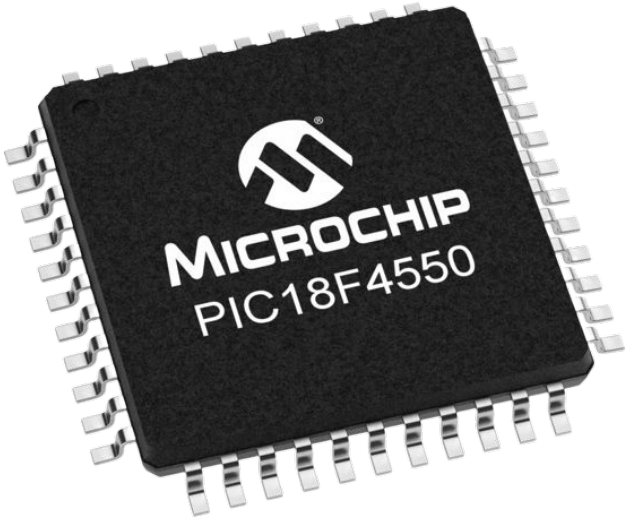
Wels

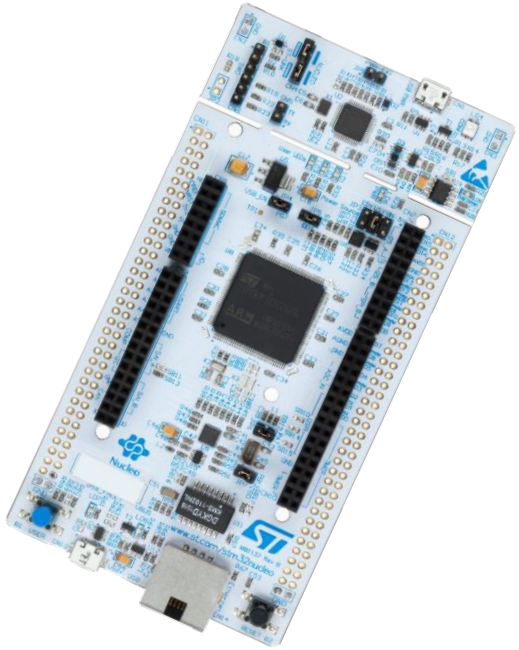
C Embebido

@welstheory
hola@welstheory.com
www.welstheory.com

Wels

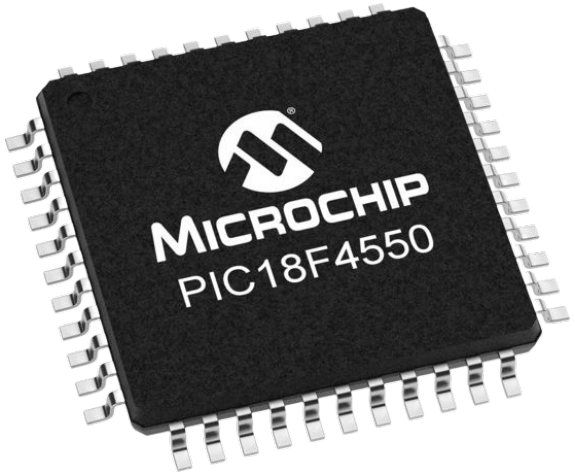
PIC18 y STM32





Vamos a utilizar punteros

Wels

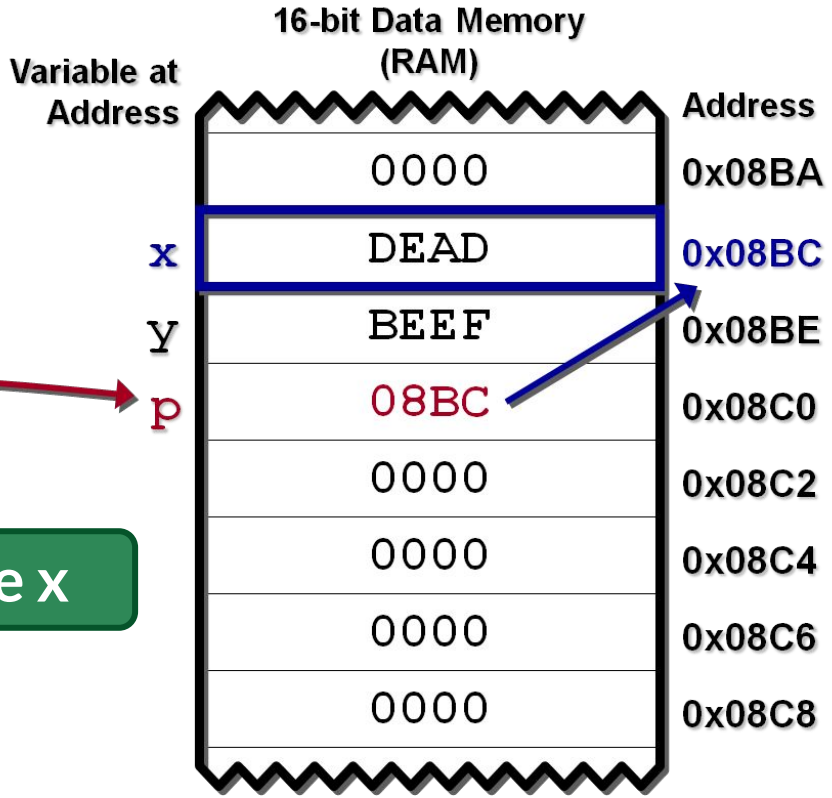


Programa

```c
#include <xc.h>
#include <stdint.h>
#include "Configuracion.h"

int main(void)
{
    TRISD = 0x00;
    while(1)
    {
        LATD = 0x01; // Encendiendo el LED
        __delay_ms(100);
        LATD = 0x00; // Encendiendo el LED
        __delay_ms(100);
    }
    return 0;
}
```

Ejemplo de Punteros

```
{
  int x, y;
  int *p;

  x = 0xDEAD;
  y = 0xBEEF;
  p = &x;

}
```

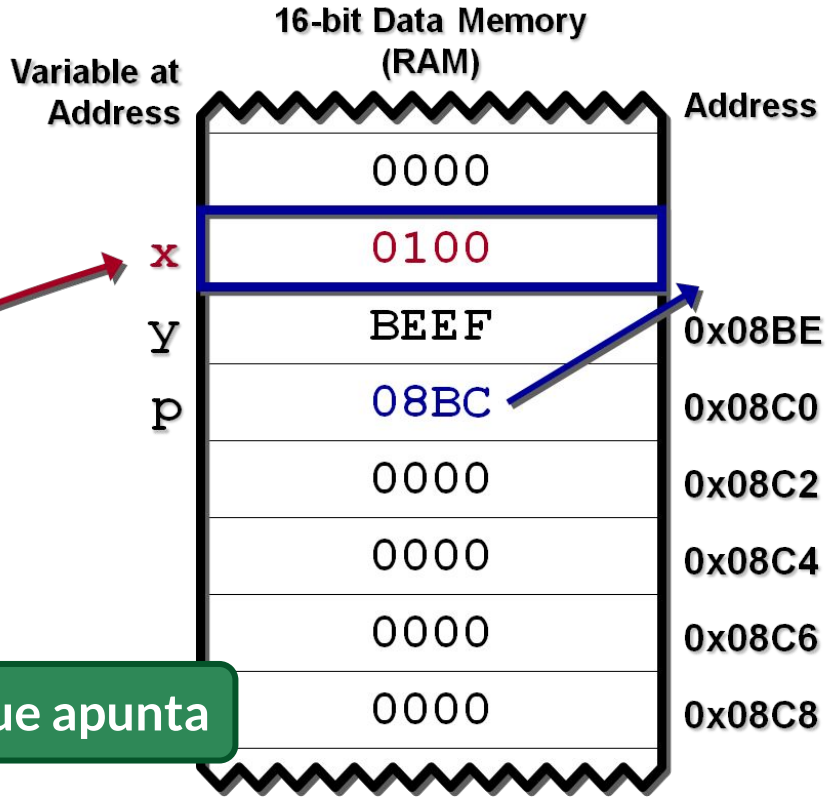


Le asignamos la dirección de x

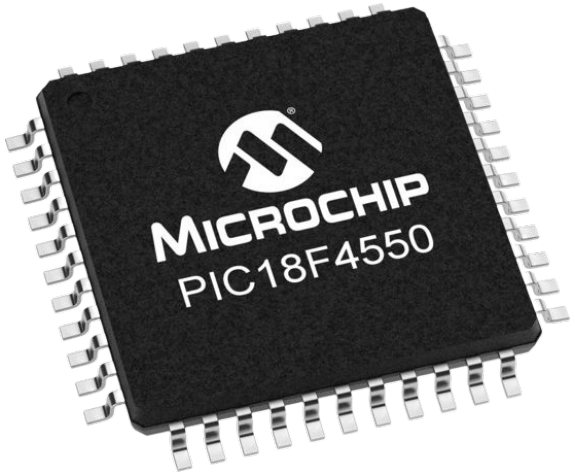
Ejemplo de Punteros

Wels

```
{
  int x, y;
  int *p;

  x = 0xDEAD;
  y = 0xBEEF;
  p = &x;

  *p = 0x0100;
```



Asignar el valor a la variable que apunta

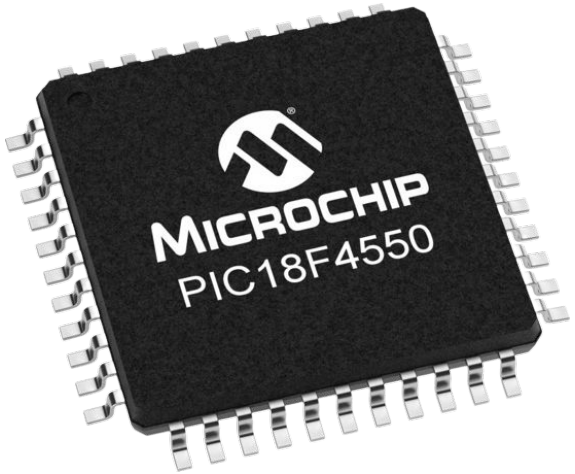


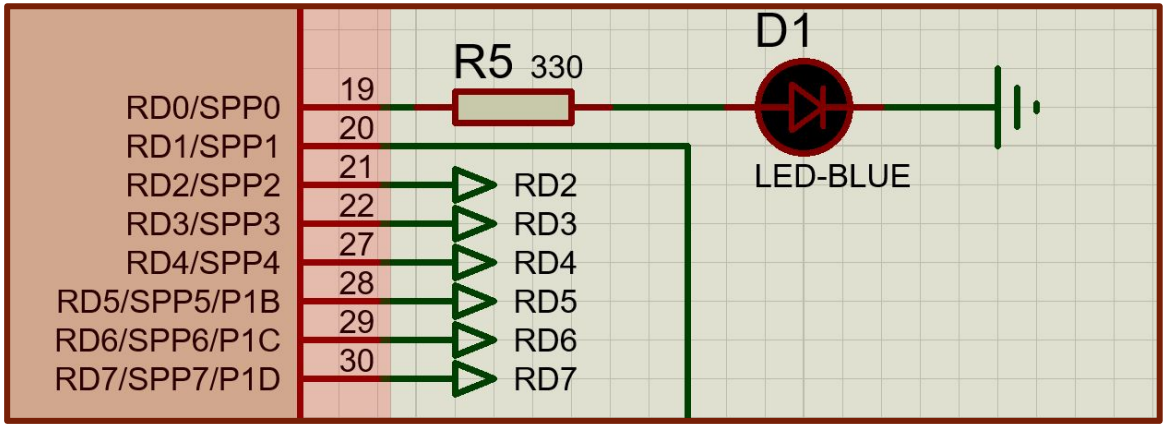
PIC184550

TABLE 5-1: SPECIAL FUNCTION REGISTER MAP

Address	Name	Address	Name	Address	Name	Address	Name	Address	Name
FFFh	TOSU	FDFh	INDF2(1)	FBFh	CCPR1H	F9Fh	IPR1	F7Fh	UEP15
FFEh	TOSH	FDEh	POSTINC2(1)	FBEh	CCPR1L	F9Eh	PIR1	F7Eh	UEP14
FFDh	TOSL	FDDh	POSTDEC2(1)	FBDh	CCP1CON	F9Dh	PIE1	F7Dh	UEP13
FFCh	STKPTR	FDCh	PREINC2(1)	FBCh	CCPR2H	F9Ch	—(2)	F7Ch	UEP12
FFBh	PCLATU	FDBh	PLUSW2(1)	FBBh	CCPR2L	F9Bh	OSCTUNE	F7Bh	UEP11
FFAh	PCLATH	FDAh	FSR2H	FBAh	CCP2CON	F9Ah	—(2)	F7Ah	UEP10
FF9h	PCL	FD9h	FSR2L	FB9h	—(2)	F99h	—(2)	F79h	UEP9
FF8h	TBLPTRU	FD8h	STATUS	FB8h	BAUDCON	F98h	—(2)	F78h	UEP8
FF7h	TBLPTRH	FD7h	TMR0H	FB7h	ECCP1DEL	F97h	—(2)	F77h	UEP7
FF6h	TBLPTRL	FD6h	TMR0L	FB6h	ECCP1AS	F96h	TRISE(3)	F76h	UEP6
FF5h	TABLAT	FD5h	T0CON	FB5h	CVRCON	F95h	TRISD(3)	F75h	UEP5
FF4h	PRODH	FD4h	—(2)	FB4h	CMCON	F94h	TRISC	F74h	UEP4
FF3h	PRODL	FD3h	OSCCON	FB3h	TMR3H	F93h	TRISB	F73h	UEP3
FF2h	INTCON	FD2h	HLVDCON	FB2h	TMR3L	F92h	TRISA	F72h	UEP2
FF1h	INTCON2	FD1h	WDTCON	FB1h	T3CON	F91h	—(2)	F71h	UEP1
FF0h	INTCON3	FD0h	RCON	FB0h	SPBRGH	F90h	—(2)	F70h	UEP0
FEFh	INDF0(1)	FCFh	TMR1H	FAFh	SPBRG	F8Fh	—(2)	F6Fh	UCFG
FEEh	POSTINC0(1)	FCEh	TMR1L	FAEh	RCREG	F8Eh	—(2)	F6Eh	UADDR
FEDh	POSTDEC0(1)	FCDh	T1CON	FADh	TXREG	F8Dh	LATE(3)	F6Dh	UCON
FECh	PREINC0(1)	FCCh	TMR2	FACh	TXSTA	F8Ch	LATD(3)	F6Ch	USTAT
FEBh	PLUSW0(1)	FCBh	PR2	FABh	RCSTA	F8Bh	LATC	F6Bh	UEIE
FEAh	FSR0H	FCAh	T2CON	FAAh	—(2)	F8Ah	LATB	F6Ah	UEIR
FE9h	FSR0L	FC9h	SSPBUF	FA9h	EEADR	F89h	LATA	F69h	UIE
FE8h	WREG	FC8h	SSPADD	FA8h	EEDATA	F88h	—(2)	F68h	UIR
FE7h	INDF1(1)	FC7h	SSPSTAT	FA7h	EECON2(1)	F87h	—(2)	F67h	UFRMH
FE6h	POSTINC1(1)	FC6h	SSPCON1	FA6h	EECON1	F86h	—(2)	F66h	UFRML
FE5h	POSTDEC1(1)	FC5h	SSPCON2	FA5h	—(2)	F85h	—(2)	F65h	SPPCON(3)
FE4h	PREINC1(1)	FC4h	ADRESH	FA4h	—(2)	F84h	PORTE	F64h	SPPEPS(3)
FE3h	PLUSW1(1)	FC3h	ADRESL	FA3h	—(2)	F83h	PORTD(3)	F63h	SPPCFG(3)
FE2h	FSR1H	FC2h	ADCON0	FA2h	IPR2	F82h	PORTC	F62h	SPPDATA(3)
FE1h	FSR1L	FC1h	ADCON1	FA1h	PIR2	F81h	PORTB	F61h	—(2)
FE0h	BSR	FC0h	ADCON2	FA0h	PIE2	F80h	PORTA	F60h	—(2)

Wels

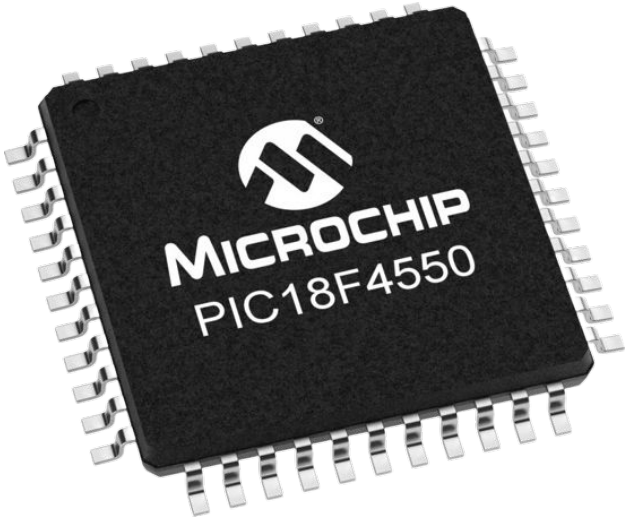


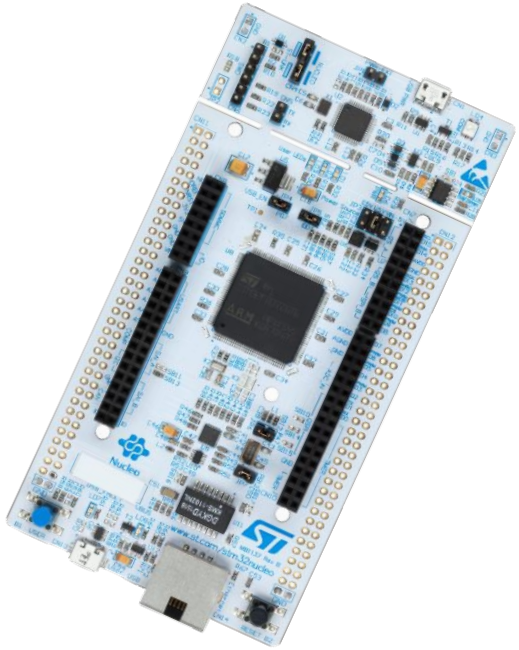


PIC184550

Wels

PIC18 y STM32



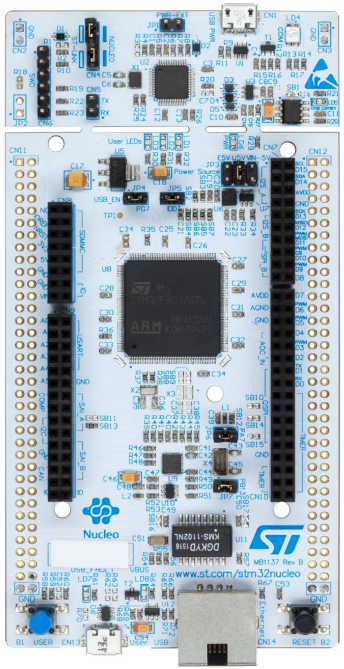


Vamos a utilizar punteros

Wels

USAREMOS EL STM32CubeIDE

STM32 - F429



- El microcontrolador de la placa tiene 144 pines.
- Cada GPIO tiene 16 pines
- Excepto el GPIO H que sólo tiene dos pines.

A - B - C - D - E - F - G - H - I

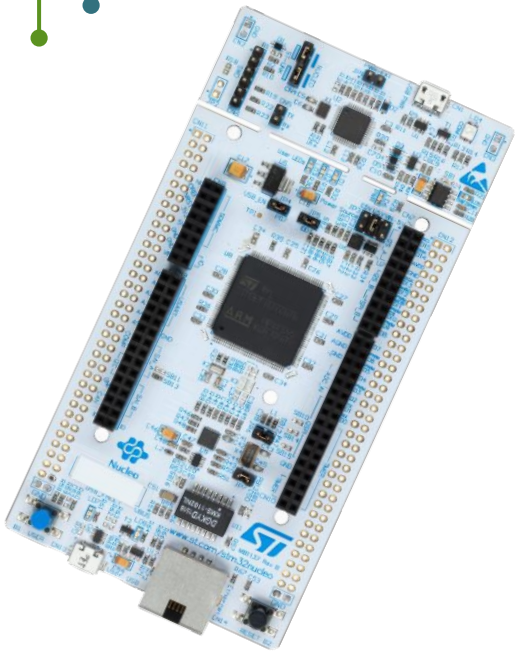
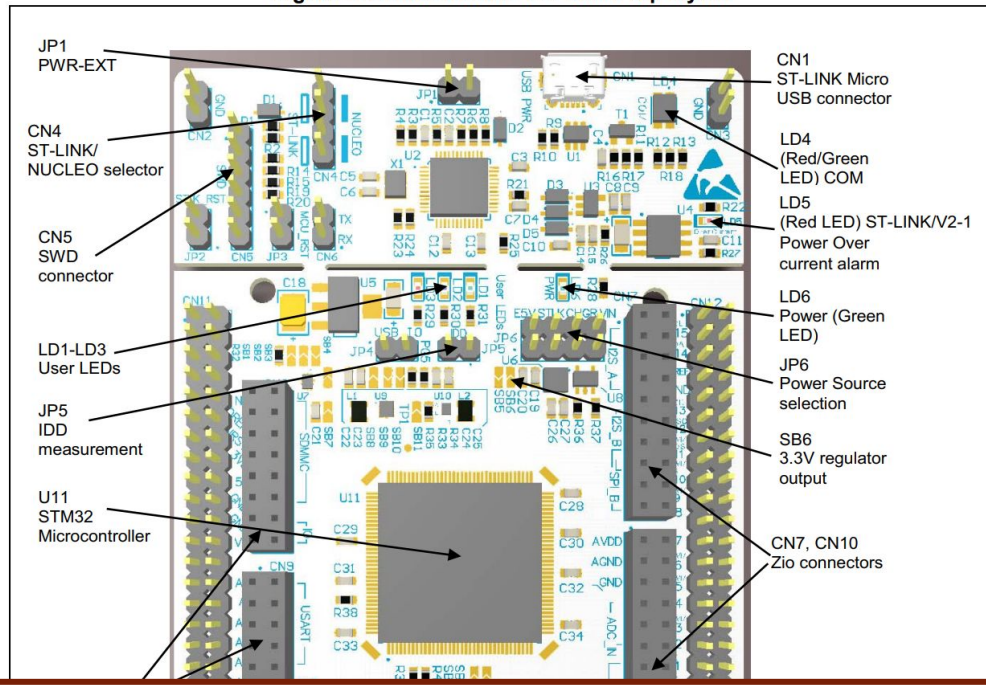
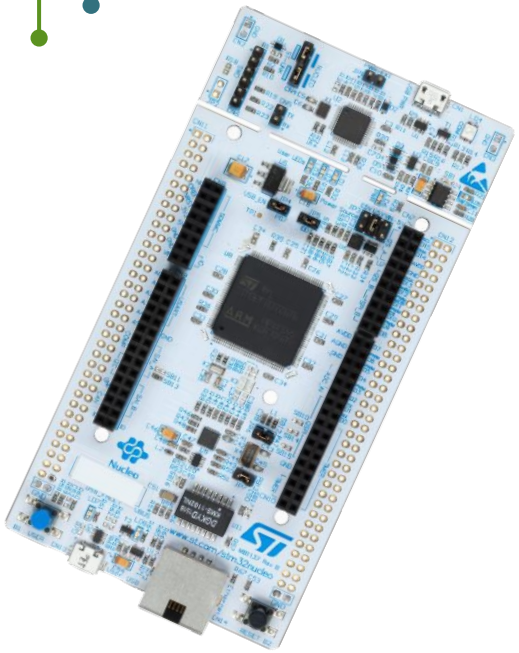


Figure 4. STM32 Nucleo-144 board top layout



LED1 : GPIOB 0

Wels



NUCLEO-F429ZI

life.augmented

RM0090
Reference manual

STM32F405/415, STM32F407/417, STM32F427/437 and STM32F429/439 advanced Arm®-based 32-bit MCUs

Introduction

This reference manual targets application developers. It provides complete information on how to use the STM32F405xx/07xx, STM32F415xx/17xx, STM32F42xxx and STM32F43xxx microcontroller memory and peripherals.

The STM32F405xx/07xx, STM32F415xx/17xx, STM32F42xxx and STM32F43xxx constitute a family of microcontrollers with different memory sizes, packages and peripherals.

For ordering information, mechanical and electrical device characteristics please refer to the datasheets.

For information on the Arm® Cortex®-M4 with FPU core, please refer to the *Cortex®-M4 with FPU Technical Reference Manual*.

Related documents

Available from STMicroelectronics web site (*http://www.st.com*):

- STM32F40x and STM32F41x datasheets
- STM32F42x and STM32F43x datasheets
- For information on the Arm® *Cortex®*-M4 with FPU, *refer* to the *STM32F3xx/F4xxx Cortex®*-M4 with FPU *programming manual (PM0214)*.

GPIO

Dentro de **GPIOB** existen diferentes registros para configurar

0x4002 0C00 - 0x4002 0FFF	GPIOD
0x4002 0800 - 0x4002 0BFF	GPIOC
0x4002 0400 - 0x4002 07FF	GPIOB

- *MODER*
- *OTYPER*
- *OSPEEDR*
- *PUPDR*
- *IDR*
- *ODR*
- *BSRR*
- *LCKR*
- *AFRH*
- *AFRL*

GPIO

Dentro de **GPIOB** existen diferentes registros para configurar

0x4002 0C00 - 0x4002 0FFF	GPIOD
0x4002 0800 - 0x4002 0BFF	GPIOC
0x4002 0400 - 0x4002 07FF	GPIOB

MODER
OTYPER
OSPEEDR
PUPDR
IDR
ODR
BSRR
LCKR
AFRH
AFRL

Cada registro del periférico tiene un ancho de **32 bits**

Wels

GPIOs Configuración

Vamos hacer la configuración para el pin 0 del GPIOB

Wels

MODER

Output

- Con el registro MODER seleccionamos entrada, salida, analógico, función alternativa

MODERy[1:0]: Port x configuration bits (y = 0..15)

These bits are written by software to configure the I/O direction mode.

00: Input (reset state)

01: General purpose output mode

10: Alternate function mode

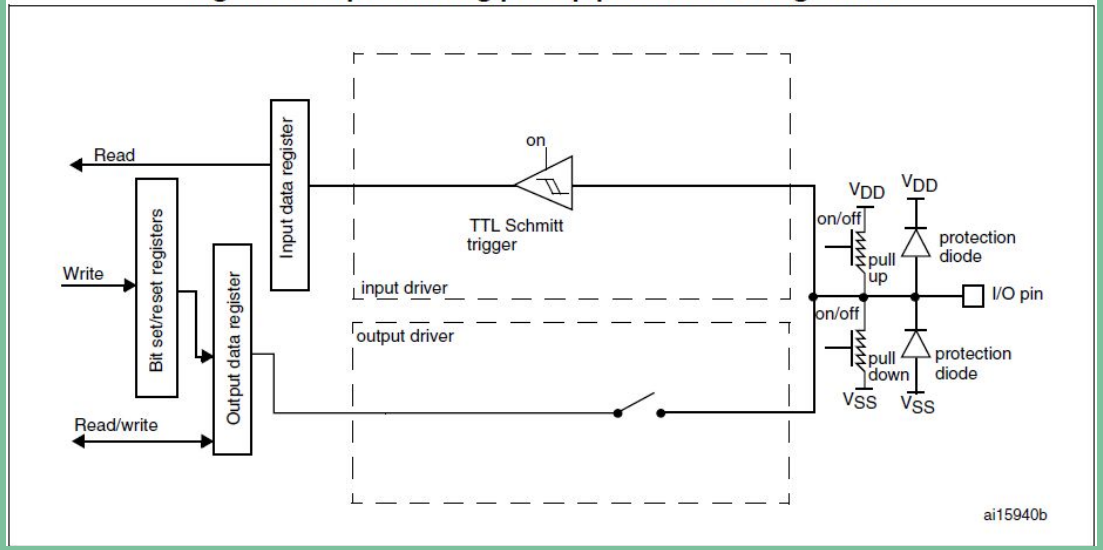
11: Analog mode

ODR

Escritura

- Registro de escritura de estado del pin.

Figure 18. Input floating/pull up/pull down configurations



Wels

GPIOB -> ODR

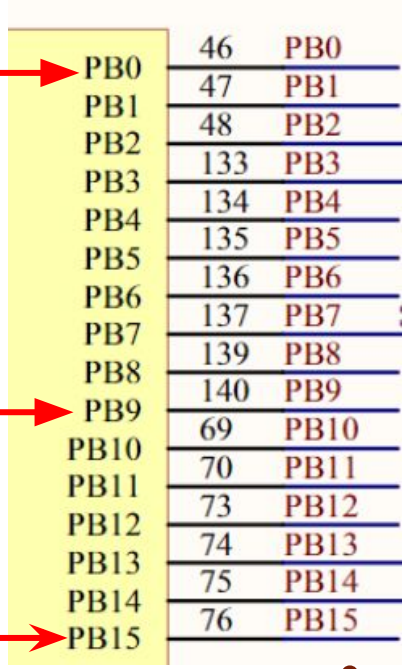
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Cada bit del registro es para cada Pin.

GPIOB -> ODR

Wels

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw



GPIOB -> MODER

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

2 bits por cada Pin

MODERy[1:0]: Port x configuration bits (y = 0..15)
These bits are written by software to configure the I/O direction mode.
00: Input (reset state)
01: General purpose output mode
10: Alternate function mode
11: Analog mode

PB0	46	PB0
PB1	47	PB1
PB2	48	PB2
PB3	133	PB3
PB4	134	PB4
PB5	135	PB5
PB6	136	PB6
PB7	137	PB7
PB8	139	PB8
PB9	140	PB9
PB10	69	PB10
PB11	70	PB11
PB12	73	PB12
PB13	74	PB13
PB14	75	PB14
PB15	76	PB15

¿Pero cuál es la dirección?

GPIOB

Wels

0x4002 0C00 - 0x4002 0FFF	GPIOD
0x4002 0800 - 0x4002 0BFF	GPIOC
0x4002 0400 - 0x4002 07FF	GPIOB

- GPIOB comienza en **0x4002 0400**. Que llamaremos la dirección base.

- A la dirección base del GPIOB se le suma el address offset: **0x00**

8.4.1 GPIO port mode register (GPIOx_MODER) (x = A..I/J/K)

Address offset: 0x00

Reset values:

- 0xA800 0000 for port A
- 0x0000 0280 for port B
- 0x0000 0000 for other ports

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

GPIOB

0x4002 0C00 - 0x4002 0FFF	GPIOD
0x4002 0800 - 0x4002 0BFF	GPIOC
0x4002 0400 - 0x4002 07FF	GPIOB

- GPIOB comienza en **0x4002 0400**. Que llamaremos la dirección base.

- A la dirección base del GPIOB se le suma el address offset: **0x14**

8.4.6 GPIO port output data register (GPIOx_ODR) (x = A..I/J/K)

Address offset: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Wels

GPIO Clock

- Debemos habilitar el clock de cada GPIO que utilicemos.
- Se encuentra en el registro **RCC->AHB1ENR**

	GPIOIEN	GPIOHEN	GPIOGEN	GPIOFEN	GPIOEEN	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN
	rw	rw	rw	rw	rw	rw	rw	rw	rw

A - B - C - D - E - F - G - H - I

RCC

Wels

0x4002 3C00 - 0x4002 3FFF	Flash interface register
0x4002 3800 - 0x4002 3BFF	RCC
0x4002 3000 - 0x4002 33FF	CRC

- RCC comienza en **0x4002 3800.** Que llamaremos la dirección base.

- A la dirección base del AHB1ENR se le suma el address offset: **0x30**

6.3.10 RCC AHB1 peripheral clock register (RCC_AHB1ENR)

Address offset: 0x30

Reset value: 0x0010 0000

Access: no wait state, word, half-word and byte access.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reser-ved	OTGHS ULPIEN	OTGHSEN	ETHMACPTPEN	ETHMACRXEN	ETHMACTXEN	ETHMACEN	Res.	**DMA2DEN**	DMA2EN	DMA1EN	CCMDATARAMEN	Res.	BKPSRAMEN	Reserved	
	rw	rw	rw	rw	rw	rw		rw	rw	rw			rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CRCEN	Res.	GPIOKEN	GPIOJEN	GPIOIEN	GPIOHEN	GPIOGEN	GPIOFEN	GPIOEEN	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN
			rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Wels

Operadores Bitwise

En C, las operaciones se pueden hacer en nivel bit. Veremos **OR (|)** y **AND (&)**.

X	Y	X & Y	X | Y
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

Operadores Bitwise

En C, las operaciones se pueden hacer en nivel bit. Veremos **OR (|)** y **AND (&)**.

```
Registro = Registro | dato;
```

```
Registro |= dato;
```

Operadores Bitwise

En C, las operaciones se pueden hacer en nivel bit. Veremos **OR (|)** y **AND (&)**.

```
Registro = Registro & dato;
```

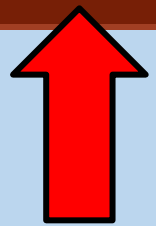
```
Registro &= dato;
```

Wels

Operadores Bitwise

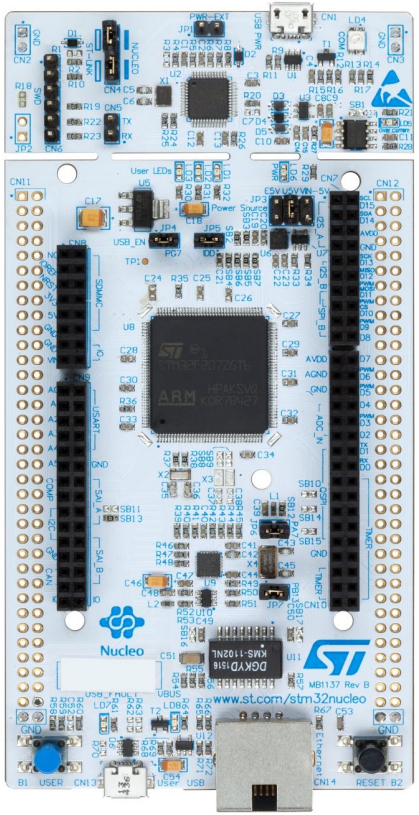
En C, las operaciones se pueden hacer en nivel bit. Veremos **OR (|)** y **AND (&)**.

```
Registro &= dato;
```



Limpiar los bits de configuración

Wels



Gracias

@welstheory
hola@welstheory.com
+51 918 899 684