

Programación de Sistemas y Dispositivos CHECKLISTS – DESARROLLO DE APLICACIONES

CREACIÓN DE UN PROYECTO, COMPILACIÓN Y DEPURACIÓN

1. Crear nuevo proyecto de tipo aplicación:

File \rightarrow New \rightarrow C Project	
Project Name	lab <num></num>
Project Type	ARM Cross Target Application \rightarrow Empty Project
Toolchains	ARM Windows GCC (Sourcery G++ Lite)
Next	

2. Configurar las propiedades de compilación del proyecto:

Advanced settings \rightarrow C/C++ Build \rightarrow Settings \rightarrow Tool Settings		
ARM Sourcery Windows GCC Assembler \rightarrow Directories		
Include paths	//BSP/include	
ARM Sourcery Windows GCC Compiler \rightarrow Directories		
Include paths	//BSP/include	
ARM Sourcery Windows GCC Linker \rightarrow General		
Script File	/lab <num>.ld</num>	
Libraries +	BSP	
Library search path +	//BSP/Debug	
Apply $\rightarrow OK$		
Finish		

- 3. Copiar los ficheros fuente a la carpeta del proyecto.
- 4. Refrescar proyecto:

File → Refresh

5. Compilar proyecto:

Project → Build all

6. Crear una configuración para depurar el proyecto:

Run \rightarrow Debug Configurations \rightarrow GDB Hardware Debugging \rightarrow New		
Select other		
Use configuration specific settings		
Launchers	Standard GDB Hardware Debugging Launcher	
Main		
Name	lab <num></num>	
C/C++ Application	Debug/lab <num>.elf</num>	
Project	lab <num></num>	
Debugger		
GDB Command	arm-none-eabi-gdb	
Use remote target		
JTAG Device	Generic TCP/IP	

Host name or IP address	localhost
Port number	3333
Startup	
Reset and Delay	
Halt	
Inicialization Commands	remote reset init
Close	

7. Arrancar el On-Chip Debugger:

Run \rightarrow External tools \rightarrow OpenOCD

8. Arrancar el depurador:

Run \rightarrow Debug Configurations \rightarrow "lab<num> Debug" \rightarrow Run

CREACIÓN DE LA BIBLIOTECA DE FIRMWARE

1. Crear nuevo proyecto de tipo biblioteca:

File \rightarrow New \rightarrow C Project	
Project Name	BSP
Project Type	ARM Cross Target Static Library → Empty Project
Toolchains	ARM Windows GCC (Sourcery G++ Lite)
Next	

2. Configurar las propiedades de compilación del proyecto:

Advanced settings \rightarrow C/C++ Build \rightarrow Settings \rightarrow Tool Settings		
ARM Sourcery Windows GCC Assembler \rightarrow Directories		
Include paths	//BSP/include	
ARM Sourcery Windows GCC Compiler \rightarrow Directories		
Include paths	//BSP/include	
Apply $\rightarrow OK$		
Finish		

- 3. Crear en la carpeta de proyecto (BSP) las carpetas src e include.
- 4. Copiar ficheros de cabecera (*.h) a la carpeta *include* y los de módulo (*.c) a la carpeta src.
- 5. Refrescar proyecto:

File → Refresh

6. Compilar proyecto:

Project → Build all

VOLCADO DE APLICACIONES SOBRE FLASH-ROM:

- 1. Arrancar un intérprete de órdenes.
- 2. Cambiar al directorio que corresponda:

> cd <ruta>\PSyD\lab<num>\Debug

3. Descargar la imagen de la aplicación sobre la Flash-ROM, ejecutando el comando:

> C:\software\electronica\ARM\EclipseARM\openocd-0.7.0\bin\openocd-0.7.0	
-f "C:\software\electronica\ARM\EclipseARM\openocd-0.7.0\scripts\test\arm-fdi-uc	n.cfg"

- -c "program lab<num>.hex 0x0"
 - 4. Una vez finalizada la prueba, restaurar el contenido original de la Flash ROM ejecutando:

> C:\software\electronica\ARM\EclipseARM\Repo\restaurarROM.bat

Adición de Bibliotecas C

1. Si fuera necesario, pueden utilizarse funciones de bibliotecas estándar modificando las propiedades de compilación del proyecto:

File → Properties C/C++ Build → Settings		
Libraries +	с	
Libraries +	gcc	
Libraries +	nosys	
Library search path +	\${eclipse_home}//sourcery-g++-lite-arm-2011.03\arm-none-eabi\lib	
Library search path +	\${eclipse_home}//sourcery-g++-lite-arm-2011.03\lib\gcc\arm-none-	
	eabi\4.5.2	
Apply $\rightarrow OK$		