

1.1 Temporizadores internos

POLICOM incorpora dos contadores que van a permitir al programador operaciones que supongan control de tiempo.

1.1.1 Prestaciones

- 2 temporizadores independientes
- Carga y habilitación programadas.
- Dos modos de funcionamiento:
 1. Modo libre: el temporizador irá decrementando su valor hasta desbordamiento, y seguirá decrementando.
 2. Modo periódico: el contador irá decrementando su valor comenzando siempre por el vlaor de la precarga.
- Frecuencia de decremento programable a partir de la frecuencia fundamental del bus de periféricos: 6 MHz.
- Generación de interrupción 0x4_IRQ y 0x5_IRQ programadas.

1.1.2 Programación

El dispositivo incorpora los siguientes registros:

Para el temporizador número 0:

| Dirección | Nombre | Descripción | Acceso | Bits | Ini. |
|-----------|-----------|--|--------|------|------|
| 0x46_0000 | timer0_LR | Registro de carga de valor inicial del contador. | R/W | 16 | 0 |
| 0x46_0004 | timer0_CV | Valor actual del temporizador | RO | 16 | 0 |
| 0x46_0008 | timer0_CI | Registro de limpieza de interrupción | R/W | -- | 0 |
| 0x46_000C | timer0_CR | Registro de configuración | R/W | 4 | 0 |

Para el temporizador 1:

| Dirección | Nombre | Descripción | Acceso | Bits | Ini. |
|-----------|-----------|--------------------------------------|--------|------|------|
| 0x46_0020 | timer1_LR | Registro de carga de valor inicial | R/W | 16 | 0 |
| 0x46_0024 | timer1_CV | Valor actual del temporizador | RO | 16 | 0 |
| 0x46_0028 | timer1_CI | Registro de limpieza de interrupción | R/W | -- | 0 |
| 0x46_002C | timer1_CR | Registro de configuración | R/W | 4 | 0 |

1.1.2.1 El registro de carga de valor inicial

El programador cargará el valor deseado en este registro. Dicho valor comenzará a decrementarse cuando se habilite el temporizador. La velocidad de decremento será programable en el registro de configuración.

1.1.2.2 El registro de valor actual

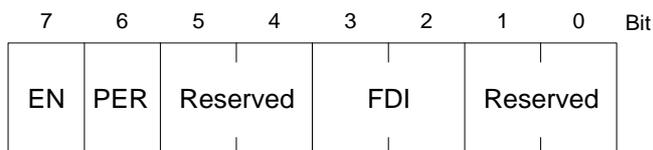
Muestra el estado actual del temporizador.

1.1.2.3 Registro de limpieza de interrupción

Cualquier valor escrito en este registro desactiva la fuente de interrupción activada en el temporizador.

1.1.2.4 Registro de configuración

Su contenido es el siguiente:



- FDI → Configuración de la frecuencia de decrecimiento del temporizador. Dicha frecuencia es el resultado de dividir 6Mhz por el número configurado según la tabla:

| FDI | Divisor de velocidad |
|------------|-----------------------------|
| 00 | 1 |
| 01 | 16 |
| 10 | 256 |
| 11 | 4094 |

- PER → Modo de funcionamiento del temporizador

| PER | Modo |
|------------|--------------------------|
| 0 | Funcionamiento libre. |
| 1 | Funcionamiento periódico |

- EN → Habilita si '1' la generación de interrupción al alcanzar el temporizador el valor '0'.

1.1.3 Modo de operación

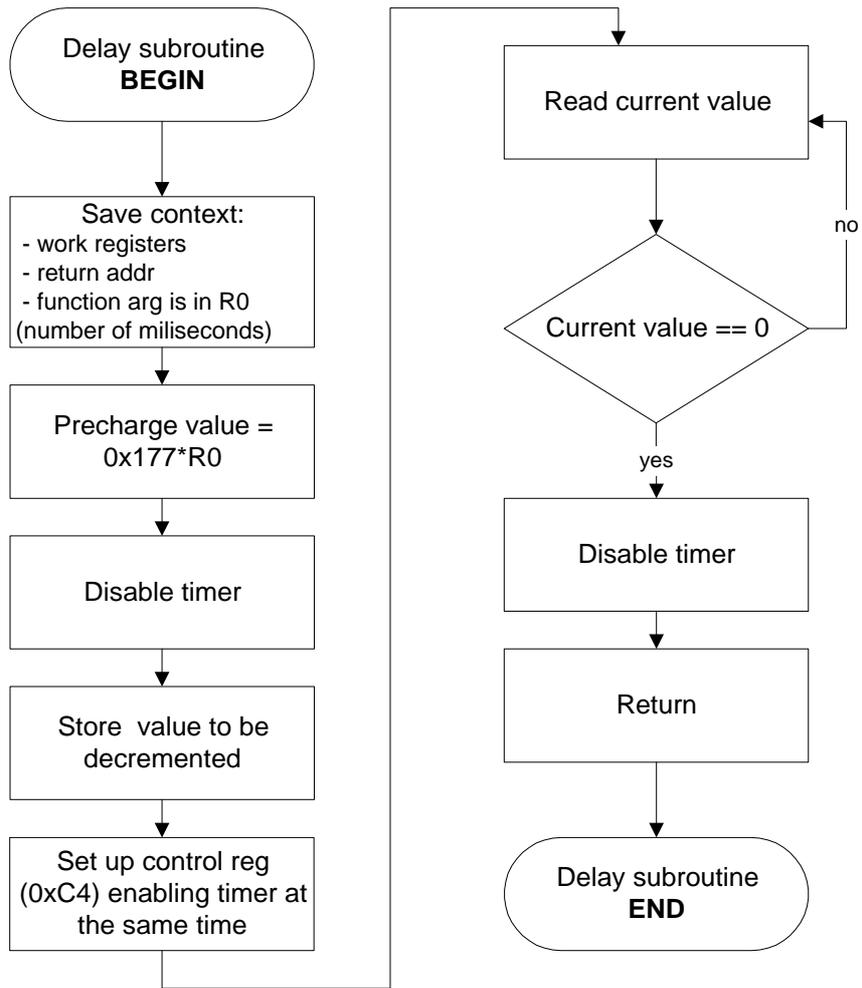
En esta sección se va describir cómo deben utilizarse cada temporizador que incorpora Policom. Los resultados expuestos han sido extraídos a partir de las pruebas de co-simulación en la fase de diseño del sistema Policom.

El uso de temporizadores en Policom se estima que sea grande ya que los protocolos que el chip implementará emplean mucho este recurso.

1.1.3.1 Configuración delay

Existen multitud de aplicaciones que requieren que un determinado proceso se quede esperando un determinado tiempo en lo que se denomina espera activa. Es el caso de la programación de los convertidores ADC y DAC que, según se vio en su sección, requieren ser habilitados durante un tiempo concreto antes de ser utilizados para realizar una conversión.

EL diagrama de flujo que precede muestra el funcionamiento de este tipo de configuración. El código corresponde con lo mostrado en el diagrama de flujo.



```

DELAY_ver_1_0
  STMDB      SP!, {R0-R12,LR}
DELAY_ver_1_0_Begin
  LDR       R1,=0x177 ;0x177 = 375 --> 1msec
  MUL      R2,R1,R0      ;
  LDR      R5,=timer_TC0_CR
  LDR      R6,=timer_TC0_LR
  LDR      R7,=timer_TC0_CV
  MOV      R4,#0x0
  STR      R4,[R5]      ; Disable and load timer
  STR      R2,[R6]
  MOV      R4,#0xC4      ; enable timer with: PCLK /16
                        ; periodic running
  STR      R4,[R5]
DELAY_ver_1_0_label_0
  LDR      R5,[R6]      ; remains into tho loop until
  CMP      R5,#0        ; timer == 0
  BGT     DELAY_ver_1_0_label_0
  MOV      R7,#0        ; disable timer
  STR      R7,[R3]
DELAY_ver_1_0_End
  LDMIA    SP!, {R0-R12,PC}
  
```

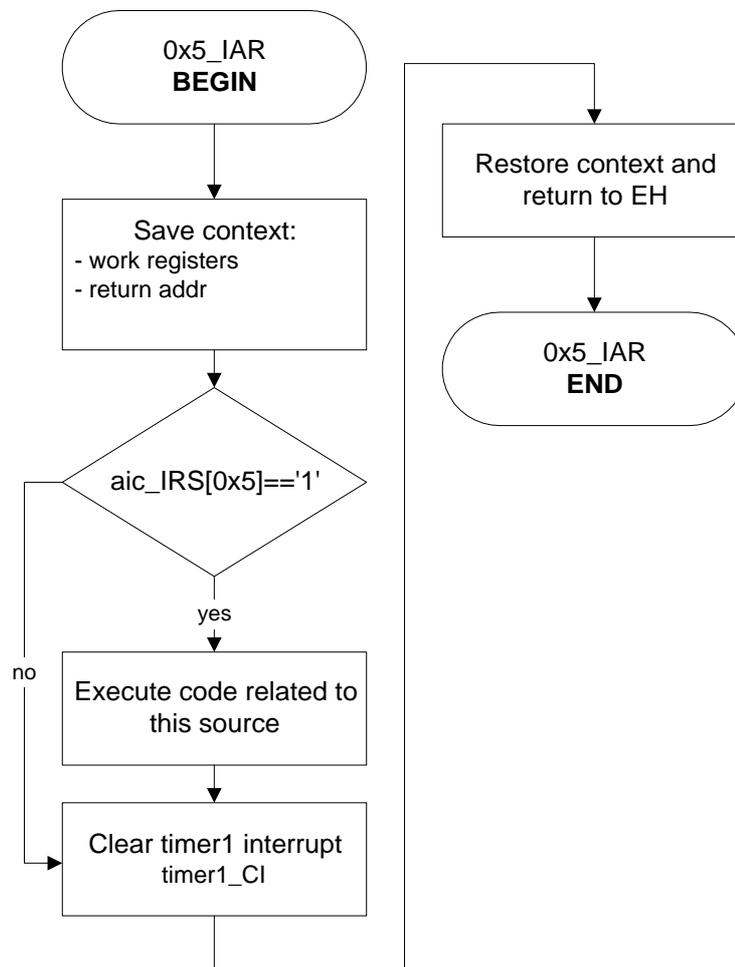
El código mostrado corresponde con una rutina cuya función consiste en esperar un número de milisegundos que se especifica en la llamada.

1.1.3.2 Gestión por interrupciones

Otro uso muy común del temporizador es para sincronizar determinadas tareas. En este caso utiliza la capacidad del temporizador para generar interrupción.

Para programar interrupción de temporizador es necesario llevar a cabo todos los pasos explicados en la sección dedicada a la gestión de interrupciones IRQ de Policom

En la interrupción de temporizador están implicadas dos rutinas que deben ser programadas por el usuario: la de configuración de interrupción (timer1_ICR) y la de atención de interrupción (timer1_IAR). En esta sección sólo se encargará de explicar un ejemplo de rutina de atención concreta. La ICR es similar a la de cualquier otro dispositivo.



```

0X5_IAR_VER_TEST_3_ver_1_0
  STMDB SP!, {LR}
0X5_IAR_VER_TEST_3_ver_1_0_Begin
  LDR R3, =timer_TC0_CR
  LDR R4, =timer_TC0_CV
  LDR R5, =timer_TC0_LR
  LDR R6, =aic_IRS
  LDR R7, =bpp_IOLR
  LDR R8, =adc_CR
  LDR R9, =adc_DR
  LDR R10, =dac_DR
  LDR R11, =dac_CR
  LDR R12, =timer_TC1_CI
  MOV R0, #0x1
  MOV R0, R0, LSL #TIMER2 ; first off all check timer2 irq source
  LDR R1, [R6]
  TST R1, R0
  BEQ 0X5_IAR_VER_TEST_3_ver_1_0_End
  LDR R0, =EXT_PER_1_Base ; code related to this source
  MOV R1, #0
  STR R1, [R0]
  MOV R0, #0x1
  STR R0, [R8]
  MOV R0, #0x78
  STR R0, [R5]
  MOV R0, #0x80
  STR R0, [R3]
0X5_IAR_VER_TEST_3_ver_1_0_Label_0
  LDR R1, [R4]
  CMP R1, #0x8000
  BLT 0X5_IAR_VER_TEST_3_ver_1_0_Label_0
  MOV R0, #0x5 ; start conversion
  STR R0, [R8]
0X5_IAR_VER_TEST_3_ver_1_0_Label_1
  LDR R1, [R8]
  TST R1, #0x4
  BEQ 0X5_IAR_VER_TEST_3_ver_1_0_Label_1;
  LDR R0, [R9] ; load value recently converted
  LDR R1, [R7] ; read bpp value --> R3
  MOV R1, R1, LSR #8
  MUL R2, R1, R0
  MOV R0, #0x1
  STR R0, [R11]
  MOV R0, #0x78
  STR R0, [R5]
  MOV R0, #0x80
  STR R0, [R3]
0X5_IAR_VER_TEST_3_ver_1_0_Label_2
  LDR R1, [R4]
  CMP R1, #0x8000 ; buscamos el overflow del contador
  BLT 0X5_IAR_VER_TEST_3_ver_1_0_Label_2
  STR R2, [R10] ; convert to ANALOG
  LDR R0, =EXT_PER_1_Base
  MOV R1, #0
  STR R1, [R0, #1]
; Clear interrupt in timer 2
  MOV R0, #0
  STR R0, [R12]
0X5_IAR_VER_TEST_3_ver_1_0_End

```

