

# PINSEL

## Registros de configuración de los pines

### LPC\_PINCON->PINSELx

David Trujillo

Selección de función de los pines. x de 0 a 9, 2 registros por puerto [R/W]

- **31:0 [0]:** 2 bits por pin.
  - 00 = Generalmente GPIO
  - 01 = Función 1
  - 10 = Función 2
  - 11 = Función 3

### LPC\_PINCON->PINMODEx

Configura las resistencias de entrada en los pines. x de 0 a 9, 2 registros por puerto [R/W]

- **31:0 [0]:** 2 bits por pin.
  - 00 = Pull-up
  - 01 = Repeater Se habilita PU o PD según el nivel del pin. Para evitar estado flotante y no consumir energía.
  - 10 = None
  - 11 = Pull-down

### LPC\_PINCON->PINMODE\_ODx

Configura los pines como de drenador abierto. x de 0 a 4 [R/W]

- **31:0 [0]:** 1 bit por pin.
  - 0 = Normal
  - 1 = Open-drain

David Trujillo

## Drivers de configuración de los pines

### Estructuras de configuración

- **PINSEL\_CFG\_Type**
  - **uint8\_t Portnum:**
    - PINSEL\_PORT\_x x de 0 a 4.
  - **uint8\_t Pinnum:**
    - PINSEL\_PIN\_x x de 0 a 31.
  - **uint8\_t Funcnum:**
    - PINSEL\_FUNC\_x x de 0 a 3.
  - **uint8\_t Pinmode:**
    - PINSEL\_PINMODE\_PULLUP
    - PINSEL\_PINMODE\_TRISTATE
    - PINSEL\_PINMODE\_PULLDOWN
  - **uint8\_t OpenDrain:**
    - PINSEL\_PINMODE\_NORMAL
    - PINSEL\_PINMODE\_OPENDRAIN

### PINSEL\_ConfigPin(PinCfg)

Configura un pin

- **PinCfg:** Puntero a estructura de configuración.

David Trujillo

## GPIO

### Pines disponibles

- **Puerto 0:** [0-11], [15-30]; pueden interrumpir
- **Puerto 1:** [0-1], [4], [8-10], [14-31]
- **Puerto 2:** [0-13]; pueden interrumpir
- **Puerto 3:** [25-26]
- **Puerto 4:** [28-29]

# Registros de los GPIO

## LPC\_GPIOx->FIODIR

David Trujillo

Dirección de los pines [R/W]

- **31:0 [0]:** 0 = Entrada , 1 = Salida , 1 bit por pin.

## LPC\_GPIOx->FIOMASK

Máscara de bits [R/W]

- **31:0 [0]:** Habilita la lectura y escritura de cada pin, 1 bit por pin.
  - 0
  - 1

## LPC\_GPIOx->FIOPIN

Estado de los pines [R/W]

- **31:0 [0]:** Estado de cada pin, 1 bit por pin.
  - 0
  - 1

## LPC\_GPIOx->FIOSET

Pone en alto los pines a los que les asigne un 1 en este registro [R/W]

- **31:0 [0]** 1 bit por pin.

David Trujillo

## LPC\_GPIOx->FIOCLR

Pone en bajo los pines a los que les asigne un 1 en este registro [W0]

- **31:0 [0]** 1 bit por pin.

## LPC\_GPIOINT->IOxIntEnR

Habilita las interrupciones por flanco de subida en el puerto x [R/W]

- **31:0 [0]**

## LPC\_GPIOINT->IOxIntEnF

Habilita las interrupciones por flanco de bajada en el puerto x [R/W]

- **31:0 [0]**

## LPC\_GPIOINT->IntStatus

Estado de las interrupciones [R0]

- **0 P0Int [0]:** Se pone en 1 cuando hay por lo menos una interrupción pendiente en el puerto 0.
- **2 P2Int [0]:** Se pone en 1 cuando hay por lo menos una interrupción pendiente en el puerto 2.

## LPC\_GPIOINT->IOxIntStatR

David Trujillo

Estado de las interrupciones por flanco de subida en el puerto x [R0]

- **31:0 [0]:** Se pone en 1 cuando hubo una interrupción por flanco de subida en el pin correspondiente.

## LPC\_GPIOINT->IOxIntStatF

Estado de las interrupciones por flanco de bajada en el puerto x [R0]

- **31:0 [0]:** Se pone en 1 cuando hubo una interrupción por flanco de bajada en el pin correspondiente.

## LPC\_GPIOINT->IOxIntClr

Limpia las interrupciones de los pines correspondientes del puerto x [w0]

- 31:0 [0]

---

## Drivers de los GPIO

David Trujillo

---

### GPIO\_SetDir(portNum, bitValue, dir)

Define la dirección de los pines

- **portNum**: Puerto a configurar. 0 al 4
- **bitValue**: 0 a 0xFFFFFFFF para elegir los pines a configurar.
- **dir**: 0 = Entrada , 1 = Salida

---

### GPIO\_SetValue(portNum, bitValue)

Pone en alto pines específicos (salidas únicamente)

- **portNum**: Puerto a configurar. 0 al 4
- **bitValue**: 0 a 0xFFFFFFFF para elegir los pines a configurar.

---

### GPIO\_ClearValue(portNum, bitValue)

Pone en bajo pines específicos (salidas únicamente)

- **portNum**: Puerto a configurar. 0 al 4
- **bitValue**: 0 a 0xFFFFFFFF para elegir los pines a configurar.

---

### uint32\_t GPIO\_ReadValue(portNum)

David Trujillo

Retorna el estado de los pines de un puerto

- **portNum**: Puerto a leer. 0 al 4

---

### FIO\_SetMask(portNum, bitValue, maskValue)

Habilita o deshabilita la máscara de pines específicos (salidas únicamente)

- **portNum**: Puerto a configurar. 0 al 4
- **bitValue**: 0 a 0xFFFFFFFF para elegir los pines a configurar.
- **maskValue**:
  - 0 : Deshabilita la máscara
  - 1 : Habilita la máscara

---

### GPIO\_IntCmd(portNum, bitValue, edgeState)

Habilita las interrupciones por flanco de subida o bajada (P0.0-P0.30, P2.0-P2.13)

- **portNum**: Puerto a configurar. 0 o 2 únicamente.
- **bitValue**: 0 a 0xFFFFFFFF para elegir los pines a configurar. No hace |= con el registro, lo sobrescribe.
- **edgeState**:
  - 0 : Flanco de subida
  - 1 : Flanco de bajada

---

### NVIC\_EnableIRQ(EINT3\_IRQn)

David Trujillo

Habilita la interrupción en el NVIC

---

### EINT3\_IRQHandler()

Handler de la interrupción de GPIO

---

### FunctionalState GPIO\_GetIntStatus(portNum, pinNum, edgeState)

Devuelve el estado de la interrupción (1 o 0) de un pin (P0.0-P0.30, P2.0-P2.13)

- **portNum**: Puerto a leer. 0 o 2 únicamente.

- **pinNum**: Pin a leer. 0 al 30 para el puerto 0 y 0 al 13 para el puerto 2.
- **edgeState**: Estado que se quiere leer.
  - 0 : Flanco de subida
  - 1 : Flanco de bajada

## GPIO\_ClearInt(portNum, bitValue)

David Trujillo

Limpia la interrupción de un pin (P0.0-P0.30, P2.0-P2.13)

- **portNum**: Puerto a limpiar. 0 o 2 únicamente.
- **bitValue**: Elección de los pines a limpiar.

# INTERRUPCIONES EXTERNAS

## Pines disponibles

- **EINT0**: P2[10]. Func 01.
- **EINT1**: P2[11]. Func 01.
- **EINT2**: P2[12]. Func 01.
- **EINT3**: P2[13]. Func 01.

## Registros de las interrupciones externas

### LPC\_SC->EXTINT

Registros de flag de interrupciones externas [R/W]

- **0 EINT0 [0]**: Se limpia escribiendo 1. En modo de nivel, solo se puede limpiar si el pin no está en el nivel configurado para interrumpir.
- **1 EINT1 [0]**: Idem.
- **2 EINT2 [0]**: Idem.
- **3 EINT3 [0]**: Idem.

David Trujillo

### LPC\_SC->EXTMODE

Registros de configuración de modo de las interrupciones externas [R/W]

- **0 EXTMODE0 [0]**: Se debe deshabilitar la interrupción antes de cambiar el modo, y limpiar el EXTINT correspondiente antes de habilitarla.
  - 0 = Nivel.
  - 1 = Flanco.
- **1 EXTMODE1 [0]**:
- **2 EXTMODE2 [0]**:
- **3 EXTMODE3 [0]**:

### LPC\_SC->EXTPOLAR

Registros de configuración de los flancos de las interrupciones externas [R/W]

- **0 EXTPOLAR0 [0]**: Se debe deshabilitar la interrupción antes de cambiar el modo, y limpiar el EXTINT correspondiente antes de habilitarla.
  - 0 = Nivel bajo o Flanco de bajada.
  - 1 = Nivel alto o Flanco de subida.
- **1 EXTPOLAR1 [0]**:
- **2 EXTPOLAR2 [0]**:
- **3 EXTPOLAR3 [0]**:

## Drivers de las interrupciones externas

David Trujillo

## Estructuras de configuración

- **EXTL\_InitTypeDef**
  - **EXTL\_Line**:
    - EXTI\_EINTx x de 0 a 3.
  - **EXTL\_Mode**:
    - EXTI\_MODE\_LEVEL\_SENSITIVE
    - EXTI\_MODE\_EDGE\_SENSITIVE
  - **EXTL\_polarity**:
    - EXTI\_POLARITY\_LOW\_ACTIVE\_OR\_FALLING\_EDGE
    - EXTI\_POLARITY\_HIGH\_ACTIVE\_OR\_RISING\_EDGE

## EXTI\_Init()

Limpia los flags de interrupción y configura todo a 0

## EXTI\_DeInit()

No hace nada

David Trujillo

## EXTI\_Config(EXTICfg)

Configura una interrupción externa

- **EXTICfg**: Estructura de configuración.

## NVIC\_EnableIRQ(EINTx\_IRQn)

Habilita la interrupción externa x en el NVIC

- x [0-3].

## EINTx\_IRQHandler(void)

Handlers de la interrupción externa x [0-3]

## EXTI\_SetMode(EXTILine, mode)

Configura el modo de una interrupción externa

- **EXTILine**: Línea a configurar.
  - **EXTI\_EINTx** x de 0 a 3.
- **mode**:
  - **EXTI\_MODE\_LEVEL\_SENSITIVE**
  - **EXTI\_MODE\_EDGE\_SENSITIVE**

David Trujillo

## EXTI\_SetPolarity(EXTILine, polarity)

Configura la polaridad de una interrupción externa

- **EXTILine**: Línea a configurar.
  - **EXTI\_EINTx** x de 0 a 3.
- **polarity**:
  - **EXTI\_POLARITY\_LOW\_ACTIVE\_OR\_FALLING\_EDGE**
  - **EXTI\_POLARITY\_HIGH\_ACTIVE\_OR\_RISING\_EDGE**

## EXTI\_ClearEXTIFlag(EXTILine)

Limpia el flag de una interrupción externa

- **EXTILine**: Interrupción a limpiar.
  - **EXTI\_EINTx** x de 0 a 3.

# SYSTICK

## Registros del SysTick

### SysTick->CTRL

David Trujillo

Registro de control del SysTick [R/W]

- **0 ENABLE [0]**: Habilita el contador.
- **1 TICKINT [0]**: Habilita la interrupción. Interrumpe cuando el contador llega a 0.
- **2 CLKSOURCE [1]**: Selecciona la fuente de clock.
  - **0** Fuente de clock externa (STCLK).
  - **1** Fuente de clock interna.
- **16 COUNTFLAG [0]**: Bandera de conteo. Se pone en 1 cuando el contador llega a 0. Se limpia leyéndola.

### SysTick->LOAD

Registro de recarga del SysTick [R/W]

- **23:0 RELOAD [0]**: Valor de recarga. Cuando el contador llega a 0, vuelve a contar desde este valor.

## SysTick->VAL

Registro de valor actual del SysTick [R/W]

- **23:0 CURRENT [0]**: Valor actual del contador.

David Trujillo

## SysTick->CALIB

Registro de calibración del SysTick [R/W]

- **23:0 TENMS [0x0F 423F]**: Valor de calibración. Contiene el valor de carga necesario para contar 10ms a 100MHz.
- **30 SKEW [0]**: Indica si con el valor de TENMS se puede considerar preciso el conteo de 10ms.
  - 0 El conteo se considera preciso.
  - 1 El conteo no es preciso.
- **31 NOREF [0]**: Indica si existe un reloj externo de referencia.
  - 0 Disponible.
  - 1 No disponible.

# Drivers del SysTick

## SYSTICK\_Internallnit(time)

Inicializa el SysTick con el CPU clock y carga  $(cclk/1000)*time - 1$  en LOAD

- **time**: Tiempo en ms. Tiempo máximo igual a  $1/SystemCoreClock * (2^{24}) * 1000$  (ms)

## SYSTICK\_Externallnit(freq, time)

David Trujillo

Inicializa el SysTick con un clock externo y carga  $(freq/1000)*time - 1$  en LOAD

- **freq**: Frecuencia del clock externo.
- **time**: Tiempo en ms. Tiempo máximo igual a  $1/freq * (2^{24}) * 1000$  (ms)

## SYSTICK\_Cmd(NewState)

Habilita o deshabilita el contador del SysTick

- **NewState**: ENABLE o DISABLE

## SYSTICK\_IntCmd(NewState)

Habilita o deshabilita la interrupción del SysTick

- **NewState**: ENABLE o DISABLE

## SysTick\_Handler()

Handler de la interrupción del SysTick

## uint32\_t SYSTICK\_GetCurrentValue()

Devuelve el valor actual del contador del SysTick (VAL)

David Trujillo

## SYSTICK\_ClearCounterFlag()

Limpia la bandera de conteo del SysTick (COUNTFLAG)

## SysTick\_Config(ticks)

Configura el SysTick con un valor de recarga, habilita las interrupciones e inicia el contador

- **ticks**: Valor de recarga. Máximo  $2^{24} - 1 = 16777215$

# TIMERS

## Pines disponibles

David Trujillo

- **Timer 0:**
  - **CAP0.0:** P1[26]. Func 11.
  - **CAP0.1:** P1[27]. Func 11.
  - **MAT0.0:** P1[28]. Func 11. P3[25] Func 10.
  - **MAT0.1:** P1[29]. Func 11. P3[26] Func 10.
- **Timer 1:**
  - **CAP1.0:** P1[18]. Func 11.
  - **CAP1.1:** P1[19]. Func 11.
  - **MAT1.0:** P1[22]. Func 11.
  - **MAT1.1:** P1[25]. Func 11.
- **Timer 2:**
  - **CAP2.0:** P0[4]. Func 11.
  - **CAP2.1:** P0[5]. Func 11.
  - **MAT2.0:** P0[6]. Func 11. P4[28] Func 10.
  - **MAT2.1:** P0[7]. Func 11. P4[29] Func 10.
  - **MAT2.2:** P0[8]. Func 11.
  - **MAT2.3:** P0[9]. Func 11.
- **Timer 3:**
  - **CAP3.0:** P0[23]. Func 11.
  - **CAP3.1:** P0[24]. Func 11.
  - **MAT3.0:** P0[10]. Func 11.
  - **MAT3.1:** P0[11]. Func 11.

## Registros de los Timers

### LPC\_SC->PCONP

David Trujillo

Power para los Timers

- **1 PCTIMO [0]:** Habilita el Timer 0.
- **2 PCTIM1 [0]:** Habilita el Timer 1.
- **22 PCTIM2 [0]:** Habilita el Timer 2.
- **23 PCTIM3 [0]:** Habilita el Timer 3.

### LPC\_SC->PCLKSELO

Clock para los Timers 0 y 1

- **3:2 PCLK\_TIMER0 [0]:**
  - **00** = CCLK/4
  - **01** = CCLK
  - **10** = CCLK/2
  - **11** = CCLK/8
- **5:4 PCLK\_TIMER1 [0]**

### LPC\_SC->PCLKSEL1

Clock para los Timers 2 y 3

- **13:12 PCLK\_TIMER2 [0]**
- **15:14 PCLK\_TIMER3 [0]**

### LPC\_TIMx->IR

Flags de interrupción del timer LPC\_TIMx

David Trujillo

- **0 MR0 [0]:** Interrupción por coincidencia con MR0.
- **1 MR1 [0]:** Interrupción por coincidencia con MR1.
- **2 MR2 [0]:** Interrupción por coincidencia con MR2.
- **3 MR3 [0]:** Interrupción por coincidencia con MR3.
- **4 CR0 [0]:** Interrupción por evento de captura en CR0.
- **5 CR1 [0]:** Interrupción por evento de captura en CR1.

### LPC\_TIMx->TCR

Control del Timer LPC\_TIMx

- **0 CE [0]:** Habilita el contador.
- **1 CR [0]:** Resetea el TC y PC en el próximo flanco de PCLK. Debe ser limpiado por software para seguir contando.

## LPC\_TIMx->CTCR

### Control de modo del timer LPC\_TIMx

David Trujillo

- **1:0 CTM [0]:**
  - 00 = Timer Mode
  - 01 = Counter Mode, captura en flanco de subida
  - 10 = Counter Mode, captura en flanco de bajada
  - 11 = Counter Mode, captura en flanco de subida y bajada
- **3:2 CIS [0]:**
  - 00 = CAPx.0
  - 01 = CAPx.1 En modo contador, los 3 bits del CAP seleccionado como captura de evento se ponen en 0 (CCR); sin embargo, el otro CAP se puede seguir usando para interrumpir o capturar el TC

## LPC\_TIMx->CCR

### Control de captura del timer LPC\_TIMx

- **0 CAP0RE [0]:** CAP0 en RE causa que TC se copie a CR0.
- **1 CAP0FE [0]:** CAP0 en FE causa que TC se copie a CR0.
- **2 CAP0I [0]:** Interrumpe en el flanco habilitado (pueden ser ambos).
- **3 CAP1RE [0]:** CAP1 en RE causa que TC se copie a CR1.
- **4 CAP1FE [0]:** CAP1 en FE causa que TC se copie a CR1.
- **5 CAP1I [0]:** Interrumpe en el flanco habilitado (pueden ser ambos).

## LPC\_TIMx->CR0

## LPC\_TIMx->CR1

### Registros de captura del timer LPC\_TIMx

David Trujillo

- **31:0 CR [0]:** Copia el valor de TC cuando ocurre el evento habilitado en CCR.

## LPC\_TIMx->TC

- **31:0 TC [0]:** Incrementa cada que PC alcanza el valor de PR, si no se reinicia cuenta hasta 0xFFFF FFFF y regresa a 0 sin interrumpir (se puede interrumpir con un match si se necesita)

## LPC\_TIMx->PR

### Prescaler del timer LPC\_TIMx

- **31:0 PR [0]:** Cada que PC alcanza PR, TC incrementa en 1, si PR=0, TC++ cada ciclo de reloj. En modo contador, se pone en 0 para contar todos los eventos.

## LPC\_TIMx->PC

### Prescaler counter del timer LPC\_TIMx

- **31:0 PC [0]:** Cuenta cada ciclo de reloj, si PC=PR, TC incrementa en 1.

## LPC\_TIMx->MR0

## LPC\_TIMx->MR1

## LPC\_TIMx->MR2

David Trujillo

## LPC\_TIMx->MR3

### Registros de match del timer LPC\_TIMx

- **31:0 MR [0]:** Se comparan continuamente con TC, si son iguales se dispara el evento controlado por MCR.

## LPC\_TIMx->MCR

### Control de match del timer LPC\_TIMx

- **0 MR0I [0]:** Interrumpe cuando MR0=TC.
- **1 MR0R [0]:** Resetea TC cuando MR0=TC.
- **2 MR0S [0]:** Detiene TC y PC cuando MR0=TC, además TCR[0]=0.



- 5:3: Idem para MR1.
- 8:6: Idem para MR2.
- 11:9: Idem para MR3.

## LPC\_TIMx->EMR

Control de salida del match del timer LPC\_TIMx

David Trujillo

- 0 **EM0 [0]**: Valor de salida para pin MATx.0, comportamiento controlado por EMC0.
- 1 **EM1 [0]**: Idem para MATx.1.
- 2 **EM2 [0]**: Idem para MATx.2 (solo en timer 2).
- 3 **EM3 [0]**: Idem para MATx.3 (solo en timer 2).
- 5:4 **EMC0 [0]**: Comportamiento de MATx.0:
  - 00 = No cambia
  - 01 = Pone a 0 en match
  - 10 = Pone a 1 en match
  - 11 = Alterna el valor en match
- 7:6 **EMC1 [0]**: Idem para MATx.1.
- 9:8 **EMC2 [0]**: Idem para MATx.2 (solo en timer 2).
- 11:10 **EMC3 [0]**: Idem para MATx.3 (solo en timer 2).

## Drivers de los Timers

### Estructuras de configuración

- **TIM\_TIMERCFG\_Type**
  - **uint8\_t PrescaleOption**:
    - TIM\_PRESCALE\_TICKVAL: Valor de prescaler en ticks.
    - TIM\_PRESCALE\_USVAL: Valor de prescaler en microsegundos.
  - **uint32\_t PrescaleValue**:
- **TIM\_COUNTERCFG\_Type**
  - **uint8\_t CounterOption**:
    - TIM\_COUNTER\_INCAP0: Captura en pin CAPx.0 para TIMEx.
    - TIM\_COUNTER\_INCAP1: Captura en pin CAPx.1 para TIMEx.
  - **uint8\_t CountInputSelect**:
    - La documentación dice que los argumentos van en el campo CounterOption, pero la función usa CountInputSelect, posiblemente mal documentado. No manipular esta estructura de forma directa.
- **TIM\_MATCHCFG\_Type**
  - **uint8\_t MatchChannel**: Canal de match.
    - 0
    - 1
    - 2
    - 3
  - **uint8\_t IntOnMatch**: Habilita interrupción en match.
    - ENABLE o DISABLE
  - **uint8\_t StopOnMatch**: Detiene el timer en match.
    - ENABLE o DISABLE
  - **uint8\_t ResetOnMatch**: Resetea el timer en match.
    - ENABLE o DISABLE
  - **uint8\_t ExtMatchOutputType**: Tipo de salida en match.
    - TIM\_EXTMATCH\_NOTHING
    - TIM\_EXTMATCH\_LOW
    - TIM\_EXTMATCH\_HIGH
    - TIM\_EXTMATCH\_TOGGLE
  - **uint8\_t MatchValue**: Valor de match.
- **TIM\_CAPTURECFG\_Type**
  - **uint8\_t CaptureChannel**: Canal de captura.
    - 0
    - 1
  - **uint8\_t RisingEdge**: Habilita captura en flanco de subida.
    - ENABLE o DISABLE
  - **uint8\_t FallingEdge**: Habilita captura en flanco de bajada.
    - ENABLE o DISABLE
  - **uint8\_t IntOnCaption**: Habilita interrupción en captura.
    - ENABLE o DISABLE

David Trujillo

### TIM\_Init(LPC\_TIMx, TimerCounterMode, TIM\_ConfigStruct)

Habilita el power para el timer, clock en CCLK/4, setea el prescaler para el rate de la struct, resetea y limpia la interrupción

- **LPC\_TIMx**: Timer a inicializar. `x = [0-3]`
- **TimerCounterMode**: Modo del timer.
  - `TIM_TIMER_MODE` Modo timer
  - `TIM_COUNTER_RISING` Modo contador, captura en flanco de subida
  - `TIM_COUNTER_FALLING` Modo contador, captura en flanco de bajada
  - `TIM_COUNTER_ANY` Modo contador, captura en flanco de subida y bajada
- **TIM\_ConfigStruct**: Puntero `TIM_TIMERCFG_Type` para timer o `TIM_COUNTERCFG_Type` para contador.

## TIM\_DeInit(LPC\_TIMx)

David Trujillo

Detiene el timer y deshabilita el power

- **LPC\_TIMx**: Timer a desinicializar. `x = [0-3]`

## TIM\_ClearIntPending(LPC\_TIMx, IntFlag)

Limpia la bandera de interrupción de un timer

- **IntFlag**: Bandera a limpiar.
  - `TIM_MR0_INT` Match 0
  - `TIM_MR1_INT` Match 1
  - `TIM_MR2_INT` Match 2
  - `TIM_MR3_INT` Match 3
  - `TIM_CR0_INT` Captura 0
  - `TIM_CR1_INT` Captura 1

## TIM\_ClearIntCapturePending(LPC\_TIMx, IntFlag)

Idem que la anterior, pero para capturas. Función totalmente al pedo IMHO

- **IntFlag**: Bandera a limpiar.
  - `0` Captura 0 (no pasar `TIM_CR0_INT` porque la función inicia desplazando 4 bits)
  - `1` Captura 1 (no pasar `TIM_CR1_INT` porque la función inicia desplazando 4 bits)

## FlagStatus TIM\_GetIntStatus(LPC\_TIMx, IntFlag)

David Trujillo

Devuelve el estado de la bandera de interrupción de un timer (1 o 0)

- **IntFlag**: Bandera a leer.
  - `TIM_MR0_INT` Match 0
  - `TIM_MR1_INT` Match 1
  - `TIM_MR2_INT` Match 2
  - `TIM_MR3_INT` Match 3
  - `TIM_CR0_INT` Captura 0
  - `TIM_CR1_INT` Captura 1

## FlagStatus TIM\_GetIntCaptureStatus(LPC\_TIMx, IntFlag)

Idem que la anterior, pero para capturas. Función totalmente al pedo IMHO

- **IntFlag**: Bandera a leer.
  - `0` Captura 0 (no pasar `TIM_CR0_INT` porque la función inicia desplazando 4 bits)
  - `1` Captura 1 (no pasar `TIM_CR1_INT` porque la función inicia desplazando 4 bits)

## TIM\_ConfigStructInit(TimerCounterMode, TIM\_ConfigStruct)

Inicializa la estructura de configuración `TIM_TIMERCFG_Type` o `TIM_COUNTERCFG_Type`

- **TimerCounterMode**: Modo del timer.
  - `TIM_TIMER_MODE` Modo timer
  - `TIM_COUNTER_RISING` Modo contador, captura en flanco de subida
  - `TIM_COUNTER_FALLING` Modo contador, captura en flanco de bajada
  - `TIM_COUNTER_ANY` Modo contador, captura en flanco de subida y bajada
- **TIM\_ConfigStruct**: Puntero `TIM_TIMERCFG_Type` para timer o `TIM_COUNTERCFG_Type` para contador.

## TIM\_ConfigMatch(LPC\_TIMx, TIM\_MatchConfigStruct)

Configura el valor del match, las interrupciones y el comportamiento de la salida

- **TIM\_MatchConfigStruct**: Puntero a estructura de configuración de tipo `TIM_MATCHCFG_Type`

## TIM\_UpdateMatchValue(LPC\_TIMx, MatchChannel, MatchValue)

Configura el valor de match

- **MatchChannel:** Canal de match.
  - 0
  - 1
  - 2
  - 3
- **MatchValue:** Nuevo valor de match.

David Trujillo

## TIM\_SetMatchExt(LPC\_TIMx, ext\_match )

Configura el comportamiento de la salida en match

- **ext\_match:** Comportamiento de la salida en match.
  - TIM\_EXTMATCH\_NOTHING
  - TIM\_EXTMATCH\_LOW
  - TIM\_EXTMATCH\_HIGH
  - TIM\_EXTMATCH\_TOGGLE

## NVIC\_EnableIRQ(TIMERx\_IRQn)

Habilita la interrupción del timer en el NVIC

- **TIMERx\_IRQn:** Interrupción del timer. x = [0-3]

## TIMERx\_IRQHandler()

Handler de la interrupción del timer x [0-3]

## TIM\_ConfigCapture(LPC\_TIMx, TIM\_CaptureConfigStruct)

Configura el flanco de captura

- **TIM\_CaptureConfigStruct:** Puntero a estructura de configuración de tipo TIM\_CAPTURECFG\_Type

David Trujillo

## TIM\_Cmd(LPC\_TIMx, NewState)

Habilita o deshabilita el contador (TCR[0]), no deshabilita el power

- **NewState:** ENABLE o DISABLE

## uint32\_t TIM\_GetCaptureValue(LPC\_TIMx, CaptureChannel)

Devuelve el valor de TC que se capturó (CRx)

- **CaptureChannel:** Canal de captura.
  - TIM\_COUNTER\_INCAP0
  - TIM\_COUNTER\_INCAP1

## TIM\_ResetCounter(LPC\_TIMx)

Resetea el timer (TCR[1])

# ADC

## Pines disponibles

David Trujillo

- **AD0:** P0[23]. Func 01.
- **AD1:** P0[24]. Func 01.
- **AD2:** P0[25]. Func 01.
- **AD3:** P0[26]. Func 01.
- **AD4:** P1[30]. Func 11.
- **AD5:** P1[31]. Func 11.
- **AD6:** P0[3]. Func 10.
- **AD7:** P0[2]. Func 10.

## Registros del ADC

## LPC\_SC->PCONP

Power para el ADC [R/W]

- **12 PCADC [0]:** Limpiar PDN antes de deshabilitar, habilitar PCADC antes del PDN.

## LPC\_SC->PCLKSELO

Clock para el ADC [R/W]

- **25:24 PCLK\_ADC [0]:**
  - 00 = CCLK/4
  - 01 = CCLK
  - 10 = CCLK/2
  - 11 = CCLK/8

## LPC\_ADC->ADCR

Configuración del ADC [R/W]

- **7:0 SEL [0x1]:** Selección de canal, en modo SW se debe habilitar 1 a la vez.
- **15:8 CLKDIV [0]:**  $[PCLK\_ADC / (CLKDIV + 1)] \leq 13 \text{ MHz}$
- **16 BURST [0]:** Si se habilita,  $START=000$  y  $ADINTEN:8=0$ .
- **21 PDN [0]:** Habilita el ADC.
- **26:24 START [0]:**
  - 001 = Inicia la conversión.
  - 010 = EINT0 (P2[10]).
  - 011 = CAP0.1 (P1[27]).
  - 100 = T0MR1.
  - 101 = T0MR3.
  - 110 = T1MR0.
  - 111 = T1MR1.
- **27 EDGE [0]:** 0 = Subida, 1 = Bajada; decide flanco de 010-111 de START.

## LPC\_ADC->ADGDR

Estado Global [R/W]

- **15:4 RESULT [X]:** Última conversión.
- **26:24 CHN [X]:** Canal que se convirtió.
- **30 OVERRUN [0]:** Resultado sobrescrito, siempre 1 en burst, se limpia leyendo.
- **31 DONE [0]:** Conversión lista, se limpia leyendo ADGDR o escribiendo ADCR. Si se escribe ADCR durante una conversión, se pone en 1 y se reinicia la conversión.

## LPC\_ADC->ADINTEN

Interrupciones [R/W]

- **7:0 ADINTENx [0]:** Habilita interrupción por canal x, 1 = habilita.
- **8 ADGINTEN [1]:** 0 = Interrumpen los canales habilitados, 1 = interrumpe DONE de ADGDR.

## LPC\_ADC->ADDRx

Estado específico del canal x [0-7] [R0]

- **15:4 RESULT [X]:** Última conversión.
- **30 OVERRUN [0]:** Resultado sobrescrito, siempre 1 en burst, se limpia leyendo.
- **31 DONE [0]:** Conversión lista, se limpia leyendo.

## LPC\_ADC->ADSTAT

Estado Global [R0]

- **7:0 DONE [0]:** Repite el DONE de ADDRx.
- **15:8 OVERRUN [0]:** Repite el OVERRUN de ADDRx.
- **16 ADINT [0]:** Se pone en 1 cuando hay algún DONE en 1, si no hay ningún DONE se limpia, por lo que si se interrumpe por software, se limpia leyendo ADGDR.

## Drivers del ADC

## ADC\_Init(LPC\_ADC, rate)

Activa el Power al adc, setea CLKDIV para tener el rate deseado y habilita el PDN del ADC

- **LPC\_ADC**: ADC a inicializar, es el único que hay
- **rate**: Frecuencia de muestreo, máximo 200 kHz

## ADC\_BurstCmd(LPC\_ADC, NewState)

Habilita o deshabilita el modo burst

- **NewState**: ENABLE o DISABLE

David Trujillo

## ADC\_StartCmd(LPC\_ADC, start\_mode)

Configura la forma de inicio de la conversión

- **start\_mode**:
  - ADC\_START\_CONTINUOUS : Para modo burst
  - ADC\_START\_NOW : Inicia una conversión
  - ADC\_START\_ON\_EINT0 : Inicia en flanco de EINT0
  - ADC\_START\_ON\_CAP01 : Inicia en flanco de CAP0.1
  - ADC\_START\_ON\_MAT01 : Inicia en flanco de TOMR1
  - ADC\_START\_ON\_MAT03 : Inicia en flanco de TOMR3
  - ADC\_START\_ON\_MAT10 : Inicia en flanco de T1MR0
  - ADC\_START\_ON\_MAT11 : Inicia en flanco de T1MR1

## ADC\_ChannelCmd(LPC\_ADC, channel, NewState)

Habilita o deshabilita un canal

- **channel**: Canal a habilitar
- **NewState**: ENABLE o DISABLE

## ADC\_EdgeStartConfig(LPC\_ADC, EdgeOption)

Configura el flanco de inicio de la conversión

David Trujillo

- **EdgeOption**:
  - ADC\_START\_ON\_RISING : Inicia en flanco de subida
  - ADC\_START\_ON\_FALLING : Inicia en flanco de bajada

## ADC\_IntConfig(LPC\_ADC, IntType, NewState)

Habilita o deshabilita la interrupción de un canal o del bit DONE de ADGDR

- **IntType**:
  - ADC\_ADINTENx : Habilita interrupción por canal x
- **NewState**: ENABLE o DISABLE

## NVIC\_EnableIRQ(ADC\_IRQn)

Habilita la interrupción del ADC en el NVIC

## ADC\_IRQHandler()

Handler de la interrupción del ADC

## ADC\_PowerdownCmd(LPC\_ADC, NewState)

Habilita o deshabilita el PDN del ADC (Innecesario si se usa ADC\_Init)

- **NewState**: ENABLE o DISABLE

David Trujillo

## uint32\_t ADC\_GlobalGetData(LPC\_ADC)

Devuelve el valor de la última conversión (ADGDR)

## uint16\_t ADC\_ChannelGetData(LPC\_ADC, channel)

Devuelve el valor de la última conversión de un canal específico (ADDRx)

- **channel:** Canal a leer

## FlagStatus ADC\_GlobalGetStatus(LPC\_ADC, StatusType)

Verifica el estado global (Overrun o Done, según StatusType)

- **StatusType:**
  - 0 : Overrun
  - 1 : Done

David Trujillo

## FlagStatus ADC\_ChannelGetStatus(LPC\_ADC, channel, StatusType)

Verifica el estado de un canal (Overrun o Done, según StatusType)

- **channel:** Canal a leer
- **StatusType:**
  - 0 : Overrun
  - 1 : Done

## DAC

### Pines disponibles

- AOUT: P0[26]. Func 10.

## Registros del DAC

### LPC\_SC->PCLKSELO

Clock para el timer asociado al DAC-DMA [R/W]

David Trujillo

- **23:22 PCLK\_DAC [0]:**
  - 00 = CCLK/4
  - 01 = CCLK
  - 10 = CCLK/2
  - 11 = CCLK/8

### LPC\_DAC->DACR

Configuración del DAC [R/W]

- **15:6 VALUE [0]:** Valor de salida.  $(VALUE \times (V_{refp} - V_{refn}) / 1024 + V_{refn})$
- **16 BIAS [0]:**
  - 0 Tiempo de establecimiento de 1us y corriente de salida de 700uA. Permite actualización de 1MHz.
  - 1 Tiempo de establecimiento de 2.5us y corriente de salida de 350uA. Permite actualización de 400kHz.

### LPC\_DAC->DACCTRL

Control del DAC [R/W]

- **0 INT\_DMA\_REQ [0]:** Habilita la solicitud de DMA.
  - 0 El bit se pone a 0 cuando se escribe en DACR .
  - 1 El bit se pone a 1 cuando se agota el timer de DMA.
- **1 DBLBUF\_ENA [0]:** Habilita el doble buffer.
  - 0 Se actualiza la salida del DAC apenas se escribe en DACR .
  - 1 Los datos pasan por un buffer antes de sacarlos, por sincronización. CNT\_ENA debe estar habilitado.
- **2 CNT\_ENA [0]:**
  - 0 El contador de time-out no se habilita.
  - 1 El contador de time-out se habilita.
- **3 DMA\_ENA [0]:** Habilita DMA.
  - 0 El acceso a DMA está deshabilitado.
  - 1 Se habilita el DMA burst request input 7.

David Trujillo

### LPC\_DAC->DACCNTVAL

Valor del contador de time-out [R/W]

- **15:0 VALUE [0]**

---

# Drivers del DAC

---

## Estructuras de configuración

---

- **DAC\_CONVERTER\_CFG\_Type**
  - **uint8\_t DBLBUF\_ENA:**
    - 0 o 1
  - **uint8\_t CNT\_ENA:**
    - 0 o 1
  - **uint8\_t DMA\_ENA:**
    - 0 o 1

David Trujillo

## DAC\_Init(LPC\_DAC)

---

Activa el Power al DAC, el CLK/4 y el bias de 700uA

## DAC\_UpdateValue (LPC\_DAC, dac\_value)

---

Actualiza el valor de salida del DAC

- **dac\_value:** Valor de salida (0-1023).

## DAC\_SetBias (LPC\_DAC, bias)

---

Setea el bias del DAC

- 0 = 700uA, 1us, 1MHz
- 1 = 350uA, 2.5us, 400kHz

## DAC\_ConfigDAConverterControl (LPC\_DAC, DAC\_ConverterConfigStruct)

---

Configuración del DMA (DACCTRL)

David Trujillo

- **DAC\_ConverterConfigStruct:** Estructura DAC\_CONVERTER\_CFG\_Type.

## DAC\_SetDMATimeOut(LPC\_DAC, time\_out)

---

Setea el time-out del DMA

- **time\_out:** Valor del contador de time-out.

# DMA

---

## Registros del DMA

---

### LPC\_GPDMA->DMACIntStat

---

Estado de las interrupciones por canal [R0]

- **7:0 [0]:** 1 bit por canal. Para limpiarla, debería preguntar si la interrupción es de TC o de error y limpiar la correspondiente.

### LPC\_GPDMA->DMACIntTCStat

---

Estado de las interrupciones por transferencia completa [R0]

- **15:0 [0]:** 1 bit por canal. Se pone en 1 cuando se completa una transferencia (todo el TransferSize).

### LPC\_GPDMA->DMACIntTCClear

---

David Trujillo

Limpia las interrupciones por transferencia completa [W0]

- **15:0 [0]:** 1 bit por canal. Escribir un 1 limpia la interrupción del canal correspondiente.

### LPC\_GPDMA->DMACIntErrStat

---

Estado de las interrupciones por error [R0]

- **15:0 [0]:** 1 bit por canal. Se pone en 1 cuando hay un error en la transferencia.

## LPC\_GPDMA->DMACIntErrClr

Limpia las interrupciones por error [W0]

David Trujillo

- **15:0 [0]:** 1 bit por canal. Escribir un 1 limpia la interrupción del canal correspondiente.

## LPC\_GPDMA->DMACRawIntTCStat

Estado de las interrupciones por transferencia completa sin enmascarar [R0]

- **15:0 [0]:** 1 bit por canal.

## LPC\_GPDMA->DMACRawIntErrStat

Estado de las interrupciones por error sin enmascarar [R0]

- **15:0 [0]:** 1 bit por canal.

## LPC\_GPDMA->DMACEnbldChns

Estado de los canales [R0]

- **7:0 [0]:** 1 bit por canal. NO habilita los canales, solo muestra su estado.
  - 0 = Canal deshabilitado.
  - 1 = Canal habilitado.

## LPC\_GPDMA->DMACSoftBReq

David Trujillo

Solicitud de transferencia burst [R/W]

Si se escribe un 1 en un bit, se solicita una transferencia burst en el canal correspondiente. Se pone en 0 cuando se completa la transferencia burst.

- **0 [0]:** SSP0 Tx
- **1 [0]:** SSP0 Rx
- **2 [0]:** SSP1 Tx
- **3 [0]:** SSP1 Rx
- **4 [0]:** ADC
- **5 [0]:** I2S Ch0
- **6 [0]:** I2S Ch1
- **7 [0]:** DAC
- **8 [0]:** UART0 Tx/MAT0.0
- **9 [0]:** UART0 Rx/MAT0.1
- **10 [0]:** UART1 Tx/MAT1.0
- **11 [0]:** UART1 Rx/MAT1.1
- **12 [0]:** UART2 Tx/MAT2.0
- **13 [0]:** UART2 Rx/MAT2.1
- **14 [0]:** UART3 Tx/MAT3.0
- **15 [0]:** UART3 Rx/MAT3.1

## LPC\_GPDMA->DMACSoftSReq

Solicitud de transferencia simple [R/W]

Si se escribe un 1 en un bit, se solicita una transferencia simple en el canal correspondiente. Se pone en 0 cuando se completa la transferencia simple.

- **15:0 [0]:** Mismas fuentes que DMACSoftBReq .

## LPC\_GPDMA->DMACSoftLBReq

David Trujillo

Solicitud de última transferencia de ráfaga [R/W]

- **15:0 [0]:** Mismas fuentes que DMACSoftBReq .

## LPC\_GPDMA->DMACSoftLSReq

Solicitud de última transferencia simple [R/W]

- **15:0 [0]:** Mismas fuentes que DMACSoftBReq .

## LPC\_GPDMA->DMACConfig



#### Configuración del DMA [R/W]

- **0 E [0]:** Habilita el DMA.
- **1 M [0]:**
  - 0 = Little endian.
  - 1 = Big endian.

## LPC\_GPDMA-&gtDMACSync

David Trujillo

#### Sincronización de las solicitudes de DMA [R/W]

- **0 [0]:** 0 = Sincroniza, 1 = no sincroniza.

## LPC\_SC-&gtDMAREQSEL

#### Selección de fuente de solicitud de DMA para las fuentes compartidas (8-15) [R/W]

- **0 DMASEL08 [0]:**
  - 0 = UART
  - 1 = Timer
- **1 DMASEL09 [0]**
- **2 DMASEL10 [0]**
- **3 DMASEL11 [0]**
- **4 DMASEL12 [0]**
- **5 DMASEL13 [0]**
- **6 DMASEL14 [0]**
- **7 DMASEL15 [0]**

## LPC\_GPDMACHx-&gtDMACCSrcAddr

#### Dirección de la fuente de la transferencia [R/W]

- **31:0 [0]:** Dirección de la fuente.

## LPC\_GPDMACHx-&gtDMACCDestAddr

David Trujillo

#### Dirección del destino de la transferencia [R/W]

- **31:0 [0]:** Dirección del destino.

## LPC\_GPDMACHx-&gtDMACCLLI

#### Dirección del siguiente LLI [R/W]

- **1:0 [0]:** Siempre 0, ya que las direcciones son múltiplos de 4 (word-aligned).
- **31:2 [0]:** Dirección del siguiente LLI.

## LPC\_GPDMACHx-&gtDMACCControl

#### Control de la transferencia [R/W]

- **11:0 TransferSize [0]:** Tamaño total de la transferencia.
- **14:12 SBSIZE [0]:** Tamaño de la transferencia de burst.
  - 000 = 1
  - 001 = 4
  - 010 = 8
  - 011 = 16
  - 100 = 32
  - 101 = 64
  - 110 = 128
  - 111 = 256
- **17:15 DBSIZE [0]:** Tamaño de la transferencia de burst.
- **20:18 SWidth [0]:** Ancho de la transferencia de fuente.
  - 000 = 8 bits
  - 001 = 16 bits
  - 010 = 32 bits
- **23:21 DWidth [0]:** Ancho de la transferencia de destino.
- **26 SI [0]:** Indica si la dirección se debe incrementar después de cada transferencia. 0 = dirección constante.
- **27 DI [0]:** Indica si la dirección se debe incrementar después de cada transferencia. 0 = dirección constante.
- **31 I [0]:** Interrumpe al final de la transferencia total (TransferSize).

David Trujillo

## LPC\_GPDMACHx-&gtDMACCConfig

- **0 E [0]:** Habilita el canal.
  - 0 = Canal deshabilitado.
  - 1 = Canal habilitado.
- **5:1 SrcPeripheral [0]:** Periférico fuente.
- **10:6 DestPeripheral [0]:** Periférico destino.
- **13:11 TransferType [0]:** Tipo de transferencia.
  - 000 = M2M
  - 001 = M2P
  - 010 = P2M
  - 011 = P2P
- **14 IE [0]:** Máscara de interrupción de error.
- **15 ITC [0]:** Máscara de interrupción de transferencia completa.
- **17 A [0]:** Estado del canal.
- **18 H [0]:** Ignora los pedidos de transferencia.

David Trujillo

## Drivers del DMA

### Estructuras de configuración

- **GPDMA\_Channel\_CFG\_Type**
  - **uint32\_t ChannelNum:**
    - 0 a 7 . Número de canal. El canal 0 tiene la mayor prioridad.
  - **uint32\_t TransferSize:**
    - Tamaño total de la transferencia.
  - **uint32\_t TransferWidth:**
    - Ancho de la transferencia.
  - **uint32\_t SrcMemAddr:**
    - Dirección de la fuente.
  - **uint32\_t DstMemAddr:**
    - Dirección del destino.
  - **uint32\_t TransferType:**
    - GPDMA\_TRANSFERTYPE\_M2M
    - GPDMA\_TRANSFERTYPE\_M2P
    - GPDMA\_TRANSFERTYPE\_P2M
    - GPDMA\_TRANSFERTYPE\_P2P
  - **uint32\_t SrcConn:** Periférico fuente.
    - GPDMA\_CONN\_SSP0\_Tx
    - GPDMA\_CONN\_SSP0\_Rx
    - GPDMA\_CONN\_SSP1\_Tx
    - GPDMA\_CONN\_SSP1\_Rx
    - GPDMA\_CONN\_ADC
    - GPDMA\_CONN\_I2S\_Channel\_0
    - GPDMA\_CONN\_I2S\_Channel\_1
    - GPDMA\_CONN\_DAC
    - GPDMA\_CONN\_UART0\_Tx
    - GPDMA\_CONN\_UART0\_Rx
    - GPDMA\_CONN\_UART1\_Tx
    - GPDMA\_CONN\_UART1\_Rx
    - GPDMA\_CONN\_UART2\_Tx
    - GPDMA\_CONN\_UART2\_Rx
    - GPDMA\_CONN\_UART3\_Tx
    - GPDMA\_CONN\_UART3\_Rx
    - GPDMA\_CONN\_MAT0\_0
    - GPDMA\_CONN\_MAT0\_1
    - GPDMA\_CONN\_MAT1\_0
    - GPDMA\_CONN\_MAT1\_1
    - GPDMA\_CONN\_MAT2\_0
    - GPDMA\_CONN\_MAT2\_1
    - GPDMA\_CONN\_MAT3\_0
    - GPDMA\_CONN\_MAT3\_1
  - **uint32\_t DstConn:** Periférico destino.
    - Idem
  - **uint32\_t DMALLI:**
    - Si no hay siguiente LLI, poner 0.
- **GPDMA\_LLI\_Type**
  - **uint32\_t SrcAddr:**
    - Dirección de la fuente.
  - **uint32\_t DstAddr:**
    - Dirección del destino.
  - **uint32\_t NextLLI:**
    - Dirección del siguiente LLI.
  - **uint32\_t Control:**
    - Control de la transferencia.

David Trujillo

David Trujillo

## GPDMA\_Init()

David Trujillo

Habilita el power, limpia los registros de configuración de los canales y limpia las interrupciones

## Status GPDMA\_Setup(GPDMAChannelConfig)

Configura un canal, no lo habilita, devuelve 1 si la configuración fue exitosa

- **GPDMAChannelConfig:** Puntero a estructura de configuración de tipo GPDMA\_Channel\_CFG\_Type

## IntStatus GPDMA\_IntGetStatus(type, channel)

Devuelve el estado de las interrupciones, 1 si existe una interrupción activa

- **type:**
  - GPDMA\_STAT\_INT
  - GPDMA\_STAT\_INTTC
  - GPDMA\_STAT\_INTERR
  - GPDMA\_STAT\_RAWINTTC
  - GPDMA\_STAT\_RAWINTERR
  - GPDMA\_STAT\_ENABLED\_CH
- **channel:** Canal a consultar.

## GPDMA\_ClearIntPending(type, channel)

Limpia las interrupciones

David Trujillo

- **type:**
  - GPDMA\_STATCLR\_INTTC
  - GPDMA\_STATCLR\_INTERR
- **channel:** Canal a limpiar.

## GPDMA\_ChannelCmd(channelNum, NewState)

Habilita o deshabilita un canal

- **channelNum:** Canal a habilitar/deshabilitar.
- **NewState:** ENABLE o DISABLE

David Trujillo