Jorge Vicente Camuñas Heredia **25/10/20**

Práctica 4. Entrada/Salida con ESP32 (ESP-IDF) Uso del ADC

En esta práctica se pide programar un cronómetro que haga uso de GPIO, del touch-pad, del sensor Hall y del ADC para el uso de un sensor infrarrojo externo. Con el GPIO se pide conectar un pin como entrada y otro como salida con la finalidad de cambiar la señal cada segundo para usar este proceso como contador del cronómetro, el touch-pad se usará como "botón" para cambiar entre el estado START y el estado STOP del cronómetro y por último los sensores Hall e infrarrojo servirán como RESET del cronómetro.

Con esta idea de diseño se pidió realizar un esquema inicial (solo contando con el sensor Hall) previo al comienzo de la práctica para poder compararse con el estado final del desarrollo. Junto a este documento se encuentra una imagen del sistema final con todos los elementos, mientras que en esta memoria se desarrollarán las diferentes funciones de forma general para explicar como funciona el diseño elegido y como interactúan sus partes con el fin de lograr cumplir los requisitos expuestos en el boletín.

En el archivo "cronometro.c" se pueden encontrar las siguientes funciones:

- Cabe mencionar antes de las funciones la definición de los valores necesarios para asignar los pines de entrada y salida, los valores del sensor infrarrojo (si se ha seleccionado) y variables globales necesarias en las distintas tareas y funciones.

- app_main(): esta tarea principal es la que inicializará los pines de salida y entra, los diferentes sensores, el ctimer utilizado, las interrupciones y las tareas necesarias para hacer todo funcionar.
- callback_timer(): esta función a la que llama el ctimer cada segundo desde el comienzo de la ejecución se encarga de cambiar el pin de salida al contrario de su valor actual. Para conocer dicho valor se utiliza una variable booleana global que cambia a la vez que el pin de salida.
- timer_isr(): esta interrupción se activa al cambiar el valor del pin de entrada, lo que indica que ha pasado 1 segundo. Cuando se da esta interrupción se reactiva la tarea RefreshTask()
- RefreshTask(): esta tarea es la encargada de actualizar el valor por pantalla cada segundo, es por ello que después de cada actualización se suspende a la espera de que ocurra la interrupción timer_isr(). Dentro de esta función se tienen dos cosas en cuenta la primera es sumar 1 al valor sec si el cronómetro esta activo (variable booleana global) y la segunda es mostrar el valor de minutos y segundos de forma correcta. Esta tarea es la de mayor prioridad para que no pueda suceder que se encuentre en otro lugar cuando acabe el ctimer.
- tp_set_thresholds(): esta función se encarga de encontrar un valor de threshold adecuado para el touch sensor. Para ello toma 10 valores al comienzo de la ejecución y con ellos se calcula el valor óptimo. La llamada a esta función sucede al comienzo del app_main() justo después de inicializar el sensor touch y antes de crear la interrupción asociada.
- tp_isr_handler(): esta interrupción se activa al captarse un cambio en alguno de los pines ligados al sensor touch. Cuando se activa indica que se requiere un cambio en el estado START/STOP del cronómetro por lo que reanuda la tarea que corresponde.
- StartStopTask(): esta tarea se encarga de cambiar el estado del cronómetro. Si se encuentra en estado parado pasa a activo y si estaba activo pasará a parado. Este cambio hace que se actualice o no el valor de sec cuando se reanuda la tarea RefreshTask(). Además, muestra por pantalla el estado en el que se encuentra el cronómetro en este momento.

- ResetTask(): esta tarea es la que se encarga de comprobar el sensor elegido, ya sea sensor Hall o sensor infrarrojo, para que cuando se active se resteé el cronómetro. El reseteo consiste en establecer el valor de min y de sec a 0 y parar el cronómetro. Esta tarea esta definida dos veces una para cada sensor y según la opción escogida en el menuconfig se ejecutara una u otra. Para el caso de sensor Hall se considera activado el reset si se ha detectado una variación de campo magnético en valor absoluto superior a 100, mientras que para el sensor infrarrojo se da por activado el reset si el valor final de distancia es inferior a 10 cm y superior a 4 cm (debido a la mala interpretación que hace por debajo de dicho valor).

Además, se pide responder a si los ADC están calibrados o no. La respuesta a esta pregunta es si están calibrados y pueden usarse directamente, algo que puede notarse en el propio ejemplo del GitHub. El valor de referencia se puede ver al ejecutar el siguiente script:

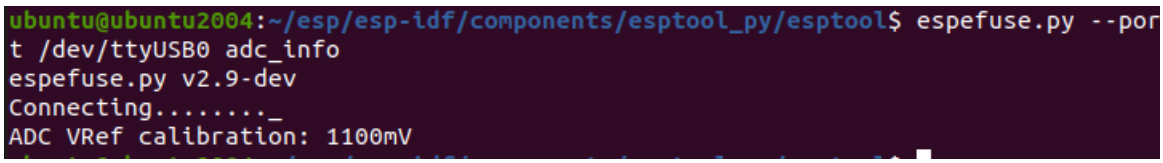

Figura 1: Obtención de VREF

Como puede verse en la anterior figura el valor VREF es 1100mV.