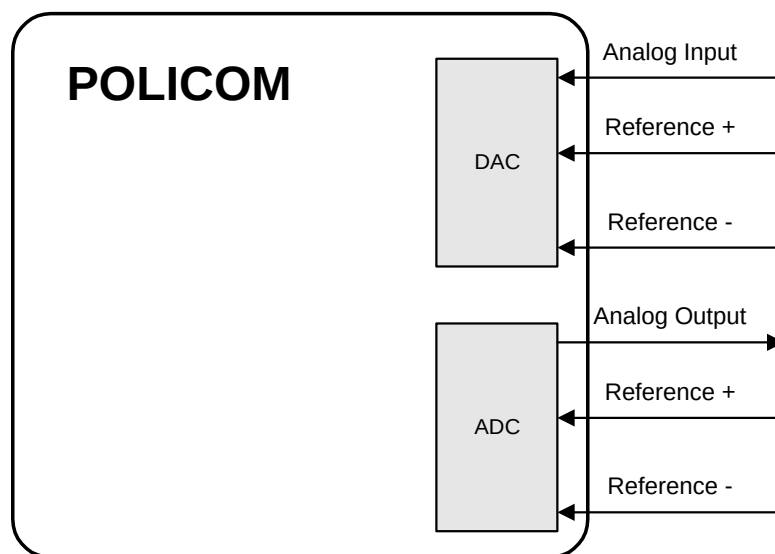


1.1 Entradas y salidas analógicas

POLICOM permite tratamiento de señales analógicas gracias a que incorpora sendos convertidores analógicos a digital y digital a analógico.

La figura muestra la interfaz analógica que sigue el chip. Esto va a permitir a Policom realizar lecturas de dispositivos que eventualmente se instalen junto a él (todo tipo de sensores), o bien controlar dispositivo analógicos que se instalen (por ejemplo actuadores).



1.1.1 El convertidor analógico digital (ADC).

1.1.1.1 Prestaciones

- Convertidor ADC por aproximaciones sucesivas de 10 bits a 1 Mmuestra/sec. El tiempo de conversión dependerá de la celda que finalmente se incluya en el diseño final.
- Modo de bajo consumo activado por software.
- Generación de interrupción al final de conversión.

1.1.1.2 Programación

La siguiente tabla muestra los registros accesibles al programador.

Dirección	Nombre	Descripción	Acceso	Bits	Ini.
0x44_0000	adc_DR	Resultado de la conversión	RO	10	-
0x44_0004	adc_CR	Registro de configuración y estado	R/W	3	0
0x44_0008	adc_CD	Inhabilitación de dispositivo	WO	3	-

El Resultado de la conversión (adc_DR)

Este registro contendrá el resultado de la conversión de la señal analógica entrante cuando se haya llegado al final de la conversión.

El resultado antes del final de conversión es impredecible.

Registro de Configuración y Estado (adc_CR)

En caso de escritura establecerá los parámetros básicos para poder llevar a cabo la conversión. Está compuesto de:

2	1	0	Bit
SC	EI	ED	

- ED → Habilitación del dispositivo.

‘1’ hace que el convertidor salga del estado de bajo consumo.

‘0’ sin efecto.

- EI → Habilitación de la generación de interrupción tras conversión

‘0’ no genera interrupción.

‘1’ genera interrupción.

- SC → Inicio de conversión.

‘0’ no tiene efecto

‘1’ inicia conversión

En caso de lectura este registro informa al programador del estado del dispositivo. Su composición es la siguiente:

2	1	0	Bit
EOC	IB	EB	

- EB → Habilitación del dispositivo.

‘1’ convertidor habilitado.

‘0’ convertidor no habilitado.

- IB → Habilitación de la generación de interrupción tras conversión

‘0’ no habilitado para generar interrupción.

‘1’ habilitado para generar interrupción.

- EOC → Final de conversión.

‘0’ conversión no finalizada.

‘1’ fin de conversión.

1.1.1.3

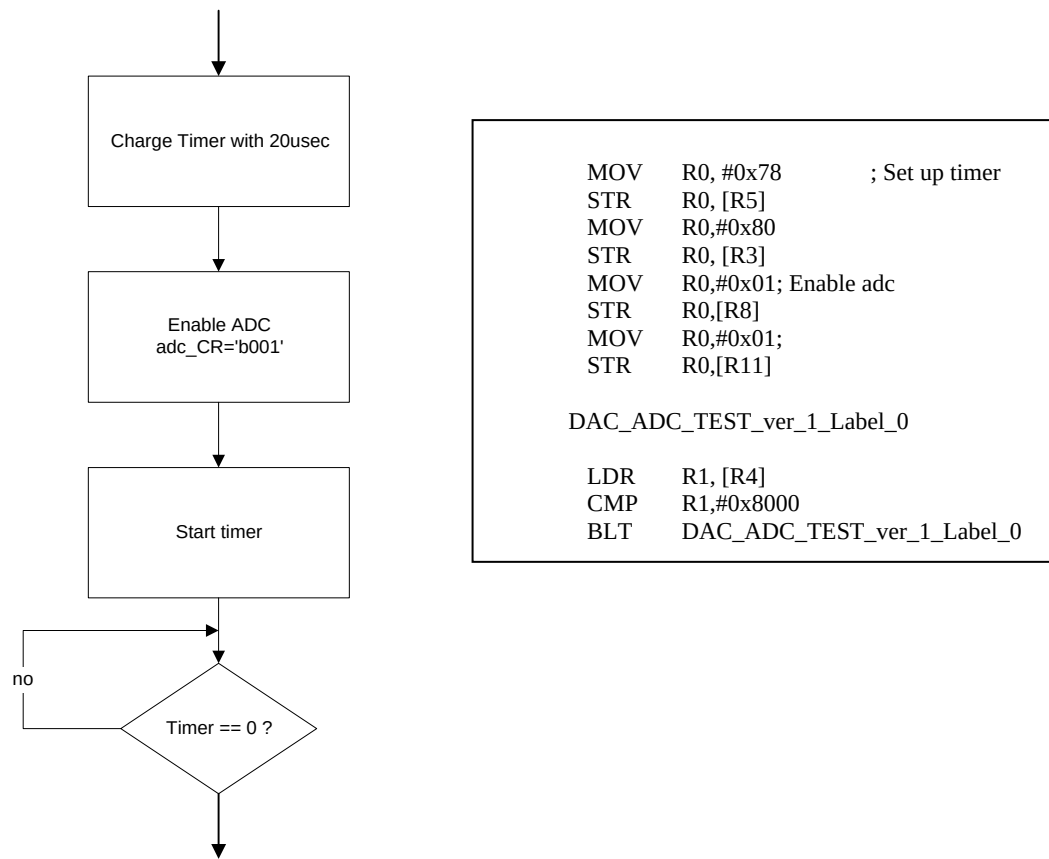
Modo de operación

Habilitación del dispositivo

El convertidor ADC necesita ser habilitado antes de proceder a realizar la primera conversión, ya que por defecto se encuentra en modo de bajo consumo.

El proceso de habilitación del dispositivo supone un retraso desde que se ordena dicha habilitación hasta que se pueda realizar la primera conversión. Dicho retraso debe ser tenido en cuenta por el programador, que deberá contabilizar un tiempo de 20 usec entre la habilitación del dispositivo y el lanzamiento de la primera conversión.

Las figuras siguientes muestran el diagrama de flujo que se sigue y el código ARM propuesto.



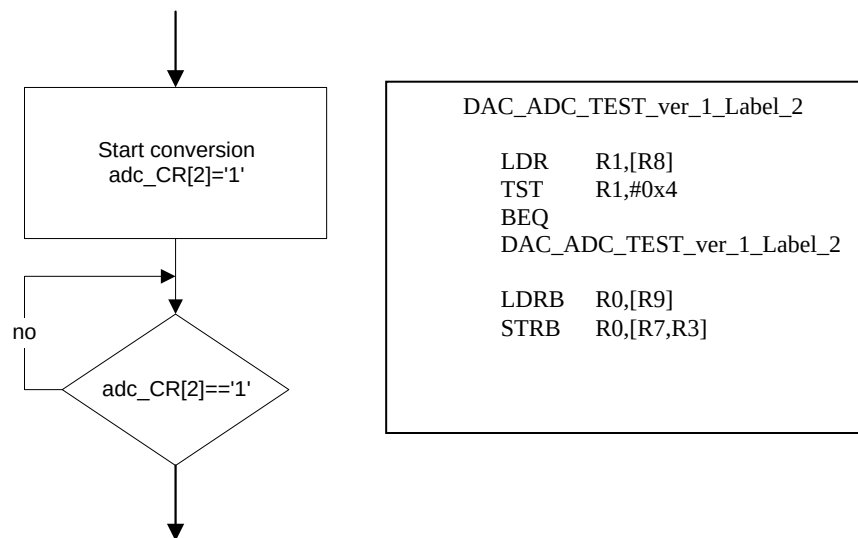
Conversión

Una vez que se ha habilitado el convertidor, para realizar la conversión habrá que ordenarle que inicie la misma.

El final de la conversión, que tardará unos 13 usec (datos del test chip), quedará reflejado en el registro de configuración.

Para el tratamiento de las conversiones se pueden tomar dos esquemas diferentes de programación:

- Gestión por espera activa. Una vez lanzada la conversión, el programador consultará el registro de estado (configuración) en busca del bit EOC (final de conversión). El diagrama de flujo junto con el código ARM se muestra a continuación.



- Gestión por interrupciones. Al igual que otros periféricos, el convertidor ADC de Policom permite al programador la posibilidad de tratar la final de conversión como si de una excepción IRQ se tratara. Para ellos deberá realizar todas las operaciones necesarias para instalar la que se denominará 0xB_IRQ. La metodología queda detallada en la sección del tratamiento de excepciones.

Como resultado de las simulaciones realizadas durante la fase de diseño, la forma más aconsejable para el tratamiento del ADC es mediante espera activa, ya que el tiempo que se optimiza de procesador en el caso de tratamiento por interrupciones es insignificante. Si futuras versiones de Policom incorporaran convertidores más lentos sí sería preferible utilizar la gestión por excepciones.

1.1.2 El convertidor digital analógico (DAC)

1.1.2.1 Prestaciones

- Convertidor de librería ATMEL de 10 bits
- Estado de bajo consumo mediante inhabilitación del dispositivo.
- Tiempo de conversión dependiendo de la celda que finalmente se incluya en el chip.

1.1.2.2 Programación

La siguiente tabla muestra los diferentes registros que componen la interfaz de programación del dispositivo:

Dirección	Nombre	Descripción	Acceso	Bits	Ini.
0x45_0000	dac_DR	Data a convertir	RO	10	-
0x45_0004	dac_CR	Registro de configuración.	RO	1	0
	dac_CE	Registro de habilitación.	WO	1	-
0x45_0008	dac_CD	Registro de Inhabuilitación.	WO	1	-

EL resultado de la conversión (dac_DR)

Este registro contiene el dato que se quiere convertir.

Registro de configuración (dac_CR)

Su único bit permite conocer si el dispositivo se encuentra habilitado o no.

dac_CR='1' → Habilitado.

dac_CR='0' → Inhabilitado.

Registro de habilitación / inhabilitación.

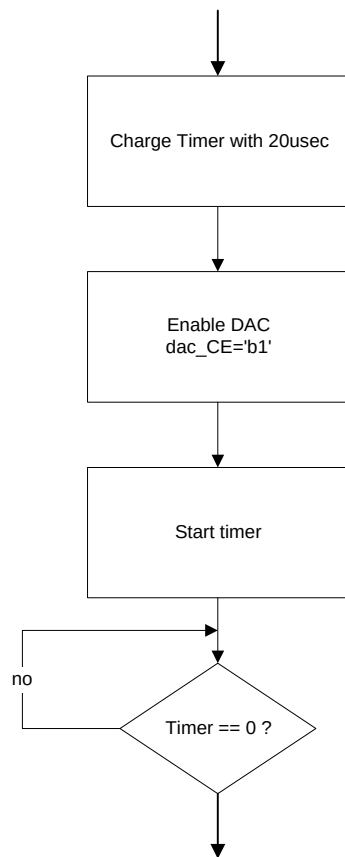
La escritura de '1' en el dac_CE permite habilitar el dispositivo. Por el contrario, si se escribe en el dac_CD '1', el dispositivo pasará a inhabilitado.

1.1.2.3 Modo de operación

Al igual que ocurre con el convertidor ADC, para poder llevar a cabo una conversión con el DAC es necesario que el programador habilite el dispositivo

20 usec antes de enviar el primer dato a convertir. El código utilizado es análogo al del convertidor ADC.

A continuación se muestra un trozo de código empleado para las pruebas y su diagrama de flujo correspondiente.



```
MOV    R0, #0x78      ; Set up timer
STR     R0, [R5]
MOV     R0, #0x80
STR     R0, [R3]
MOV     R0, #0x01      ;enable dac
STR     R0, [R11]

DAC_ADC_TEST_ver_1_Label_0

LDR     R1, [R4]
CMP     R1, #0x8000
BLT     DAC_ADC_TEST_ver_1_Label_0
```