

Tarea 1

Pregunta 1:

Conecta el pin marcado "AD0" del sensor a la línea 3.3V. ¿Qué cambia con respecto a no conectarlo? ¿Para qué puede ser útil esta funcionalidad?

Sin conectarlo, el valor decimal que corresponde a la dirección del dispositivo en registro es de 104(0x68), al conectar el VCC al AD0 la dirección del dispositivo cambia a 105, esto es útil cuando necesitas usar varios sensores del mismo **tipo** en un mismo microcontrolador, (la dirección de registro suele venir predeterminada por el fabricante, por lo que al usar dos dispositivos del mismo modelo habrá un conflicto).

Tarea 2

Pregunta 1:

"El máster es el ESP32 y el esclavo (puede haber múltiples esclavos, sin embargo, en nuestro motherboard para esta práctica solo tenemos conectado uno) es el MPU6050 en este caso"

1Cada registro en el MPU6050 tiene 8 bits, para leer cada registro es necesario que el maestro mande un bit ACK al esclavo para que este sepa que puede continuar enviando bits, es decir, por cada registro son necesarios 9 bits o 9 ciclos, como leemos 14 registros entonces necesitamos $9 * 14 = 126$ ciclos de reloj(en el último registro que se lee, el máster en lugar de mandar ACK, enviará NACK y con ello el esclavo sabrá que debe dejar de mandar bits) son los necesarios para leer todos los datos de los registros, ahora, para empezar a leer registros necesitamos los 7 bits que identifican la dirección del esclavo en el bus I2C + el bit de escritura (primero el maestro escribe la dirección del esclavo, esclavo envía bit 0 de ACK, luego la dirección del registro (8 bits), el esclavo envía bit ACK para confirmar que la dirección existe), el maestro envía dirección del esclavo I2C para lectura con bit 1, esclavo envía ACK si puede), entonces $9 \text{ ciclos} * 3 = 27$. Entonces tendremos 153 ciclos para leer una muestra completa. Según el SCL, podemos leer 400k ciclos por segundo. Entonces tardaremos 0.0003825 segundos en leer una muestra completa.

Pregunta 2:

Completamos una muestra en 0.0003825 segundos, entonces la cantidad de muestras que podemos completar en un segundo es el inverso: 2614, esa es nuestra frecuencia de muestreo.

Tarea 3

Preguntas 1:

El sensor captura los valores medidos en g(unidades de gravedad terrestre: 9.802 m/s^2) y son de 0 g para X e Y, 1g para Z, estos son los valores medidos con el sensor en reposo en posición horizontal, al girarlo sobre un lado (suponemos 90°) hacemos que otro eje, ya sea el X o Y sea paralelo a Z, con lo cual, adquirirá el valor que tiene Z. Es decir, dependiendo de cómo lo hayamos girado, ya sea o el eje X o el eje Y tendrá el valor de 1g o -1g. Con el sensor en reposo o en no reposo, mide la aceleración en los tres ejes espaciales, suponiendo que lo dejamos en reposo horizontal, X e Y medirán 0g porque no hay aceleración en esos ejes y Z medirá 1g debido a la gravedad

Pregunta 2:

Devuelve, -10.54° en X, 1.54° en Y y -1.13° en Z aprox. No son para nada los esperados, al estar en reposo la velocidad de giro es nula porque pues está en reposo, no está girando y sin embargo los valores que devuelve el MPU6050 no son exactamente esos, esto se debe al offset del MPU6050.

Pregunta 3:

Listo.

Tarea 4

“Cumplir con lo que dice el pdf” listo

Tarea 5

“””

“Pin.OUT y Pin.IN se ven desde el punto de vista del ESP32 porque nos estamos conectando directamente con él vía código, o sea, Pin.OUT implica que el ESP32 desde ese PIN envía voltaje/datos y Pin.IN implica que el ESP32 desde ese PIN recibe voltaje/datos.”

“””

Pregunta 1:

Ocurre que se detiene la impresión de datos por pantalla, al volverlo a conectar la impresión se reanuda. Esto es correcto que sea de esta manera porque nosotros ya no estamos leyendo vía el bus i2c en cada iteración un registro que nos diga si hay o no muestras disponibles como en el ejercicio 4, aquí estamos conectando un cable fuera del bus que une al ESP32 y al MPU6050 en donde el pin del ESP32 (12 en este caso) se establece como input, o sea, no manda electricidad/información, sino que recibe. La data que recibe es del MPU6050 y esta le dice con un high (1) si hay nueva data o con un low (0) si no hay. Y nosotros nos basamos en estas condiciones para mostrar datos en pantalla, por lo que al desconectar el cable se deja de enviar esta confirmación, jamás podrá ser True que tenga nuevas muestras y no accederá al condicional que las printea, al volverlo a conectar esto se soluciona porque reanudas el proceso normal.

Pregunta2:

La condición del if evaluada sin interrupciones en el flujo del programa (en condiciones normales) es una contradicción, es siempre falso para cualquier valor de verdad. Sin embargo, en nuestro caso sabemos que la interrupción “pausa” la ejecución del programa rompiendo el flujo normal y pasa a la función handler, que se encarga de setear la new_sample variable a True (si se lanzó handler es porque ya hay una nueva muestra disponible). Al volver al flujo principal del programa lo hace en el punto exacto desde donde fue interrumpida, esto puede ocasionar la aparente paradoja que se nos presenta. Lo que ocurre cuando se satisface la expresión del condicional es que al empezar a evaluar el if, el intérprete toma la primera parte del and con new_sample False, pero justo en ese momento, cuando va a evaluar la segunda parte del and, salta la interrupción, se cambia el valor de new_sample a True y al volver a nuestro flujo de programa lo hace en el punto exacto donde lo dejó antes de ser interrumpida, o sea, a evaluar la segunda parte del and pero ahora new_sample es True, esto provoca que se ejecute lo que hay dentro del condicional. Además, esto se puede comprobar imprimiendo el new_sample antes del condicional y veremos que es False y justo después de entrar si lo imprimimos veremos que es True. O sea, el valor de new_sample fue modificado mientras se evaluaba la condición del if.

Si antes del condicional hacemos v = new_sample y evaluamos con v en lugar de con new_sample. Nunca se ejecutará el código. Esto ocurre porque v no cambiará su valor hasta que se ejecute de nuevo esa línea de código en la próxima iteración, aunque se interrumpa el flujo y cambie el valor de new_sample v no lo hará, así que nunca se cumplirá ese condicional

