

Práctica 2b: Sistema de E/S y dispositivos básicos

Proyecto Hardware

Índice

- **Objetivos y descripción de la práctica 3**
- Sistema de E/S de la placa LPC2105
- Ayudas código

Objetivo: Desarrollar código para un sistema empujado real

- En esta placa el procesador está acompañado de muchos dispositivos, principalmente de entrada/salida
- Vamos a aprender a interaccionar con ellos trabajando en C, pero empleando ensamblador cuando sea necesario
- Depurar un código con varias fuentes de interrupción activas
- Depurar un código en ejecución (no sólo paso a paso)

¿Qué tenemos que aprender?

- Entender la configuración de la placa:
 - Registros de configuración de los elementos utilizados
 - Acceso a los registros desde C y desde el entorno
- Gestión del hardware del sistema utilizando C:
 - Utilización de las bibliotecas de la placa
 - Gestión de las interrupciones en C
 - Entender las estructuras que genera el compilador a partir del código fuente (especialmente la pila de programa)

Descripción de la práctica

- Estudiar y aprender a usar:
 - controlador de interrupciones vectorizadas (VIC)
 - GPIO (General-Purpose I/O)
 - botones (emulados con Int. Externas)
 - dormir y a despertar al procesador (uso eficiente energía)
 - FSM – Maquinas de estados finitos
 - estructura modular

Reducir el consumo de energía

- Cuando el procesador no tiene trabajo se queda esperando en un bucle while.
- ¿Qué sentido tiene consumir energía cuando el procesador no tiene nada que hacer?
- En el 2020 el sector TIC fue responsable del 15% de las emisiones de CO₂!
- **Hay que mejorar la eficiencia energética de nuestros diseños**

Reducir el consumo de energía

- En lugar de tirar energía ejecutando un bucle de vamos a utilizar los modos de **bajo consumo** del procesador
 - **Objetivo:**
 - Reducir el consumo manteniendo el estado
 - Si no hay trabajo entramos en modo **idle** (el procesador para)
 - Si el juego no se usa durante mucho tiempo entramos en **power-down** (el procesador se duerme)

Descripción de la práctica

- Incluir la interacción con el usuario y la E/S
 - Múltiples alarmas con un reloj
 - Botones – Interrupciones externas
 - Bajo consumo
- Se reutilizará todo en la P3.
¡Es importante que todo sea modular!

Fechas de entrega

- Entrega
 - Aproximadamente semana 10
 - Los turnos de corrección aparecerán en Moodle

Material disponible

- En Moodle disponéis de:
 - Un proyecto ejemplo:
 - Documentación original de la placa

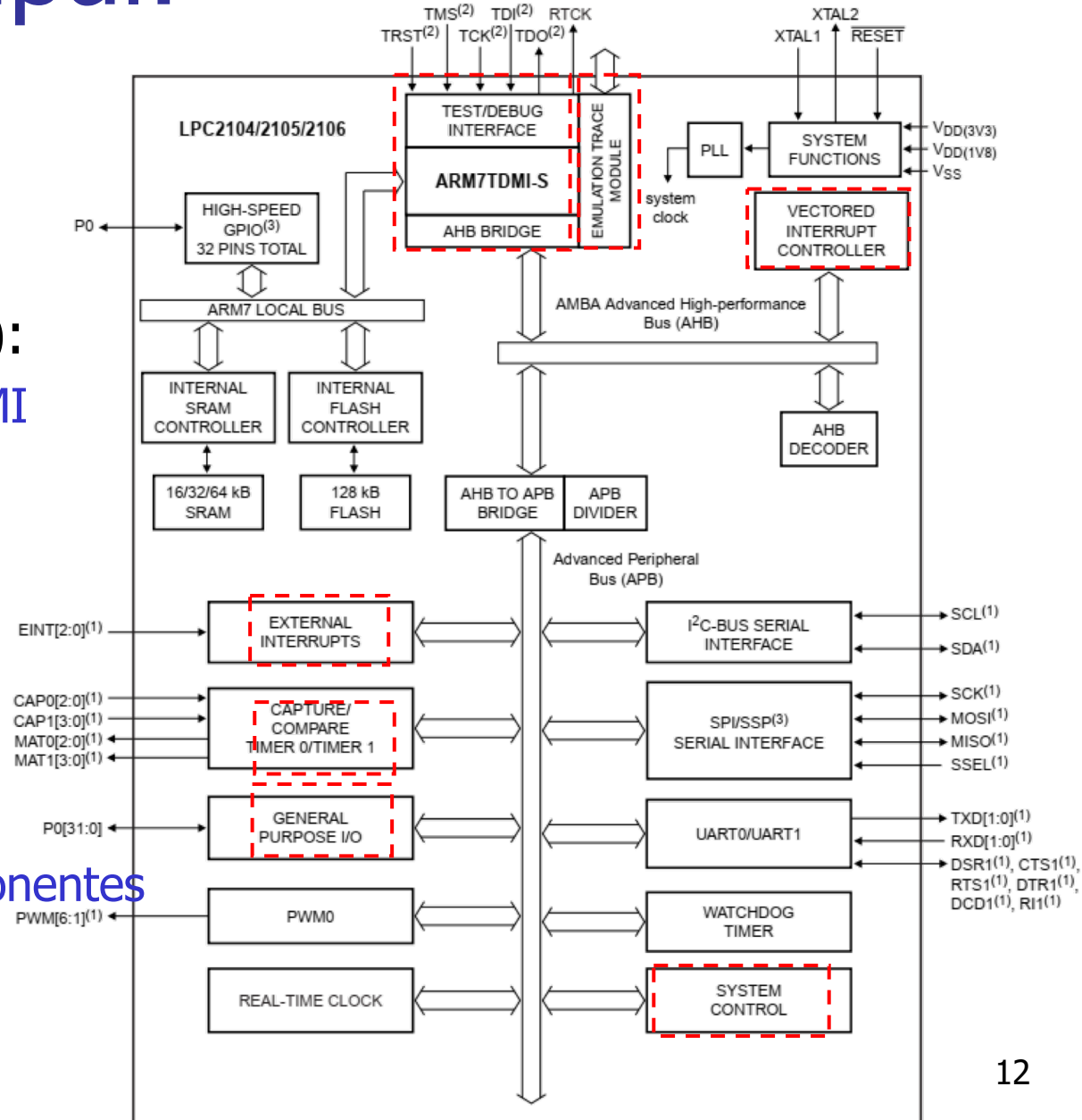
Índice

- Objetivos y descripción de la práctica
- **Sistema de E/S**
- Ayudas código

Chip principal: LPC2105

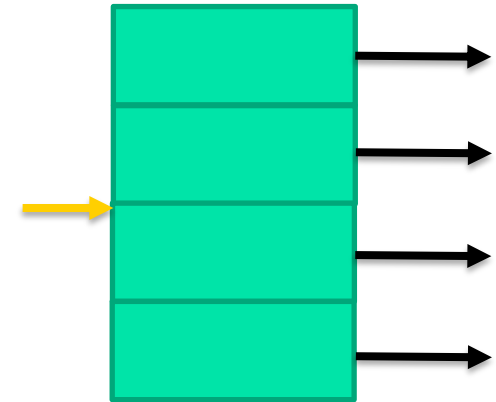
■ System-on-Chip (Soc):

- Procesador ARM7TDMI
- Controladores y E/S
 - GPIO
 - UART
 - Timers
 - Power control
- Controlador INT. Vectorizadas
- Y muchos más componentes



Alarmas

- Activación periódica
 - Un temporizador hardware
- Alarmas
 - Múltiples alarmas software vivas
 - **alarma_activar**, Acciones:
 - Activar, reprogramar, desactivar
 - Cada alarma debe guardar:
 - activa, ID_evento, auxData, retardo_ms, periódica
 - **alarma_tratar_evento**



GPIO

- 32 GPIO (1 puerto)
 - P0.0-P0.31
- Selección
- Dirección
- Bit-level set & clear

Table 54. Pin description ...continued

Symbol	Pin	Type	Description
P0.16/EINT0/MAT0.2	46	I/O	P0.16 — Port 0 bit 16.
		I	EINT0 — External interrupt 0 input.
		O	MAT0.2 — Match output for Timer 0, channel 2.

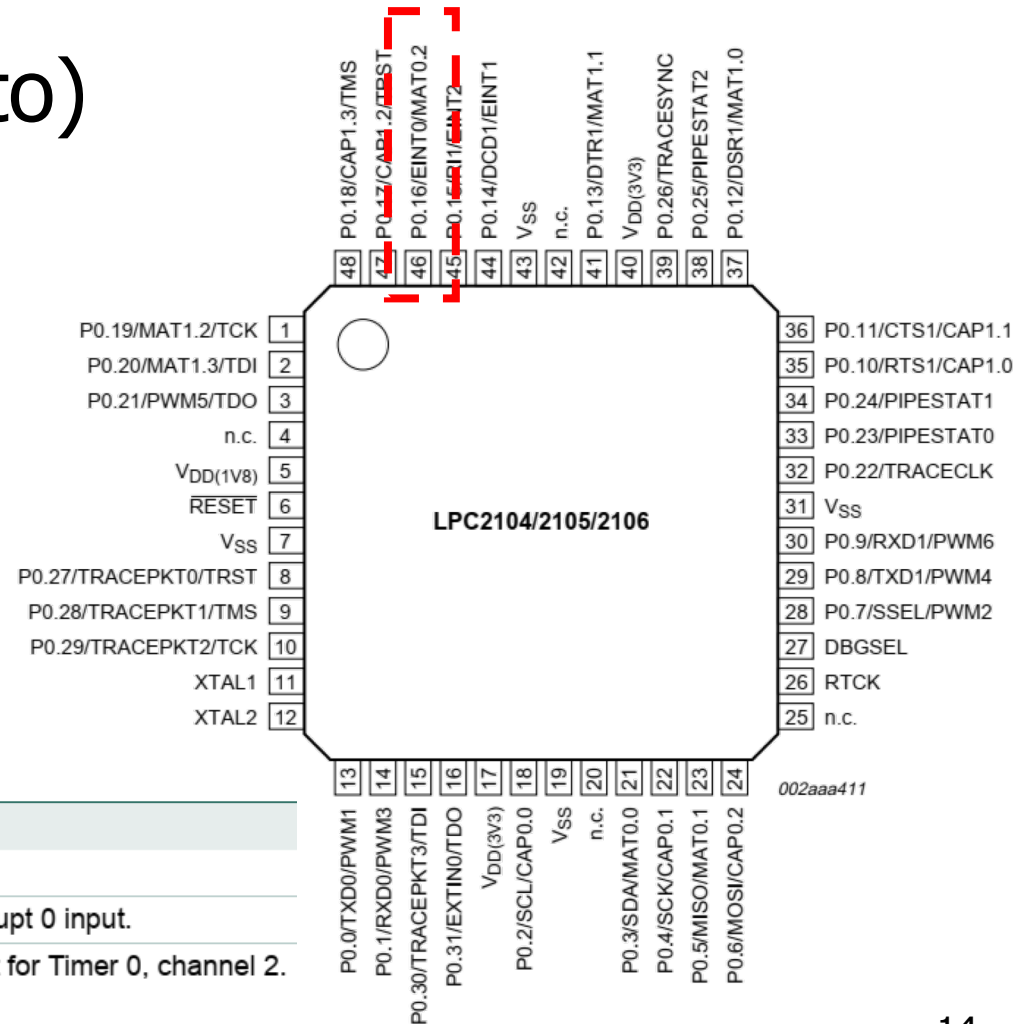


Fig 14. LPC2104/05/06 LQFP48 pin package

GPIO

- Asignar PINS a un dispositivo
 - cap 6 y 7.

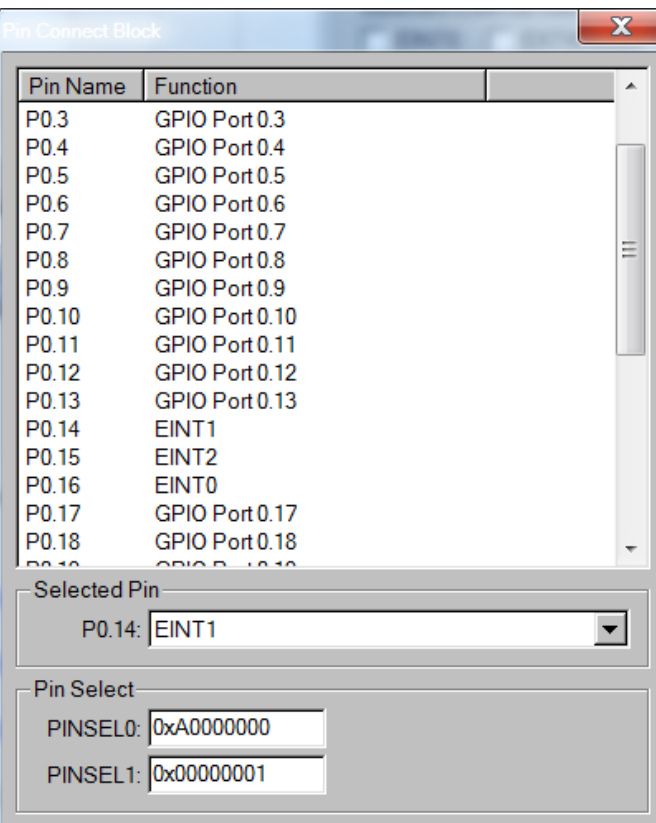


Table 55. Pin connect block register map

Name	Description	Access	Reset value ^[1]	Address
PINSEL0	Pin function select register 0.	Read/Write	0x0000 0000	0xE002 C000
PINSEL1	Pin function select register 1.	Read/Write	0x0000 0000	0xE002 C004

[1] Reset value reflects the data stored in used bits only. It does not include reserved bits content.

Table 57. Pin function select register 1 (PINSEL1 - 0xE002 C004)

PINSEL1	Pin Name	Value	Function	Value after reset
1:0	P0.16	0	0	GPIO Port 0.16
		0	1	EINT0
		1	0	Match 0.2 (Timer 0)

GPIO

- P0xDIR
- P0xVAL
- P0xSET
- P0xCLR

General Purpose Input/Output (GPIO)

GPIO

IODIR: 0x00000000

IOSET: 0x00000000

IOCLR: 0x00000000

IOPIN: 0xFFFFFFFF

Pins: 0xFFFFFFFF

31 Bits 24 23 Bits 16 15 Bits 8 7 Bits 0

Table 60. GPIO register map (legacy APB accessible registers)

Generic Name	Description	Access	Reset value ^[1]	PORT0 Address & Name
IOPIN	GPIO Port Pin value register. The current state of the GPIO configured port pins can always be read from this register, regardless of pin direction.	R/W	NA	0xE002 8000 IO0PIN
IOSET	GPIO Port Output Set register. This register controls the state of output pins in conjunction with the IOCLR register. Writing ones produces HIGHS at the corresponding port pins. Writing zeroes has no effect.	R/W	0x0000 0000	0xE002 8004 IO0SET
IODIR	GPIO Port Direction control register. This register individually controls the direction of each port pin.	R/W	0x0000 0000	0xE002 8008 IO0DIR
IOCLR	GPIO Port Output Clear register. This register controls the state of output pins. Writing ones produces LOW at the corresponding port pins and clears the corresponding bits in the IOSET register. Writing zeroes has no effect.	WO	0x0000 0000	0xE002 800C IO0CLR

[1] Reset value reflects the data stored in used bits only. It does not include reserved bits content.

Table 62. GPIO port 0 Direction register (IO0DIR - address 0xE002 8008) bit description

Bit	Symbol	Value	Description	Reset value
31:0	P0xDIR		Slow GPIO Direction control bits. Bit 0 controls P0.0 ... bit 30 controls P0.30.	0x0000 0000
		0	Controlled pin is input.	
		1	Controlled pin is output.	

EXT INT

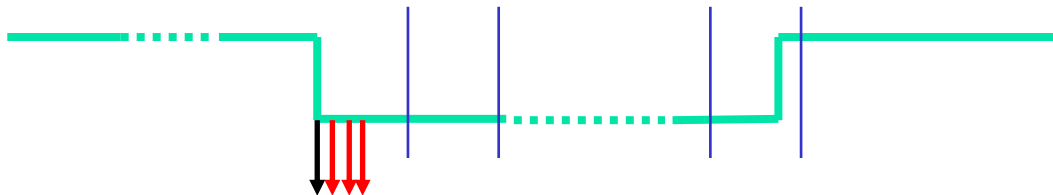
- Cap3, 6.1

Table 9. External interrupt registers

Name	Description	Access	Reset value ^[1]	Address
EXTINT	The External Interrupt Flag Register contains interrupt flags for EINT0, EINT1, EINT2. See Table 3-10 .	R/W	0	0xE01F C140
EXTWAKE	The External Interrupt Wakeup Register contains four enable bits that control whether each external interrupt will cause the processor to wake up from Power-down mode. See Table 3-11 .	R/W	0	0xE01F C144
EXTMODE	The External Interrupt Mode Register controls whether each pin is edge- or level sensitive.	R/W	0	0xE01F C148
EXTPOLAR	The External Interrupt Polarity Register controls which level or edge on each pin will cause an interrupt.	R/W	0	0xE01F C14C

[1] Reset value reflects the data stored in used bits only. It does not include reserved bits content.

- EXTMODE y EXTPOLAR no están en el sistema: las interrupciones se activan por nivel con un '0'



EXT INT

External Interrupts

Name	Int	Wake
EINT0	0	0
EINT1	0	0
EINT2	0	0

Selected External Interrupt

☐ EINT0 ☐ EXTWAKE0

EXTINT: 0x00

EXTWAKE: 0x00

Toolbox

Update Windows

1	Play
2	Pass
3	F0
4	F1
5	...
6	C7

Pin Connect Block

Pin Name	Function
P0.3	GPIO Port 0.3
P0.4	GPIO Port 0.4
P0.5	GPIO Port 0.5
P0.6	GPIO Port 0.6
P0.7	GPIO Port 0.7
P0.8	GPIO Port 0.8
P0.9	GPIO Port 0.9
P0.10	GPIO Port 0.10
P0.11	GPIO Port 0.11
P0.12	GPIO Port 0.12
P0.13	GPIO Port 0.13
P0.14	EINT1
P0.15	EINT2
P0.16	EINT0
P0.17	GPIO Port 0.17
P0.18	GPIO Port 0.18

Selected Pin

P0.14: EINT1

Pin Select

PINSEL0: 0xA0000000

PINSEL1: 0x00000001

EXT INT

- Cap3, 6.1
 - permiten despertar al procesador desde power-down

EXTMODE y EXTPOLAR no están en el sistema: las interrupciones se activan por nivel con un '0'

Table 9. External interrupt registers

Name	Description	Access	Reset value ^[1]	Address
EXTINT	The External Interrupt Flag Register contains interrupt flags for EINT0, EINT1, EINT2. See Table 3–10 .	R/W	0	0xE01F C140
EXTWAKE	The External Interrupt Wakeup Register contains four enable bits that control whether each external interrupt will cause the processor to wake up from Power-down mode. See Table 3–11 .	R/W	0	0xE01F C144
EXTMODE	The External Interrupt Mode Register controls whether each pin is edge- or level sensitive.	R/W	0	0xE01F C148
EXTPOLAR	The External Interrupt Polarity Register controls which level or edge on each pin will cause an interrupt.	R/W	0	0xE01F C14C

[1] Reset value reflects the data stored in used bits only. It does not include reserved bits content.

Power

■ 10.1

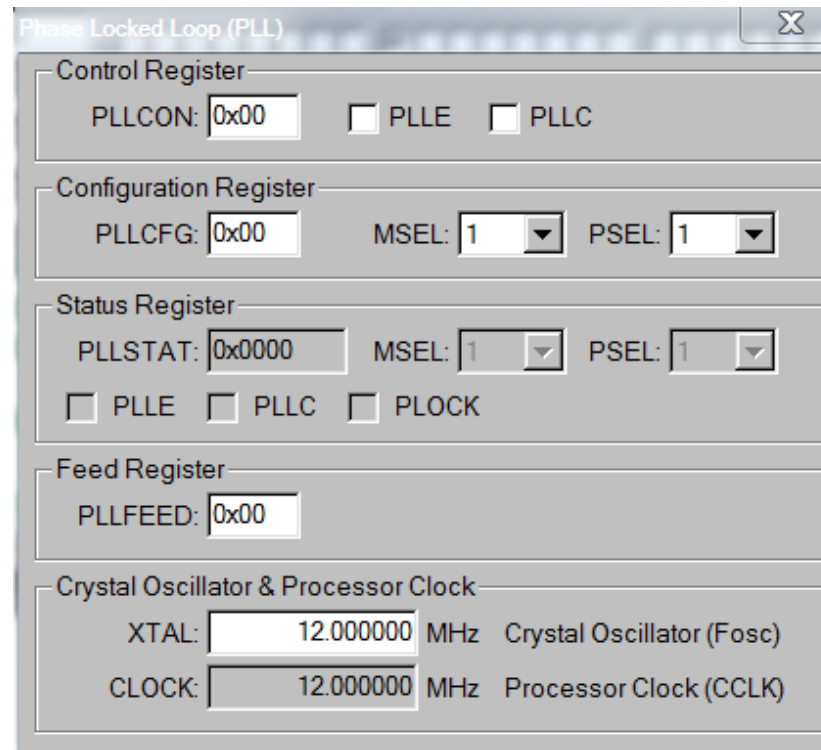
Table 25. Power control registers

Name	Description	Access	Reset value ^[1]	Address
PCON	Power Control Register. This register contains control bits that enable the two reduced power operating modes of the microcontroller. See Table 3–26 .	R/W	0x00	0xE01F C0C0
PCONP	Power Control for Peripherals Register. This register contains control bits that enable and disable individual peripheral functions, allowing elimination of power consumption by peripherals that are not needed.	R/W	0x0000 1FBE	0xE01F C0C4

[1] Reset value reflects the data stored in used bits only. It does not include reserved bits content.

PLL - Phase Locked Loop

- CCLK - CPU clock
 - 60 MHz startup
 - PowerDown lo vuelve a 12MHz
- PCLK – Peripheral Clock
 - CPU/VPB
 - VPB Divider (4)



The screenshot shows a software window titled "Phase Locked Loop (PLL)" with a close button (X) in the top right corner. The window is divided into several sections for configuring the PLL:

- Control Register:** Contains a text field for "PLLCON" with the value "0x00" and two checkboxes, "PLLE" and "PLL", both of which are currently unchecked.
- Configuration Register:** Contains a text field for "PLLCFG" with the value "0x00", and two dropdown menus for "MSEL" and "PSEL", both set to "1".
- Status Register:** Contains a text field for "PLLSTAT" with the value "0x0000", and two dropdown menus for "MSEL" and "PSEL", both set to "1". Below these are three checkboxes: "PLLE", "PLL", and "PLOCK", all of which are unchecked.
- Feed Register:** Contains a text field for "PLLFEED" with the value "0x00".
- Crystal Oscillator & Processor Clock:** Contains two rows of configuration:
 - The first row has a text field for "XTAL" with the value "12.000000" followed by "MHz" and the label "Crystal Oscillator (Fosc)".
 - The second row has a text field for "CLOCK" with the value "12.000000" followed by "MHz" and the label "Processor Clock (CCLK)".

Demostrador

Orientado a eventos

juego

hello_world

aplicación

Planificador

μkernel

visualizar

FIFO

alarmas

temporizador_drv

drivers

HAL

dependiente
del hardware

int_ext_hal

power_hal

gpio_hal

temporizador_hal

Índice

- Objetivos y descripción de la práctica
- Sistema de E/S
- **Ayudas código**

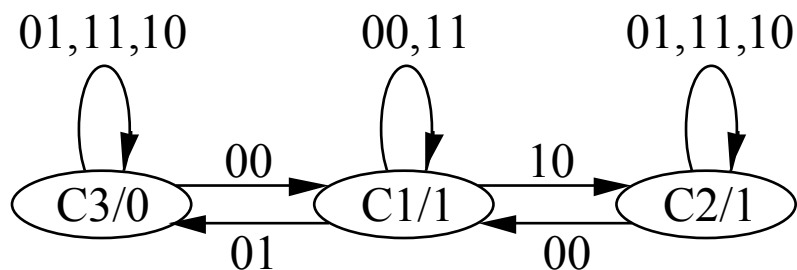
Autómatas. Implementación

- Ej.: Detección sentido contrario

- **MOORE**

- Entradas nivel muestreadas (síncronas)

- Salidas asíncronas

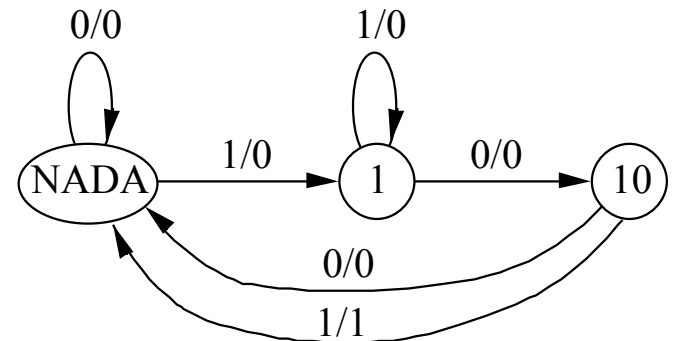


```
// Estado pertenece al conjunto de estados
Entrada = Leer_Entrada ();
switch (Estado)
{
    case C1 : Salida(NO_ALARMA) ;
        switch (Entrada) {
            case I01 : Estado = C3 ; break ;
            case I10 : Estado = C2 ; break ;
            default : }
        break ;
    case C2 : Salida (NO_ALARMA) ;
        if (Entrada == I00) Estado = C1 ;
        break ;
    case C3 : Salida (ALARMA) ;
        if (Entrada == I00) Estado = C1 ;
        break ;
}
```


Autómatas. Implementación

```
Espera_Sincronismo () ;  
Entrada = Leer_Bit () ;  
switch (Estado)  
{  
  case NADA : if (Entrada==0) {Salida=0; Estado=NADA;}  
              else if (Entrada==1) {Salida=0; Estado=E1;}  
              break ;  
  case E1 :   if (Entrada==0) {Salida=0; Estado=E10;}  
              else if (Entrada==1) {Salida=0; Estado=E1;}  
              break ;  
  case E10 :  if (Entrada==0) {Salida=0; Estado=NADA;}  
              else if (Entrada==1) {Salida=1; Estado=E101;}  
              break ;  
  case E101 : if (Entrada==0) {Salida=0; Estado=NADA;}  
              else if (Entrada==1) {Salida=0; Estado=E1;}  
              break ;  
}  
Genera (Salida) ;
```

Ej: reconocedor de cadenas: **MEALY**



*Podéis encontrar una forma más eficiente de implementar MSF en:
<http://johnsantic.com/comp/state.html>*