

## HOWTO: Running Linux on MIPSfpga-plus SoC deployed on Altera MAX10 FPGA board

This short document contains some information about running Linux on MIPSfpga-plus[\[link\]](#) system that is deployed on Terasic DE10-Lite development board with Altera MAX10 chip and SDRAM. It is based on the MIPSfpga labs.

### System and PC configuration

- Terasic DE10-Lite development board (Altera MAX10 FPGA chip + SDRAM 64MB);
- working JTAG hardware debugger (FTDI MPSSE based board in my case [\[link\]](#));
- Quartus Prime 16.1 on Windows 7 (x64) PC for FPGA configuration synthesis;
- Ubuntu 17.04 (x64) virtual machine for Linux and RAM-disk build.

### MIPSfpga-plus RTL settings (Windows PC)

- those lines should be uncommented in mfp\_ahb\_lite\_matrix\_config.vh:
 

```
`define MFP_USE_SDRAM_MEMORY
`define MFP_USE_DUPLEX_UART
`define MFP_USE_MPSSE_DEBUGGER
```
- you should build the MIPSfpga-plus SoC and program the FPGA device.

### Kernel and RAM-disk building (Linux console)

- installing the requirements for building the kernel:
 

```
sudo apt-get update
sudo apt-get install -y build-essential git libncurses5-dev bc unzip
```
- working directory creating
 

```
mkdir ~/mipsfpga
cd ~/mipsfpga
```
- downloading the MIPS toolchain
 

```
wget http://codescape-mips-sdk.imgtec.com/components/toolchain/2016.05-06/Codescape.GNU.Tools.Package.2016.05-06.for.MIPS.MTI.Linux.CentOS-5.x86_64.tar.gz
```
- and installing it
 

```
mkdir ~/mipsfpga/toolchain
tar -xvf Codescape.GNU.Tools.Package.2016.05-06.for.MIPS.MTI.Linux.CentOS-5.x86_64.tar.gz -C ~/mipsfpga/toolchain
```
- adding it to the \$PATH variable (in ~/.profile)
 

```
pathadd() {
    if [ -d "$1" ] && [[ ":$PATH:" != *"$1:"* ]]; then
        PATH="$1${PATH:+":$PATH"}"
    fi
}

pathadd "$HOME/bin"
pathadd "$HOME/.local/bin"
pathadd "$HOME/mipsfpga/toolchain/mips-mti-linux-gnu/2016.05-06/bin"
```
- getting Linux and Buildroot source
 

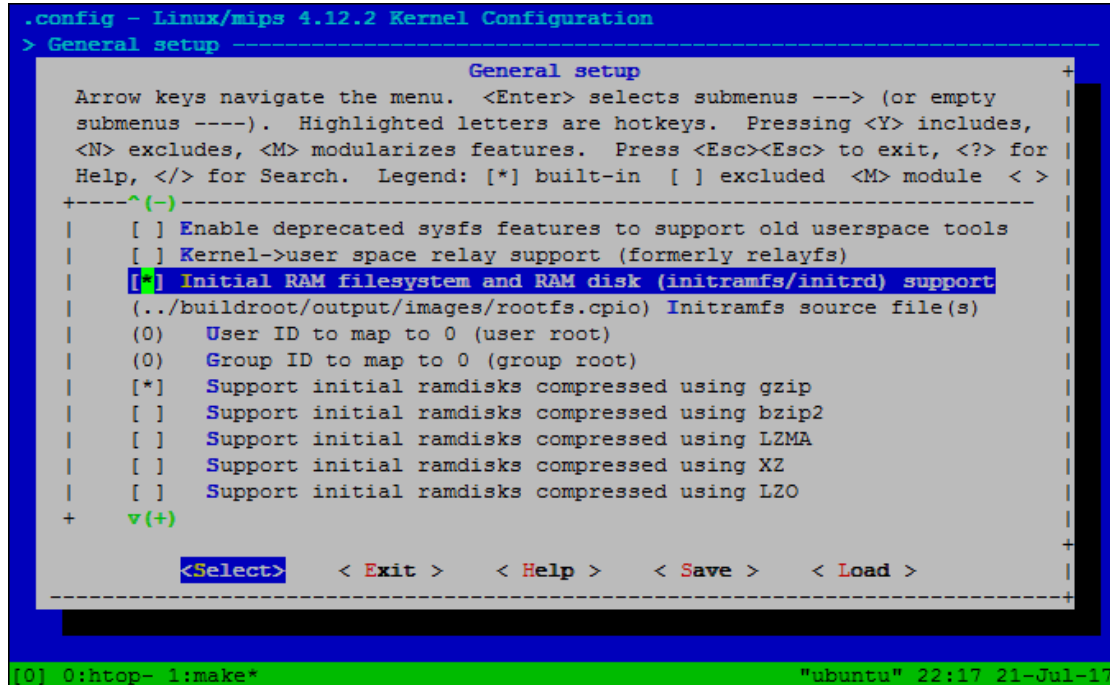
```
git clone git://git.kernel.org/pub/scm/linux/kernel/git/stable/linux-stable.git kernel
git clone git://git.buildroot.net/buildroot
```
- creating the folder for Linux and Buildroot patches;
 

```
mkdir patches
```
- then you need to download those patches from [\[link\]](#) and save them in this folder;
- buildroot configure and build (using 2017.05.1 tag = commit f3d8beeb3694):
 

```
cd buildroot
git checkout 2017.05.1
git apply ../patches/MIPSfpga_buildroot.patch
make xilfpga_static_defconfig
make
```

- Linux configure using v4.12.1 tag = commit cb6621858813)
 

```
cd ../kernel/
ls -l ../buildroot/output/images/rootfs.cpio
git checkout v4.12.1
git apply ../patches/MIPSFPGA_linux.patch
make ARCH=mips xilfpga_de10lite_defconfig
make ARCH=mips menuconfig
```
- RAM-disk support enable:



- Linux kernel build:
 

```
make ARCH=mips CROSS_COMPILE=~/.mipsfpga/toolchain/mips-mti-linux-gnu/2016.05-06/bin/mips-mti-linux-gnu-
```

### Kernel loading and running (Linux console)

- to see the system log you need to connect to the UART port;
- loading and running the kernel image using the gdb:
 

```
mips-mti-linux-gnu-gdb ./vmlinux
(gdb) target remote <OpenOCD host>:<port>
(gdb) mo reset halt
(gdb) set endian little
(gdb) load
(gdb) continue
```
- after the short delay (because of EARLY\_PRINTK disabled) the log that is looking like this will be shown:
 

```
Linux version 4.12.1+ (stas@ubuntu) (gcc version 4.9.2 (Codescape GNU Tools 2016.05-06 for MIPS MTI Linux) ) #1 Sat Jul 22 14:35:05 MSK 2017
CPU0 revision is: 00019e60 (MIPS M14KEc)
MIPS: machine is terasic,de10lite
Determined physical RAM map:
  memory: 04000000 @ 00000000 (usable)
Initrd not found or empty - disabling initrd
Primary instruction cache 4kB, VIPT, 2-way, linesize 16 bytes.
Primary data cache 4kB, 2-way, VIPT, no aliases, linesize 16 bytes
Zone ranges:
  Normal [mem 0x0000000000000000-0x0000000003ffffff]
Movable zone start for each node
Early memory node ranges
  node 0: [mem 0x0000000000000000-0x0000000003ffffff]
Initmem setup node 0 [mem 0x0000000000000000-0x0000000003ffffff]
Built 1 zonelists in Zone order, mobility grouping on. Total pages: 16256
Kernel command line: console=ttyS0,115200
PID hash table entries: 256 (order: -2, 1024 bytes)
Dentry cache hash table entries: 8192 (order: 3, 32768 bytes)
```

```

Inode-cache hash table entries: 4096 (order: 2, 16384 bytes)
Memory: 60512K/65536K available (1830K kernel code, 99K rwd data, 320K rodata, 944K init,
185K bss, 5024K reserved, 0K cma-reserved)
NR_IRQS:8
clocksource: MIPS: mask: 0xffffffff max_cycles: 0xffffffff, max_idle_ns: 38225208935 ns
sched_clock: 32 bits at 50MHz, resolution 20ns, wraps every 42949672950ns
Console: colour dummy device 80x25
Calibrating delay loop... 10.81 BogoMIPS (lpj=21632)
pid_max: default: 32768 minimum: 301
Mount-cache hash table entries: 1024 (order: 0, 4096 bytes)
Mountpoint-cache hash table entries: 1024 (order: 0, 4096 bytes)
devtmpfs: initialized
clocksource: jiffies: mask: 0xffffffff max_cycles: 0xffffffff, max_idle_ns:
7645041785100000 ns
futex hash table entries: 256 (order: -1, 3072 bytes)
clocksource: Switched to clocksource MIPS
random: fast init done
workingset: timestamp_bits=30 max_order=14 bucket_order=0
Serial: 8250/16550 driver, 4 ports, IRQ sharing disabled
console [ttyS0] disabled
b0400000.serial: ttyS0 at MMIO 0xb0401000 (irq = 0, base_baud = 3125000) is a 16550A
console [ttyS0] enabled
Freeing unused kernel memory: 944K
This architecture does not have kernel memory protection.
mount: mounting devpts on /dev/pts failed: No such device
mount: mounting tmpfs on /dev/shm failed: Invalid argument
mount: mounting tmpfs on /tmp failed: Invalid argument
mount: mounting tmpfs on /run failed: Invalid argument
Starting logging: OK
Initializing random number generator... done.
Starting network: ip: socket: Function not implemented
ip: socket: Function not implemented
FAIL

Welcome to MIPSfpga
mipsfpga login:

```

### GPIO test (MIPSfpga console)

- working with GPIO for system test:

```

ls /sys/class/gpio/
export      gpiochip480  unexport
mount -t debugfs none /sys/kernel/debug
cat /sys/kernel/debug/gpio
gpiochip0: GPIOs 480-511, parent: platform/bf800000.gpio, bf800000.gpio:
echo 480 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio480/direction
echo 1 > /sys/class/gpio/gpio480/value

```
- after those actions LED0 should be ON.