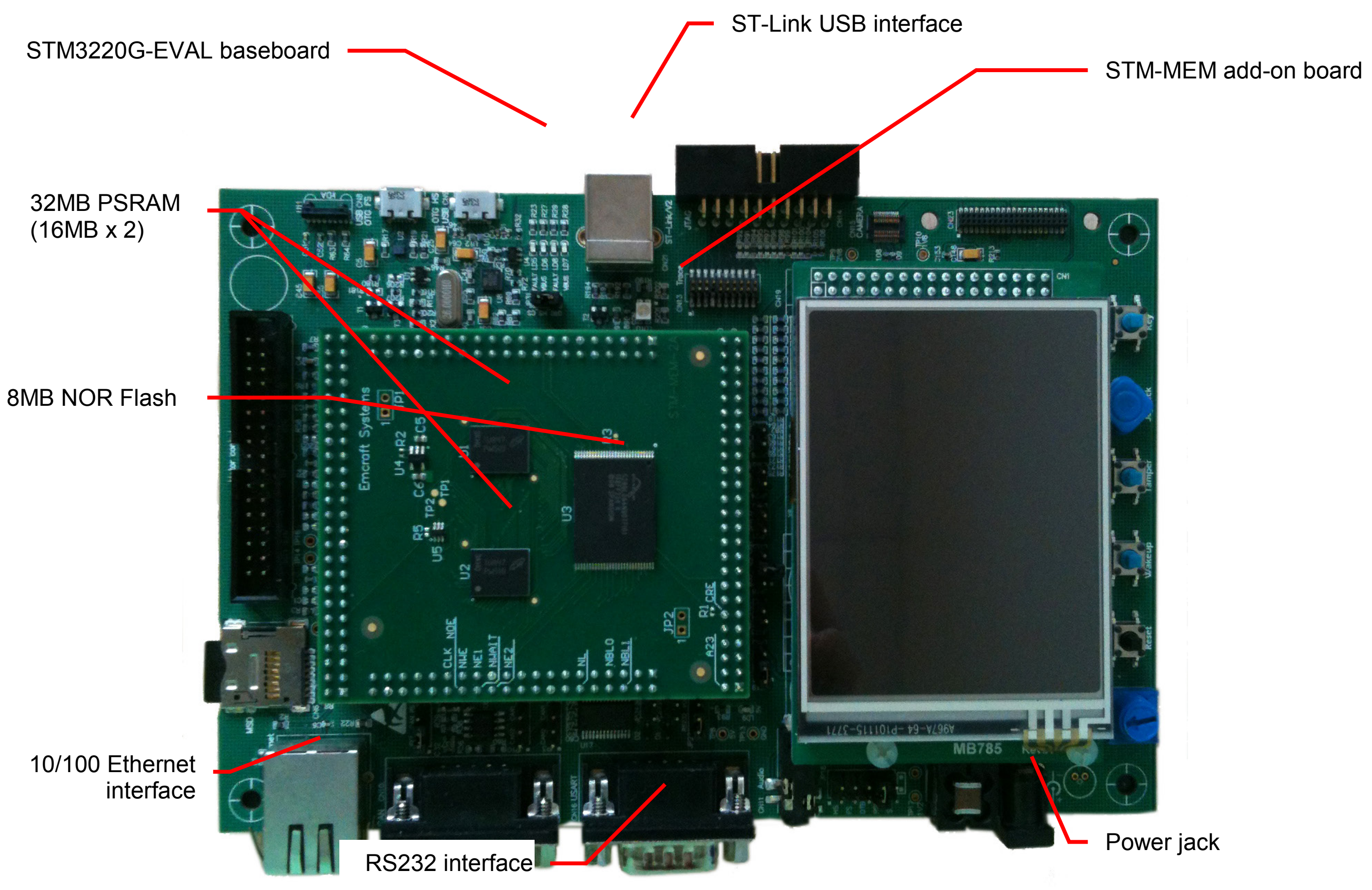
1. `CONFIG_KERNEL_IN_ENVM` requires disabling `CONFIG_ARM_UNWIND` and `CONFIG_EARLY_PRINTK`.
   ID: RT 74683.
   Workaround: When enabling `CONFIG_KERNEL_IN_ENVM` in the kernel, disable `CONFIG_ARM_UNWIND` and `CONFIG_EARLY_PRINTK`.
2. Debugging of multi-threaded applications using GDB is not supported.
   ID: RT 77243.
   Workaround: None. This issue will be resolved in future releases of the product.

4. Hardware Setup

This section explains how to set up the STMicroelectronics STM3240G-EVAL board in harness with the Emcraft STM-MEM add-on board.

4.1. Hardware Interfaces

The STM3240G-EVAL board in harness with the Emcraft STM-MEM add-on board provides the following components and interfaces:



4.2. Jumpers

The following jumpers must be configured on the STM3240G-EVAL board to allow its operation in harness with the STM-MEM add-on board:

Jumper	Configuration	Notes
JP1	2-3 closed	
JP2	2-3 closed	
JP3	open	
JP5	1-2 closed	
JP10	open	
JP18	PSU contact closed, other contacts open	

4.3. Installation of STM-MEM Add-On onto STM3240G-EVAL

To install the Emcraft STM-MEM add-on board onto the STMicroelectronics STM3240G-EVAL board, follow the step-wise procedure below:

1. Disconnect all power sources from the STM3240G-EVAL board;
2. Remove R47, R52, and R141 resistors on the STM3240G-EVAL board. Note that when the R141 is removed, the LED3 on the STM3240G-EVAL board is disconnected from the PI9 port of the STM32F407 device;
3. Check that SB14 and SB15 solder bridges on the STM3240G-EVAL board are open;
4. Connect the STM-MEM add-on board connectors to the corresponding connectors on the STM3240G-EVAL board (CN1 to CN1, CN2 to CN2 etc.).

4.4. Board Connections

To provide the power to the STM3240G-EVAL board, connect a power cable to the power jack of the board (CN18).

To provide a serial interface to a PC / notebook plug a null-modem RS-232 cable to the CN16 connector on the board. On the PC side, the serial link provides a serial console device to the STM3240G-EVAL. The software installed on the board is configured for a 115.2 K terminal. On the Linux host, the serial console is available using a `/dev/ttyS`*n* device.

To provide network connectivity to the board, connect it into your LAN by plugging a standard Ethernet cable into the 10/100 Ethernet connector. The board is pre-configured with an IP address of 192.168.0.2.

Apply the power to the STM3240G-EVAL board using the on-board power switch.

5. STM3240G-EVAL Board Linux Software Set-up

5.1. U-Boot Installation

To install U-boot onto the STMicroelectronics STM3240G-EVAL, follow the step-wise procedure documented below:

1. On the Windows development host download and install the ST-LINK/V2 software from the STMicroelectronics web site (`Design Support` in http://www.st.com/internet/evalboard/product/251168.jsp). The following software and documents are required:
   a) Driver: `ST-LINK/V2 USB driver for Windows 7, Vista and XP`;
   b) ST-LINK utility: `STM32 ST-LINK utility`;
   c) ST-LINK User Manual: `UM0892: STM32 ST-LINK Utility`.
2. Connect the on-board ST-LINK programmer device to the Windows development host over the USB link;
3. Power on the STM3240G-EVAL board;
4. On the Windows development host run the ST-LINK utility:
   a) Connect to the target the `Target` -> `Connect` menu;
   b) Open the `Target -> Option Bytes...` menu and select the `WDG_SW`, `nRST_STDBY` and `nRST_STOP` check-boxes, if not already selected;
   c) Program the STM3240G-EVAL board as described in Section 3.4, "Device Programming" of the *ST-LINK Utility User Manual*. File to program is `u-boot.bin`, the start address is the default 0x08000000.
5. Connect the UART port of the STM3240G-EVAL board to a host;
6. Run a terminal program (e.g. `HyperTerminal` on Windows, or `kermit` on Linux) and create a serial connection to the STM3240G-EVAL board with the following COM-port settings: `115200 8N1`;
7. Reset the STM3240G-EVAL board and see the U-Boot start-up banner in the terminal program.

5.2. U-Boot Environment

When the STM3240G-EVAL board is reset, the Linux bootstrap will proceed to boot the U-Boot firmware from the built-in Flash printing the following output to the serial console:

```
U-Boot 2010.03-linux-cortexm-1.12.0 (Dec 06 2013 - 16:23:19)

CPU  : STM32F4 (Cortex-M4)
Freqs: SYSCLK=168MHz,HCLK=168MHz,PCLK1=42MHz,PCLK2=84MHz
Board: STM3240G-EVAL board + STM-MEM add-on,Rev 2.A
```

```
DRAM:  32 MB
Flash:  8 MB
*** Warning - bad CRC, using default environment

In:    serial
Out:   serial
Err:   serial
Net:   STM32_MAC
Hit any key to stop autoboot:  0
STM3240G-EVAL>
```

U-boot makes use of the so-called environment variables to define various aspects of the system functionality. Parameters defined by the U-boot environment variables include: target IP address, target MAC address, address in RAM where a Linux bootable images will be loaded, and many more. To examine the current settings of the environment variables, run `printenv` from the U-Boot command interface.

The reason for the warning about the "bad CRC" is that U-Boot is configured to store its environment variables in the external Flash. However, this being the first time when you boot Linux STM32F4 on the development board, obviously there is no U-Boot environment programmed to the external Flash. U-Boot goes to the external Flash, fails to find its environment there, prints the warning message and resorts to using the default environment integrated into the U-Boot image at build time.

U-Boot provides a command called `saveenv` that stores the up-to-date run-time environment to the persistent storage, which will be the external Flash for the U-Boot configuration used on the STM3240G-EVAL board. You need to call `saveenv` any time when you want to copy current settings of the environment variables to the persistent storage in Flash. This is how you can write the current U-Boot environment to the external Flash:

```
STM3240G-EVAL> saveenv
Saving Environment to Flash...
...
STM3240G-EVAL>
```

Reset the STM3240G-EVAL board and check that there is no warning about the bad CRC in the boot-up messages. This is expected since now U-Boot successfully finds its environment in the external Flash:

```
STM3240G-EVAL> reset
resetting ...

U-Boot 2010.03-linux-cortexm-1.12.0 (Dec 06 2013 - 19:43:45)
...
Hit any key to stop autoboot:  0
STM3240G-EVAL>
```

5.3. Ethernet MAC Address

In Linux STM32F4, the MAC address of the Ethernet interface is defined by the `ethaddr` U-Boot environment variable. The value of the MAC address can be examined from the U-Boot command line monitor as follows:

```
STM3240G-EVAL> printenv ethaddr
ethaddr=C0:B1:3C:88:88:88
STM3240G-EVAL>
```

The default U-Boot environment for the STM3240G-EVAL board sets `ethaddr` to a fixed MAC address. This address should work for you in a general case, however if you have more than two STM3240G-EVAL boards in your LAN, use of the same address on multiple boards may result in packet collisions in your LAN and overall may render your LAN mal-functioning.

If you have more than one STM3240G-EVAL boards in your LAN, you have to assign a unique MAC address to each board.

The MAC address can be changed by modifying the `ethaddr` variable as follows:

```
STM3240G-EVAL> setenv ethaddr C0:B1:3C:88:88:89
```

Don't forget to store your update in the persistent storage using `saveenv` so it is remembered across resets and power cycles.

5.4. Network Configuration

You will have to update the network configuration of your board to match settings of your local environment.

Typically, all you have to allow loading images over network from a TFTP server is update the U-Boot environment variables `ipaddr` (the board IP address) and `serverip` (the IP address of the TFTP server). Here is how it is done.

Update `ipaddr` and `serverip`:

```
STM3240G-EVAL> setenv ipaddr 192.168.0.2
STM3240G-EVAL> setenv serverip 192.168.0.1
```

and then save the updated environment to the external Flash so that your changes are persistent across resets/power cycles.

5.5. Running Pre-installed Linux Image

The STM-MEM add-on board comes with a Linux bootable image for the `networking` project installed into external Flash. To boot this Linux configuration onto the STM3240G-EVAL board just reset the board and let U-Boot perform the autoboot sequence.

Detailed information on functionality of the pre-installed Linux image can be found in *Linux Cortex-M User's Manual*, Section 3.

5.6. Loading Linux Images

At this point, you are able to load Linux bootable images to the board over TFTP and either boot them directly or install them to the external Flash to allow booting Linux from Flash on power-up/reset.

On the host, activate the Linux STM32F4 development environment and build the `networking` project:

```
-bash-3.2$ . ACTIVATE.sh
-bash-3.2$ cd projects/networking/
-bash-3.2$ make
...
-bash-3.2$
```

Copy the Linux bootable image to the TFTP download directory:

```
-bash-3.2$ cp networking.uImage /tftpboot/vlad/
-bash-3.2$
```

To load the image directly, use the `netboot` U-Boot macro:

```
STM3240G-EVAL> setenv image vlad/networking.uImage
STM3240G-EVAL> run netboot
...
TFTP from server 172.17.0.1; our IP address is 172.17.5.100
Filename 'vlad/networking.uImage'.
...
Loading: #################################################################
         #################################################################
         #############
done
Bytes transferred = 2084704 (1fcf60 hex)
...
   Image Name:   Linux-2.6.33-arm1
   Image Type:   ARM Linux Kernel Image (uncompressed)
...
   Verifying Checksum ... OK
   Loading Kernel Image ... OK
OK

Starting kernel ...

Linux version 2.6.33-arm1 (vlad@ocean.emcraft.com) (gcc version 4.4.1 (Sourcery G++ Lite
2010q1-189) ) #1 Fri Dec 06 15:43:44 MSK 2013
...
```

To load the image into the Flash, use the U-Boot `update` macro:

```
STM3240G-EVAL> setenv image vlad/networking.uImage
STM3240G-EVAL> run update
...
TFTP from server 172.17.0.1; our IP address is 172.17.5.100
Filename 'vlad/networking.uImage'.
...
Loading: #################################################################
         #################################################################
         #############
done
Bytes transferred = 2084704 (1fcf60 hex)
................................ done
Un-Protected 32 sectors

................................ done
Erased 32 sectors
Copy to Flash... done
STM3240G-EVAL>
```

Reset the board and verify that the newly programmed image boots on the target in the autoboot mode:

```
STM3240G-EVAL> reset
resetting ...

U-Boot 2010.03-linux-cortexm-1.12.0 (Dec 06 2013 - 17:19:37)
...
Starting kernel ...
...
init started: BusyBox v1.17.0 (Dec 06 2013 - 17:19:37)
~ #
```

5.7. U-Boot Build

The BSP distribution comes with U-Boot pre-built for the STM3240G-EVAL board. If however you need to re-build U-Boot for your board, please follow the instructions below:

1. Install the Linux STM32F4 distribution to the development host, as described in the Linux Cortex-M User's Manual.
2. From the top of the Linux STM32F4 installation, activate the Linux STM32F4 cross-compile environment by running `. ACTIVATE.sh`.
3. Go to the U-Boot source directory (`cd u-boot/`).
4. Run the following commands:

```
[psl@pvr u-boot]$ make stm3240g-eval_config
Configuring for stm3240g-eval board...
[psl@pvr u-boot]$ make -s
```

6. Further Materials

Refer to *Linux Cortex-M User's Manual* for detailed information on the software architecture of the Linux STM32F4 distribution.

Visit Emcraft Systems' web site at www.emcraft.com to obtain additional materials related to Linux STM32F4.

7. Support

We appreciate your review of our product and welcome any and all feedback. Comments can be sent directly by email to:

a2f-linux-support@emcraft.com

The following level of support is included with your purchase of this product:

- Email support for installation, configuration and basic use scenarios of the product during 3 months since the product purchase;
- Free upgrade to new releases of the downloadable materials included in the product during 3 months since the product purchase.

If you require support beyond of what is described above, we will be happy to provide it using resources of our contract development team. Please contact us for details.