GPIO

1. REGISTROS DEL MICRO CONTROLADOR

IO

Los registros que controlan la GPIO van desde la posición 0x400F 4000 a la hasta la 0x400F 6320. En este sector se encuentran 9 grupos: B, W, DIR, MASK, PIN, MPIN, SET, CLR, NOT.

En el caso de los grupos B y W, Modificar el registro BX modifica el pin Z del puerto Y donde X = Y\*32 + Z. Para todos los otros registros Modificar el bit Z del registro Y modifica el pin Z del puerto Y.

B: Son registros de 8 bits. Hay 256 de estos, cada uno controla un pin. De cada uno de estos registros solo se van a leer o escribir el bit menos significativo. Un 1 en ese bit corresponde a estado alto. El offset va desde 0x0000 hasta 0x00FF.

W: Son registros de 32 bits. Hay 256 de estos, cada uno controla un pin. En la lectura se lee 0x FFFF FFFF si esta en alto y 0 si está en bajo. En la escritura, se escribe 0 para que esté en estado bajo y cualquier otro valor para que está en alto. El offset va desde 0x1000 hasta 0x13FC.

DIR: Son registros de 32 bits. Hay 8 de estos, uno por cada puerto. Escribir un 1 en un bit pone a un pin como salida y si se escribe un 0 se establece como entrada. El offset va desde 0x2000 hasta 0x201C.

MASK: Son registros de 32 bits. Hay 8 de estos, uno por cada puerto. Poner un 1 en el bit X del registro MASKY hace que no se pueda leer ni escribir en el bit X del registro MPINY. El offset va desde 0x2080 hasta 0x209C.

PIN: Son registros de 32 bits. Hay 8 de estos, uno por cada puerto. Se puede leer o escribir el estado de un pin. Un 1 corresponde a estado alto. El offset va desde 0x2100 hasta 0x211C.

MPIN: Son registros de 32 bits. Hay 8 de estos, uno por cada puerto. Si El bit correspondiente de MASK está en 1, entonces el bit no se puede escribir y siempre se lee como 0. Si el bit de MASK está en 0 entonces funciona como PIN. El offset va desde 0x2180 hasta 0x219C.

SET: Son registros de 32 bits. Hay 8 de estos, uno por cada puerto. Escribir un 1 setea el pin y escribir un 0 no genera cambios. No se puede leer estos registros. El offset va desde 0x2200 hasta 0x221C.

CLR: Son registros de 32 bits. Hay 8 de estos, uno por cada puerto. Al leer se lee el estado del pin. Escribir un 1 pone el pin en bajo y escribir un 0 no genera cambios. El offset va desde 0x2280 hasta 0x229C.

NOT: Son registros de 32 bits. Hay 8 de estos, uno por cada puerto. Escribir un 1 togglea el estado del pin y escribir un 0 no genera cambios. No se puede leer estos registros. El offset va desde 0x2300 hasta 0x231C.

INTERRUPCIONES POR PIN

Existen dos tipos de interrupciones que se pueden generar por GPIO, las interrupciones asociadas a pin, análogas a INT0 e INT1 en el atmega328p, y las interrupciones asociadas por grupo, análogas PCINT0 a PCINT2 en el atmega328p.

Existen 8 interrupciones de pines de GPIO, PIN\_INT0 a PIN\_INT8. Al contrario de lo que ocurre con el atmega328p uno puede elegir que pines de GPIO son los que van a formar parte de este tipo de interrupción.

Las interrupciones por pin estás asociadas a 2 grupos de registro, uno asociado a SCU y otro a GPIO.

Los registros asociados al SCU permiten asociar el numero de interrupción al pin de GPIO y existen 2 registros: PINTSEL0 y PINTSEL1. La dirección base de estos registros es 0x4008 6000 y el offset de estos registros son 0xE00 y 0xE04 respectivamente.

Tabla 1.1:

|  |  |  |
| --- | --- | --- |
| Registro | Bits | Función |
| PINTSEL0 | 4:0 | Selecciona el número de pin de GPIO para la interrupción PIN\_INT0. |
| 7:5 | Selecciona el número de port de GPIO para la interrupción PIN\_INT0. |
| 8:12 | Selecciona el número de pin de GPIO para la interrupción PIN\_INT1. |
| 13:15 | Selecciona el número de port de GPIO para la interrupción PIN\_INT1. |
| 16:20 | Selecciona el número de pin de GPIO para la interrupción PIN\_INT2. |
| 21:23 | Selecciona el número de port de GPIO para la interrupción PIN\_INT2. |
| 24:28 | Selecciona el número de pin de GPIO para la interrupción PIN\_INT3. |
| 29:31 | Selecciona el número de port de GPIO para la interrupción PIN\_INT3. |
| PINTSEL1 | 4:0 | Selecciona el número de pin de GPIO para la interrupción PIN\_INT4. |
| 7:5 | Selecciona el número de port de GPIO para la interrupción PIN\_INT4. |
| 8:12 | Selecciona el número de pin de GPIO para la interrupción PIN\_INT5. |
| 13:15 | Selecciona el número de port de GPIO para la interrupción PIN\_INT5. |
| 16:20 | Selecciona el número de pin de GPIO para la interrupción PIN\_INT6. |
| 21:23 | Selecciona el número de port de GPIO para la interrupción PIN\_INT6. |
| 24:28 | Selecciona el número de pin de GPIO para la interrupción PIN\_INT7. |
| 29:31 | Selecciona el número de port de GPIO para la interrupción PIN\_INT7. |

Los registros asociados al GPIO lo que haces es definir las características de las interrupciones. Estos registros son registros de 32 bits en donde el bit n de cada registro está asociada a una característica de la interrupción PIN\_INTn. La dirección base de estos registros es 0x4008 7000.

Tabla 1.2:

|  |  |  |  |
| --- | --- | --- | --- |
| Registro | Tipo | offset | Función (valor del bit n) |
| ISEL | R/W | 0x00 | 0 activa interrupción por flanco, 1 activa por nivel. |
| IENR | R/W | 0x04 | Si es por nivel: 0 deshabilita interrupción, 1 habilita interrupción.  Si es por flanco: 0 deshabilita interrupción por flanco ascendente, 1 habilita interrupción por flanco ascendente. |
| SIENR | WO | 0x08 | 0 no genera cambio.  Si es por nivel: 1 habilita interrupción.  Si es por flanco: 1 habilita interrupción por flanco ascendente. |
| CIENR | WO | 0x0C | 0 no genera cambio.  Si es por nivel: 1 deshabilita interrupción.  Si es por flanco: 1 deshabilita interrupción por flanco ascendente. |
| IENF | R/W | 0x10 | Si es por nivel: 0 setea el nivel de activación en LOW, setea el nivel de activación en HIGH.  Si es por flanco: 0 deshabilita interrupción por flanco descendente, 1 habilita interrupción por flanco descendente. |
| SIENF | WO | 0x14 | 0 no genera cambio.  Si es por nivel: 1 setea el nivel de activación en LOW.  Si es por flanco: 1 habilita interrupción por flanco descendente. |
| CIENF | WO | 0x18 | 0 no genera cambio.  Si es por nivel: 1 setea el nivel de activación en HIGH.  Si es por flanco: 1 deshabilita interrupción por flanco descendente. |
| RISE | R/W | 0x1C | Lectura: 0 no se detectó ningún flanco ascendente desde que se limpió el bit, 1 se detectó un flanco ascendente.  Escritura: 0 no genera cambio, 1 limpia el bit del registro. |
| FALL | R/W | 0x20 | Lectura: 0 no se detectó ningún flanco descendente desde que se limpió el bit, 1 se detectó un flanco descendente.  Escritura: 0 no genera cambio, 1 limpia el bit del registro. |
| IST | R/W | 0x24 | Lectura: 0 no se espera una interrupción, 1 se espera interrupción  Escritura: 0 no genera cambios, 1 si es por nivel, cambia el nivel de activación (togglea), si es por flaco limpia los bits de los registros RISE y FALL. |

INTERRUPCIONES POR GRUPO

Las interrupciones por grupo se generan cuando una o todos los pines de un grupo, dependiendo de la configuración, envían la señal adecuada. Existen 2 interrupciones de este tipo: GINT0, GINT1. Ambas interrupciones presentan la misma estructura de registros. La diferencia radica en que la dirección base de GINT0 es 0x4008 8000 mientras que la de GINT1 es 0x4008 9000.

Tabla 1.3:

|  |  |  |
| --- | --- | --- |
| Registro | offset | Función |
| CTRL | 0x00 | Bit 0: leer un 0 indica que no hay interrupción pendiente mientras que leer un 1 indica que si la hay. Escribir un 0 no genera cambios y escribir un 1 pone en 0 el bit  Bit 1: 0 función OR, indica que uno de los pines del grupo tiene que cumplir la condición de trigger para que se genere la interrupción. 1 función AND, indica que todos los pines tienen que cumplir la condición.  Bir 2: 0 la interrupción se genera por flanco, 1 la interrupción se genera por nivel |
| PORT\_POLn | 0x20 +n\*0x4 | Bit m: 0 el pin m del gpio port n se activa en LOW, 1 se activa en HIGH. |
| PORT\_ENAn | 0x40 +n\*0x4 | Bit m: 0 el pin m del gpio port n no contribuye a la interrupción, 1el pin contribuye a la interrupción. |

1. Estructura de datos que "matchean" el GPIO

IO

La estructura que marche los registros de GPIO con su número de puerto y pin es LPC\_GPIO\_T. que se muestra en la figura 2.1.

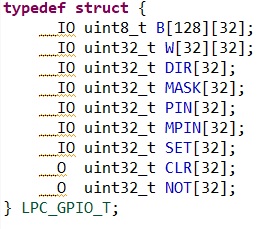


Figura 2.1: Estructura que representa los registros GPIO.

En la figura 2.1 se puede observar que IO corresponde a los registros de lectura y escritura y O corresponde a los de solo escritura. También se puede observar que cada grupo se agregan elementos de más para que cada agrupo posea el offset correcto.

También existe una etiqueta LPC\_GPIO\_PORT\_BASE a la que se le asigna el valor 0x400F4000, que corresponde a la dirección de inicio de los registros GPIO. Además, se encuentra la línea **#define** LPC\_GPIO\_PORT ((LPC\_GPIO\_T \*) LPC\_GPIO\_PORT\_BASE) que genera un puntero a LPC\_GPIO\_T con el valor de la dirección de inicio.

Como es necesario saber el numero de puerto y pin para poder gestionar el GPIO, existen otras 3 estructuras que sirven para obtener el numero de pin y puerto a partir de la etiqueta que le asigna la EDU-CIAA a cada pin del microcontrolador.

La primera estructura es un **enum** gpioMap\_t, que se encuentra en el archivo sapi\_peripheral\_map.h. enum. Lo que hace es asignarle a cada etiqueta de la EDU-CIAA un valor de posición. La segunda estructura es un vector de pinInitGpioLpc4337\_t, llamado gpioPinsInit, que se encuentra en el archivo sapi\_gpio.c. Lo que hace es obtener un pinInitGpioLpc4337\_t, a partir de la posición asignada a cada etiquta. Por último, el pinInitGpioLpc4337\_t contiene 5 int8\_t, correspondientes al puerto físico, pin físico, número de función asignada a gpio, puerto gpio y pin gpio. En conjunto estas 3 estructuras lo que hacen en conjunto es apartir de una estiqueta poder manejar la SCU para poner la función gpio y poder manejar la gpio. En las figuras 2.2 a 2.4 se puede observar la estructura de pinInitGpioLpc4337\_t.

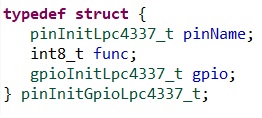


Figura 2.2: Estructura de pinInitGpioLpc4337\_t

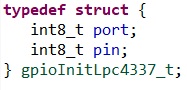


Figura 2.3: estructura de gpioInitLpc4337\_t

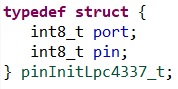


Figura 2.4: estructura de pinInitLpc4337\_t

INTERRUPCIONES POR PIN

La estructura que representa los registros de la SCU es LPC\_SCU\_T que .se muestra en la figura 2.5, donde lo importante para las interrupciones es el atributo PINTSEL.

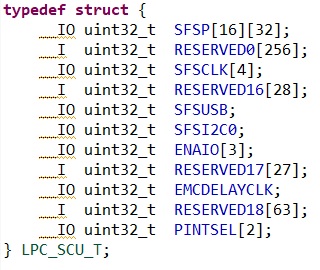


Figura 2.5: estructura que representa los registros en SCU.

Para poder acceder a estos registros se encuentra la etiqueta LPC\_SCU\_BASE que posee el valor 0x40086000 que es la dirección base de dichos registros. También existe la linea **#define** LPC\_SCU ((LPC\_SCU\_T\*) LPC\_SCU\_BASE) que genera un puntero a LPC\_SCU\_T con el valor de la dirección de inicio.

La estructura que representa los registros de GPIO para el manejo de interrupción por pin es LPC\_PIN\_INT\_T, que se muestra en la figura 2.6.

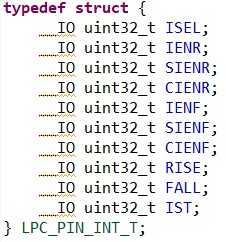


Figura 2.6: estructura que representa los registros de interrupción por pin.

Las líneas

**#define** LPC\_PIN\_INT\_BASE 0x40087000 y

**#define** LPC\_GPIO\_PIN\_INT ((LPC\_PIN\_INT\_T\*) LPC\_PIN\_INT\_BASE)

Generan un puntero a la dirección inicial de dichos registros.

INTERRUPCIONES POR GRUPO

La estructura que representa los registros de GPIO para el manejo de interrupción por grupo es LPC\_GPIOGROUPINT\_T, que se muestra en la figura 2.7.

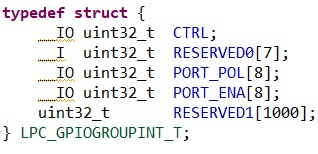


Figura 2.6: estructura que representa los registros de interrupción por grupo.

En el caso del grupo 0, las etiquetas

**#define** LPC\_GPIO\_GROUP\_INT0\_BASE 0x40088000 y

**#define** LPC\_GPIOGROUP ((LPC\_GPIOGROUPINT\_T\*) LPC\_GPIO\_GROUP\_INT0\_BASE)

generan un puntero a la dirección inicial de dichos registros.

En el caso del grupo 1 está la etiqueta

**#define** LPC\_GPIO\_GROUP\_INT1\_BASE 0x40089000

Que marca el inicio de los registros de este grupo.

1. Primitivas

IO: Nivel Chip (NXP)

**void** **Chip\_GPIO\_Init**(LPC\_GPIO\_T \*pGPIO): inicializa el bloque GPIO. pGPIO debe contener la dirección de inicio de los registros de gpio.

Archivo: gpio\_18xx\_43xx.h

**void** **Chip\_GPIO\_DeInit**(LPC\_GPIO\_T \*pGPIO): des inicializa el bloque GPIO.

Archivo: gpio\_18xx\_43xx.h

STATIC INLINE **void** **Chip\_GPIO\_SetDir**(LPC\_GPIO\_T \*pGPIO, uint8\_t portNum, uint32\_t bitValue, uint8\_t out): En base a los bits que están en 1 en bitvalue pone los pines como entrada (out = 0), o salida (out = 1).

Archivo: gpio\_18xx\_43xx.h

STATIC INLINE **void** **Chip\_GPIO\_SetPinState**(LPC\_GPIO\_T \*pGPIO, uint8\_t port, uint8\_t pin, **bool** setting): Escribe el estado del pin modificando el grupo B.

Archivo: gpio\_18xx\_43xx.h

STATIC INLINE **bool** **Chip\_GPIO\_ReadPortBit**(LPC\_GPIO\_T \*pGPIO, uint32\_t port, uint8\_t pin): Lee el estado del pin a partir del grupo B de registros.

Archivo: gpio\_18xx\_43xx.h

Funciones Útiles:

STATIC INLINE **void** **Chip\_GPIO\_SetPinToggle**(LPC\_GPIO\_T \*pGPIO, uint8\_t port, uint8\_t pin): Togglea el valor de un pin.

Archivo: gpio\_18xx\_43xx.h

IO: Nivel Sapi

**static** **void** **gpioObtainPinInit**( gpioMap\_t pin, int8\_t \*pinNamePort, int8\_t \*pinNamePin, int8\_t \*func, int8\_t \*gpioPort, int8\_t \*gpioPin ): obtiene el valor de pin físico, puerto físico, función gpio, pin gpio y puerto gpio a partir de la etiqueta de la EDU-CIAA.

Archivo: sapi\_gpio.c

De uso común:

bool\_t **gpioInit**( gpioMap\_t pin, gpioInit\_t config ): Si se envía “ENABLE” en config, activa la función de gpio del microcontrolador. En otro caso establece la función gpio del pin seleccionado, en base a config lo puede poner como salida o como entrada. Si es entrada se puede poner con o sin resistencia de pullup y/o pulldown. Devuelve false si si pasan VCC o GND como pin o si la configuración no es correcta.

Archivo: sapi\_gpio.c

bool\_t **gpioWrite**( gpioMap\_t pin, bool\_t value ): Escribe el valor de un pin. Se debe haber puesto como salida con **gpioInit** antes de usarla.

Archivo: sapi\_gpio.c

bool\_t **gpioToggle**( gpioMap\_t pin ): Alterna el valor del pin si está como salida. Se debe haber puesto como salida con **gpioInit** antes de usarla.

Archivo: sapi\_gpio.c

bool\_t **gpioRead**( gpioMap\_t pin ): Lee el valor del pin. Se debe haber inicializado con **gpioInit** antes de usarla.

Archivo: sapi\_gpio.c

INTERRUPCIONES POR PIN: Nivel Chip (NXP)

STATIC INLINE **void** **Chip\_SCU\_GPIOIntPinSel**(uint8\_t PortSel, uint8\_t PortNum, uint8\_t PinNum): Asocia una interrupción por pin a un pin de la GPIO.

Archivo: scu\_18xx\_43xx.h

STATIC INLINE **void** **Chip\_PININT\_Init**(LPC\_PIN\_INT\_T \*pPININT): Inicializa el bloque de interrupciones por pin. Se debe usar despué de la función Chip\_GPIO\_Init.

Archivo: pinint\_18xx\_43xx.h

STATIC INLINE **void** **Chip\_PININT\_DeInit**(LPC\_PIN\_INT\_T \*pPININT): des inicializa el bloque de interrupciones por pin.

Archivo: pinint\_18xx\_43xx.h

STATIC INLINE **void** **Chip\_PININT\_SetPinModeEdge**(LPC\_PIN\_INT\_T \*pPININT, uint32\_t pins): setea el trigger por flanco.

Archivo: pinint\_18xx\_43xx.h

STATIC INLINE **void** **Chip\_PININT\_SetPinModeLevel**(LPC\_PIN\_INT\_T \*pPININT, uint32\_t pins): setea el trigger por nivel.

Archivo: pinint\_18xx\_43xx.h

STATIC INLINE **void** **Chip\_PININT\_EnableIntHigh**(LPC\_PIN\_INT\_T \*pPININT, uint32\_t pins): habilita interrupción por flanco ascendente o por nivel.

Archivo: pinint\_18xx\_43xx.h

STATIC INLINE **void** **Chip\_PININT\_DisableIntHigh**(LPC\_PIN\_INT\_T \*pPININT, uint32\_t pins): deshabilita interrupción por flanco ascendente o por nivel.

Archivo: pinint\_18xx\_43xx.h

STATIC INLINE **void** **Chip\_PININT\_EnableIntLow**(LPC\_PIN\_INT\_T \*pPININT, uint32\_t pins): habilita interrupción por flanco descendente o pone el nivel en LOW para interrupción por nivel.

Archivo: pinint\_18xx\_43xx.h

STATIC INLINE **void** **Chip\_PININT\_DisableIntLow**(LPC\_PIN\_INT\_T \*pPININT, uint32\_t pins): deshabilita interrupción por flanco descendente o pone el nivel en HIGH para interrupción por nivel.

Archivo: pinint\_18xx\_43xx.h

STATIC INLINE uint32\_t **Chip\_PININT\_GetIntStatus**(LPC\_PIN\_INT\_T \*pPININT): obtiene el estado de la interrupción.

Archivo: pinint\_18xx\_43xx.h

INTERRUPCIONES POR GRUPO: Nivel Chip (NXP)

STATIC INLINE **bool** **Chip\_GPIOGP\_GetIntStatus**(LPC\_GPIOGROUPINT\_T \*pGPIOGPINT, uint8\_t group): obtiene el estado de la interrupción.

Archivo: pinint\_18xx\_43xx.h

STATIC INLINE **void** **Chip\_GPIOGP\_SelectOrMode**(LPC\_GPIOGROUPINT\_T \*pGPIOGPINT, uint8\_t group) : pone la interruption en modo OR.

Archivo: pinint\_18xx\_43xx.h

STATIC INLINE **void** **Chip\_GPIOGP\_SelectAndMode**(LPC\_GPIOGROUPINT\_T \*pGPIOGPINT, uint8\_t group) : pone la interruption en modo AND.

Archivo: pinint\_18xx\_43xx.h

STATIC INLINE **void** **Chip\_GPIOGP\_SelectEdgeMode**(LPC\_GPIOGROUPINT\_T \*pGPIOGPINT, uint8\_t group) : pone el modo de activación en flanco.

Archivo: pinint\_18xx\_43xx.h

STATIC INLINE **void** **Chip\_GPIOGP\_SelectLevelMode**(LPC\_GPIOGROUPINT\_T \*pGPIOGPINT, uint8\_t group) : pone el modo de activación en nivel.

Archivo: pinint\_18xx\_43xx.h

STATIC INLINE **void** **Chip\_GPIOGP\_SelectLowLevel**(LPC\_GPIOGROUPINT\_T \*pGPIOGPINT, uint8\_t group, uint8\_t port, uint32\_t pinMask) : setea el modo de activación en estado LOW.

Archivo: pinint\_18xx\_43xx.h

STATIC INLINE **void** **Chip\_GPIOGP\_SelectHighLevel**(LPC\_GPIOGROUPINT\_T \*pGPIOGPINT, uint8\_t group, uint8\_t port, uint32\_t pinMask) : setea el modo de activación en estado HIGH.

Archivo: pinint\_18xx\_43xx.h

STATIC INLINE **void** **Chip\_GPIOGP\_DisableGroupPins**(LPC\_GPIOGROUPINT\_T \*pGPIOGPINT, uint8\_t group, uint8\_t port, uint32\_t pinMask) : Desasocia un pin a la interrupción.

Archivo: pinint\_18xx\_43xx.h

STATIC INLINE **void** **Chip\_GPIOGP\_EnableGroupPins**(LPC\_GPIOGROUPINT\_T \*pGPIOGPINT, uint8\_t group, uint8\_t port, uint32\_t pinMask) : Asocia un pin a la interrupción.

Archivo: pinint\_18xx\_43xx.h

1. Diagrama

