



Universidad
de Alcalá



Práctica 4

Configuración y uso de bloques ADC/DAC

ASIGNATURA

Sistemas Electrónicos

Grado en Ingeniería en Tecnologías Industriales

Universidad de Alcalá

Curso 3º – Cuatrimestre 1º



Dpto. Electrónica



1. INTRODUCCIÓN

En esta práctica se abordará la puesta en marcha, configuración y uso de bloques ADC y DAC de los que dispone el sistema electrónico ESP32. También se realiza la calibración de un módulo ADC del sistema ESP32 real.

2. OBJETIVOS

Con la realización de esta práctica se pretende complementar las cuestiones teóricas con una aplicación real de adquisición de valores de tensión analógicos. Para la configuración correcta del bloque ADC se revisarán conceptos como referencia de tensión, resolución de bits, fondo de escala, atenuación de la señal, etc. Se comprobará la no linealidad del sistema de conversión, debiendo aplicar un procedimiento de calibración previo a la realización de medidas.

El estudiante hará las comprobaciones externas necesarias, a través de un multímetro para conocer la precisión de los bloques ADC, valores digitales y fondos de escala para conocer el rango de tensión (valores mínimo y máximo). Se asumirá que el multímetro utilizado es el sistema de medida de referencia, sin que introduzca errores.

3. RESUMEN DE LA PRÁCTICA

La práctica se divide en las siguientes tareas a realizar por el estudiante:

1. Revisión de los distintos bloques ADC y DAC de los que dispone el dispositivo ESP32.
2. Uso del bloque DAC para configurar salidas de tensión.
3. Estudio de las distintas funciones disponibles para la configuración y adquisición de datos de un canal ADC. Se configurará la lectura ADC para distintas resoluciones de bits y rango de medida.
4. Calibración del bloque ADC. Se utilizará un potenciómetro a la entrada del ADC que permita proporcionar distintas tensiones medidas con un polímetro para realizar los ajustes de calibración vistos en teoría. Verificación de la calibración.
5. Ejercicio de filtrado de una señal. Desde el propio ESP32 se genera la señal a través del DAC. Esta señal se captura con un bloque ADC y se filtra o no en función del estado de un pulsador.



4. BLOQUES ADC EN EL ESP32

La adquisición de señales analógicas desde un dispositivo ESP32 se puede realizar mediante dos ADCs de aproximación sucesiva (SAR-ADCs). Existen diferentes posibilidades de control del funcionamiento de estos ADCs. Revisar <https://github.com/espressif/arduino-esp32/blob/master/docs/source/api/adc.rst>

- Selección de canales de entrada: permite leer datos analógicos de diferentes pines. El ESP32 permite medir señales analógicas a través de hasta 18 pines (aunque sólo usa dos ADCs, mediante la multiplexación analógica interna de dichos pines). También es posible medir señales internas, como la señal de alimentación (Vdd33) lo que permite conocer el valor actual de la tensión de alimentación del circuito ESP32, que debería ser 3.3V.
- Resolución ADC: permite configurar la resolución de las conversiones de 9 a 12 bits. Mayor resolución proporciona mayor precisión, pero puede requerir más tiempo de conversión.
- Referencia de voltaje: permite configurar la referencia de voltaje utilizada para las conversiones ADC entre una referencia interna o externa. Por defecto la tensión de referencia, fondo de escala de la medida, es 3.3 voltios.
- Frecuencia de muestreo (*sampling rate*): se puede configurar la velocidad de muestreo del ADC para determinar cuántas conversiones se realizarán por segundo.
- Modo de operación: se puede optar entre el modo de una sola conversión (*single-shot*) o el modo de conversión continua (*continuous conversion*). El modo single-shot realiza una sola conversión y luego se detiene, mientras que el modo continuo realiza conversiones repetidamente a una velocidad predefinida.

En la Figura 2 se muestran los bloques internos que conectan y controlan el funcionamiento de los ADC de aproximaciones sucesivas del ESP32.

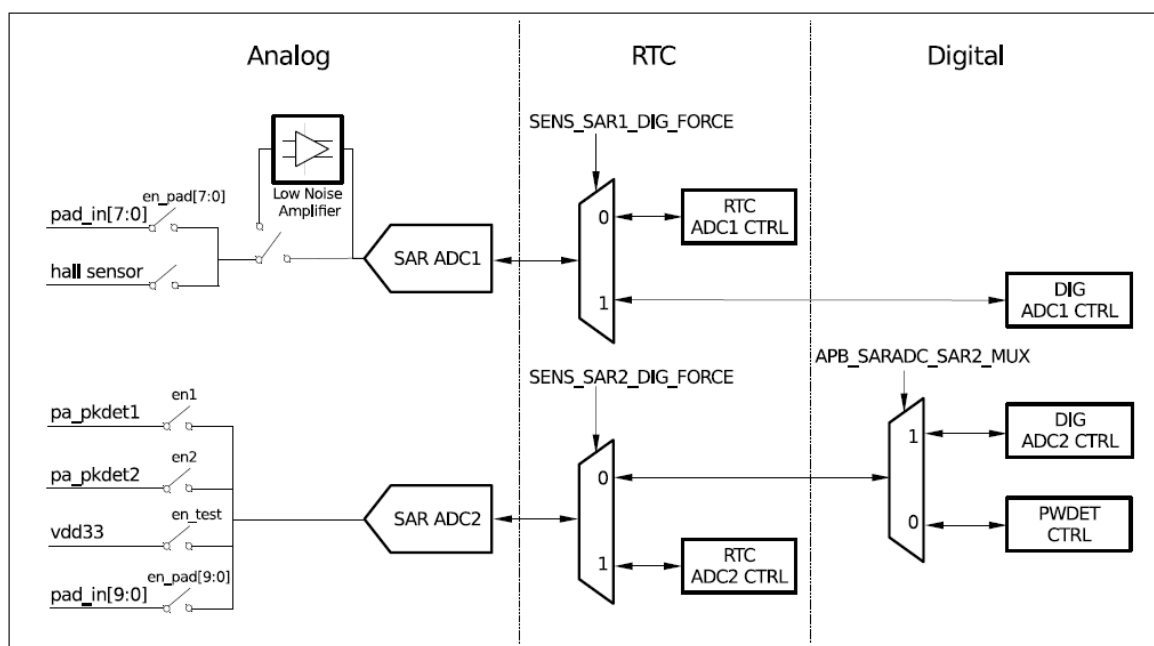


Figura 1. Bloques internos de módulos ADC del ESP32



5. BLOQUES DAC EN EL ESP32

El ESP32 también es capaz de generar señales analógicas, utilizando dos DAC independientes y un generador de formas de onda de coseno.

Se pueden utilizar dos canales DAC de 8 bits para convertir los valores digitales en señales de salida analógicas. La estructura de diseño está compuesta por cadenas de resistencias integradas y un buffer. Admiten una fuente de alimentación externa para utilizarla como referencia de tensión.

En la Figura 3 se muestra un diagrama de conexiones del bloque DAC.

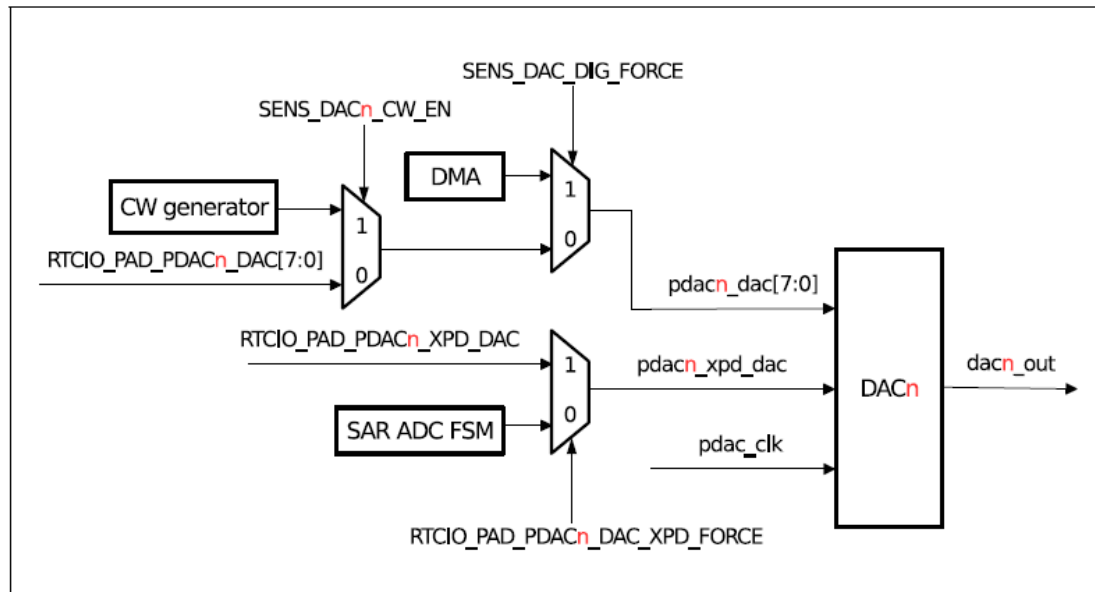


Figura 2. Bloques internos DAC del ESP32

Las salidas de ambos bloques DAC1 y DAC2, están en los pines GPIO 25 y 26 respectivamente. Los DAC son de 8 bits, configurando con valores digitales de 0 a 255 un valor de tensión de salida en el pin asociado de 0V a 3.3V.

El valor del voltaje de salida es por tanto igual a $volt_out = (3.3 * valor_digital) / 256$;

Revisar <https://github.com/espressif/arduino-esp32/blob/master/docs/source/api/dac.rst>

6. EJEMPLO DE CONFIGURACIÓN Y FUNCIONAMIENTO ADC/DAC ESP32

A continuación, se muestra un ejemplo para configurar y utilizar los bloques ADC y DAC.

<https://wokwi.com/projects/380733829667456001>

Nota: la salida DAC no funciona en el simulador

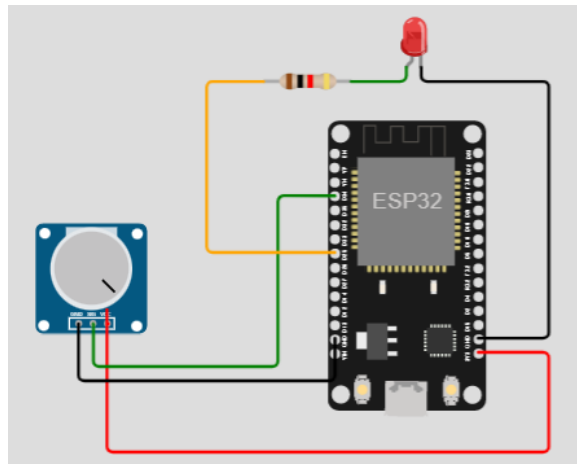


Figura 3. Esquema del ejercicio potenciómetro-ADC-DAC-LED.

A través de un potenciómetro (p.e. de 10 kOhm) se puede variar el valor de tensión de entrada variable. En el ejemplo el pin intermedio del potenciómetro se conecta al canal 6 del ADC1 a través del pin 34 del ESP32.

Girando el potenciómetro se captura desde el ADC una tensión distinta. También se puede hacer uso del joystick, conectando Vrx o Vry al pin 34 del ADC ESP32.

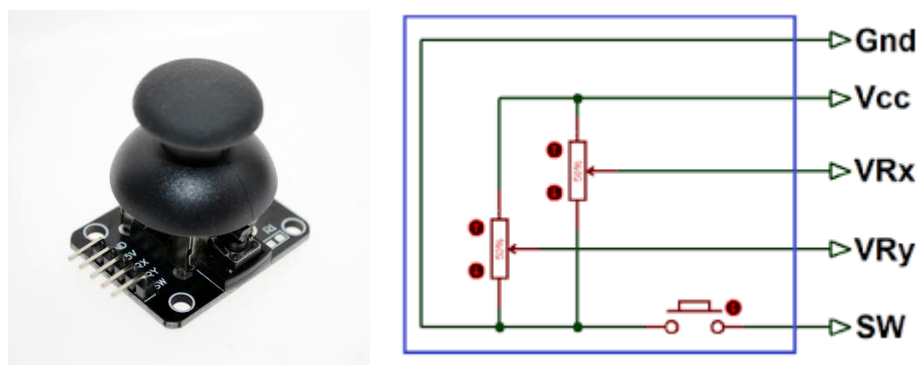


Figura 4. Esquema funcionamiento del joystick resistivo.

El bloque ADC se configura en el rango de medida más amplio (seleccionando la atenuación mayor) y con una resolución en el dato digital de 12 bits. Atendiendo a la función de conversión ideal del ADC, $3.3 \cdot \text{dato_digital_capturado} / (2^{12})$, se obtiene el valor de tensión de entrada.

A partir de ese valor obtenido por el ADC, en la salida del DAC1 en el pin 25 se genera una tensión variable entre 1.5 y 2.5 voltios conectada a un diodo LED a través de una resistencia. Por tanto, modificando el potenciómetro se tiene un control de la luminosidad del diodo LED.

Finalmente, por el puerto serie se visualizará el valor digital capturado, el voltaje según la función de conversión teórica, y el valor digital de la salida DAC.



El código es el que se muestra a continuación:

```
// https://wokwi.com/projects/380733829667456001
// el modulo DAC no genera tensiones diferentes en el simulador
#define pin2ADC 34
#define DAC2pin 25
void setup() {
    //set the resolution to 12 bits (0-4096)
    analogReadResolution(12);
    // voltage full-range 150mV - 3100 mV
    analogSetAttenuation(ADC_11db); // ADC_0db, ADC_2_5db, ADC_6db, ADC_11db
    // DAC channel 1 is attached to GPIO25, DAC channel 2 is attached to GPIO26
    // acepta valores de 0 a 255 -> Vout = 0 a 3.3v

    Serial.begin(115200);
    Serial.printf("Ejemplo de configuracion y uso ADC-DAC \n");
}

int dacVal, adcVal;
float voltageADC, tensionDAC;
void loop() {
    adcVal = analogRead(pin2ADC); // leemos el valor de tensión en el pin del bloque ADC
    voltageADC = adcVal / 4095.0 * 3.3; // conversion de tension ideal
    Serial.printf("Voltage entrada: %1.3f V => valor ADC: %d, \n", voltageADC, adcVal);
    // en funcion del valor adcVal 0 a 4095 la tensión a generar en el DAC debe ser 1.5 a 2.5
    tensionDAC = 1.5 + 1.0*adcVal/4095;
    dacVal = 255.0 * ( tensionDAC / 3.3); // tensión de salida en el DAC
    dacWrite(DAC2pin, dacVal); // ilumina más o menos el LED: Iled=(Vdac-1.1)/R

    Serial.printf("Valor DAC: %d, Tension generada: %1.3f V \n\n", dacVal, tensionDAC);

    delay(3000);
}
```



7. CALIBRACION BLOQUE ADC DISPOSITIVO ESP32 REAL

Se pide que calibre la función de conversión de un bloque ADC de su dispositivo ESP32 real.

Para el ADC debe obtener una función gráfica que relacione para distintos puntos el *código_ADC*= $f(\text{voltaje_entrada})$. Para realizar este ejercicio, fije a la entrada del ADC distintos valores de tensión obtenidos con el potenciómetro en el sistema anterior. Realice incrementos de tensión de 0.2 v (valores reales medidos con un multímetro) y apunte el código digital capturado desde el ADC (promedie varias medidas).

Calcule el error de offset y ganancia de la función de conversión linealizada en los tramos que mejor considere, p.e. en 3 tramos de 0-1v-2v-3v. Realice la función para que a partir del dato capturado el sistema obtenga la tensión real de entrada. Finalmente, genere el código que modifica el valor capturado para tener calibrado el ADC.

8. ENTREGABLE 3: ADQUISICIÓN Y FILTRADO DE UNA SEÑAL

En este ejercicio/práctica se debe usar el bloque ADC calibrado anteriormente, capturar una señal de entrada y filtrarla usando técnicas de procesamiento digital de señal. Para ello se proporcionar en un fichero “include” *datos_filtrado.h* todos los datos y constantes a utilizar en el sistema electrónico.

La Figura 1 muestra un esquema de funcionamiento del sistema a desarrollar.

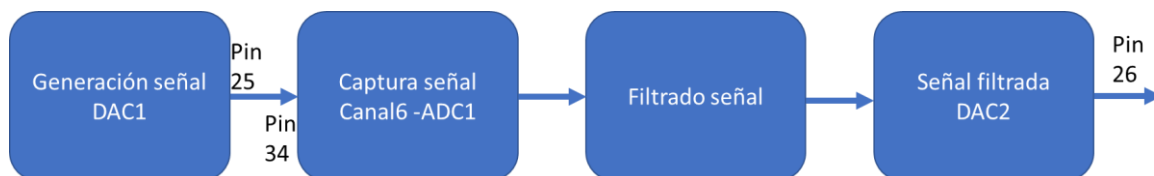


Figura 5. Bloques de procesamiento del sistema propuesto.

La generación de la señal por el pin 25 se proporciona a través del siguiente código inicial que contiene todos los datos de la señal, generación y constantes del filtro:

```

#define PL 11 // PL -> cambiar por el puesto de laboratorio del estudiante

#include "datos_filtrado24.h" // contiene constantes del filtro y señal a filtrar

void setup(){
  arranca_generacion(); // genera la señal DAC pin 25

  Serial.begin(115200);
  delay(100);
  Serial.printf("\n\n Ejercicio Filtrado \n");
}

void loop()
{
}

```



Se recomienda en primer término visualizar en el osciloscopio la señal que genera el anterior programa en el pin 25, que simula a una señal externa con ruido.

A continuación, conecte con un cable la salida del DAC1 (pin25) a la entrada de un canal del bloque ADC1, p.e pin 34, para capturar la señal externa a una **frecuencia de muestreo de 1 kSample/s (1000 muestras/seg)**.

Para obtener los valores de tensión capturados en los instantes de tiempo fijados por esa frecuencia, se configurará un nuevo bloque *Timer*. El fichero incluye “datos_filtrado24.h” no se debe modificar.

Compruebe el correcto funcionamiento obteniendo la señal capturada a través del segundo bloque DAC2, pin 26, y visualice la señal en el osciloscopio. Si usa 2 canales de osciloscopio en las salidas DAC1 y DAC2, debe comprobar que efectivamente se tiene adquirida la misma señal que existe generada por DAC1. Obtenga el valor máximo y mínimo de la tensión de la señal de entrada.

Para filtrar la señal se debe procesar una ventana de datos capturados de la señal de entrada de tamaño igual al número de coeficientes del filtro. La función debe devolver el valor filtrado de la señal muestreada. Se adjunta la función a ejecutar:

```
// Función de filtrado FIR
int filtrado(int input) {
    static int buffer[ncoefs];
    int output = 0;
    // Desplazar las muestras en el buffer
    for (int i = ncoefs - 1; i > 0; i--)
        buffer[i] = buffer[i - 1];
    buffer[0] = input; // nueva entrada

    // Aplicar el filtro FIR
    for (int i = 0; i < ncoefs; i++)
        output += coef[i] * buffer[i];
    output = output >> 16; // coefs enteros - 2^16

    // asegurar valores ADC permitidos en el rango establecido
    if(output < 0) output = 0;
    if(output > 4095) output = 4095;

    return output;
}
```

Se obtendrá el valor filtrado simplemente llamando a la función similar a:

```
int valor_filtrado = filtrado(valorADC);
```

La función de filtrado puede proporcionar valores fuera de rango, por lo que se ha limitado la salida entre 0 – 4095, considerando que se está utilizando valores con una resolución de 12 bits en el ADC.

Para comprobar el filtrado se debe obtener una tensión de salida en DAC2 con los valores filtrados obtenidos.

En la siguiente imagen se muestra la captura del osciloscopio con la señal original de entrada y la señal de salida filtrada.

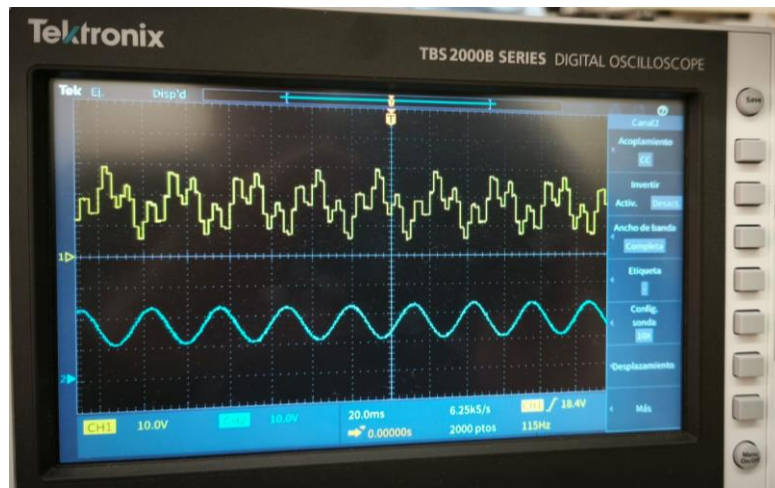


Figura 6. Captura de las señales generada y filtrada en el osciloscopio

Se pide que incluya un botón para utilizar o no los datos de calibración del ADC. Si lo desea puede añadir otras opciones, previa consulta con el profesor.

Visualice la señal de entrada y la señal de salida en los 2 canales del osciloscopio del laboratorio comprobando que efectivamente se está realizando correctamente el filtrado.

