

ACTIVACIÓN DE GPIOs

En la mayoría de microcontroladores los puertos I/O están mapeados a memoria, lo que significa que se puede acceder a una entrada o salida leyendo o escribiendo en direcciones establecidas. Para la configuración correcta de estos en ensamblador, es necesario tener en cuenta los siguientes registros principales:

AMSEL

Selección de modo analógico, conecta los pines a ADC o comparador analógico.

Un set lo activa.

PCTL

Selección de función alternativa. Existen quince posibles modos en que un pin puede funcionar, por lo que cuatro bits en este registro pueden especificar cuál desea utilizarse (I/O, UART, PWM, etc.) para activar el multiplexor de puerto o PMx.

En el caso de código que sólo utiliza I/Os ha de considerarse que el valor en este registro siempre será 0.

En la siguiente tabla se presentan los modos de funcionamiento de todos los pines de la Tiva C, excluyéndose PC3-PC0 por estar reservados para JTAG.

IO	Am	0	1	2	3	4	5	6	7	8	9	14
PA0		Port	U0Rx							CANIRx		
PA1		Port	U0Tx							CANITx		
PA2		Port		SSI0Clk								
PA3		Port		SSI0Fss								
PA4		Port		SSI0Rx								
PA5		Port		SSI0Tx								
PA6		Port			I ₂ C1SCL		MIPWM2					
PA7		Port			I ₂ C1SDA		MIPWM3					
PB0		Port	U1Rx						T2CCP0			
PB1		Port	U1Tx						T2CCP1			
PB2		Port			I ₂ C0SCL				T3CCP0			
PB3		Port			I ₂ C0SDA				T3CCP1			
PB4	Am10	Port		SSI2Clk		MOPWM2			T1CCP0	CAN0Rx		
PB5	Am11	Port		SSI2Fss		MOPWM3			T1CCP1	CAN0Tx		
PB6		Port		SSI2Rx		MOPWM0			T0CCP0			
PB7		Port		SSI2Tx		MOPWM1			T0CCP1			
PC4	C1-	Port	U4Rx	U1Rx		MOPWM6		IDX1	WT0CCP0	U1RTS		
PC5	C1+	Port	U4Tx	U1Tx		MOPWM7		PhA1	WT0CCP1	U1CTS		
PC6	C0+	Port	U3Rx					PhB1	WT1CCP0	USB0open		
PC7	C0-	Port	U3Tx						WT1CCP1	USB0pflt		
PD0	Am7	Port	SSI3Clk	SSI1Clk	I ₂ C3SCL	MOPWM6	MIPWM0		WT2CCP0			
PD1	Am6	Port	SSI3Fss	SSI1Fss	I ₂ C3SDA	MOPWM7	MIPWM1		WT2CCP1			
PD2	Am5	Port	SSI3Rx	SSI1Rx		M0Fault0			WT3CCP0	USB0open		
PD3	Am4	Port	SSI3Tx	SSI1Tx				IDX0	WT3CCP1	USB0pflt		
PD4	USB0DM	Port	U6Rx						WT4CCP0			
PD5	USB0DP	Port	U6Tx						WT4CCP1			
PD6		Port	U2Rx			M0Fault0		PhA0	WT5CCP0			
PD7		Port	U2Tx					PhB0	WT5CCP1	NMI		
PE0	Am3	Port	U7Rx									
PE1	Am2	Port	U7Tx									
PE2	Am1	Port										
PE3	Am0	Port										
PE4	Am9	Port	U5Rx		I ₂ C2SCL	MOPWM4	MIPWM2			CAN0Rx		
PE5	Am8	Port	U5Tx		I ₂ C2SDA	MOPWM5	MIPWM3			CAN0Tx		
PF0		Port	U1RTS	SSI1Rx	CAN0Rx		MIPWM4	PhA0	T0CCP0	NMI	C0o	
PF1		Port	U1CTS	SSI1Tx			MIPWM5	PhB0	T0CCP1		C1o	TRD1
PF2		Port		SSI1Clk		M0Fault0	MIPWM6		T1CCP0			TRD0
PF3		Port		SSI1Fss	CAN0Tx		MIPWM7		T1CCP1			TRCLK
PF4		Port					M1Fault0	IDX0	T2CCP0	USB0open		

DEN

Habilitación del modo digital, un set lo activa.

DIR

Especificación de dirección. Set configura output y clear, input.

AFSEL

Función alternativa. Especifica si ha sido seleccionado un modo alternativo, actuando junto a la selección de PCTL.
Un clear implica ninguna función alternativa seleccionada (modo GPIO).

PUR

Configuración de resistencias pull up en inputs.

DATA

Encierra las direcciones del puerto completo (8 pines).

SYSCTL_RCGC2_R

Reloj. A cada puerto del A al F le corresponde un bit de este registro destinado a activar su reloj.

A continuación se presenta un resumen de las características de los registros principales para configuración de puertos

Registro	Función	Offset
DATA	Información de puerto	0x3FC
AMSEL	Analógica	0x528
PCTL	Alternativa	0x52C
DEN	Digital	0x51C
DIR	Dirección	0x400
AFSEL	Desempeño	0x420

El registro SYSCTL_RCGC2_R se encuentra en la dirección 0x400FE108, sin uso de offsets.

En desarrollo de software, estos registros serán declarados como constantes con el formato `GPIO_PORTx_registro_R`. Por ejemplo, para la selección de función analógica en el puerto C, el nombre a leerse será `GPIO_PORTC_AMSEL_R`.

El término offset está ligado a conceptos básicos de informática, donde se define como el desplazamiento desde un punto central (arbitrario) en un arreglo hasta un objeto de interés.

En lenguaje ensamblador se habla de offsets como desplazamientos en posiciones de memoria. Así, para cada puerto existe una dirección base (especificadas luego) y al pretender utilizar alguno de los registros de la tabla simplemente se sumará a esta el offset requerido.

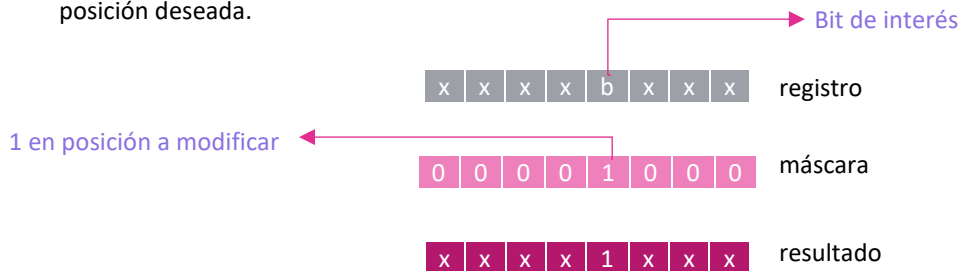
FRIENDLY CODE

En el diseño de sistemas embebidos es importante mantener la eficiencia del código al máximo. Para esto existen técnicas que evitan la sobre escritura innecesaria de bits en registros de interés, el mantra de la programación amigable se basa en el “do not harm”, que implica la modificación solamente de los bits necesarios.

Esta línea de diseño utiliza dos métodos, según el valor que se desee dar un bit.

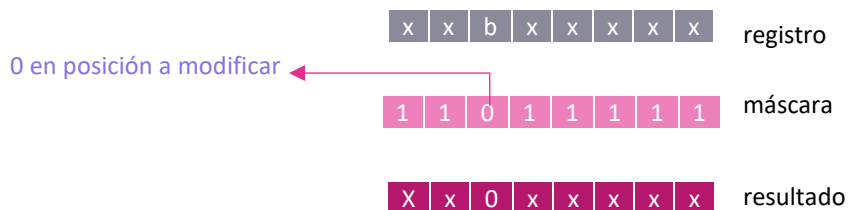
1. Or to set

Utiliza una operación lógica OR entre el registro a modificar y una máscara de 1s para hacer set en la posición deseada.



2. And to clear

Utiliza una operación lógica AND entre el registro y una máscara de 0s para hacer clear en la posición deseada.



Para generar software amigable existen dos métodos principales, según como se desea manejar el registro. Estos son:

READ-MODIFY-WRITE

Utiliza los principios de la sección anterior en cuatro pasos: cargar el contenido del registro DATA (este contiene todas las direcciones de pines en el puerto), trasladar a un registro temporal, realizar la operación or para establecer un 1 o bit clear para establecer un 0 en los bits requeridos y por último volver a guardar el valor.

Para escribir 1s se recurre al **READ-OR-WRITE** con la instrucción ORR

```
LDR R1, =GPIO_PORTA_DATA_R
LDR R0, [R1]
ORR R0, R0, #0x80
STR R0, [R1]
```

Para escribir 0s se recurre al **READ-AND-WRITE** con la instrucción BIC

```
LDR R1, =GPIO_PORTA_DATA_R
LDR R0, [R1]
BIC R0, R0, #0x80
STR R0, [R1]
```

BIT SPECIFIC ADDRESSING

Dado que repetir los pasos de read-modify-write para todos los pines en un puerto haría al código excesivamente largo, se recurre a una estrategia más de friendly code que utiliza las direcciones base para cada puerto y suma los offsets referentes a los pines que interesa activar. Así, por cada registro a modificar se accederá una sola vez pero sin afectar más bits de los necesarios.

Las direcciones base para cada puerto son:

Puerto	Dirección
A	0x40004000
B	0x40005000
C	0x40006000
D	0x40007000
E	0x40024000
F	0x40025000

Como se mencionó anteriormente, según los pines de cada puerto que se desee utilizar, se sumará a la dirección base los offset necesarios. Este desplazamiento de memoria para cada pin está dado por $4 * 2^b$, donde b es la posición del bit.

De esta forma, los bits corresponden a las siguientes constantes:

# bit	Constante
7	0x0200
6	0x0100
5	0x0080
4	0x0040
3	0x0020
2	0x0010
1	0x0008
0	0x0004

Es importante considerar que la suma de todas las constantes da como resultado 0x03FC, correspondiente al offset del registro DATA. Así se demuestra que este contiene las direcciones pertinentes a todos los pines de un puerto.

RITUAL DE INICIALIZACIÓN

Todas las consideraciones de diseño expuestas en este resumen están encaminadas a la configuración adecuada de pines GPIO en la Tiva C. Por lo que, como último, se especifica el orden correcto en que debe accederse a los registros necesarios conformado el *ritual de inicialización de puertos GPIO*:

1. Inicializar reloj
2. Desbloqueo de pines (PC3-PC0, PD7, PF0)
3. Deshabilitar función analógica
4. Limpiar bits en PCTL (ya que 0 corresponde a I/Os)
5. Especificar dirección (input/output)
6. Limpiar bits en función alternativa
7. Habilitar función digital