PiLUP

Operative System and firmware upgrading

User and developer manual

This document is a work in progress.
The most current version is kept in the github repository:
https://github.com/assiro/PiLUP-project

Standard PiLUP system configuration

Default system setting parameters

- MAC address 00:0A:35:00:1E:53
- Eth0 dhcp enabled
- Eth0:0 192.168.168.168 - net mask 255.255.255.0 (fi

Written by Roberto Assiro

Manual version 1.0 – witten for PiLUP

INFN - SEZIONE DI LECCE
March 2018

Repository's files for PiLUP:

- **pilup.ub** Compressed image of the Petalinux System
- **pilup.bit** bitstream file for FPGA
- **pilup.bin** Image of first 16Mb of flash memory. It includes uboot, image.ub and fpga bitstream

Implemented commands in Petalinux

- **mountboot** mounts itbs volume (system image folder) under /boot
- **umountboot** unmounts the volume
- **version (**or **ver**) shows version of system
- **image** replace petalinux image.ub from tftp server (no bitstream) **example: image 172.16.17.198**
- **upgrade** replace petalinux (pilup.ub) and bitstrem (pilup.bit) from tftp **example: upgrade 172.16.17.198**

How to program a brand new PiLUP

store PiLUP.bin to flash memory (Jtag only)

1. Connect jtag programmer to PILUP (Digilent or xilinx)
2. Remove the jumper (flash/jtag) and turn on the PILUP
3. Launch SDK on computer and select "**Program flash**" from the Xilinx Tools menu
4. Select **PiLUP.bin** as image file (49 Mb)
5. Click "Program" button to store the file into the flash memory. The programming phase requires 45 minutes. Restart PiLUP

How to update the operative system or bitstream on PILUP

To update bitstream and linux image to the latest version there are two ways:

1. **PiLUP update by TFTP server. This solution require ethernet connection to the network**

1 - TFTP server:

- Connect PiLUP to the network by LAN and turn it on
- At command line type: **upgrade <TFT server** IP Number> **example: upgrade 172.16.17.198**

  PiLUP will download and overwrite the image system and bitstream files on flash memory
- Check the new version number of operative system after reboot

Changing the PILUP's MAC address

Technical documentation about PILUP's MAC address definition

U-boot uses the environment variable "ethaddr" as the MAC address.

How to?

Customization of the board MAC address using u-boot's command to change the value of the environment variable "ethaddr"

- Turn on the PILUP, the u-boot will start. Before the counting down finishes press enter, you will get the prompt :

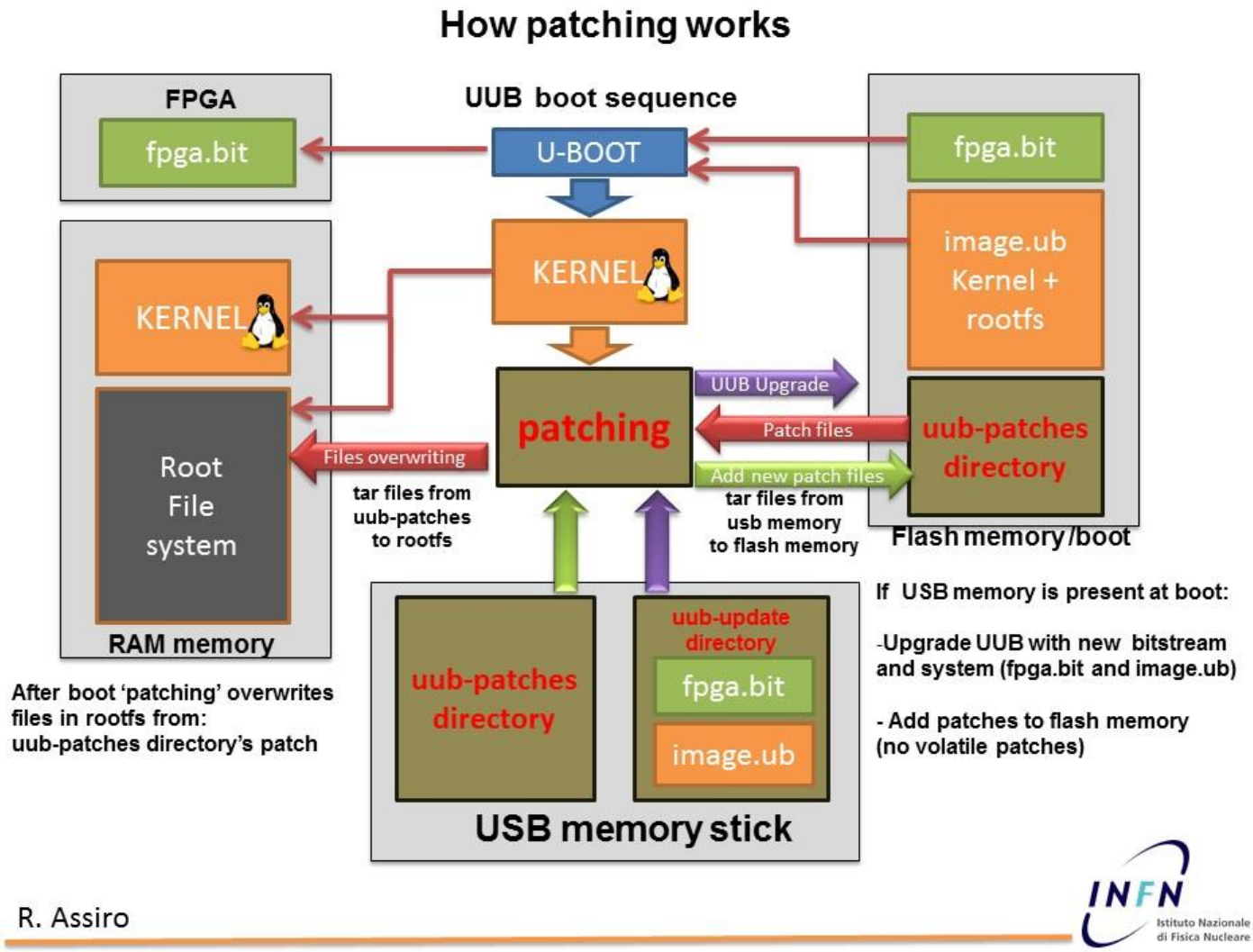
```
U-boot-PiLUP>

U-boot-PiLUP> set ethaddr 00:11:22:33:44:55    (example of new MAC to provide)

U-boot-PiLUP> saveenv          save the setting into the flash memory's environment

U-boot-PiLUP> saveenv          repeat the command for the second partition

Reboot the PILUP
```

The new MAC address is saved into the environments of flash memory and petalinux will start with new MAC.
The value in device tree is overwritten
In order to use several PILUP's connected to the same network is suggested to define different MAC address for each board during first check.

"Patching" the solution to add software and settings to the rootfs

"patching" runs after kernel start, it will:

1. write or overwrite files in rootfs from patches contents in flash memory (pilup-patches folder) if USB memory stick is connected "patching" writes or overwrites files in rootfs from patches contents (pilup-patches folder). If patch's name starts with a "**P**", patch file will be stored in flash memory too (permanent patching).

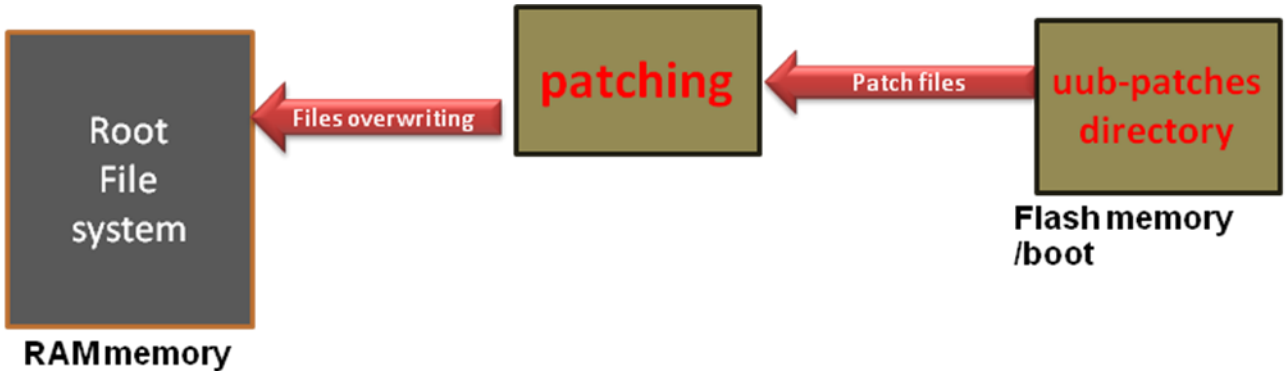


R. Assiro

The patch

"Patch" is a compressed file which contains programs and scripts with path destination in root file system.
The patch's name must start with **"P"** followed by a sequence number from 00 to 99
In the flash memory the directory containing patch's files is **/boot/pilup-patches**

"Patching" is the main script that runs, after Kernel starts, and it can write and overwrite files in rootfs



How build a no volatile patch into flash memory (/boot)

command to create a patch from PILUP's command line:

Example:

1. Tranfer your application software in your workspace in file system (example: bin/hello.elf)

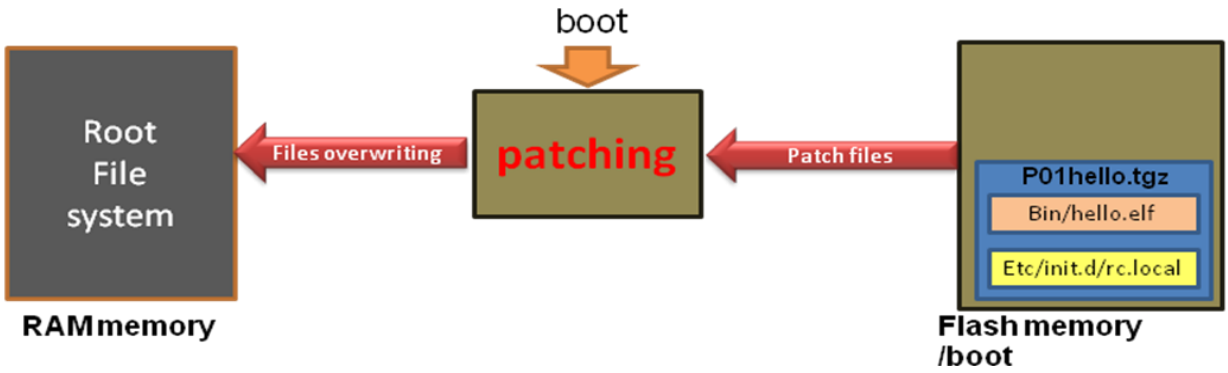
2. mount flash memory partition **/boot**:
   root@PiLUP:~# **mountboot**

3. create tar file from /root:
   root@PiLUP:~# **cd /**
   root@PiLUP:~# **tar cvzf /boot/PiLUP-patches/P01hello.tgz bin/hello.elf**

4. unmount partition /boot:
   root@PiLUP:~# **umountboot**

How to build a patch to run automatically a program at boot

rc.local is a script file to edit to set an auto start process



1. edit and modify rc.local :
   root@PiLUP:~# **vi etc/init.d/rc.local**

2. insert command a line to run your application (example hello.elf in flash memory)

   #!/bin/bash
   #Please put follow your application programs you want run at boot
   #example: name2run > /dev/null 2>/dev/null &

   hello.elf > /dev/null 2>/dev/null &
   echo "hello is running at boot!"

   exit and save from vi

3. build the patch file :
   root@PiLUP:~# **cd /**
   root@PiLUP:~# **mountboot**
   root@PiLUP:~# **tar cvzf /usb/PiLUP-patches/P01hello.tgz bin/hello.elf etc/init.d/rc.local**
   root@PiLUP:~# **umountboot**

Reboot the PILUP, hello.elf will run automatically after boot

How to connect a laptop to the PILUP by LAN without network

Standard LAN configuration is DHCP (eth0) and alias address **192.168.168.168** (eth0:0)

direct connection LAN to LAN:

Set your laptop LAN configuration as fixed IP number :

IP address : 192.168.168.1
IP mask : 255.255.255.0

Open your browser and connect to: 192.168.168.168

How to create a TFTP Server on your Ubuntu machine

Install and Setup

1. Install following packages.

```
sudo apt-get install xinetd tftpd tftp
```

3. Create /etc/xinetd.d/tftp and put this entry

```
service tftp
{
protocol        = udp
port            = 69
socket_type     = dgram
wait            = yes
user            = nobody
server          = /usr/sbin/in.tftpd
server_args     = /tftpboot
disable         = no
}
```

15. Create a folder /tftpboot this should match whatever you gave in server_args. mostly it will be tftpboot

```
sudo mkdir /tftpboot
sudo chmod -R 777 /tftpboot
sudo chown -R nobody /tftpboot
```

19. Restart the xinetd service.

newer systems:

```
sudo service xinetd restart
```

older systems:

```
sudo /etc/init.d/xinetd restart
```

Now our tftp server is up and running.

Testing our tftp server

5. Create a file named test with some content in /tftpboot path of the tftp server

Obtain the ip address of the tftp server using ifconfig command

6. Now in some other system follow the following steps.

```
7.      tftp 192.168.1.2
8.      tftp> get test
9.      Sent 159 bytes in 0.0 seconds
10.
11.     tftp> quit
12.
```

```
cat test

how to restart tftp service : sudo service tftpd-hpa restart
```