



## Instrucciones para la instalación del entorno para las prácticas de E/S en Raspberry Pi

El objetivo de estas prácticas es gestionar directamente el sistema de Entrada/Salida (E/S) de una Raspberry Pi (RPI). Para ello debemos evitar que el Sistema Operativo (SO) interfiera con la E/S y, por tanto, arrancaremos la RPi sin SO en una configuración que en inglés se denomina *Bare Metal*. Desgraciadamente el SO también es responsable de proporcionar un sistema de ficheros, editores, compiladores, etc. por lo que tendremos que encargarnos nosotros de todo. Para ello usaremos un sistema anfitrión (*host*), por ejemplo un PC de sobremesa o un portátil, corriendo Windows para poder editar y compilar nuestros programas, y un programa cargador (bootloader) en nuestra RPi se encargará de cargar y ejecutar esos programas.

Durante el proceso de arranque de la RPi (los detalles concretos los puedes encontrar en el manual o buscando por internet), un fichero de imagen llamado `kernel.img` es cargado desde la tarjeta SD. Esta imagen contiene el ejecutable del programa que quieras correr y debe estar almacenado en una tarjeta SD formateada con el sistema FAT, junto con otros dos ficheros, `bootcode.bin` y `start.elf`. Ambos ficheros debes descargarlos de los enlaces adjuntos y no tienes que modificarlos.

El código del programa puedes editarlo y compilarlo desde el *host* usando los programas que prefieras. No obstante os recomendamos usar los dos programas siguientes que ya hemos probados:

- [Scite](#), a **SC**Intilla based Text Editor.
- [Yagarto](#), Yet Another **G**nu **AR**m **TO**olchain, un entorno de compilación cruzada para la arquitectura ARM.

Para instalarlos simplemente descarga los programas de los enlaces anteriores y sigue las instrucciones que aparezcan. A continuación, puedes seguir los siguientes pasos para automatizar el proceso de compilación:

1. Modifica el fichero `cpp.properties` que encontrarás en el directorio de instalación de Scite y añade al final las dos líneas siguientes:

```
command.build.*.s=yagarto-path\bin\make $(FileName)
command.go.*.s=yagarto-path\bin\send$(FileDir)\$(FileName).img
```

donde `yagarto-path` es el directorio donde se encuentra Yagarto, por ejemplo  
`C:\yagarto-20121222`

2. Crea un fichero llamado `make.bat` en el subdirectorio `bin` dentro del directorio de instalación de Yagarto. Este fichero debe incluir las siguientes tres líneas:

```
arm-none-eabi-as -o tmp.o %1.s
```



## Estructura de Computadores Departamento Arquitectura de Computadores



Grado en Ing. Informática, en Ing. de Computadores  
e Ing. del Software

```
arm-none-eabi-ld -e 0 -Ttext=0x8000 tmp.o  
arm-none-eabi-objcopy a.out -O binary %1.img
```

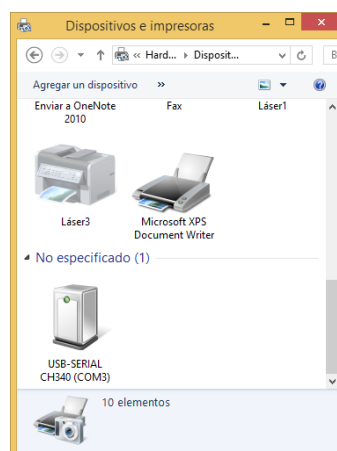
donde estamos indicando al compilador que cree un ejecutable cuyo código empiece a partir de la dirección 0x800. El fichero resultante `.img` es el fichero que debemos renombrar como `kernel.img` para almacenarlo en la tarjeta SD.

Para hacer las prácticas basta con abrir Scite, editar allí un fichero con el código fuente, guardarlo con el nombre que prefieras, y pulsar la tecla **'F7'** para ensamblarlo. En el caso de que haya errores puedes consultarlos en el panel derecho de Scite para arreglarlos en tu fichero. Si todo es correcto se creará un nuevo fichero con la extensión `.img` que debes renombrar como `kernel.img` y almacenarlo en la tarjeta SD para que el bootloader lo cargue. A continuación puedes insertar la tarjeta SD en la RPi y encenderla para ejecutar el programa.

### Adaptador USB-serie

Si dispones de un adaptador USB-serie se puede acelerar el proceso de edición-ensamblado-ejecución mediante el envío automático del ejecutable `.img` a través del puerto serie de la RPi. Para ello debes copiar en la tarjeta SD el fichero `kernel.img` que se os proporciona, de forma que tome el control en cada nuevo arranque del sistema. Este programa, tras un pitido inicial, quedará escuchando el puerto serie y lanzará a ejecutar lo que por él reciba. Habrá que reiniciar la RPi cada vez que quieras enviar un nuevo fichero para su ejecución.

Debes conectar el adaptador a un puerto USB del host y se consulta en el administrador de dispositivos el puerto COM asignado:



Mediante un emulador de terminal que soporte el protocolo XModem de comunicación serie,



## Estructura de Computadores Departamento Arquitectura de Computadores

Grado en Ing. Informática, en Ing. de Computadores  
e Ing. del Software



por ejemplo [Tera Term](#), se puede automatizar todo el proceso. Para instalarlo descarga el fichero .zip o el instalador de la dirección anterior, y añade un fichero llamado `send.ttl` al directorio donde se instale TeraTerm. El fichero debe incluir lo siguiente:

```
connect '/C=3'  
setbaud 115200  
setdtr 0  
xmodemsend param2 1  
closett
```

donde el número 3 se corresponde con el puerto COMM del adaptador. Además, crea otro fichero llamado `send.bat` en el subdirectorio `bin` donde se encuentra Yagarto y escribe en él:

```
TeraTermPath\ttpmacro.exe TeraTermPath\send.ttl "%*"
```

donde *TeraTermPath* es el directorio donde se encuentra Tera Term y almacenamos `send.ttl`. Si el directorio tuviera espacios en blanco usa dobles comillas, por ejemplo `"C:\Program Files\teraterm\ttpmacro.exe"`.

Ahora, desde Scite, cada vez que compiles un fichero puedes a continuación presionar la tecla **'F5'** para enviar el fichero `.img` que quieres ejecutar a la RPi. No olvides que hay que reiniciar la RPi cada vez que quieras enviar un nuevo fichero.

### Anexo: Instrucciones para hacer prácticas en casa utilizando Linux

Vuestro compañero Arturo José Jiménez, nos ha pasado la siguiente secuencia de comandos para quien quiera pasarle los archivos a la Rpi desde un host linux:

```
sudo apt install -y lrzsz gcc-arm-none-eabi  
sudo usermod -a -G dialout $USER  
sudo usermod -a -G dialout root  
cat > send << EOL  
arm-none-eabi-as -o tmp.o $1.s  
arm-none-eabi-ld -e 0 -Ttext=0x8000 tmp.o  
arm-none-eabi-objcopy a.out -O binary kernel.img  
stty -F /dev/ttyUSB0 115200  
sx kernel.img < /dev/ttyUSB0 > /dev/ttyUSB0  
rm -f a.out tmp.o kernel.img  
EOL  
  
chmod +x send  
  
./send Full_test
```