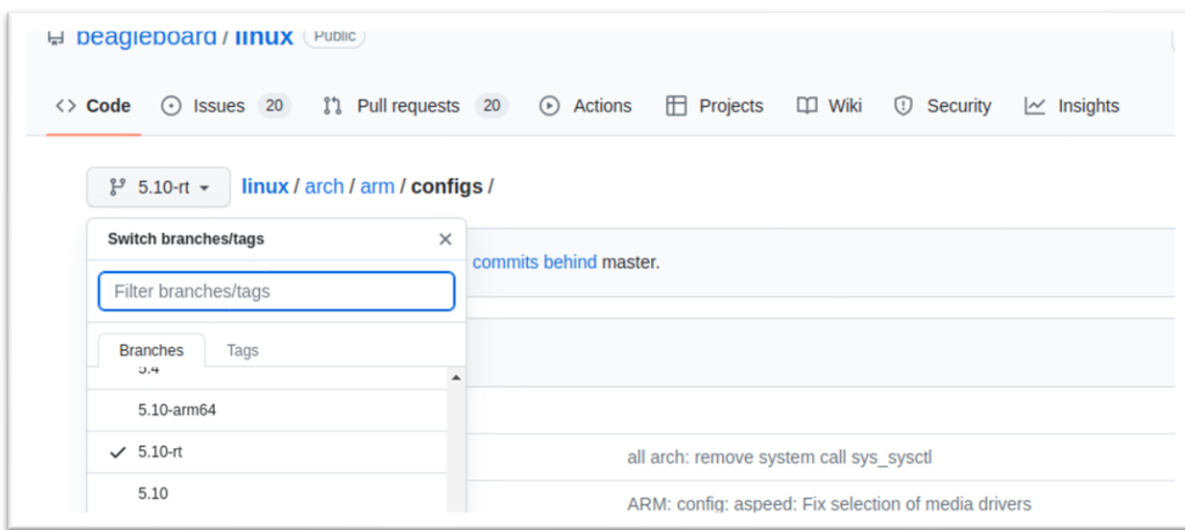


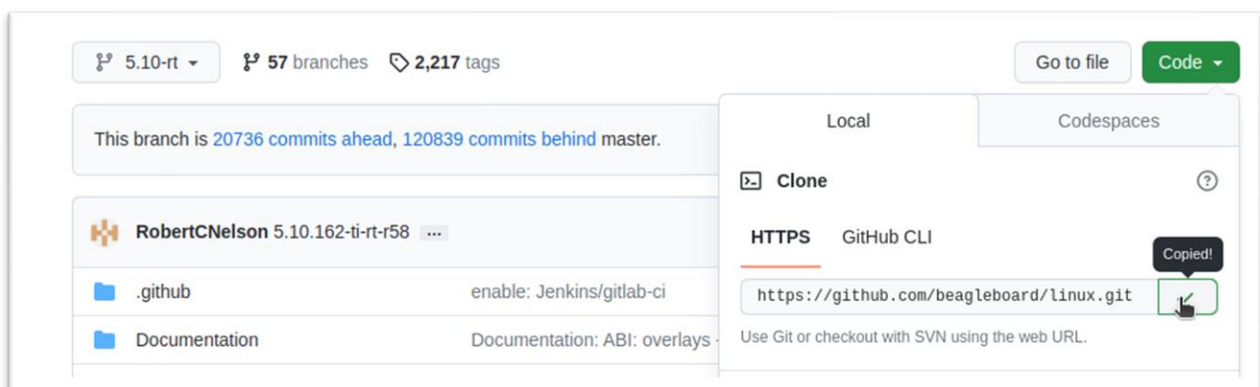
# Updating latest Kernel image v5.10

## Step 1: Selecting Latest Kernel Source.

- Go to the Beagle Board GitHub repository located at <https://github.com/beagleboard/linux>.
- Switch to the branch 5.10-rt (Here you will find bb.org\_defconfig)

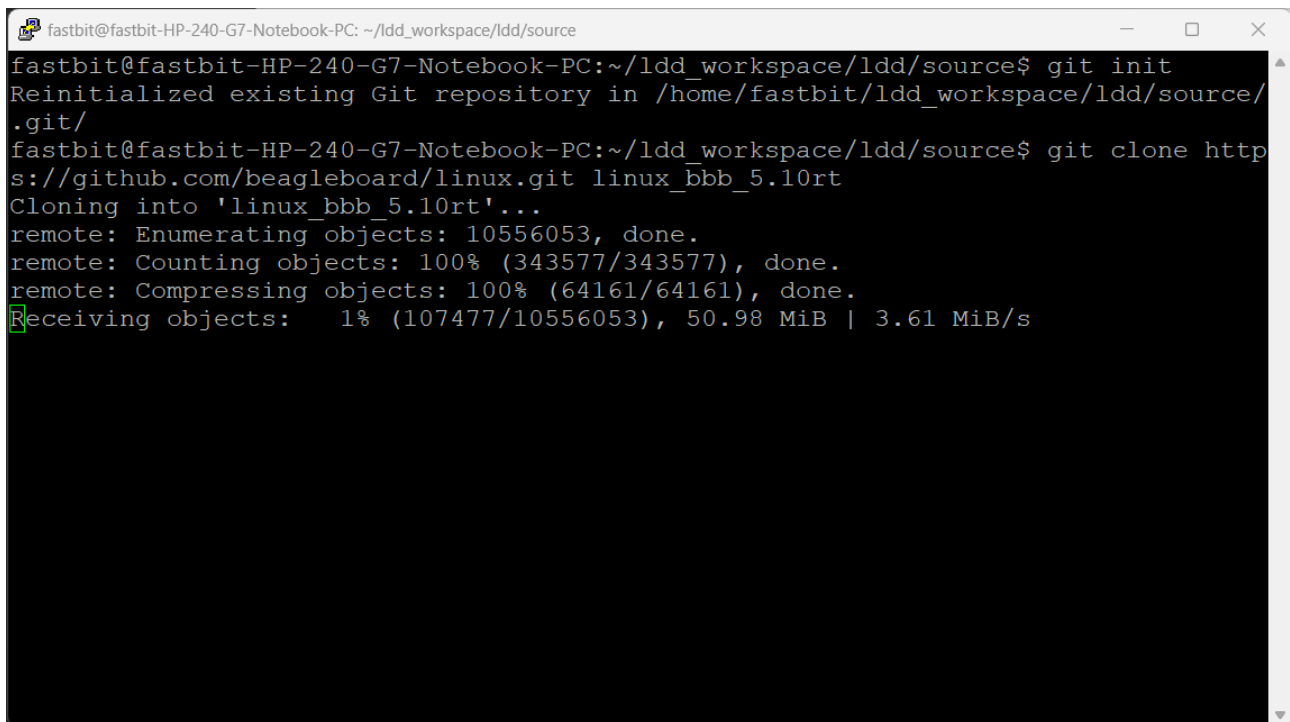


- Copy the link in the code option.



## Step 2: Clone the Kernel Source.

- Open the terminal and navigate to the source folder of your workspace. Once you are in the desired directory, run the command `git init`
- After initializing the Git repository, you can proceed to clone the repository by pasting the copied URL from Git and providing a name for the cloned repository. Here is the command: `git clone https://github.com/beagleboard/linux.git`
- Execute the command `git checkout 5.10-rt` to switch to the desired branch in the Git repository.

A terminal window screenshot showing the execution of Git commands. The window title is 'fastbit@fastbit-HP-240-G7-Notebook-PC: ~/ldd\_workspace/ldd/source'. The terminal output shows 'git init' being run, which reinitializes an existing repository. Then, 'git clone https://github.com/beagleboard/linux.git linux\_bbb\_5.10rt' is run, cloning the repository into a new directory. The output shows progress for enumerating, counting, and compressing objects, and finally receiving objects at 50.98 MiB/s.

```
fastbit@fastbit-HP-240-G7-Notebook-PC: ~/ldd_workspace/ldd/source$ git init
Reinitialized existing Git repository in /home/fastbit/ldd_workspace/ldd/source/.git/
fastbit@fastbit-HP-240-G7-Notebook-PC:~/ldd_workspace/ldd/source$ git clone https://github.com/beagleboard/linux.git linux_bbb_5.10rt
Cloning into 'linux_bbb_5.10rt'...
remote: Enumerating objects: 10556053, done.
remote: Counting objects: 100% (343577/343577), done.
remote: Compressing objects: 100% (64161/64161), done.
Receiving objects: 1% (107477/10556053), 50.98 MiB | 3.61 MiB/s
```

## Step 3: Kernel Compilation steps.

Install these 2 interface libraries using the below commands.

```
sudo apt-get install-y libgmp-dev
sudo apt-get install libmpc-dev
```

1. Removes all the temporary folder, object files, images generated during the previous build. This step also deletes the .config file if created previously.

```
make make ARCH=arm distclean
```

2. creates a .config file by using default config file given by the vendor.

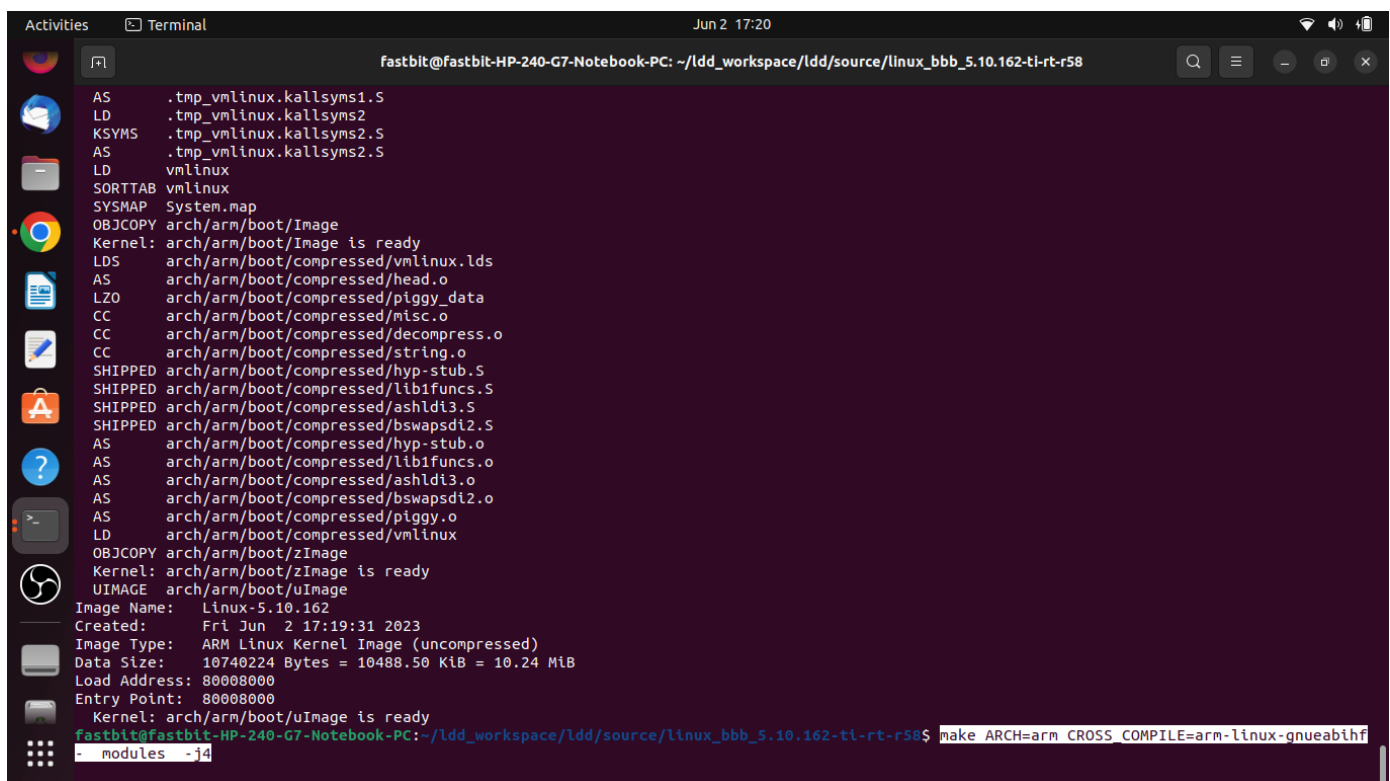
```
make ARCH=arm bb.org_defconfig
```

3. This step is optional. Run this command only if you want to change some kernel settings before compilation.

```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- menuconfig
```

4. Kernel source code compilation. This stage creates a kernel image "ulmage" also all the device tree source files will be compiled, and dtbs will be generated.

```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- ulmage dtbs
LOADADDR=0x80008000-j4
```



```
Activities  Terminal  Jun 2 17:20
fastbit@fastbit-HP-240-G7-Notebook-PC: ~/ldd_workspace/ldd/source/linux_bbb_5.10.162-tl-rt-r58

AS      .tmp_vmlinux.kallsyms1.S
LD      .tmp_vmlinux.kallsyms2
KSYMS   .tmp_vmlinux.kallsyms2.S
AS      .tmp_vmlinux.kallsyms2.S
LD      vmlinux
SORTTAB vmlinux
SYSMAP  System.map
OBJCOPY arch/arm/boot/Image
Kernel: arch/arm/boot/Image is ready
LDS     arch/arm/boot/compressed/vmlinux.lds
AS      arch/arm/boot/compressed/head.o
LZO     arch/arm/boot/compressed/piggy_data
CC      arch/arm/boot/compressed/misc.o
CC      arch/arm/boot/compressed/decompress.o
CC      arch/arm/boot/compressed/string.o
SHIPPED arch/arm/boot/compressed/hyp-stub.S
SHIPPED arch/arm/boot/compressed/libifuncs.S
SHIPPED arch/arm/boot/compressed/ashldi3.S
SHIPPED arch/arm/boot/compressed/bswapsdi2.S
AS      arch/arm/boot/compressed/hyp-stub.o
AS      arch/arm/boot/compressed/libifuncs.o
AS      arch/arm/boot/compressed/ashldi3.o
AS      arch/arm/boot/compressed/bswapsdi2.o
AS      arch/arm/boot/compressed/piggy.o
LD      arch/arm/boot/compressed/vmlinux
OBJCOPY arch/arm/boot/zImage
Kernel: arch/arm/boot/zImage is ready
UIIMAGE arch/arm/boot/uImage
Image Name: Linux-5.10.162
Created:   Fri Jun 2 17:19:31 2023
Image Type: ARM Linux Kernel Image (uncompressed)
Data Size: 10740224 Bytes = 10488.50 KiB = 10.24 MiB
Load Address: 80008000
Entry Point: 80008000
Kernel: arch/arm/boot/uImage is ready
fastbit@fastbit-HP-240-G7-Notebook-PC: ~/ldd_workspace/ldd/source/linux_bbb_5.10.162-tl-rt-r58$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf-
- modules -j4
```

- ```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- modules-j4
```

```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- modules-j4
```

The screenshot shows a terminal window on a Linux system. The title bar indicates the user is 'fastbit' and the window title is 'Jun 2, 15:33'. The terminal prompt is 'fastbit@fastbit-HP-240-G7-Notebook-PC: ~/ldd\_workspace/ldd/source/linux\_bbb\_5.10.162-ti-rt-r58'. The user has run 'sudo make ARCH=arm modules\_install', which has installed various modules. The output shows a list of modules being installed, including sound, usb, and crypto modules. The terminal window has a dark background with a light-colored text. The left sidebar shows the Ubuntu desktop environment with various application icons.

```
LD [M] sound/soc/ti/snd-soc-omap-hdmi.ko
LD [M] sound/soc/ti/snd-soc-omap-mcbsp.ko
LD [M] sound/soc/ti/snd-soc-omap-mcpdm.ko
LD [M] sound/soc/ti/snd-soc-ti-edma.ko
LD [M] sound/soc/ti/snd-soc-ti-sdma.ko
LD [M] sound/soc/ti/snd-soc-ti-udma.ko
LD [M] sound/soundcore.ko
LD [M] sound/usb/6fire/snd-usb-6fire.ko
LD [M] sound/usb/caiaq/snd-usb-caiaq.ko
LD [M] sound/usb/bcd2000/snd-bcd2000.ko
LD [M] sound/usb/hiface/snd-usb-hiface.ko
LD [M] sound/usb/line6/snd-usb-line6.ko
LD [M] sound/usb/line6/snd-usb-pod.ko
LD [M] sound/usb/line6/snd-usb-podhd.ko
LD [M] sound/usb/line6/snd-usb-toneport.ko
LD [M] sound/usb/line6/snd-usb-variax.ko
LD [M] sound/usb/misc/snd-ua101.ko
LD [M] sound/usb/snd-usb-audio.ko
LD [M] sound/usb/snd-usbmidi-lib.ko
fastbit@fastbit-HP-240-G7-Notebook-PC:~/ldd_workspace/ldd/source/linux_bbb_5.10.162-ti-rt-r58$ sudo make ARCH=arm modules_install
[sudo] password for fastbit:
INSTALL arch/arm/crypto/aes-arm-bs.ko
INSTALL arch/arm/crypto/aes-arm.ko
INSTALL arch/arm/crypto/ghash-arm-ce.ko
INSTALL arch/arm/crypto/nhpoly1305-neon.ko
INSTALL arch/arm/crypto/sha1-arm-neon.ko
INSTALL arch/arm/crypto/sha1-arm.ko
INSTALL arch/arm/crypto/sha256-arm.ko
INSTALL arch/arm/crypto/sha512-arm.ko
INSTALL block/bfq.ko
INSTALL block/kyber-tosched.ko
INSTALL crypto/adiantum.ko
INSTALL crypto/aegis128.ko
INSTALL crypto/af_alg.ko
INSTALL crypto/algif_aead.ko
INSTALL crypto/algif_hash.ko
INSTALL crypto/algif_rng.ko
INSTALL crypto/algif_skcipher.ko
```

- ```
sudo make ARCH=arm modules_install
```

```
sudo make ARCH=arm modules_install
```

The image shows a terminal window on a Linux system. The title bar at the top reads "Activities" and "Terminal". The terminal prompt is "fastbit@fastbit-HP-240-G7-Notebook-PC: ~/ldd\_workspace/ldd/source/linux\_bbb\_5.10.162-tl-rt-r58". The terminal output consists of a list of kernel modules being installed, each preceded by the word "INSTALL". The modules are:

- sound/soc/codecs/snd-soc-tlv320aic23.ko
- sound/soc/codecs/snd-soc-tlv320aic31xx.ko
- sound/soc/codecs/snd-soc-tlv320aic3x.ko
- sound/soc/codecs/snd-soc-tpa6130a2.ko
- sound/soc/codecs/snd-soc-ts3a227e.ko
- sound/soc/codecs/snd-soc-wm8753.ko
- sound/soc/codecs/snd-soc-wm8804-i2c.ko
- sound/soc/codecs/snd-soc-wm8804.ko
- sound/soc/codecs/snd-soc-wm8903.ko
- sound/soc/codecs/snd-soc-wm8904.ko
- sound/soc/codecs/snd-soc-wm8960.ko
- sound/soc/generic/snd-soc-audio-graph-card.ko
- sound/soc/generic/snd-soc-simple-card-utils.ko
- sound/soc/generic/snd-soc-simple-card.ko
- sound/soc/snd-soc-core.ko
- sound/soc/ti/snd-soc-davinci-mcasp.ko
- sound/soc/ti/snd-soc-omap-dmic.ko
- sound/soc/ti/snd-soc-omap-hdmi.ko
- sound/soc/ti/snd-soc-omap-mcbsp.ko
- sound/soc/ti/snd-soc-omap-mcpdm.ko
- sound/soc/ti/snd-soc-ti-edma.ko
- sound/soc/ti/snd-soc-ti-sdma.ko
- sound/soc/ti/snd-soc-ti-udma.ko
- sound/soundcore.ko
- sound/usb/6fire/snd-usb-6fire.ko
- sound/usb/bcd2000/snd-bcd2000.ko
- sound/usb/caiaq/snd-usb-caiaq.ko
- sound/usb/hiface/snd-usb-hiface.ko
- sound/usb/line6/snd-usb-line6.ko
- sound/usb/line6/snd-usb-pod.ko
- sound/usb/line6/snd-usb-podhd.ko
- sound/usb/line6/snd-usb-toneport.ko
- sound/usb/line6/snd-usb-variax.ko
- sound/usb/misc/snd-ua101.ko
- sound/usb/snd-usb-audio.ko
- sound/usb/snd-usbmidi-lib.ko

The terminal ends with the command "DEPMOD 5.10.162" and the prompt "fastbit@fastbit-HP-240-G7-Notebook-PC:~/ldd\_workspace/ldd/source/linux\_bbb\_5.10.162-tl-rt-r58\$".

## Step 4: Updating ulmage, DTB and Copying Modules to SD Card.

1. Insert the SD card into a card reader and open it on your computer.

- Access the boot partition.
- Rename the existing ulmage file to a different name.

2. Open the terminal and navigate to the source directory:

```
cd linux_bbb_5.10.162-ti-rt-r58/arch/arm/boot/
```

3. Copy the ulmage file to the SD card's boot partition:

```
cp ulmage /media/<username>/BOOT/
```

4. Open the /lib/modules/ directory and - Copy the newly created folder to the SD card's root file system:

```
cd /lib/modules/
```

```
ls
```

```
sudo cp -a 5.10.162-ti-rt-r58/ /media/<username>/ROOTFS/lib/modules/
```

5. Sync the changes to ensure data is written to the SD card properly:

```
sync
```

6. Open the source directory in the terminal.

- Navigate to /arch/arm/boot/dts
- Copy the **dtb** file to the **BOOT** partition using the command "`cp am335x-boneblack.dtb /media/<username>/BOOT/`".

