

Técnicas Digitales II

Entrada / Salida

Entrada / Salida (I/O)

 Las entradas/salidas en un sistema, permite conectar el mismo con el exterior por medio de dispositivos denominados periféricos.

Periféricos en una PC

Teclados, Monitores, impresoras, redes cableadas o inalámbricas, etc.

Periféricos en un SE

Resistencias para calentar, motores, inyectores de combustibles, sensores de temperatura, etc.

E/S Mapeada en Memoria y E/S independiente

Son dos modelos diferentes de mapear los dispositivos de E/S

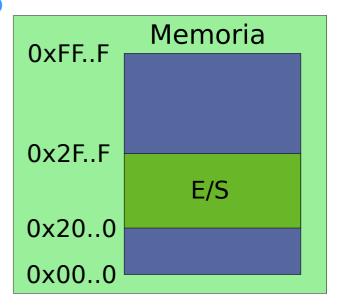
E/S Mapeada en Memoria

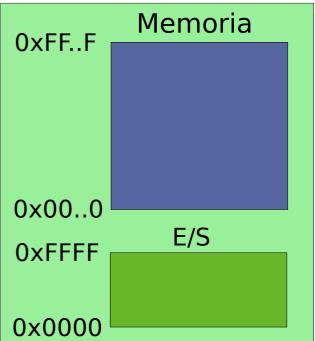
Este modelo utiliza el mismo bus de direcciones para memoria y para E/S. Ejemplo Arquitectura ARM.

E/S independiente

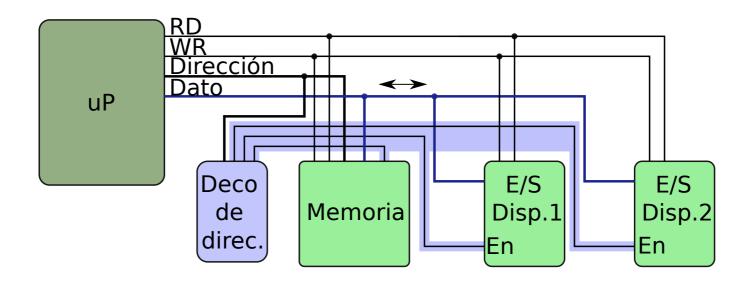
Utiliza un conjunto independiente de instrucciones para acceder al mapa de E/S.

Ejemplo Arquitectura x86.

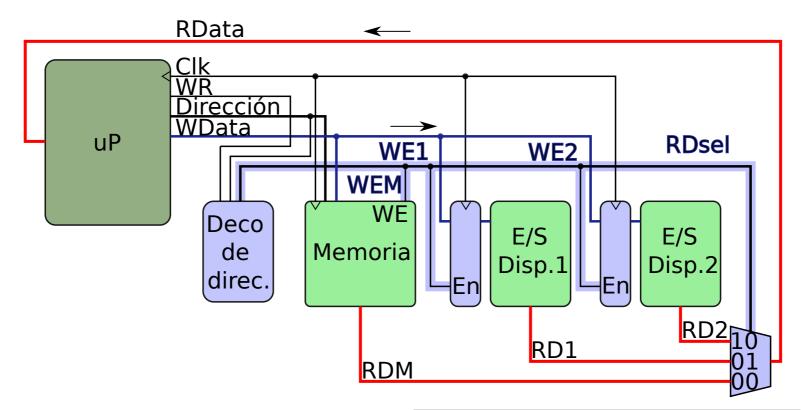




Mapeo con dispositivos preparados para bus



Mapeo con bus de escritura y lectura separados



Escritura

MOV R1,#7 LDR R2,=ioadr STR R1,[R2] .word 0x20001000

ioadr:

Lectura

LDR R2,=ioadr LDR R1,[R2] ioadr: .word 0x20001000

E/S Mapeada en Memoria

- Las direcciones asociadas con los dispositivos de E/S se llaman registros.
- El software que se comunica con los dispositivos de E/S se llaman controlador de dispositivo (device driver).

E/S en Sistemas Embebidos

- Los sistemas embebidos utilizan en general un procesador para interaccionar con el entorno físico.
- Esto se realiza con el uso de microcontroladores, ellos son una combinación de un microprocesador con un conjunto de periféricos:
 - Pines de E/S digitales y analógicos
 - Puertos seriales
 - Timers
 - Etc.

E/S en Sistemas Embebidos

- Estos microcontroladores pueden llegar a ser bastante económicos (de unos centavos a un par de dolares) y consumir unos pocos mW
- Bajan el costo y tamaño del sistema integrando la mayoría de los componentes necesarios en un solo chip.
- Pueden clasificarse según el tamaño de su ALU
 - 8 bits pequeños, básicos y menos costosos.
 - 32 bits mayor memoria memoria y mayor rendimientos.

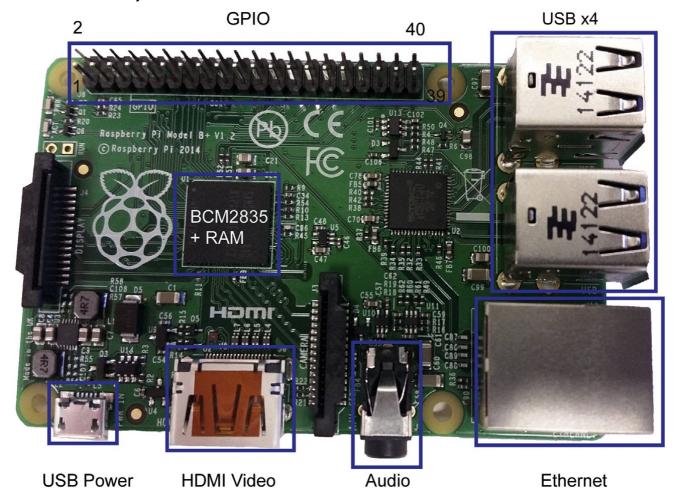
Hardware utilizado

Para explicar los diferentes dispositivos de E/S, utilizaremos un hardware en concreto

- Placa Raspberry Pi 1 B+ System-on-chip (SoC) Broadcom BCM2835 con un procesador de 32-bit ARM1176JZ-F a 700 MHz (Año 2012)
- Actualmente esta placa ya posee varias evoluciones
 - Placa Raspberry Pi 2 B System-on-chip (SoC) Broadcom BCM2836 con un procesador de 32-bit ARM Cortex A7 quadcore a 900 MHz
 - Placa Raspberry Pi 3 B System-on-chip (SoC) Broadcom BCM2837 con un procesador de 64-bit core ARMv8 quad-core 1.2GHz
 - Placa Raspberry Pi 3 B+ System-on-chip (SoC) Broadcom BCM2837 con un procesador de 64-bit core ARMv8 quad-core 1.4GHz

Hardware utilizado

 Placa Raspberry Pi 1 B+ System-on-chip (SoC) Broadcom BCM2835 con un procesador de 32-bit ARM1176JZ-F a 700 MHz (Año 2012)

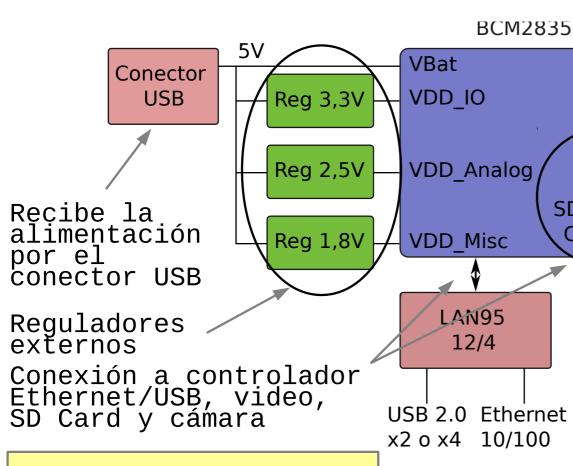


BCM2835

- El BCM2835 es un System on Chips (SoC), es decir integra un sistema completo formado por:
 - Un microprocesador ARM de la familia de los procesadores de aplicación
 - Un procesador VideoCore para video, gráficos y cámara
 - Una serie de periféricos como USB, GPIO, I2C, PWM UART, etc.

Un SoC integra dispositivos mas complejos que un microcontrolador junto con procesadores mas potentes y de arquitectura mas compleja

BCM2835 y Raspberry



Una SD con un
Raspbian Linux + HDMI
+ USB permiten
trabajar como un
computador completo

Una fracción de las 54 Conexiones de E/S disponibles

GPIO →

HDMI

SD Card

Cámara

RCA

Solo

B+

3,3V • • 5V SDA1/GPIO2 SCL1/GPIO3 ● GND GCLK/GPIO4 ● GPIO14/TXDO GND ● GPIO15/RXDO GPIO17 ● GPIO18/PWM0 GPIO27 GND GPIO22 • • GPIO23 3.3V ● GPIO24 MOSI/GPIO10 GND MISO/GPIO9 • GPIO25 SCLK/GPIO11 ● GPIO8/SPI CE0 GND ● GPIO7/SPI CE1 ID SD ● ID SC GPIO5 GND GPIO6 ● GPIO12 GPIO13 GND **GPIO19** • • GPIO16 GPIO26 • • GPIO20

GND ● GPIO21

Con.Audio

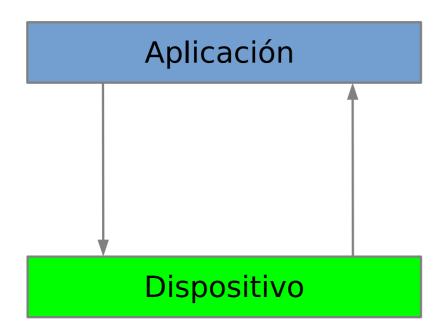
PWM0/GPIO40

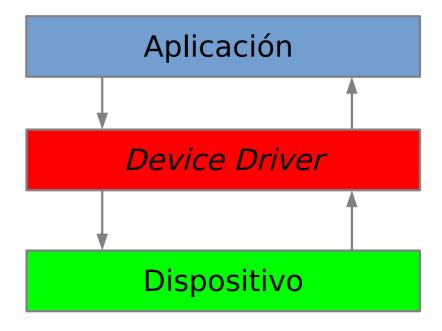
PWM1/GPIO45

GPIO47

Controlador de Dispositivo (device driver)

- Los dispositivos de E/S pueden ser accedidos directamente por sus registros.
- Sin embargo es de buena practica de programación, hacerlo a través de funciones llamadas controladores de dispositivos.





Controlador de Dispositivo (device driver)

Aplicación

Device Driver

Dispositivo

El Software que utiliza al dispositivo a través de los driver, no requiere conocer el funcionamiento del mismo, solo lo utiliza con funciones de alto nivel

El driver requiere de los detalles del funcionamiento del dispositivo, las direcciones de los registros y el comportamiento de cada uno de ellos

Controlador de Dispositivo (device driver)

Aplicación

Device Driver

Dispositivo

El uso de funciones en lugar de modificar bits en registros, convierte el código en mas legible

En caso de cambiar de hardware la sola reescritura de los driver es suficiente

Si el driver es parte de un SO, se puede gestionar acceso desde múltiples aplicaciones además de administrar la seguridad

Direcciones de E/S

- 0x20000000 0x20FFFFFF es el rango donde se encuentran los Dispositivos de E/S mapeados en memoria
- En general los dispositivos poseen varios registros que se encuentran direccionados a partir de una dirección base

Ejemplo GPIO

Dirección	Descripción
0x20200000	Dirección Base
0x20200034	Lectura de valores pines 31:00

Direcciones de E/S

Dirección	Base	Periféricos
0x20003000		System Timer
0x2000B200		Interrupts
0x2000B400		ARM Timer
0x20200000		GPIO
0x20201000		UART0
0x20203000		PCM Audio
0x20204000		SPI0
0x20205000		I2C Master #1
0x2020C000		PWM
0x20214000		I2C Slave
0x20215000		miniUART1, SPI1, SPI2
0x20300000		SD Card Controller
0x20804000		I2C Master #2
0x20805000		I2C Master #3

Diseño de un Driver - EasyPIO

Bibliografía

Harris & Harris. Digital design and computer architecture: ARM edition. Elsevier, 2015. Capítulo 9.

¿ Preguntas ?