

Guía instalación Linux-CIAA en placa

Requisitos

- Distro basada en Debian (Ubuntu 14 recomendado)
- Linux ya compilado con buildroot

Instalamos algunos paquetes requeridos:

```
sudo aptitude install php5-cli
sudo aptitude install libftdi-dev
sudo aptitude install libusb-1.0-0-dev
sudo aptitude install pkg-config
```

Bajamos y compilamos *OpenOCD*

```
mkdir openocd
wget http://ufpr.dl.sourceforge.net/project/openocd/openocd/0.8.0/openocd-0.8.0.tar.bz2
tar -xvzf openocd-0.8.0.tar.bz2
cd openocd-0.8.0
./configure --enable-ftdi
```

Resultado:

```
OpenOCD configuration summary
-----
MPSSE mode of FTDI based devices      yes
ST-Link JTAG Programmer                yes (auto)
TI ICDI JTAG Programmer                yes (auto)
Keil ULINK JTAG Programmer             yes (auto)
Altera USB-Blaster II Compatible      yes (auto)
Segger J-Link JTAG Programmer          yes (auto)
OSBDM (JTAG only) Programmer          yes (auto)
eStick/opendous JTAG Programmer        yes (auto)
Andes JTAG Programmer                  yes (auto)
Versaloon-Link JTAG Programmer         yes (auto)
USBProg JTAG Programmer                yes (auto)
Raisonance RLink JTAG Programmer       yes (auto)
Olimex ARM-JTAG-EW Programmer          yes (auto)
CMSIS-DAP Compliant Debugger           no
```

Ahora compilamos e instalamos:

```
make
sudo make install
```

Ir al repositorio de Firmware-CIAA y bajar el archivo de configuración para la ciao-nxp

```
wget
https://raw.githubusercontent.com/ciaa/Firmware/master/modules/tools/openocd/cfg/cortexM4/lpc43xx/1pc4337/ciao-nxp.cfg
```

ahora podemos ejecutar openocd

```
sudo openocd -f ciao-nxp.cfg
```

Lanzará un error porque no enchufamos la placa.

Enchufar la CIAA por el usb debug

Ejecutar *dmesg* para ver si detectó el chip FTDI

```
[ 1444.408307] usb 1-2: new full-speed USB device number 3 using ohci-pci
[ 1445.068247] usb 1-2: New USB device found, idVendor=0403, idProduct=6010
[ 1445.068253] usb 1-2: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[ 1445.068256] usb 1-2: Product: Dual RS232-HS
[ 1445.068258] usb 1-2: Manufacturer: FTDI
[ 1445.166288] usbcore: registered new interface driver usbserial
[ 1445.166592] usbcore: registered new interface driver usbserial_generic
[ 1445.166681] usbserial: USB Serial support registered for generic
[ 1445.204078] usbcore: registered new interface driver ftdi_sio
[ 1445.205913] usbserial: USB Serial support registered for FTDI USB Serial Device
[ 1445.205960] ftdi_sio 1-2:1.0: FTDI USB Serial Device converter detected
[ 1445.206053] usb 1-2: Detected FT2232H
[ 1445.206055] usb 1-2: Number of endpoints 2
[ 1445.206056] usb 1-2: Endpoint 1 MaxPacketSize 512
[ 1445.206058] usb 1-2: Endpoint 2 MaxPacketSize 512
[ 1445.206059] usb 1-2: Setting MaxPacketSize 512
[ 1445.219445] usb 1-2: FTDI USB Serial Device converter now attached to ttyUSB0
[ 1445.219480] ftdi_sio 1-2:1.1: FTDI USB Serial Device converter detected
[ 1445.219545] usb 1-2: Detected FT2232H
[ 1445.219547] usb 1-2: Number of endpoints 2
[ 1445.219549] usb 1-2: Endpoint 1 MaxPacketSize 512
[ 1445.219550] usb 1-2: Endpoint 2 MaxPacketSize 512
[ 1445.219552] usb 1-2: Setting MaxPacketSize 512
[ 1445.225525] usb 1-2: FTDI USB Serial Device converter now attached to ttyUSB1
```

Ahora ejecutamos *openocd* nuevamente y lo dejamos corriendo

```
sudo openocd -f ciaa-nxp.cfg
```

Abrir otra terminal y dirigirse al directorio donde buildroot dejó la toolchain, y ejecutamos GDB:

```
cd buildroot/output/host/opt/ext-toolchain/bin
./arm-cortexm3-uclinuxeabi-gdb --eval-command="target remote localhost:3333"
```

Ejecutamos los comandos dentro de gdb, para copiar la imagen de u-boot:

```
monitor halt
monitor flash erase_sector 0 0 last
monitor flash write_image erase unlock /home/usuario/ciaa/buildroot/output/images/u-boot.bin
0x1A000000 bin
```

***NOTA:** Lo marcado en negrita puede variar según el usuario del lector

Nos aparece:

```
auto erase enabled
auto unlock enabled
wrote 131072 bytes from file /home/usuario/ciaa/buildroot/output/images/u-boot.bin in 12.463983s
(10.270 KiB/s)
```

Ya grabamos la imagen de u-boot. Y podemos cerrar la consola de gdb y la de openocd

Ahora debemos conectarnos a la TTYUSB que se creó al conectar la placa por medio de minicom o miniterm

En otra consola:

```
sudo miniterm.py -p /dev/ttyUSB1 -b 115200
```

Presionamos SW1 en la placa para resetar el micro y luego una tecla para parar el autoboot: veremos:

```
U-Boot 2010.03-07124-g1677bf6 (abr 19 2015 - 02:19:50)

CPU : LPC43xx series (Cortex-M4/M0)
Freqs: SYSTICK=144MHz,CCLK=144MHz
Board: CIAA NXP
DRAM: 8 MB
Using default environment

In: serial
Out: serial
Err: serial
Net: LPC18XX_MAC
Hit any key to stop autoboot: 0
ciaa >
```

Esta en la consola de comandos de u-boot. Deberemos montar un servidor TFTP en nuestra PC para pasarle la imagen de Linux desde nuestra PC al u-boot para que lo grabe en la memoria de la CIAA.

Creamos el servidor TFTP

```
sudo aptitude install xinetd tftpd tftp
sudo nano /etc/xinetd.d/tftp
```

Poner el contenido:

```
service tftp
{
protocol      = udp
port          = 69
socket_type   = dgram
wait         = yes
user         = noody
server        = /usr/sbin/in.tftpd
server_args   = /srv/tftp
disable      = no
}
```

Luego creamos la carpeta /srv/tftp donde vamos a poner el archivo de imagen que servimos y le damos permiso a cualquier usuario:

```
sudo mkdir /srv/tftp
sudo chmod -R 777 /srv/tftp
```

Luego copiamos la imagen creada con buildroot a este directorio:

```
sudo cp ciaa/buildroot/output/images/uImage /srv/tftp/uImage-ciaa-nxp
```

Por último reiniciamos el servidor:

```
sudo service xinetd restart
```

Cargando la imagen de linux

Nos fijamos cual es la IP de nuestra PC:

```
ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:3b:e0:d7
          inet addr:192.168.1.102  Bcast:192.168.1.255  Mask:255.255.255.0
          ...
```

En este ejemplo, es 192.168.1.102

Volvemos a la consola de u-boot, CONECTAMOS EL CABLE DE RED A LA PLACA y resetamos con SW1

Cargamos la ip de la placa y la de nuestro server tftp seteando las variables de entorno:

```
ciaa>
setenv serverip 192.168.1.102
setenv ipaddr 192.168.1.105    <-- le ponemos una ip libre de nuestra red
```

Chequeamos con el comando *printenv* que se hayan cargado correctamente.

Luego ejecutamos:

```
run update
```

Para que se copie la imagen desde el servidor tftp, el resultado debe ser:

```
ciaa > run update
Auto-negotiation...completed.
LPC18XX_MAC: link UP (100/Full)
Using LPC18XX_MAC device
TFTP from server 192.168.1.102; our IP address is 192.168.1.105
Filename 'uImage-ciaa-nxp'.
Load address: 0x28000000
Loading: #####
          #####
          #####
          #####
done
Bytes transferred = 1240762 (12eeba hex)
```

Resetamos presionando SW1 y deberá iniciar el sistema operativo