Entradas y Salidas Digitales

Sistemas con Microprocesadores

Ing. Esteban Volentini (evolentini@herrera.unt.edu.ar)

http://www.microprocesadores.unt.edu.ar/procesadores

Cronograma

Actividad	Inicio	Descripción	Fin
Presentación	19/08	Reglamento de la Materia	✔
Tema 1	19/08	Estructura de las computadoras	✔
Tema 2	26/08	Proyecto con un microcontrolador	✔
Tema 3	30/08	Descripción funcional de microprocesador	✔
Tema 4	13/09	Programación en lenguaje ensablador	✔
Tema 5	25/09	Descripción general de un microcontrolador	✔
Tema 6	27/09	Estructura general de microcontrolador	✔
Parcial	09/10	Primer examen parcial	✔
Tema 7	14/10	Sistema de Interrupciones	✔
Tema 8	21/10	Entradas y salidas digitales	←
Tema 9	28/10	Entrada/salida con perifericos	
Tema 10	06/11	Temporizadores	
Proyectos	25/11	Seminarios de Proyectos	
Parcial	04/12	Segundo examen parcial	

Entradas y salidas digitales (GPIO)

- Un microcontrolador incorpora diferentes tipos de dispositivos:
  - Entradas y salidas digitales
  - Entradas y salidas analógicas
  - Puertos de comunicaciones
  - Temporizadores
- Las entradas y salidas digitales suelen ser los dispositivos más usados y mas fáciles de utilizar.

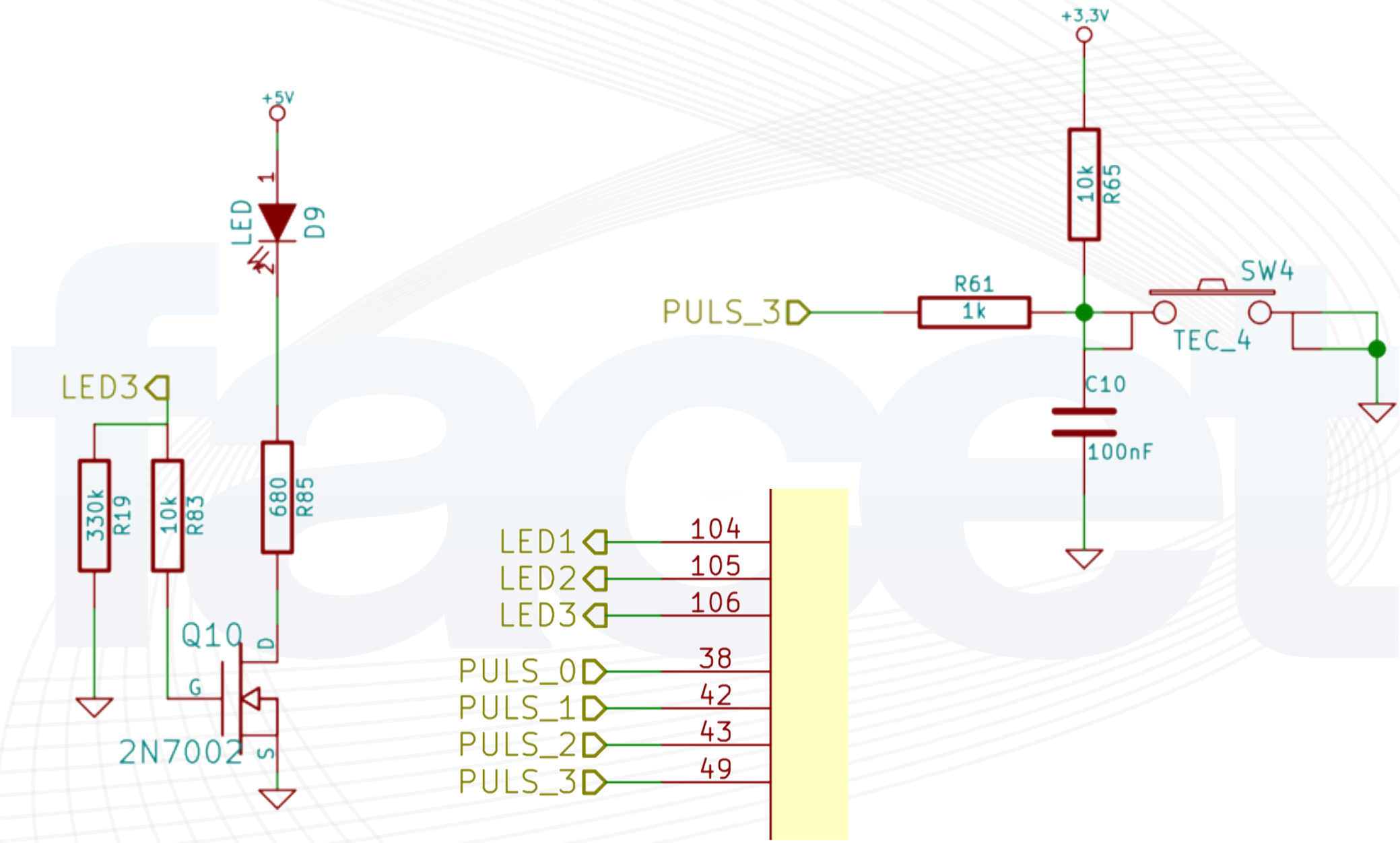
Muchas funciones en un solo terminal

- El costo de un chip depende mucho del área de silicio y de la cantidad de terminales
- Los fabricantes incluyen mas dispositivos que terminales
- Habitualmente un mismo terminal puede tener entre cuatro y ocho funciones diferentes
- En el LPC4337 los terminales físicos son totalmente independientes de los puertos GPIO.

Entradas/Salidas Digitales GPIO

- Primero se debe configurar los terminales que conforman el puerto y su conexión eléctrica.
  - Capítulo 17 Manual NXP LPC4337.
- Luego configurar las entradas/salidas digitales GPIO para su funcionamiento.
  - Capítulo 19 Manual NXP LPC4337.
  - En siguiente tema se verá configuración de sus interrupciones.

Ejemplo sobre la EDU-CIAA



Ejemplo sobre la EDU-CIAA

- Del diagrama esquemático de la placa
  - Tecla 4 conectado al terminal 49 y a masa
  - Led 3 conectado al terminal 106 y a masa
  - Procesador LPC4337 LQFP144
- Distribución de Terminales
  - Capitulo 16, Tabla 187

Información de los terminales

Pin name	LBGA256	TFBGA100	LQFP208	LQFP144		Reset state [1]	Type	Description
P2_12	E15	B9	153	106	[2]	N; PU	I/O	**GPIO1[12] —** General purpose digital input/output pin.
							O	**CTOUT_4 —** SCT output 4. Match output 3 of timer 3.
							-	**R —** Function reserved.
							I/O	**EMC_A3 —** External memory address line 3.
							-	**R —** Function reserved.
							-	**R —** Function reserved.
							I/O	**U2_UCLK —** Serial clock input/output for USART2 in synchronous mode.
P1_6	T4	K4	67	49	[2]	N; PU	I/O	**GPIO1[9] —** General purpose digital input/output pin.
							I	**CTIN_5 —** SCT input 5. Capture input 2 of timer 2.
							-	**R —** Function reserved.
							O	**$\overline{\text{EMC\_WE}}$ —** LOW active Write Enable signal.
							-	**R —** Function reserved.
							O	**$\overline{\text{EMC\_BLS0}}$ —** LOW active Byte Lane select signal 0.
							I/O	**SGPIO14 —** General purpose digital input/output pin.
							I/O	**SD_CMD —** SD/MMC command signal.

Ejemplo sobre la EDU-CIAA

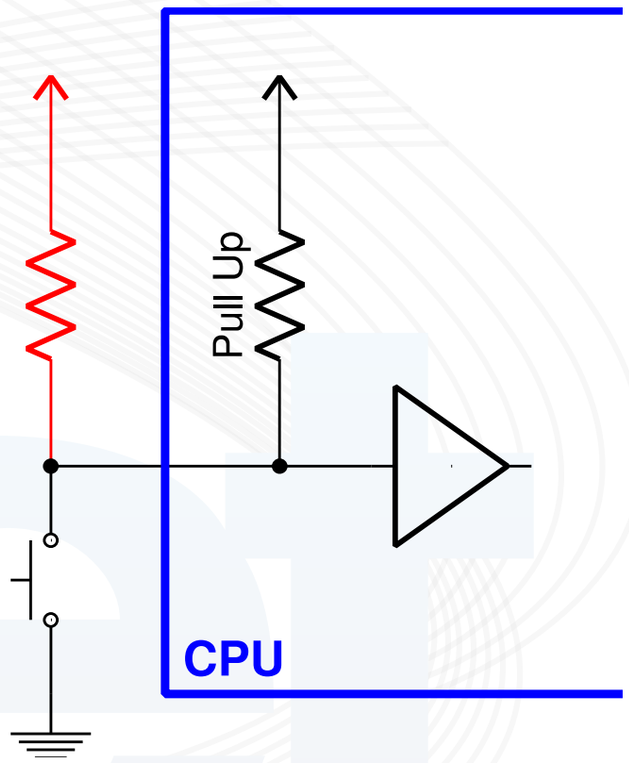
- Tecla 4 conectado al terminal 49
  - P1_6: Puerto 1, Terminal 6
  - En Función 0 es GPIO1[9]
  - Bit 9 del GPIO 1
- Led 3 conectado al terminal 106
  - P2_12: Puerto 2, Terminal 12
  - En Función 0 es GPIO1[12]
  - Bit 12 del GPIO 1

Configuración de los Terminales

- El comportamiento de cada terminal del microcontrolador se configura por medio de un registro que define la función y el comportamiento del mismo
- Existen dos tipos de registros para configurar el comportamiento de los pines
  - Terminales normales
  - Terminales con alta capacidad de corriente
- Los registros difieren según el tipo.

Resistencia de Pull Up y Pull Down

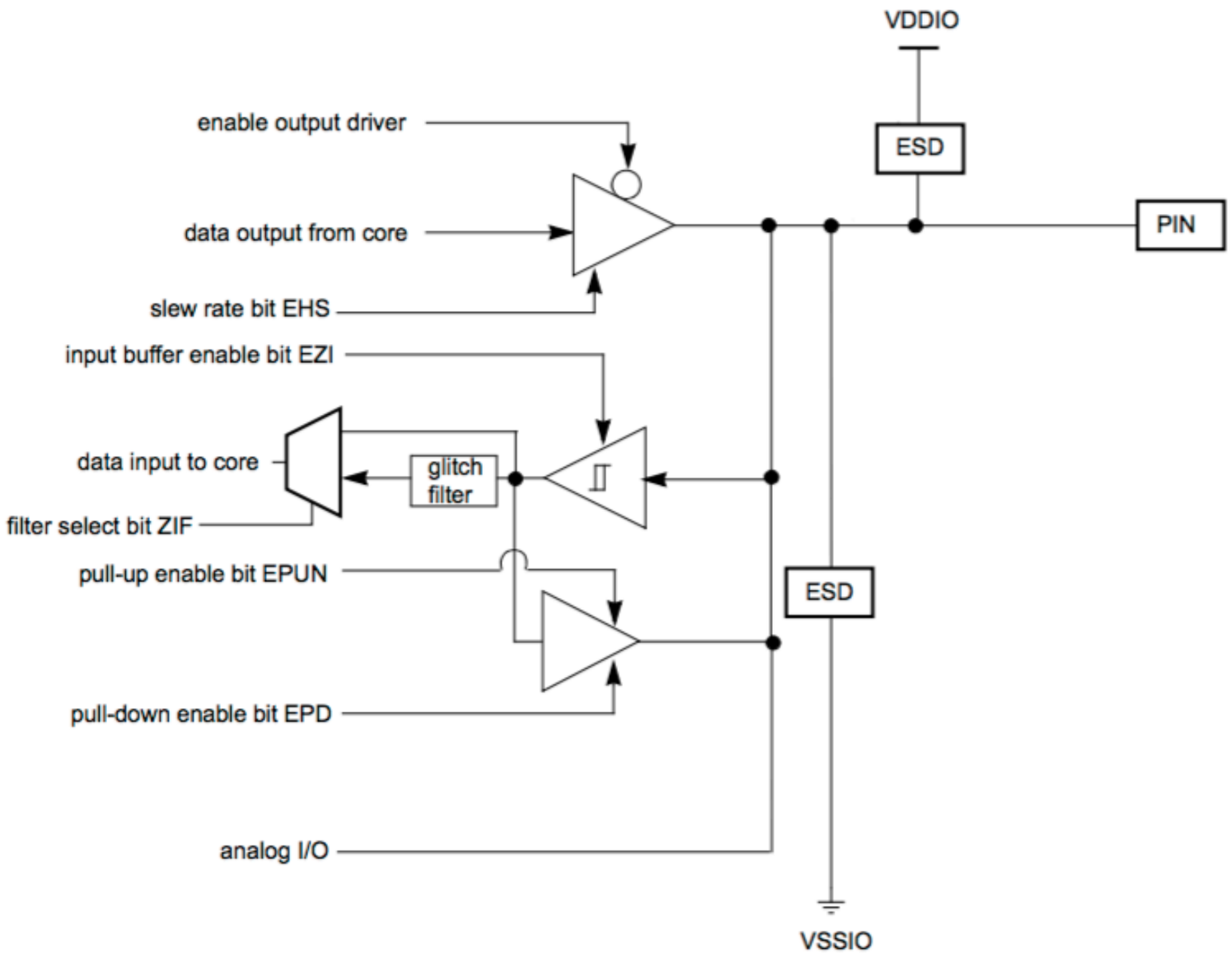
- Se usa **PU** para entradas open collector, generando un wired AND (lógica positiva).
- También se usa para tener una entrada con nivel de tensión alto.
  - Sirve por ejemplo para conectar un switch
  - Ahorra una R externa (en rojo).
- **PD** se usa para entrada con nivel bajo.
  - Sirve también para conectar un Switch y ahorrar una R externa.



Configuración de Terminales

- En reset quedan configurados
  - GPIO, entrada.
  - Pull up
  - Puerto de entrada, inhabilitado.
- Cada terminal tiene un registro de configuración asociado al mismo.
  - SFSP0_0 – para bit 0 de puerto 0.
  - SFSP1_0 – para bit 1 de puerto 0.
  - …
  - System Function Select Port_#

Conexión de una Terminal del MCU



Configuración Terminales: MODE

- Cada terminal puede configurarse para una función entre 8 posibles (tabla 189). Por ej:
  - GPIO pin
  - Timer
  - USART
  - …
- En Hw equivale a un MUX con un campo de control de 3 bits, llamado MODE.
- **Mode** es un campo en el reg. de configuración del pin.
- Slew rate no debe usarse en freq > 80 MHz.

Ejemplo sobre la EDU-CIAA

- Resumen de funciones (Tabla 189)

Pin	FUNC0	FUNC1	FUNC2	FUNC3	FUNC4	FUNC5	FUNC6	FUNC7	ANALOG SEL
P1_6	GPIO1[9]	CTIN_5	R	EMC_WE	R	R	SGPIO14	SD_CMD	
P2_12	GPIO1[12]	CTOUT_4	R	EMC_A3	R	R	R	U2_UCLK	

- Dirección y tipo de registro para cada terminal

Table 190. Register overview: System Control Unit (SCU) (base address 0x4008 6000)

Name	Access	Address offset	Description	Reset value	Reset value after EMC boot	Reset value after UART boot	Reference
Pins P1_n							
SFSP1_6	R/W	0x098	Pin configuration register for pin P1_6	0x00	0xD3	0x00	Table 191
Pins P2_n							
SFSP2_6	R/W	0x118	Pin configuration register for pin P2_6	0x00	0xD2	0x00	Table 191

Config. Terminales Normales (SFSPx_b)

31..8	7	6	5	4	3	2:0
Reservados	ZIF	EZI	EHS	EPUN	EPD	MODE

EPUN	Valor
Habilita la resistencia de pull-up	0
Inhabilita la resistencia de pull-up	1

EPD	Valor
Inhabilita la resistencia de pull-down	0
Habilita la resistencia de pull-down	1

MODE	Valor
Selecciona la función del pin	0x0 a 0x7

Configuración Terminales Normales

31..8	7	6	5	4	3	2:0
Reservados	ZIF	EZI	EHS	EPUN	EPD	MODE

Valor	EHS
0	Salida de pendiente media
1	Salida de alta velocidad

Valor	EZI
0	Entrada desconectada
1	Entrada conectada

Valor	ZIF
0	Filtro de ruido habilitado
1	Sin filtro de ruido

Terminales Alta capacidad Corriente

31..10	9:8	7	6	5	4	3	2:0
Reservados	EHD	ZIF	EZI	Reservado	EPUN	EPD	MODE

Valor	EHD
0x0	Salida normal (4 mA)
0x1	Salida corriente media (8 mA)
0x2	Salida corriente alta (14 mA)
0x3	Salida corriente muy alta (20 mA)

En estos pins no se puede usar slew rate.

Ejemplo sobre la EDU-CIAA

- Tecla 4
  - P1_6: Puerto 1, Terminal 6
  - Se configura mediante el registro SFSP1_6
  - En Función 0 es GPIO1[9] (MODE = 0)
  - Entrada con pull-up (EPUN = 0, EPD = 0)
  - Se puede leer la entrada (EZI = 1)
  - Entrada sin filtro de ruido (ZIF = 0)

31..8	7	6	5	4	3	2:0
Reservados	ZIF	EZI	EHS	EPUN	EPD	MODE
0	0	1	0	0	0	000

Ejemplo sobre la EDU-CIAA

- Led 3
  - P2_12: Puerto 2, Terminal 12
  - Se configura mediante el registro SFSP2_12
  - En Función 0 es GPIO1[12] (MODE = 0)
  - Salida no utiliza pull-up ni pull-down (EPUN = 1, EPD = 0)
  - Salida de baja velocidad y poco ruido (EHS = 0)
  - Se puede leer la salida sin filtro (EZI = 1, ZIF = 0)

31..8	7	6	5	4	3	2:0
Reservados	ZIF	EZI	EHS	EPUN	EPD	MODE
0	0	1	0	1	0	000

Puertos de Entrada/Salida GPIO

- El procesador soporta hasta 8 puertos de GPIO
- Cada puerto GPIO puede tener hasta 32 líneas en futuras versiones.
- El μC LPC4337 posee 83 líneas de entrada/salida digitales.
- Los puertos GPIO0, GPIO1, GPIO2 y GPIO3 tienen 16 líneas cada uno.
- El puerto GPIO4 tiene una sola línea.
- El puerto GPIO5 tiene 18 líneas.

Configuración del GPIO

- Existen tres grupos de registros para la configuración de los puertos de GPIO
  a) Los que controlan el comportamiento y el estado de las líneas de entrada/salida digitales
  b) Los que configuran interrupciones individuales en hasta ocho líneas digitales
  c) Los que configuran interrupciones grupales en hasta dos grupos de líneas digitales
- Interrupciones para próximo tema.

Control y estado líneas digitales

- Se configura mediante los siguientes registros, correspondencia uno a uno con las lineas del GPIO. Reg(n) ⇨ configura linea (n) del GPIO.
- **DIR**: Determina si la línea es entrada o salida.
- **PIN**: Devuelve el estado actual de cada línea y fija el mismo en las salidas.
- **MASK**: Almacena la mascara para operaciones en bits particulares en el puerto.
- **MPIN**: Realiza una lectura o escritura solo de los pines enmascarados del puerto.

Estado de las líneas Digitales

- **SET**: Fija las salidas en un valor alto.
- **CLR**: Fija las salidas en un valor bajo.
- **NOT**: Cambia las salidas al valor contrario al valor actual.
- Acceso Bit Banded:
  - BYTE: Devuelve el estado de cada línea de GPIO en el bit 0 de un byte individual.
  - WORD: Devuelve el estado de cada línea de GPIO en el bit 0 de una palabra.

Registros de Dirección

- El registro DIR determina la dirección de cada una de las líneas del puerto GPIO

0x400F 6000	DIR0
0x400F 6004	DIR1
0x400F 6008	DIR2
…	…
0x400F 6018	DIR6
0x400F 601C	DIR7

31	…	1	0
DIRn[31]		DIRn[1]	DIRn[0]

Dirección DIRn[m]	Valor
Entrada	0
Salida	1

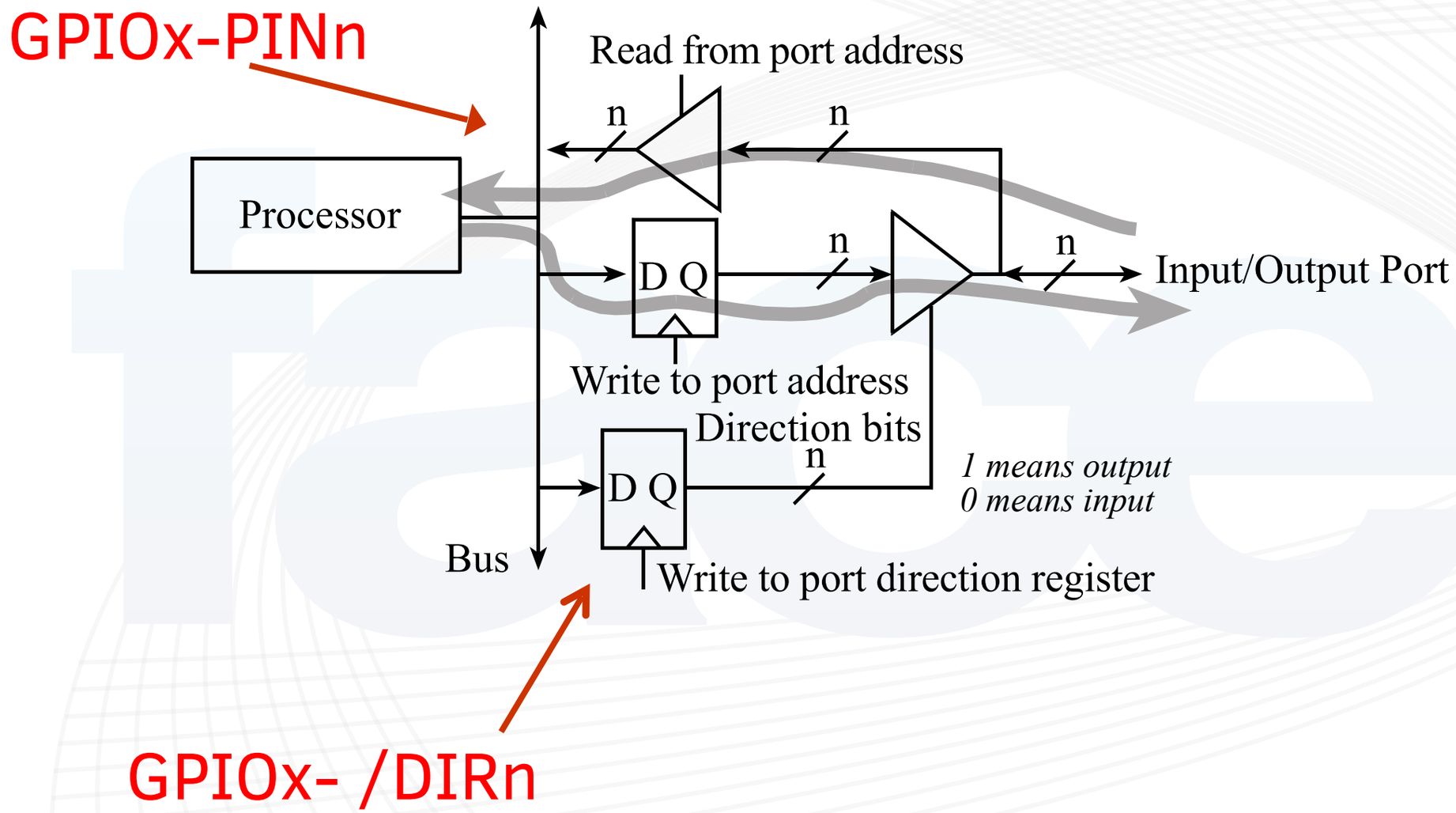
Registros de Entrada/Salida

- El registro PIN permite leer el valor actual de cada línea y escribir las salidas

0x400F 6100	PIN0
0x400F 6104	PIN1
0x400F 6108	PIN2
…	…
0x400F 6118	PIN6
0x400F 611C	PIN7

31	…	1	0
GPIOn[31]		GPIOn[1]	GPIOn[0]

Conexión 1 bit – simplificado.



Mascara de Entrada/Salida

- El registro MASK permite enmascarar lineas:
  no se puede escribir en ellas y se lee siempre cero.

0x400F 6080	MASK0
0x400F 6084	MASK1
0x400F 6088	MASK2
...	...
0x400F 6098	MASK6
0x400F 609C	MASK7

31	…	1	0
MASKn[31]		MASKn[1]	MASKn[0]

Efecto	Valor
No enmascarada	0
Enmascarada	1

Entrada/Salida Enmascaradas

- El registro MPIN permite leer o escribir PIN con la máscara aplicada

0x400F 6180	MPIN0
0x400F 6184	MPIN1
0x400F 6188	MPIN2
…	…
0x400F 6198	MPIN6
0x400F 619C	MPIN7

31	…	1	0
MPINn[31]		MPINn[1]	MPINn[0]

MPIN = PIN AND /MASK

En cada bit enmascarado:

- Se lee siempre cero
- Se ignora la escritura

Port Set Register

- Para poner a uno algunos pines del GPIO y dejar inalterado el resto. **PIN=PIN | SET**

0x400F 6200	SET0
0x400F 6204	SET1
0x400F 6208	SET2
…	…
0x400F 6218	SET6
0x400F 621C	SET7

31	…	1	0
SETn[31]		SETn[1]	SETn[0]

Efecto	Valor
GPIOn[b] no varía	0
GPIOn[b] ← 1	1

Port Clear Register

- Para poner a cero algunos pines del GPIO y dejar inalterado el resto. **PIN=PIN & /CLR**

0x400F 6280	CLR0
0x400F 6284	CLR1
0x400F 6288	CLR2
...	...
0x400F 6298	CLR6
0x400F 629C	CLR7

31	…	1	0
CLRn[31]		CLRn[1]	CLRn[0]

Efecto	Valor
GPIOn[b] no varía	0
GPIOn[b] ← 0	1

Port Toggle Register

- Para invertir algunos pines del GPIO y dejar inalterado el resto. **PIN=PIN ^ NOT**

0x400F 6300	NOTP0
0x400F 6304	NOTP1
0x400F 6308	NOTP2
…	…
0x400F 6318	NOTP6
0x400F 631C	NOTP7

31	…	1	0
NOTPn[31]		NOTPn[1]	NOTPn[0]

Efecto	Valor
GPIOn[b] no varía	0
GPIOn[b] ← /GPIOn[b]	1

Registros de bits en Palabras

- El registro W permite leer el valor actual o escribir el estado de una línea individual

0x400F 5000	W0
0x400F 5004	W1
…	…
0x400F 507C	W31
…	…
0x400F 53FC	W225

31	…	1	0
0		0	GPIOn[m]

Se lee y escribe sin máscara

Registros de bits en Bytes

- El registro B permite leer el valor actual o escribir el estado de una línea individual

0x400F 4000	B0
0x400F 4001	B1
…	…
0x400F 401F	B31
…	…
0x400F 40FF	B255

31:25	24	23:17	16	15:9	8	7:1	0
0	GPIOn [m+3]	0	GPIOn [m+2]	0	GPIOn [m+1]	0	GPIOn [m]

Se lee y escribe sin máscara

Atención: manual con errores en direcciones

Configurando la placa EDU-CIAA

- Tecla 4
  - Bit 9 del GPIO 1
  - Se debe configurar como entrada
- Led 3
  - Bit 12 del GPIO 1
  - Se debe configurar como salida
- Se configura en el registro DIR1
  - El bit 9 se fija en cero
  - El bit 12 se fija en uno

Configurando la placa EDU-CIAA

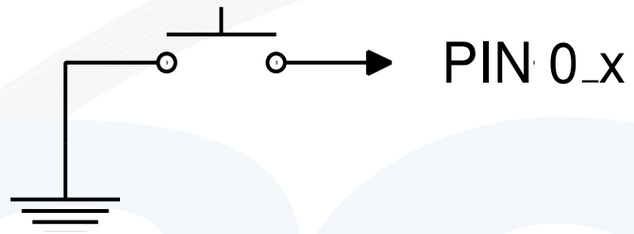
- Tecla 4
  - Para saber el estado se lee el registro PIN1
  - Hay que enmascarar el bit 9 del valor leído
  - También se puede leer el registro B42
  - También se puede leer el registro W42

Configurando la placa EDU-CIAA

- Led 3
  - Para prenderlo hay que poner en 1 el bit 12 del registro PIN1
  - Para apagarlo hay que poner en 0 el bit 12 del registro PIN1
  - También se puede escribir el registro B45
  - También se puede escribir el registro W45
  - También se puede escribir el bit 9 de los registros SET1, CLR1 y NOT1

Ejemplo

- 4 llaves informan un número BCD.
- Esquema de una llave, ¿cómo ingresa un 1?



- Leerlas y enviarlas a un display BCD
  - El display requiere entrada BCD
  - Al comienzo del programa van los EQU’s
- Usaremos GPIO0, 8 pins menos significativos.
- Los 4 lsb serán entradas, el resto salidas.

Registro de configuración de terminales

- Cada uno de los 8 terminales tiene un registro de configuración SFSPi.
  - Elegirlos para GPIO_0 a GPIO_7 de tabla 189.
  - Configuramos los 8 terminales de la misma manera.
- MODE=000 → GPIO
- EPD=0 → sin pull down.
- EPUN=0 → con pull up.
- EHS=0 → pendiente de salida media.
- EZI=1 → Entrada conectada.
- ZIF=0 → filtro de ruido conectado.
- SFSPi= 01000000B = Ox0000.0040

Configuración GPIO0

- Máscara de Entrada/Salida:
  - MASK0=0xFFFFxFF00
  - Solo deja los 8 pins.
  - ¿Por qué enmascaro hasta bit 31, si solo hay 16 bits máximo en GPIO0?
- Registro de Dirección (DIR0):
  - Último byte: 11110000 (binario)
    - 4 lsb entradas (0), los siguientes salidas (1).
  - DIR0=0x0000.00F0
  - ¿Por qué resto entradas?

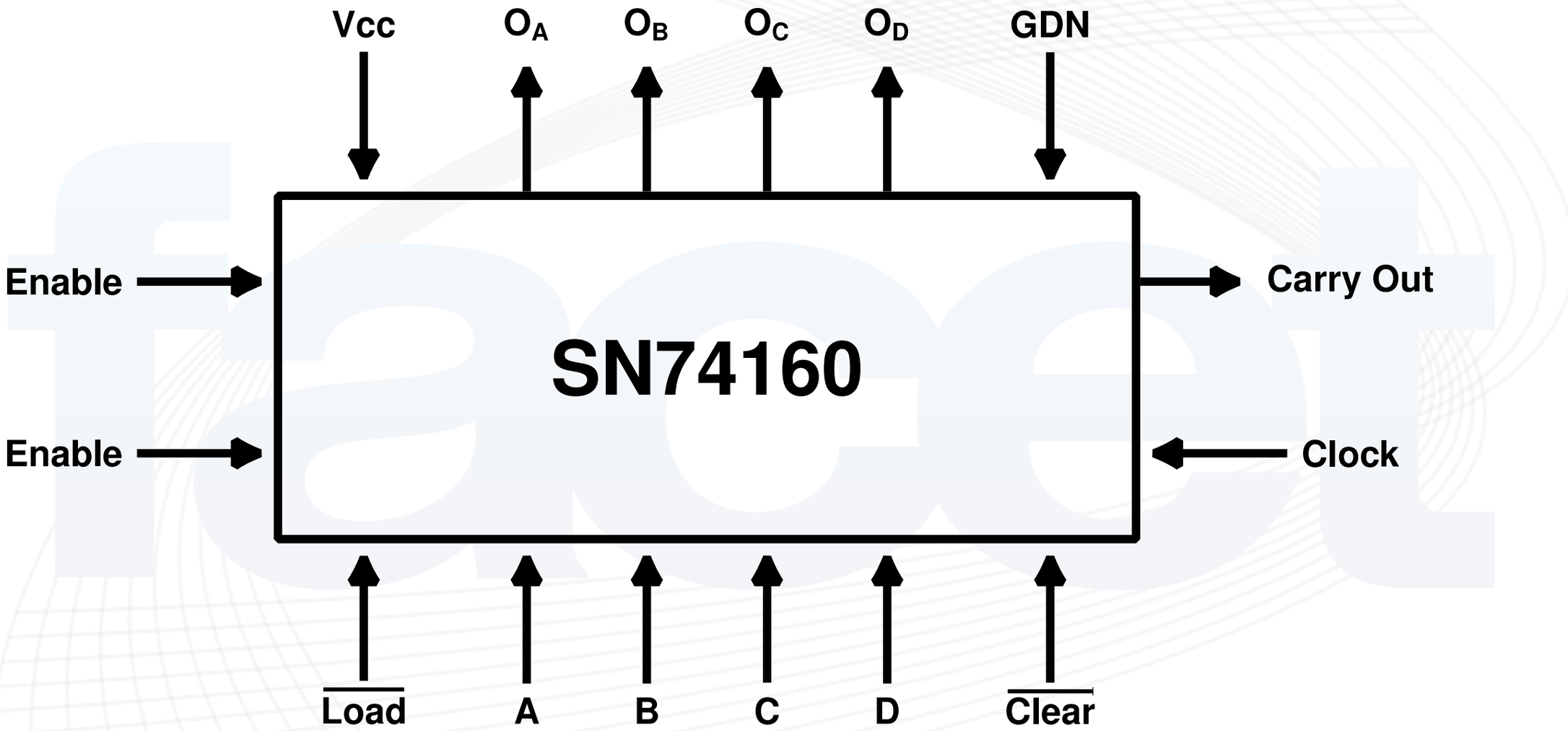
Solución con GPIO0 configurado

```
...
LDR R0,=GPIO_PIN0
LDR R1,[R0]
LSL R1,R1,#4
STR R1,[R0]
...
```

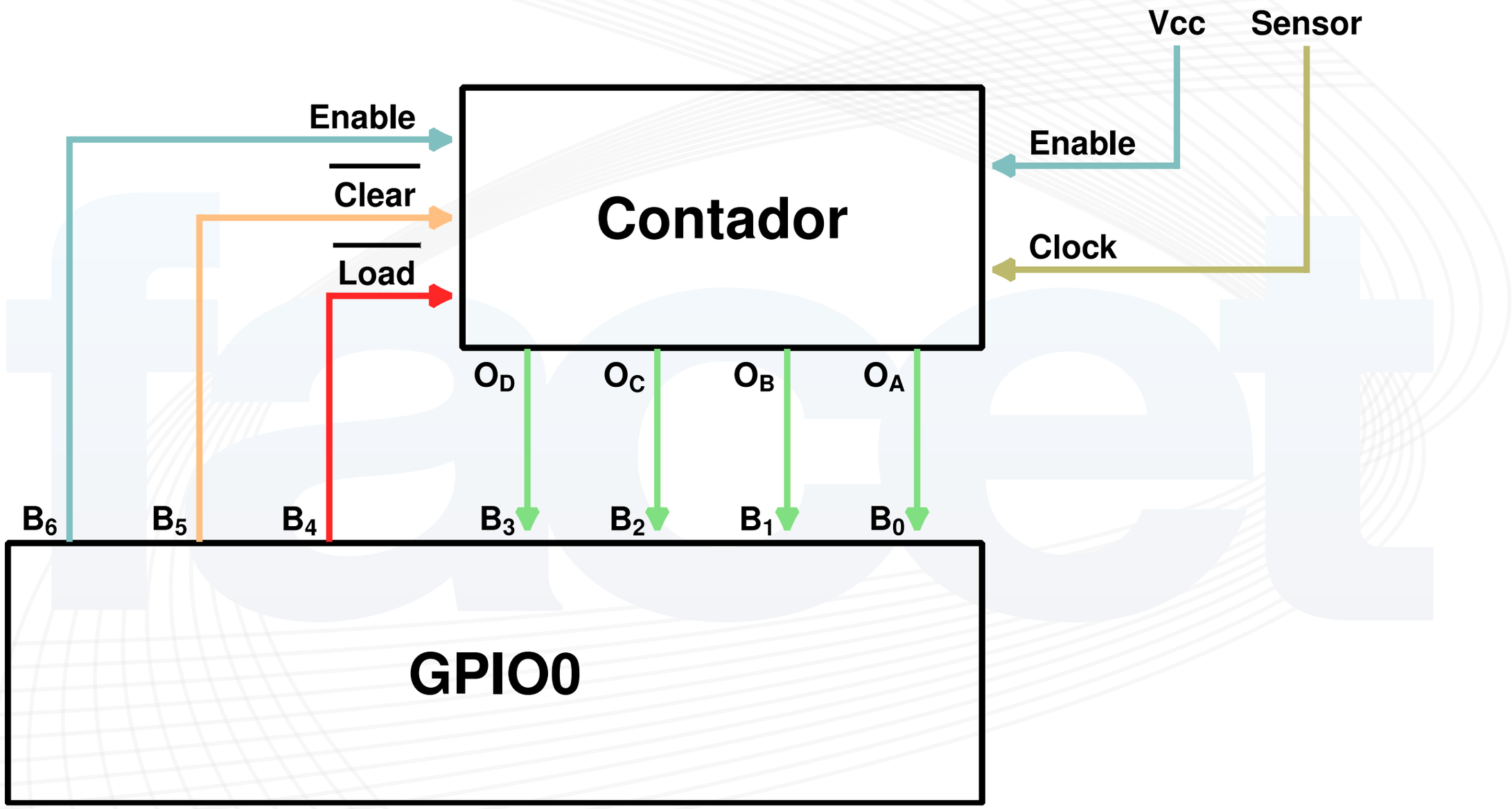
Ejemplo de Aplicación

- Se desea medir las rpm de un motor.
- Disponemos de un sensor que emite un pulso por cada vuelta del motor.
- La velocidad mínima es de 0 rpm.
- La velocidad máxima es de 3000 rpm.
- Se desea una precisión de 200 rpm.
- ¿cuántos bits se necesitan?
- ¿en qué ventana de tiempo? 15/3000=0,3 seg

Conexiones de un contador



Conexión al Microcontrolador



Algoritmo

- Configurar GPIO0 (SFSPx's, DIR0,MASK0).
- Clear CONTADOR y Enable.
- Esperar la ventana de tiempo (que cuente 15 para el caso de Vmax, 300 ms)
  - 15/3000 min = 60/200 s = 0,3 s.
- Parar, Leer Contador y Mostrar valor.
- Vaya a 2.

Inicialización

- Configurar 7 pins (SPFPi), idem caso anterior, sin pull-up ni pull-down.
  - SPFPi=0x0000.0050
- DIR0=0x0000.0070
- MASK0=0xFFFF.FF80

```
LDR R0,=GPIO_DIR0
MOV R1,#0x70
STR R1,[R0]
LDR R0,=GPIO_MASK0
MVN R1,#0x7F          // Mueve inverso 7F=FFFFFF80
STR R1,[R0]
LDR R0,=GPIO_MPIN0
```

Parar, borrar y contrar

- GPIO[6]=Enable
- GPIO[5]=/Clear
- GPIO[4]=/Load

- 001=1 → parar y resetear el contador
- 111=7 → contar.
- 011=3 → parar.

Parar, borrar y contrar

```
lazo:   MOV R1,#0x10
        STR R1,[R0]      // Parar y borrar el contador
        MOV R1,#0x70
        STR R1,[R0]      // Habilitar la cuenta

        BL  delay        // Esperar 300 milisegundos

        MOV R1,#0x30
        STR R1,[R0]      // Inhabilitar la cuenta
        LDR R1,[R0]      // Leer el valor contador
        AND R1,#0x0F     // Descartar bits no usados
        LSL R1,#1        // Multiplicar por dos

        BL   mostrar     // Mostrar el resultado
        B    lazo        // Repetir la captura
```

Observaciones

- ¿Puede haber errores por el tiempo que pasa desde que transcurre el tiempo de x ms y el tiempo en que lo paro?
- Se soluciona haciendo que la rutina de x ms lo pare también y eso esté dentro de la cuenta.
- Pero los errores son muy pequeños frente a los 300 ms de espera!!
- Esto es “tiempo real” – no olvidar.