

Programación en Linux embebido

SBC Intel Galileo

Gonzalo F. Pérez Paina



Universidad Tecnológica Nacional
Facultad Regional Córdoba
UTN-FRC

– 2017 –

SBC Intel Galileo

Descripción general

Propósito de la SBC (Single Board Computer) Intel Galileo:

1. Enfoque Arduino: manipulación sencilla del Hw
2. Enfoque Linux: potencia de un SO completo

SBC Intel Galileo

Descripción general

Propósito de la SBC (Single Board Computer) Intel Galileo:

1. Enfoque Arduino: manipulación sencilla del Hw
2. Enfoque Linux: potencia de un SO completo

Como placa Arduino

- ▶ La mayoría de los *sketch* escritos para *Arduino UNO*, *Leonardo*, y otras placas se pueden portar directamente a la placa Galileo.
- ▶ Se tiene acceso a las bibliotecas Arduino tales como SD, Ethernet, WiFi, EEPROM, SPI y Wire, pero también se puede acceder a SO Linux de la placa mediante llamadas a `system()`

SBC Intel Galileo

Descripción general

Propósito de la SBC (Single Board Computer) Intel Galileo:

1. Enfoque Arduino: manipulación sencilla del Hw
2. Enfoque Linux: potencia de un SO completo

Como placa Arduino

- ▶ La mayoría de los *sketch* escritos para *Arduino UNO*, *Leonardo*, y otras placas se pueden portar directamente a la placa Galileo.
- ▶ Se tiene acceso a las bibliotecas Arduino tales como SD, Ethernet, WiFi, EEPROM, SPI y Wire, pero también se puede acceder a SO Linux de la placa mediante llamadas a `system()`

Formas de interactuar con la terminal Linux

Es posible interactuar con el SO utilizando la terminal, ejecutando comandos en una interfaz de línea de comandos

1. puerto USB (necesita un sketch Arduino)
2. puerto RS232 del jack estéreo de 3.5mm (necesita cable especial)

SBC Intel Galileo

Descripción general

Tiene un *SoC Quark X1000* de arquitectura x86, mientras que la mayoría de las demás SBC tienen procesadores ARM (Raspberry Pi, Beaglebone Black, etc.).

SBC Intel Galileo

Descripción general

Tiene un *SoC Quark X1000* de arquitectura x86, mientras que la mayoría de las demás SBC tienen procesadores ARM (Raspberry Pi, Beaglebone Black, etc.).

Diferencia fundamental entre la arquitectura ARM y x86: conjunto de instrucciones.

SBC Intel Galileo

Descripción general

Tiene un *SoC Quark X1000* de arquitectura x86, mientras que la mayoría de las demás SBC tienen procesadores ARM (Raspberry Pi, Beaglebone Black, etc.).

Diferencia fundamental entre la arquitectura ARM y x86: conjunto de instrucciones.

- ▶ ARM es de arquitectura **RISC** (Reduced Instruction Set Computer): pequeño conjunto de instrucciones simples
- ▶ x86 es de arquitectura **CISC** (Complex Instruction Set Computer): conjunto de intrucciones potentes y más complejo

SBC Intel Galileo

Descripción general

Tiene un *SoC Quark X1000* de arquitectura x86, mientras que la mayoría de las demás SBC tienen procesadores ARM (Raspberry Pi, Beaglebone Black, etc.).

Diferencia fundamental entre la arquitectura ARM y x86: conjunto de instrucciones.

- ▶ ARM es de arquitectura **RISC** (Reduced Instruction Set Computer): pequeño conjunto de instrucciones simples
- ▶ x86 es de arquitectura **CISC** (Complex Instruction Set Computer): conjunto de intrucciones potentes y más complejo

Otras diferencias:

- ▶ Algo que en x86 lleva un ciclo de reloj, puede tomar hasta tres en un procesador ARM.
- ▶ Un conjunto de instrucciones más grande requiere más Hw, lo cual requiere mayor potencia.

SBC Intel Galileo

Descripción general

En este curso...

Se utiliza la SBC Intel Galileo como sistema embebido Linux

SBC Intel Galileo

Descripción general

En este curso...

Se utiliza la SBC Intel Galileo como sistema embebido Linux

Material necesario

- ▶ Placa Intel Galileo
- ▶ Cable USB de celular (micro USB)
- ▶ Tarjeta mini-SD, capacidad > 2GB
- ▶ Cable ethernet
- ▶ Opcional: cable plug serial + adaptador USB-serie

SBC Intel Galileo

Descripción general

En este curso...

Se utiliza la SBC Intel Galileo como sistema embebido Linux

Material necesario

- ▶ Placa Intel Galileo
- ▶ Cable USB de celular (micro USB)
- ▶ Tarjeta mini-SD, capacidad > 2GB
- ▶ Cable ethernet
- ▶ Opcional: cable plug serial + adaptador USB-serie

¡Cuidado!

Siempre conectar el cable de alimentación antes que el USB cliente

SBC Intel Galileo

Contenido del curso

- ▶ Introducción a GNU/Linux y al shell

SBC Intel Galileo

Contenido del curso

- ▶ Introducción a GNU/Linux y al shell
 - ▶ Comandos útiles, sistemas de archivos, variables de entorno, soft links, permisos, y procesos. Scripts de bash.

SBC Intel Galileo

Contenido del curso

- ▶ Introducción a GNU/Linux y al shell
 - ▶ Comandos útiles, sistemas de archivos, variables de entorno, soft links, permisos, y procesos. Scripts de bash.
- ▶ Introducción al Linux embebido

SBC Intel Galileo

Contenido del curso

- ▶ Introducción a GNU/Linux y al shell
 - ▶ Comandos útiles, sistemas de archivos, variables de entorno, soft links, permisos, y procesos. Scripts de bash.
- ▶ Introducción al Linux embebido
 - ▶ Construcción de Linux embebido, etapas, y herramientas

SBC Intel Galileo

Contenido del curso

- ▶ Introducción a GNU/Linux y al shell
 - ▶ Comandos útiles, sistemas de archivos, variables de entorno, soft links, permisos, y procesos. Scripts de bash.
- ▶ Introducción al Linux embebido
 - ▶ Construcción de Linux embebido, etapas, y herramientas
- ▶ Placa Intel Galileo

SBC Intel Galileo

Contenido del curso

- ▶ Introducción a GNU/Linux y al shell
 - ▶ Comandos útiles, sistemas de archivos, variables de entorno, soft links, permisos, y procesos. Scripts de bash.
- ▶ Introducción al Linux embebido
 - ▶ Construcción de Linux embebido, etapas, y herramientas
- ▶ Placa Intel Galileo
 - ▶ Descripción y puesta en funcionamiento, Linux SPI, instalación de Linux en memoria SD. Conexión PC y Galileo (serie, telnet, ssh). Linux embebido en Galileo.

SBC Intel Galileo

Contenido del curso

- ▶ Introducción a GNU/Linux y al shell
 - ▶ Comandos útiles, sistemas de archivos, variables de entorno, soft links, permisos, y procesos. Scripts de bash.
- ▶ Introducción al Linux embebido
 - ▶ Construcción de Linux embebido, etapas, y herramientas
- ▶ Placa Intel Galileo
 - ▶ Descripción y puesta en funcionamiento, Linux SPI, instalación de Linux en memoria SD. Conexión PC y Galileo (serie, telnet, ssh). Linux embebido en Galileo.
- ▶ Programación en lenguaje C sobre Intel Galileo

SBC Intel Galileo

Contenido del curso

- ▶ Introducción a GNU/Linux y al shell
 - ▶ Comandos útiles, sistemas de archivos, variables de entorno, soft links, permisos, y procesos. Scripts de bash.
- ▶ Introducción al Linux embebido
 - ▶ Construcción de Linux embebido, etapas, y herramientas
- ▶ Placa Intel Galileo
 - ▶ Descripción y puesta en funcionamiento, Linux SPI, instalación de Linux en memoria SD. Conexión PC y Galileo (serie, telnet, ssh). Linux embebido en Galileo.
- ▶ Programación en lenguaje C sobre Intel Galileo
 - ▶ Etapas de compilación (gcc), compilación cruzada, main con parámetros (parser), acceso al hardware/puertos disponibles en la placa. Programación de drivers.

SBC Intel Galileo

Contenido del curso

- ▶ Introducción a GNU/Linux y al shell
 - ▶ Comandos útiles, sistemas de archivos, variables de entorno, soft links, permisos, y procesos. Scripts de bash.
- ▶ Introducción al Linux embebido
 - ▶ Construcción de Linux embebido, etapas, y herramientas
- ▶ Placa Intel Galileo
 - ▶ Descripción y puesta en funcionamiento, Linux SPI, instalación de Linux en memoria SD. Conexión PC y Galileo (serie, telnet, ssh). Linux embebido en Galileo.
- ▶ Programación en lenguaje C sobre Intel Galileo
 - ▶ Etapas de compilación (gcc), compilación cruzada, main con parámetros (parser), acceso al hardware/puertos disponibles en la placa. Programación de drivers.
- ▶ Construcción de proyectos

SBC Intel Galileo

Contenido del curso

- ▶ Introducción a GNU/Linux y al shell
 - ▶ Comandos útiles, sistemas de archivos, variables de entorno, soft links, permisos, y procesos. Scripts de bash.
- ▶ Introducción al Linux embebido
 - ▶ Construcción de Linux embebido, etapas, y herramientas
- ▶ Placa Intel Galileo
 - ▶ Descripción y puesta en funcionamiento, Linux SPI, instalación de Linux en memoria SD. Conexión PC y Galileo (serie, telnet, ssh). Linux embebido en Galileo.
- ▶ Programación en lenguaje C sobre Intel Galileo
 - ▶ Etapas de compilación (gcc), compilación cruzada, main con parámetros (parser), acceso al hardware/puertos disponibles en la placa. Programación de drivers.
- ▶ Construcción de proyectos
 - ▶ Make y archivo makefile.

SBC Intel Galileo

Contenido del curso

- ▶ Introducción a GNU/Linux y al shell
 - ▶ Comandos útiles, sistemas de archivos, variables de entorno, soft links, permisos, y procesos. Scripts de bash.
- ▶ Introducción al Linux embebido
 - ▶ Construcción de Linux embebido, etapas, y herramientas
- ▶ Placa Intel Galileo
 - ▶ Descripción y puesta en funcionamiento, Linux SPI, instalación de Linux en memoria SD. Conexión PC y Galileo (serie, telnet, ssh). Linux embebido en Galileo.
- ▶ Programación en lenguaje C sobre Intel Galileo
 - ▶ Etapas de compilación (gcc), compilación cruzada, main con parámetros (parser), acceso al hardware/puertos disponibles en la placa. Programación de drivers.
- ▶ Construcción de proyectos
 - ▶ Make y archivo makefile.
- ▶ Programación de puerto serie

SBC Intel Galileo

Contenido del curso

- ▶ Introducción a GNU/Linux y al shell
 - ▶ Comandos útiles, sistemas de archivos, variables de entorno, soft links, permisos, y procesos. Scripts de bash.
- ▶ Introducción al Linux embebido
 - ▶ Construcción de Linux embebido, etapas, y herramientas
- ▶ Placa Intel Galileo
 - ▶ Descripción y puesta en funcionamiento, Linux SPI, instalación de Linux en memoria SD. Conexión PC y Galileo (serie, telnet, ssh). Linux embebido en Galileo.
- ▶ Programación en lenguaje C sobre Intel Galileo
 - ▶ Etapas de compilación (gcc), compilación cruzada, main con parámetros (parser), acceso al hardware/puertos disponibles en la placa. Programación de drivers.
- ▶ Construcción de proyectos
 - ▶ Make y archivo makefile.
- ▶ Programación de puerto serie
 - ▶ Dispositivos en linux. Manejo de archivos. Comunicación PC y Galileo. Terminales.

SBC Intel Galileo

SBC Intel Galileo – primeros pasos

- ▶ Instalar software Arduino (compatible Galileo) [arduino-1.8.2], y el soporte para la Intel Galileo

SBC Intel Galileo

SBC Intel Galileo – primeros pasos

- ▶ Instalar software Arduino (compatible Galileo) [arduino-1.8.2], y el soporte para la Intel Galileo
- ▶ Conectar fuente de alimentación (Galileo sin SD)

SBC Intel Galileo

SBC Intel Galileo – primeros pasos

- ▶ Instalar software Arduino (compatible Galileo) [arduino-1.8.2], y el soporte para la Intel Galileo
- ▶ Conectar fuente de alimentación (Galileo sin SD)
- ▶ Ejecutar IDE Arduino. Probar ejemplo **Blink**

SBC Intel Galileo

SBC Intel Galileo – primeros pasos

- ▶ Instalar software Arduino (compatible Galileo) [arduino-1.8.2], y el soporte para la Intel Galileo
- ▶ Conectar fuente de alimentación (Galileo sin SD)
- ▶ Ejecutar IDE Arduino. Probar ejemplo **Blink**
- ▶ Grabar imagen de Linux en SD

SBC Intel Galileo

SBC Intel Galileo – primeros pasos

- ▶ Instalar software Arduino (compatible Galileo) [arduino-1.8.2], y el soporte para la Intel Galileo
- ▶ Conectar fuente de alimentación (Galileo sin SD)
- ▶ Ejecutar IDE Arduino. Probar ejemplo **Blink**
- ▶ Grabar imagen de Linux en SD

Conexión Intel Galileo con PC host

- ▶ Cable serial: necesita plug serial + adaptador USB/serie

SBC Intel Galileo

SBC Intel Galileo – primeros pasos

- ▶ Instalar software Arduino (compatible Galileo) [`arduino-1.8.2`], y el soporte para la Intel Galileo
- ▶ Conectar fuente de alimentación (Galileo sin SD)
- ▶ Ejecutar IDE Arduino. Probar ejemplo `Blink`
- ▶ Grabar imagen de Linux en SD

Conexión Intel Galileo con PC host

- ▶ Cable serial: necesita plug serial + adaptador USB/serie
- ▶ Cable USB (mini USB) (sketch Arduino `Linux_USBShell.ino` y `Poor_Mans_Telnet.ino`)

SBC Intel Galileo

SBC Intel Galileo – primeros pasos

- ▶ Instalar software Arduino (compatible Galileo) [`arduino-1.8.2`], y el soporte para la Intel Galileo
- ▶ Conectar fuente de alimentación (Galileo sin SD)
- ▶ Ejecutar IDE Arduino. Probar ejemplo `Blink`
- ▶ Grabar imagen de Linux en SD

Conexión Intel Galileo con PC host

- ▶ Cable serial: necesita plug serial + adaptador USB/serie
- ▶ Cable USB (mini USB) (sketch Arduino `Linux_USBShell.ino` y `Poor_Mans_Telnet.ino`)
- ▶ [SPI Linux](#) (sin SSH) conexión por Telnet: sketch Arduino `Linux_TelnetShell`

SBC Intel Galileo

SBC Intel Galileo – primeros pasos

- ▶ Instalar software Arduino (compatible Galileo) [`arduino-1.8.2`], y el soporte para la Intel Galileo
- ▶ Conectar fuente de alimentación (Galileo sin SD)
- ▶ Ejecutar IDE Arduino. Probar ejemplo `Blink`
- ▶ Grabar imagen de Linux en SD

Conexión Intel Galileo con PC host

- ▶ Cable serial: necesita plug serial + adaptador USB/serie
- ▶ Cable USB (mini USB) (sketch Arduino `Linux_USBShell.ino` y `Poor_Mans_Telnet.ino`)
- ▶ **SPI Linux** (sin SSH) conexión por Telnet: sketch Arduino `Linux_TelnetShell`
- ▶ **SD Linux** conexión por `ssh`: necesita configurar la interfaz de red
 1. sketch Arduino para ver dirección IP
 2. terminal serie (`ifconfig`)

SBC Intel Galileo

Linux embebido en Galileo

- ▶ Toolchain
- ▶ Bootloader
- ▶ Kernel
- ▶ Root Filesystem

SBC Intel Galileo

Linux embebido en Galileo – Toolchain

- ▶ La placa Intel Galileo corre una distribución Linux Poky
- ▶ El SoC es un procesador compatible con el set de instrucciones ISA de 32-bits de Pentium, sin soporte de la clase de instrucciones SSE
- ▶ Los binarios tiene que ser construidos/cross-compilados con los flags apropiados para el target SoC Quark

SBC Intel Galileo

Linux embebido en Galileo – Toolchain

- ▶ La placa Intel Galileo corre una distribución Linux Poky
- ▶ El SoC es un procesador compatible con el set de instrucciones ISA de 32-bits de Pentium, sin soporte de la clase de instrucciones SSE
- ▶ Los binarios tiene que ser construidos/cross-compilados con los flags apropiados para el target SoC Quark

Toolchain de Intel

- ▶ compilador: `i586-poky-linux-uclibc-gcc`

SBC Intel Galileo

Linux embebido en Galileo – Toolchain

- ▶ La placa Intel Galileo corre una distribución Linux Poky
- ▶ El SoC es un procesador compatible con el set de instrucciones ISA de 32-bits de Pentium, sin soporte de la clase de instrucciones SSE
- ▶ Los binarios tiene que ser construidos/cross-compilados con los flags apropiados para el target SoC Quark

Toolchain de Intel

- ▶ compilador: `i586-poky-linux-uclibc-gcc`
- ▶ Biblioteca: `uClibc`

SBC Intel Galileo

Linux embebido en Galileo – Toolchain

- ▶ La placa Intel Galileo corre una distribución Linux Poky
- ▶ El SoC es un procesador compatible con el set de instrucciones ISA de 32-bits de Pentium, sin soporte de la clase de instrucciones SSE
- ▶ Los binarios tiene que ser construidos/cross-compilados con los flags apropiados para el target SoC Quark

Toolchain de Intel

- ▶ compilador: `i586-poky-linux-uclibc-gcc`
- ▶ Biblioteca: `uClibc`
- ▶ Sysroot (`$ gcc -print-sysroot`)

SBC Intel Galileo

Linux embebido en Galileo – Toolchain

- ▶ La placa Intel Galileo corre una distribución Linux Poky
- ▶ El SoC es un procesador compatible con el set de instrucciones ISA de 32-bits de Pentium, sin soporte de la clase de instrucciones SSE
- ▶ Los binarios tiene que ser construidos/cross-compilados con los flags apropiados para el target SoC Quark

Toolchain de Intel

- ▶ compilador: `i586-poky-linux-uclibc-gcc`
- ▶ Biblioteca: `uClibc`
- ▶ Sysroot (`$ gcc -print-sysroot`)
- ▶ Bibliotecas dinámicas utilizadas (`$ readelf -a app_bin | grep ...`)

SBC Intel Galileo

Linux embebido en Galileo – Toolchain

- ▶ La placa Intel Galileo corre una distribución Linux Poky
- ▶ El SoC es un procesador compatible con el set de instrucciones ISA de 32-bits de Pentium, sin soporte de la clase de instrucciones SSE
- ▶ Los binarios tiene que ser construidos/cross-compilados con los flags apropiados para el target SoC Quark

Toolchain de Intel

- ▶ compilador: `i586-poky-linux-uclibc-gcc`
- ▶ Biblioteca: `uClibc`
- ▶ Sysroot (`$ gcc -print-sysroot`)
- ▶ Bibliotecas dinámicas utilizadas (`$ readelf -a app_bin | grep ...`)
- ▶ Linker en tiempo de ejecución

SBC Intel Galileo

Linux embebido en Galileo – Bootloader, Kernel, y root filesystem

1. Bootloader: documentación de Intel

SBC Intel Galileo

Linux embebido en Galileo – Bootloader, Kernel, y root filesystem

1. Bootloader: documentación de Intel
2. Kernel: `$ uname -a`

SBC Intel Galileo

Linux embebido en Galileo – Bootloader, Kernel, y root filesystem

1. Bootloader: documentación de Intel
2. Kernel: `$ uname -a`
3. Root filesystem BusyBox

SBC Intel Galileo

Compilación cruzada

Toolchain instalado con el IDE arduino 1.8.2 (`i586-poky-linux-uclibc`)

SBC Intel Galileo

Compilación cruzada

Toolchain instalado con el IDE arduino 1.8.2 (i586-poky-linux-uclibc)

Buscar

- ▶ cross-compiler: `$ locate i586 | grep '\-gcc'`
- ▶ sysroot: `$ locate i586 | grep stdio.h`

SBC Intel Galileo

Compilación cruzada

Toolchain instalado con el IDE arduino 1.8.2 (i586-poky-linux-uclibc)

Buscar

- ▶ cross-compiler: `$ locate i586 | grep '\-gcc'`
- ▶ sysroot: `$ locate i586 | grep stdio.h`

Se puede configurar el entorno

- ▶ Agregando el path del toolchain a `PATH`
- ▶ Exportando una variable de entorno `SYSROOT`

SBC Intel Galileo

Compilación cruzada

Toolchain instalado con el IDE arduino 1.8.2 (i586-poky-linux-uclibc)

Buscar

- ▶ cross-compiler: `$ locate i586 | grep '\-gcc'`
- ▶ sysroot: `$ locate i586 | grep stdio.h`

Se puede configurar el entorno

- ▶ Agregando el path del toolchain a `PATH`
- ▶ Exportando una variable de entorno `SYSROOT`

`($ i586-poky-linux-uclibc-gcc --v)`

SBC Intel Galileo

Compilación cruzada

Toolchain instalado con el IDE arduino 1.8.2 (i586-poky-linux-uclibc)

Buscar

- ▶ cross-compiler: `$ locate i586 | grep '\-gcc'`
- ▶ sysroot: `$ locate i586 | grep stdio.h`

Se puede configurar el entorno

- ▶ Agregando el path del toolchain a `PATH`
- ▶ Exportando una variable de entorno `SYSROOT`

`($ i586-poky-linux-uclibc-gcc --v)`

Compilar ejemplo y luego

- ▶ `$ i586-poky-linux-uclibc-readelf -a <bin_file> | grep "Shared library"`
- ▶ `$ i586-poky-linux-uclibc-readelf -a <bin_file> | grep "program interpreter"`

SBC Intel Galileo – Acceso al Hw

–GPIO y ADC–

- ▶ Acceso a través de la interfaz de sistema de archivos **sysfs**
- ▶ Se maneja mediante E/S basado en archivos

SBC Intel Galileo – Acceso al Hw

–GPIO y ADC–

- ▶ Acceso a través de la interfaz de sistema de archivos **sysfs**
- ▶ Se maneja mediante E/S basado en archivos

Puertos de entrada/salida generales (GPIO)

1. Exportar los GPIO al **sysfs**
2. Configurar la dirección del GPIO
3. Configurar el driver del GPIO
4. Escribir y leer los valores
5. Des-exportar los GPIO del **sysfs**

SBC Intel Galileo – Acceso al Hw

–GPIO y ADC–

- ▶ Acceso a través de la interfaz de sistema de archivos **sysfs**
- ▶ Se maneja mediante E/S basado en archivos

Puertos de entrada/salida generales (GPIO)

1. Exportar los GPIO al **sysfs**
2. Configurar la dirección del GPIO
3. Configurar el driver del GPIO
4. Escribir y leer los valores
5. Des-exportar los GPIO del **sysfs**

Convertor analógico/digital (ADC)

1. Configurar mediante salida GPIO el uso del ADC (multiplexor)
2. Leer el valor del ADC

SBC Intel Galileo – Acceso al Hw

–GPIO y ADC–

- ▶ Acceso a través de la interfaz de sistema de archivos **sysfs**
- ▶ Se maneja mediante E/S basado en archivos

Puertos de entrada/salida generales (GPIO)

1. Exportar los GPIO al **sysfs**
2. Configurar la dirección del GPIO
3. Configurar el driver del GPIO
4. Escribir y leer los valores
5. Des-exportar los GPIO del **sysfs**

Convertor analógico/digital (ADC)

1. Configurar mediante salida GPIO el uso del ADC (multiplexor)
2. Leer el valor del ADC

[Intel Galileo - Programming GPIO From Linux \(Sergey's Blog\)](#)

SBC Intel Galileo – Acceso al Hw

Ejemplo con el LED onboard

Comandos desde la terminal

- ▶ Exportar el GPIO
`$ echo -n "3" > /sys/class/gpio/export`
- ▶ Configurar como salida
`$ echo -n "out" > /sys/class/gpio/gpio3/direction`
- ▶ Configurar el drive a **strong**
`$ echo -n "strong" > /sys/class/gpio/gpio3/drive`
- ▶ Prender el LED
`$ echo -n "1" > /sys/class/gpio/gpio3/value`
- ▶ Apagar el LED
`$ echo -n "0" > /sys/class/gpio/gpio3/value`
- ▶ Leer el valor de GPIO
`$ cat /sys/class/gpio/gpio3/value`

SBC Intel Galileo – Acceso al Hw

Ejemplo con el LED onboard

Comandos desde la terminal

- ▶ Exportar el GPIO
`$ echo -n "3" > /sys/class/gpio/export`
- ▶ Configurar como salida
`$ echo -n "out" > /sys/class/gpio/gpio3/direction`
- ▶ Configurar el drive a strong
`$ echo -n "strong" > /sys/class/gpio/gpio3/drive`
- ▶ Prender el LED
`$ echo -n "1" > /sys/class/gpio/gpio3/value`
- ▶ Apagar el LED
`$ echo -n "0" > /sys/class/gpio/gpio3/value`
- ▶ Leer el valor de GPIO
`$ cat /sys/class/gpio/gpio3/value`

Script para toogle: `toggle_gpio_onboard_led`

SBC Intel Galileo – Acceso al Hw

Ejemplo con entrada analógica

- ▶ Configurar la entrada analógica con el multiplexor
 - ▶ GPIO 37 como salida

```
$ echo -n "out" > /sys/class/gpio/gpio37/direction
```
 - ▶ GPIO 37 en 1

```
$ echo -n "0" > /sys/class/gpio/gpio37/value
```
- ▶ Leer el valor del ADC A0

```
$ cat /sys/bus/iio/devices/iio:device0/in_voltage0_raw
```

SBC Intel Galileo – Acceso al Hw

Ejemplo con entrada analógica

- ▶ Configurar la entrada analógica con el multiplexor
 - ▶ GPIO 37 como salida

```
$ echo -n "out" > /sys/class/gpio/gpio37/direction
```
 - ▶ GPIO 37 en 1

```
$ echo -n "0" > /sys/class/gpio/gpio37/value
```
- ▶ Leer el valor del ADC A0

```
$ cat /sys/bus/iio/devices/iio:device0/in_voltage0_raw
```

Script para leer el valor del ADC: `analog_a0`