

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:** 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

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)

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.

David Trujillo

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.

LPC_SC->EXTMODE

David Trujillo

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

Estructuras de configuración

- **EXTI_InitTypeDef**
 - **EXTI_Line**:
 - EXTI_EINTx x de 0 a 3.
 - **EXTI_Mode**:
 - EXTI_MODE_LEVEL_SENSITIVE
 - EXTI_MODE_EDGE_SENSITIVE
 - **EXTI_polarity**:
 - EXTI_POLARITY_LOW_ACTIVE_OR_FALLING_EDGE
 - EXTI_POLARITY_HIGH_ACTIVE_OR_RISING_EDGE

David Trujillo

EXTI_Init()

Limpia los flags de interrupción y configura todo a 0

EXTI_DeInit()

No hace nada

EXTI_Config(EXTICfg)

Configura una interrupción externa

- **EXTICfg**: Estructura de configuración.

David Trujillo

NVIC_EnableIRQ(EINTx_IRQn)

Habilita la interrupción externa x en el NVIC

- 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**

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**

David Trujillo

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

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

David Trujillo

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.

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.

David Trujillo

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)

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

David Trujillo

SYSTICK_IntCmd(NewState)

Habilita o deshabilita la interrupción del SysTick

- **NewState:** ENABLE o DISABLE

uint32_t SYSTICK_GetCurrentValue()

Devuelve el valor actual del contador del SysTick (VAL)

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 $2^{24} - 1 = 16777215$

ADC

David Trujillo

Pines disponibles

- **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

David Trujillo

Power para el ADC [R/W]

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

LPC_SC->PCLKSEL0

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**.

David Trujillo

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]

David Trujillo

- **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)

David Trujillo

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

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

David Trujillo

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

- **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

David Trujillo

ADC_PowerdownCmd(LPC_ADC, NewState)

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

- **NewState**: ENABLE o DISABLE

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

David Trujillo

- **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

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

David Trujillo

DAC_ConfigDAConverterControl (LPC_DAC, DAC_ConverterConfigStruct)

Configuración del DMA (DACCTRL)

- **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

David Trujillo

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

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]

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

LPC_GPDMA->DMACRawIntTCStat

David Trujillo

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

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

David Trujillo

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

Solicitud de última transferencia de ráfaga [R/W]

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

David Trujillo

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.

David Trujillo

LPC_GPDMA->DMACSync

Sincronización de las solicitudes de DMA [R/W]

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

LPC_SC->DMAREQSEL

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->DMACCSrcAddr

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

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

David Trujillo

LPC_GPDMAChx->DMACCDestAddr

Dirección del destino de la transferencia [R/W]

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

LPC_GPDMAChx->DMACCLLI

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->DMACCControl

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->DMACCConfig

Configuración de la transferencia [R/W, 17=R0]

- **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

David Trujillo

David Trujillo