

Guía rápida U-Boot

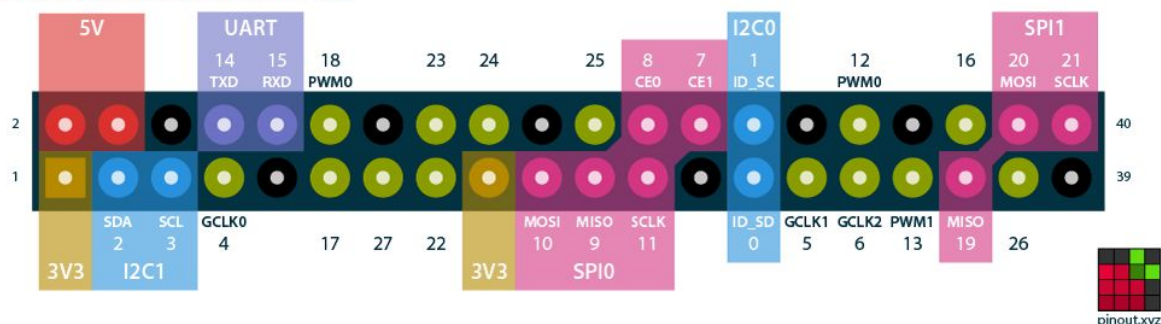
Esta guía se basa en los documentos enviados por el profesor sobre trabajos de alumnos de años anteriores, y contempla hasta la configuración de U-Boot.

1. Configuración inicial

Dado que la RaspBerry no cuenta con memoria flash interna, el sistema de booteo u-boot debe instalarse en una memoria microsd. La forma más simple de hacer esto es la siguiente:

- Descargar una versión de Raspbian (se recomienda la mas chica)
 - https://downloads.raspberrypi.org/raspbian_lite_latest
- Grabar la imagen ISO en una microsd (de 4Gb en adelante)
 - Con el programa [Etcher](#) es sumamente fácil (linux/windows)
- Probar el booteo de la memoria grabada
 - Luego del primer reinicio en donde acondiciona las particiones, debe poder cargar correctamente el sistema Debian y el usuario por defecto es **pi** con clave **raspberry**.
- Probar la comunicación por el puerto serie
 - Utilizando el único puerto serie disponible (es de 3V3, desconozco si tiene un regulador), conectarlo a la PC utilizando 115200 y configuración por defecto

Raspberry Pi GPIO BCM numbering



•

- Verificar que cuando la rasp inicia, se visualiza por el puerto serie todo el log (dmesg) y luego se habilita la terminal. Por defecto, está habilitado pero igualmente se puede usar el comando rasp-config -> interfaces para activarlo. Igual en la rasp no es necesario conectar el puerto serie porque se le puede conectar teclado y monitor (hasta resulta más práctico a veces).
- Reemplazar el kernel de linux por el binario de u-boot
 - git clone --depth 1 -b v2016.09.01 git://git.denx.de/u-boot.git
 - sudo apt-get install gcc-arm-linux-gnueabi
 - [make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi-rpi_3_32b_defconfig || make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi-rpi_defconfig] (según versión Rpi)
 - make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi-sunxi_defconfig
 - **Consejo:** Con el comando file de linux se puede verificar la arquitectura del archivo binario generado antes de continuar.
 - lsblk (ver el nombre asignado a la microSD, normalmente mmcblk)
 - sudo mount /dev/mmcblk0p1 /mnt (esta es la partición boot)
 - sudo mv /mnt/kernel.img /mnt/kernel_img (para no perder la original)
 - sudo cp -iv ./u-boot.bin /mnt/kernel_img (con pwd en donde compilamos)
 - sudo mv /mnt/kernel7.img /mnt/kernel7_img (backup, renombrar porque sino la usa en lugar de la que le copiamos).
 - sync && sudo umount /mnt
- Probar U-boot
 - Colocar la memoria en la rasp y al energizar, presionar el enter (por el puerto serie o local) para que automáticamente no pase a buscar iniciar por NFS.
- Configurar U-boot
 - bootargs=root=/dev/nfs rw ip=192.168.1.100 console=ttyS0,115200 nfsroot=192.168.1.21:/home/tito/linux-kernel-labs/modules/nfsroot/
 - bootcmd=tftp 0x42000000 zImage; tftp 0x43000000 sun8i-h2-plus-orangepi-zero.dtb; bootz 0x42000000 - 0x43000000
 - saveenv
 - printenv (mostrar variables de entorno)
 - kernel_addr_r=0x42000000
 - fdt_addr_r=0x43000000
 - (entre la rasp, orange pi y la Beagle; estas direcciones cambian)