Registros y GPIO

Mariana Prieto
Informática II - R2004

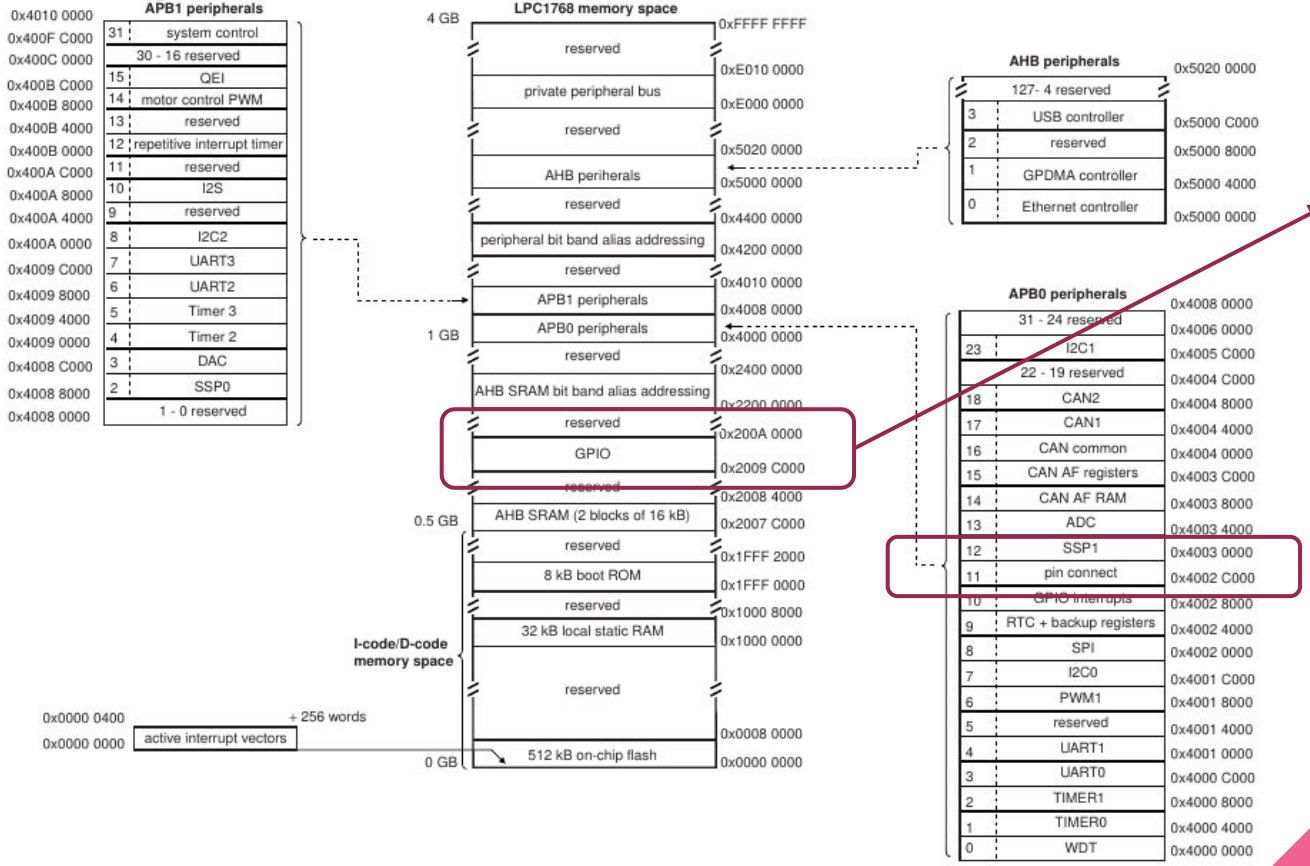
Recordando...

Memoria → ***ALMACENAMIENTO***

Registro →

- ***CONFIGURACIÓN***
- ***SU VALOR SE REFLEJA EN ALGUNA ACTIVIDAD INTERNA O SUCESO EXTERNO***

Arquitectura Cortex M3 - GPIO



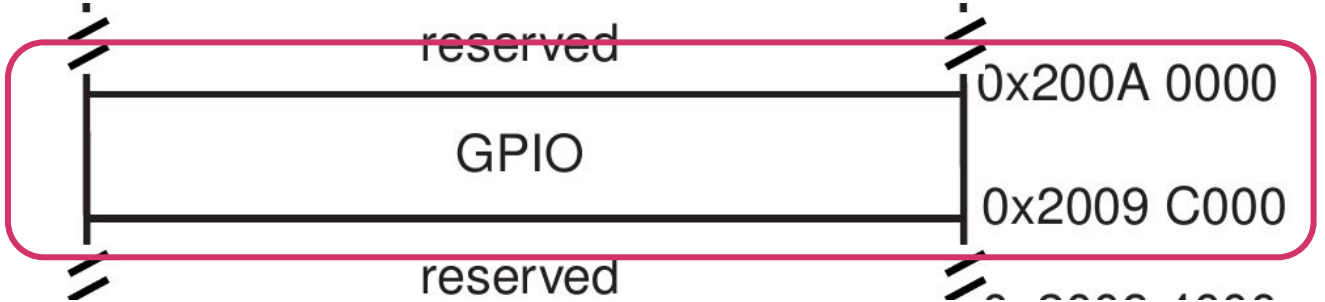
Vamos a la hoja de datos sección GPIO

Registros de configuración

Registros GPIO

Configuración usos pines

Configuración y Acceso GPIO



Configuración: ¿Para que voy a usar el pin?

Table 74. Summary of PINSEL registers

Register	Controls	Table
PINSEL0	P0[15:0]	Table 79
PINSEL1	P0 [31:16]	Table 80
PINSEL2	P1 [15:0] (Ethernet)	Table 81
PINSEL3	P1 [31:16]	Table 82
PINSEL4	P2 [15:0]	Table 83
PINSEL5	P2 [31:16]	not used
PINSEL6	P3 [15:0]	not used
PINSEL7	P3 [31:16]	Table 84
PINSEL8	P4 [15:0]	not used
PINSEL9	P4 [31:16]	Table 85
PINSEL10	Trace port enable	Table 86

Configuración cada puerto

Table 75. Pin function select register bits

PINSEL0 to PINSEL9 Values	Function
00	Primary (default) function, typically GPIO port
01	First alternate function
10	Second alternate function
11	Third alternate function

Opciones de configuración

Configurando PINSEL

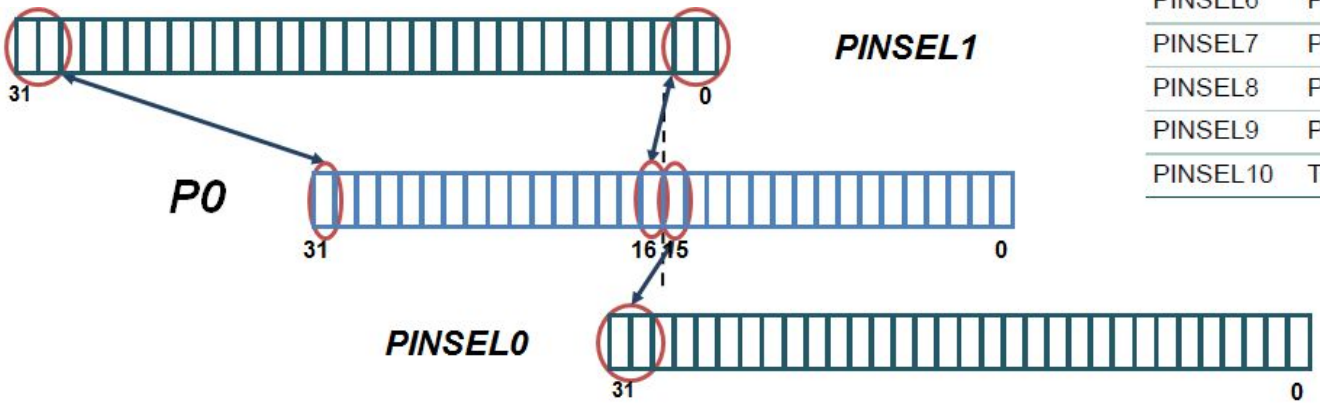
Pin Function Select Register 0 *(PINSEL0 - 0x4002C000)*

Table 75. Pin function select register bits

PINSEL0 to PINSEL9 Values	Function
00	Primary (default) function, typically GPIO port
01	First alternate function
10	Second alternate function
11	Third alternate function

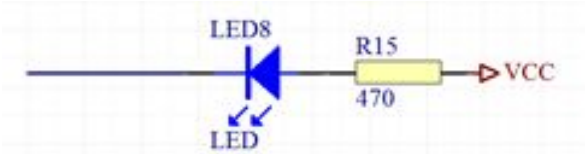
Table 74. Summary of PINSEL registers

Register	Controls	Table
PINSEL0	P0[15:0]	Table 79
PINSEL1	P0 [31:16]	Table 80
PINSEL2	P1 [15:0] (Ethernet)	Table 81
PINSEL3	P1 [31:16]	Table 82
PINSEL4	P2 [15:0]	Table 83
PINSEL5	P2 [31:16]	not used
PINSEL6	P3 [15:0]	not used
PINSEL7	P3 [31:16]	Table 84
PINSEL8	P4 [15:0]	not used
PINSEL9	P4 [31:16]	Table 85
PINSEL10	Trace port enable	Table 86



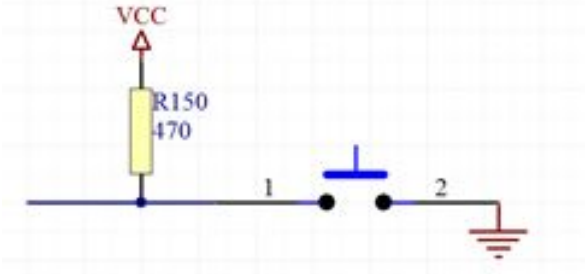
¿Para que uso el pin GPIO?

Entrada



0 prende, 1 apaga

Salida



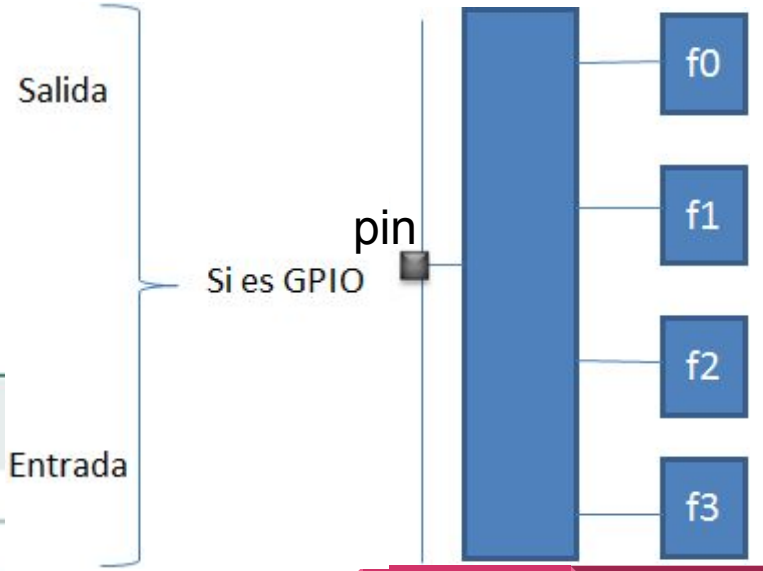
Siempre 1, 0 pulsado

VCC	GND
1	0

Tipos de entradas y salidas

PINMODE_OD0 to PINMODE_OD4 Values	Function
0	Pin is in the normal (not open drain) mode.
1	Pin is in the open drain mode.

PINMODE0 to PINMODE9 Values	Function
00	Pin has an on-chip pull-up resistor enabled.
01	Repeater mode
10	Pin has neither pull-up nor pull-down resistor enabled.
11	Pin has an on-chip pull-down resistor enabled.



Como entradas - PINMODEx

PINMODE0	Pin mode select register 0	R/W	0	0x4002 C040
PINMODE1	Pin mode select register 1	R/W	0	0x4002 C044
PINMODE2	Pin mode select register 2	R/W	0	0x4002 C048
PINMODE3	Pin mode select register 3.	R/W	0	0x4002 C04C
PINMODE4	Pin mode select register 4	R/W	0	0x4002 C050
PINMODE5	Pin mode select register 5	R/W	0	0x4002 C054
PINMODE6	Pin mode select register 6	R/W	0	0x4002 C058
PINMODE7	Pin mode select register 7	R/W	0	0x4002 C05C
PINMODE9	Pin mode select register 9	R/W	0	0x4002 C064
PINMODE_OD0	Open drain mode control register 0	R/W	0	0x4002 C068

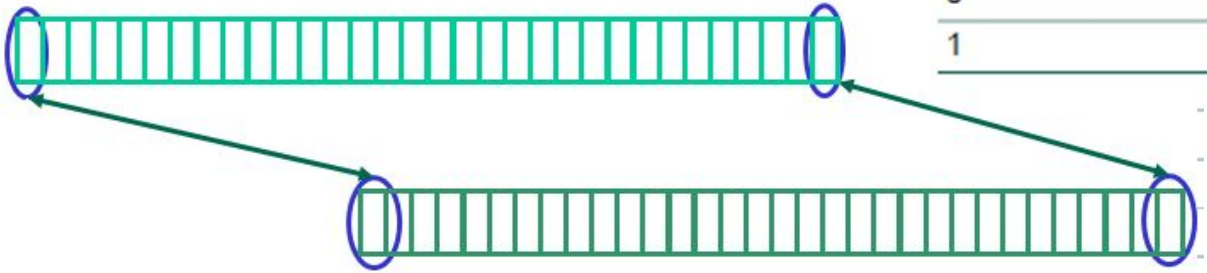
Pin Mode Select register 0 *(PINMODE0 - 0x4002C040)*

En los registros PINMODEx al igual que en los PINSELx se requieren 2 bits por cada pin a configurar. Por ello para P0 usaremos PINMODE0 y PINMODE1, para P1 PINMODE2 y PINMODE3, etc…

Como Salidas PINMODE_ODx

PINMODE9	Pin mode select register 9	R/W	0	0x4002 C064
PINMODE_OD0	Open drain mode control register 0	R/W	0	0x4002 C068
PINMODE_OD1	Open drain mode control register 1	R/W	0	0x4002 C06C
PINMODE_OD2	Open drain mode control register 2	R/W	0	0x4002 C070
PINMODE_OD3	Open drain mode control register 3	R/W	0	0x4002 C074
PINMODE_OD4	Open drain mode control register 4	R/W	0	0x4002 C078
I2CPADCFG	I^2C Pin Configuration register	R/W	0	0x4002 C07C

Ejemplo para P0



PINMODE_OD0 to PINMODE_OD4 Values	Function
0	Pin is in the normal (not open drain) mode.
1	Pin is in the open drain mode.

Name	Description	Access	Reset Value[1]	Address
PINSEL0	Pin function select register 0.	R/W	0	0x4002 C000
PINSEL1	Pin function select register 1.	R/W	0	0x4002 C004
PINSEL2	Pin function select register 2.	R/W	0	0x4002 C008
PINSEL3	Pin function select register 3.	R/W	0	0x4002 C00C
PINSEL4	Pin function select register 4	R/W	0	0x4002 C010
PINSEL7	Pin function select register 7	R/W	0	0x4002 C01C
PINSEL8	Pin function select register 8	R/W	0	0x4002 C020
PINSEL9	Pin function select register 9	R/W	0	0x4002 C024
PINSEL10	Pin function select register 10	R/W	0	0x4002 C028
PINMODE0	Pin mode select register 0	R/W	0	0x4002 C040
PINMODE1	Pin mode select register 1	R/W	0	0x4002 C044
PINMODE2	Pin mode select register 2	R/W	0	0x4002 C048
PINMODE3	Pin mode select register 3.	R/W	0	0x4002 C04C
PINMODE4	Pin mode select register 4	R/W	0	0x4002 C050
PINMODE5	Pin mode select register 5	R/W	0	0x4002 C054
PINMODE6	Pin mode select register 6	R/W	0	0x4002 C058
PINMODE7	Pin mode select register 7	R/W	0	0x4002 C05C
PINMODE9	Pin mode select register 9	R/W	0	0x4002 C064
PINMODE_OD0	Open drain mode control register 0	R/W	0	0x4002 C068
PINMODE_OD1	Open drain mode control register 1	R/W	0	0x4002 C06C
PINMODE_OD2	Open drain mode control register 2	R/W	0	0x4002 C070
PINMODE_OD3	Open drain mode control register 3	R/W	0	0x4002 C074
PINMODE_OD4	Open drain mode control register 4	R/W	0	0x4002 C078
I2CPADCFG	I^2C Pin Configuration register	R/W	0	0x4002 C07C

Registros en zona de memoria *pin connect*

Como podemos ver PINSELx, PINMODEx, PINMODE_ODx en la memoria se encuentran consecutivos permitiendo acceder al registro n desde el registro 0.

¿Cómo accedemos a los registros?

Como hemos visto los registros que configuran cada puerto son consecutivos en memoria, tanto para PINSEL como para PINMODE y PINMODE_OD, por ello con saber la dirección de comienzo de cada uno de ellos tendremos acceso a la configuración de todos los puertos

```
//!< ////////////////Registros PINSEL////////////////////////////
//!< 0x4002C000UL : Direccion de inicio de los registros PINSEL
#define     PINSEL      ( (__RW uint32_t *) 0x4002C000UL )

#define     PINSEL0     PINSEL[0]   //!< PINSEL0------>P0[15:0]        (0x4002C000)
#define     PINSEL1     PINSEL[1]   //!< PINSEL1------>P0[31:16]       (0x4002C004)
#define     PINSEL2     PINSEL[2]   //!< PINSEL2------>P1[15:0]        (0x4002C008)
#define     PINSEL3     PINSEL[3]   //!< PINSEL3------>P1[31:16]       (0x4002C00C)
#define     PINSEL4     PINSEL[4]   //!< PINSEL4------>P2[15:0]        (0x4002C010)
#define     PINSEL5     PINSEL[5]   //!< PINSEL5------>P2[31:16]       NOT USED
#define     PINSEL6     PINSEL[6]   //!< PINSEL6------>P3[15:0]        NOT USED
#define     PINSEL7     PINSEL[7]   //!< PINSEL7------>P3[31:16]       (0x4002C01C)
#define     PINSEL8     PINSEL[8]   //!< PINSEL8------>P4[15:0]        NOT USED
#define     PINSEL9     PINSEL[9]   //!< PINSEL9------>P4[31:16]       (0x4002C024)
```

Registros en la zona de memoria GPIO

Generic Name	Description	Access	Reset value[1]	PORTn Register Name & Address
FIODIR	Fast GPIO Port Direction control register. This register individually controls the direction of each port pin.	R/W	0	FIO0DIR - 0x2009 C000 FIO1DIR - 0x2009 C020 FIO2DIR - 0x2009 C040 FIO3DIR - 0x2009 C060 FIO4DIR - 0x2009 C080
FIOMASK	Fast Mask register for port. Writes, sets, clears, and reads to port (done via writes to FIOPIN, FIOSET, and FIOCLR, and reads of FIOPIN) alter or return only the bits enabled by zeros in this register.	R/W	0	FIO0MASK - 0x2009 C010 FIO1MASK - 0x2009 C030 FIO2MASK - 0x2009 C050 FIO3MASK - 0x2009 C070 FIO4MASK - 0x2009 C090
FIOPIN	Fast Port Pin value register using FIOMASK. The current state of digital port pins can be read from this register, regardless of pin direction or alternate function selection (as long as pins are not configured as an input to ADC). The value read is masked by ANDing with inverted FIOMASK. Writing to this register places corresponding values in all bits enabled by zeros in FIOMASK. **Important:** if an FIOPIN register is read, its bit(s) masked with 1 in the FIOMASK register will be read as 0 regardless of the physical pin state.	R/W	0	FIO0PIN - 0x2009 C014 FIO1PIN - 0x2009 C034 FIO2PIN - 0x2009 C054 FIO3PIN - 0x2009 C074 FIO4PIN - 0x2009 C094
FIOSET	Fast Port Output Set register using FIOMASK. This register controls the state of output pins. Writing 1s produces highs at the corresponding port pins. Writing 0s has no effect. Reading this register returns the current contents of the port output register. Only bits enabled by 0 in FIOMASK can be altered.	R/W	0	FIO0SET - 0x2009 C018 FIO1SET - 0x2009 C038 FIO2SET - 0x2009 C058 FIO3SET - 0x2009 C078 FIO4SET - 0x2009 C098
FIOCLR	Fast Port Output Clear register using FIOMASK. This register controls the state of output pins. Writing 1s produces lows at the corresponding port pins. Writing 0s has no effect. Only bits enabled by 0 in FIOMASK can be altered.	WO	0	FIO0CLR - 0x2009 C01C FIO1CLR - 0x2009 C03C FIO2CLR - 0x2009 C05C FIO3CLR - 0x2009 C07C FIO4CLR - 0x2009 C09C

Establezco Dirección
(0 = entrada -- 1 = salida)

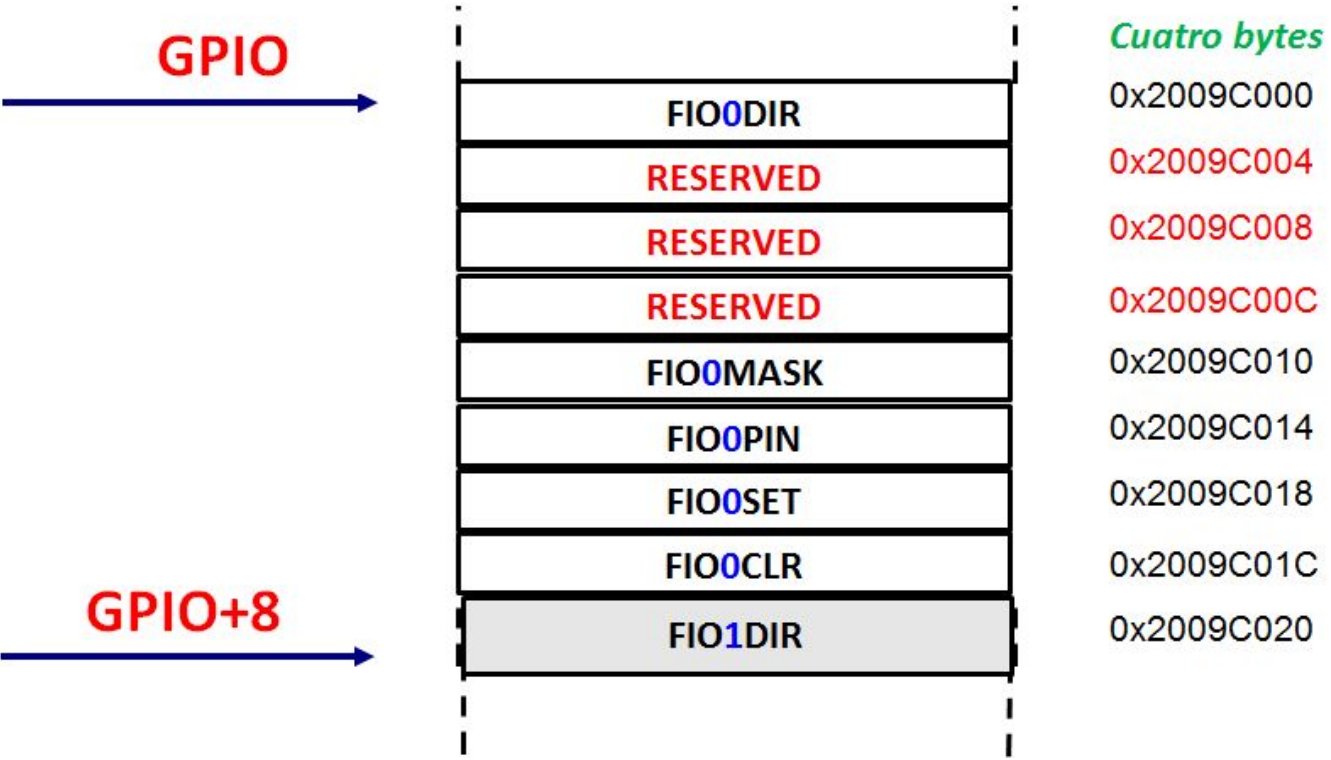
máscara (0 = enable)

Lectura/escritura de pin
(FIOMASK = 0, todo habilitado)

Con un “1” Escribo un 1

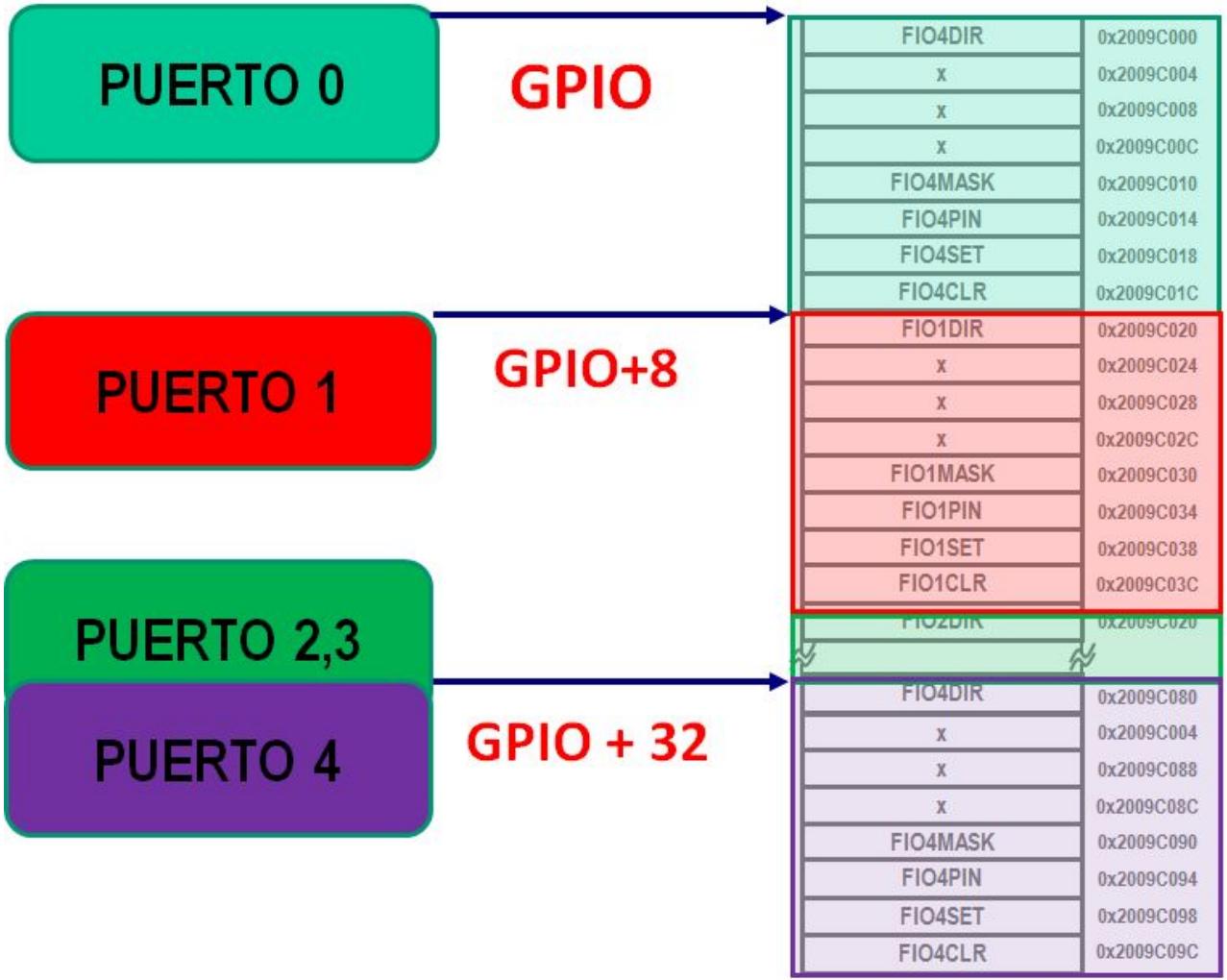
Con un “1” escribo un 0

Mapa de memorias de las GPIO



#define **GPIO**
((**uint32_t***)
0x2009C000UL)

Mapa de memoria de las GPIO



¿Cómo accedemos a los registros?

```
//!< ////////////////// REGISTROS GPIOs ////////////////////////////
//!< 0x2009C000UL : Direccion de inicio de los registros de GPIOs
#define GPIOs       ( (__RW uint32_t *) 0x2009C000UL )


#define     FIO0DIR     GPIOs[0 + 0*8]  //!< 0x2009C000
#define     FIO0MASK    GPIOs[4 + 0*8]  //!< 0x2009C010
#define     FIO0PIN     GPIOs[5 + 0*8]  //!< 0x2009C014
#define     FIO0SET     GPIOs[6 + 0*8]  //!< 0x2009C018
#define     FIO0CLR     GPIOs[7 + 0*8]  //!< 0x2009C01C

#define     FIO1DIR     GPIOs[0 + 1*8]  //!< 0x2009C020
#define     FIO1MASK    GPIOs[4 + 1*8]  //!< 0x2009C030
#define     FIO1PIN     GPIOs[5 + 1*8]  //!< 0x2009C034
#define     FIO1SET     GPIOs[6 + 1*8]  //!< 0x2009C038
#define     FIO1CLR     GPIOs[7 + 1*8]  //!< 0x2009C03C
```

Resumiendo

Para lograr que salga o entre la señal deseada deberemos:

* Configurar el puerto

  Registros **FIODIR PINMODE PINMODE_OD PINSEL**

* Enviar el dato para que se refleje en los pines correspondientes

  Registros **FIOPIN FIOSET FIOCLR**

Hagamos nuestras funciones

```
/*****************************************************************************
    \fn  void SetPINSEL(uint8_t port, uint8_t pin, uint8_t sel)
    \brief: Configuracion de funciones de los pines
    \details: Configura el registro PINSEL correspondiente para configurar el pin con la funcion requerida
    \author: Pablo Irrera Condines
    \param  [in] port Puerto a configurar
    \param  [in] pin Pin a configurar
    \param  [in] sel Funcion del pin. Puede ser:
            PINSEL_GPIO
            PINSEL_FUNC1
            PINSEL_FUNC2
            PINSEL_FUNC3
    \return void
*/
void SetPINSEL(uint8_t port, uint8_t pin, uint8_t sel);

/*****************************************************************************
    \fn  void SetPINMODE(uint8_t port, uint8_t pin, uint8_t mode)
    \brief: Configuracion de pull-up/pull-down
    \details: Configura el registro PINMODE correspondiente para configurar el pin requerido
    \author: Pablo Irrera Condines
    \param  [in] port Puerto a configurar
    \param  [in] pin Pin a configurar
    \param  [in] mode Funcion del pin. Puede ser:
            PINMODE_PULLUP
            PINMODE_REPEAT
            PINMODE_NONE
            PINMODE_PULLDOWN
    \return void
*/
void SetPINMODE(uint8_t port, uint8_t pin, uint8_t mode);
```

Hagamos nuestras funciones

```
/*****************************************************************************
    \fn  void SetDIR(uint8_t port, uint8_t pin, uint8_t dir)
    \brief: Configuracion de GPIO como entrada o salida
    \details: Configura el registro FIODIR correspondiente para configurar el pin requerido
    \author: Pablo Irrera Condines
    \param  [in] port Puerto a configurar
    \param  [in] pin Pin a configurar
    \param  [in] dir Direccion del pin GPIO. Puede ser:
            GPIO_INPUT
            GPIO_OUTPUT
    \return void
*/
void SetDIR(uint8_t port, uint8_t pin, uint8_t dir);

/*****************************************************************************
    \fn  void SetPIN(uint8_t port, uint8_t pin, uint8_t value)
    \brief: Setear valor de pin GPIO
    \details: Escribe un valor en un pin de GPIO configurado como salida
    \author: Pablo Irrera Condines
    \param  [in] port Puerto a configurar
    \param  [in] pin Pin a configurar
    \param  [in] value Valor del pin GPIO. Valores posibles: 0 o 1
    \return void
*/
void SetPIN(uint8_t port, uint8_t pin, uint8_t value);
```

Hagamos nuestras funciones

```
/**************************************************************************************
    \fn  uint8_t GetPIN(uint8_t port, uint8_t pin)
    \brief: Leer valor de pin GPIO
    \details: Lee el valor de un pin de GPIO presente en el registro FIOPIN correspondiente
    \author: Pablo Irrera Condines
    \param  [in] port Puerto a configurar
    \param  [in] pin Pin a configurar
    \return 0 o 1
*/
uint8_t GetPIN(uint8_t port, uint8_t pin);
```