Lab Session 4: Encóder

Estimated time: 1.5h (1 session)

Descripción

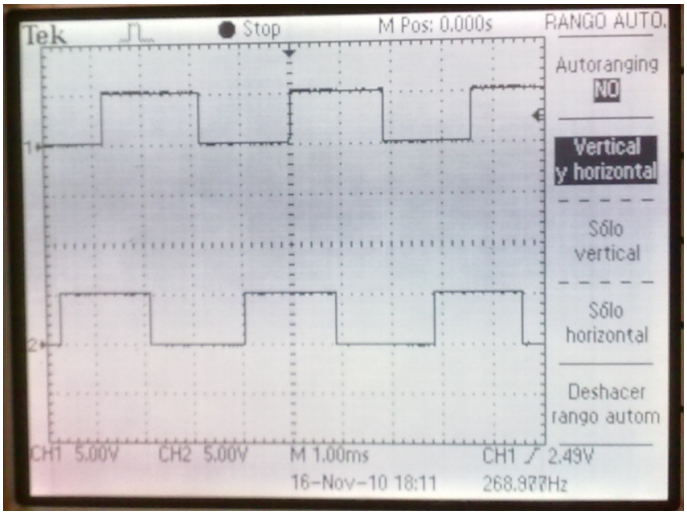
Para conocer la posición en la que se encuentra el eje de un motor se suelen emplear encoders incrementales. El objetivo de esta práctica es la programación de la lectura de este tipo de sensores, de manera que podamos conocer la posición exacta (asumiendo el error debido a la resolución del propio sensor) en la que se encuentra el eje del motor en cada instante de tiempo.

El encoder incremental, normalmente posee 4 señales:

- GND: Masa.
- Vcc: Alimentación (3-5V).
- Canal A: Canal A del encoder
- Canal B: Canal B del encoder.

IMPORTANTE: En la práctica, el encóder se alimenta directamente desde un regulador interno de 5V del que dispone la electrónica del equipo. Eso quiere decir que no es necesario alimentar el encóder externamente. Sin embargo, sí es necesario tener en cuenta GND como referencia para las señales recibidas por los canales A y B.

Si conectamos estos dos canales a un osciloscopio y movemos el eje del motor, podremos apreciar que la señal que se generan en estos dos canales es como la que se muestra en la imagen.



IMPORTANTE: Si no se mueve el eje del motor, no se verá ningún cambio en las señales.

Trabajando con el hardware real

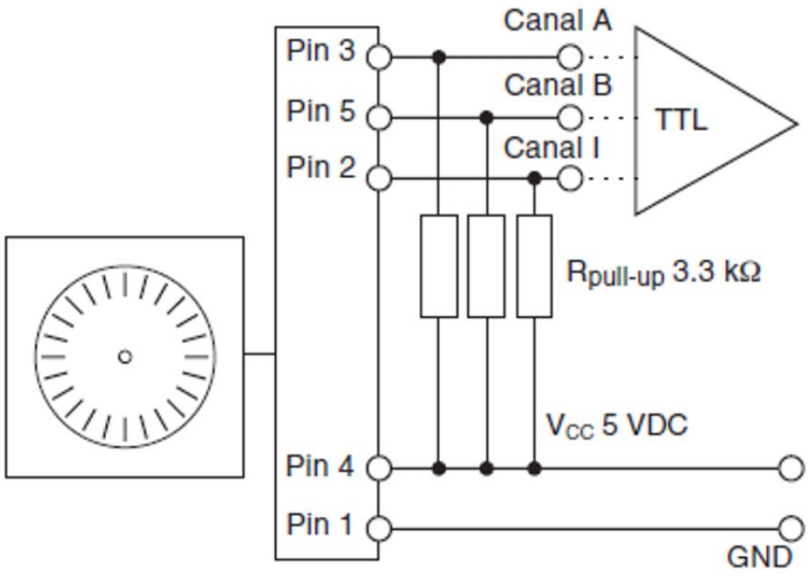
Parte 1 - Identificación de señales

Conectar el osciloscopio al encoder e identificar el color de cada una de señales descritas anteriormente en el motor que se va a emplear. Apuntar esta relación color-señal para poder usarla posteriormente con el microcontrolador. Conectar las señales de los canales A y B, así como la masa al osciloscopio y comprobar

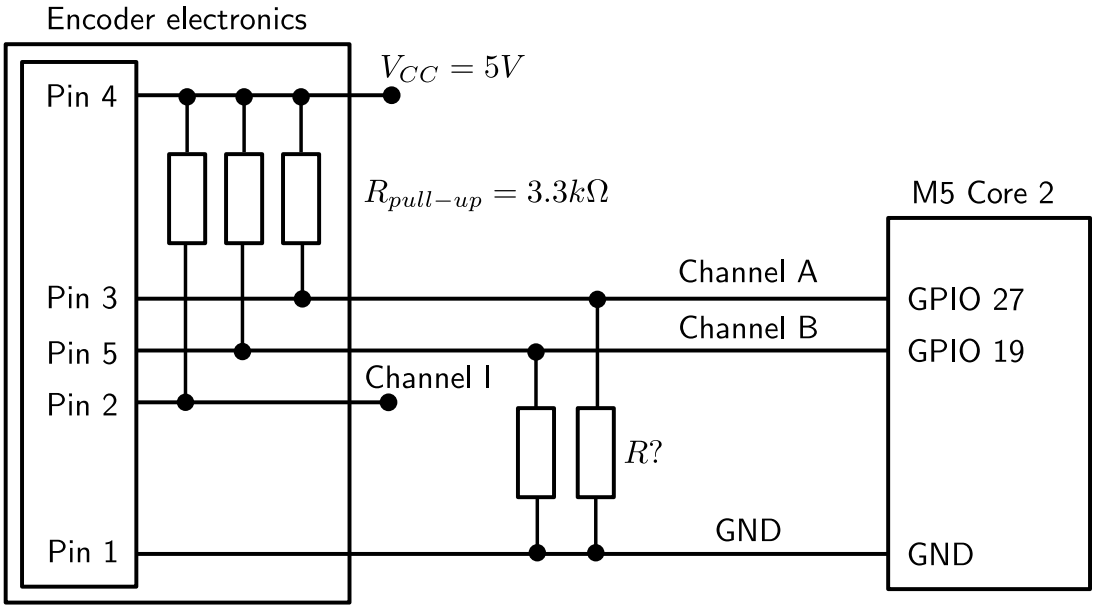
que la respuesta de los dos canales es como la de la imagen anterior. Se recomienda utilizar el botón *Run/Stop* del osciloscopio para detener la lectura mientras el motor se encuentra girando y así poder observarla bien.

Conectar la masa del encoder (GND) al microcontrolador, y los canales A y B a las entradas digitales `GPIO 27` y `GPIO 19`. Esta conexión no se puede hacer directamente tal y como se describe a continuación.

IMPORTANTE: El encóder está alimentado a una tensión de 5V. Eso quiere decir que las señales de los canales A y B en estado alto y sin carga externa proporcionan una tensión de 5V. Sin embargo, la electrónica del ESP32 funciona a 3.3V. Es decir, no se deberían conectar los canales A y B a las entradas digitales del microcontrolador directamente. Para resolver este problema se puede utilizar un divisor de tensión. Para implementarlo, hay que tener en cuenta la electrónica interna del encóder (ver imagen siguiente).



Los canales A y B ya disponen de una resistencia pull-up de $3.3 k\Omega$. Por lo tanto, para realizar el divisor de tensión sólo será necesario incluir la resistencia de la parte baja del divisor de tensión.



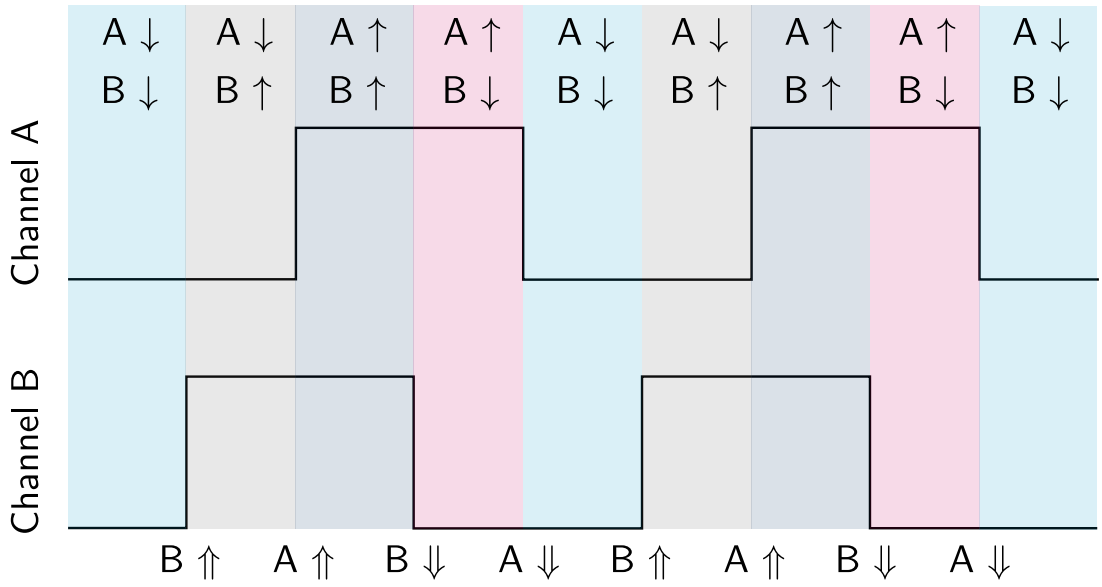
PREGUNTA: ¿Cuál es el valor teórico de la resistencia R que hay que colocar en la parte baja del divisor de tensión para que la tensión de los canales A y B pase de 5V a 3.3V cuando estén en estado alto?

Montar el circuito con el divisor de tensión y asegurarse midiendo con el osciloscopio que la tensión de los canales A y B en estado alto no supera los 3.3V.

Parte 2 - Contador de pulsos en el encoder

Desarrollar un código Arduino que permita contar los pulsos generados por ambas señales. Almacenar esta cuenta en una variable global, y tener en cuenta los cambios de sentido para incrementar o decrementar el contador. Emplear las interrupciones de las entradas digitales para ello.

Para contar los pulsos es necesario tener en cuenta los flancos de subida y/o de bajada de cada canal. Además, será necesario ver el estado de los canales después del flanco. Por ejemplo: Si hay un flanco de subida de A, y tanto A como B están en estado alto, se está girando en un sentido (positivo, por ejemplo, por lo que hay que aumentar el contador). Si, por el contrario, después del flanco de subida de A, A está en estado alto pero B está en bajo, significa que se está girando en sentido contrario (negativo, en este caso, por lo que habría que disminuir el contador). A continuación se representa un ejemplo en el que se está girando en un sentido determinado. Se puede ver cómo la señal B va "adelantada" a la A. Si se girase en sentido contrario, se vería cómo A "adelanta" a B. Esto se puede comprobar en el osciloscopio y el motor real girándolo con la mano en un sentido u otro.



Ejemplo de implementación de manejador de interrupción:

```
void IRAM_ATTR ISR_Ejemplo()
{
  ...
}
```

El valor del contador se puede enviar a través del Puerto Serie de la siguiente forma:

```
void setup()
{
  ...
  Serial.begin(9600);
}
```

```
void loop()
{
  ...
  Serial.println(contador);
}
```

Parte 3 - Contar pulsos por vuelta

Comprobar los pulsos por vueltas que genera el encoder. Buscar esta información en los detalles técnicos del motor y el encoder y comprobarlo con el encóder real de forma experimental.

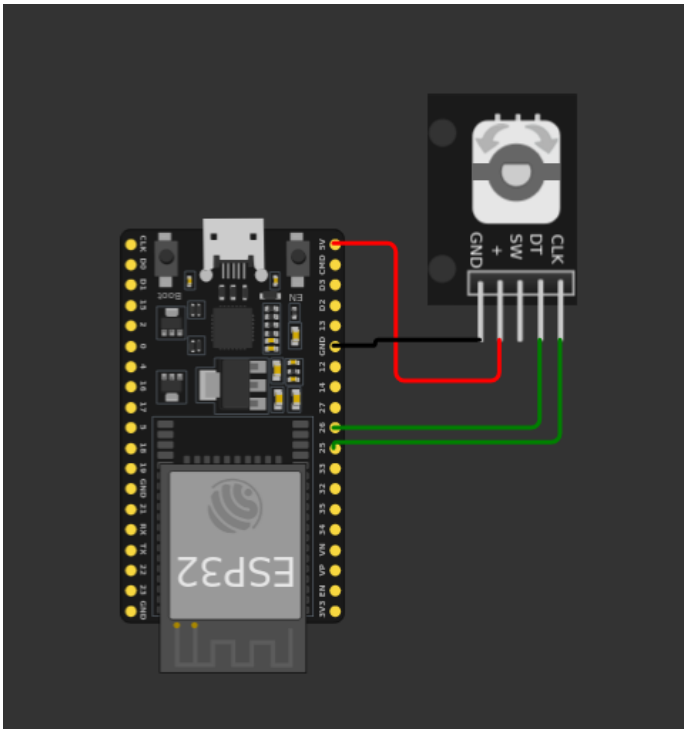
PREGUNTA:

- Si se tienen en cuenta los flancos de subida del canal A, ¿cuántos pulsos por vuelta podríamos medir?
- ¿Y si se tienen en cuenta los flancos de subida de los canales A y B?
- ¿Y si se tienen en cuenta los flancos de subida y de bajada de los canales A y B?

PREGUNTA: ¿Podríamos mandar la posición en grados en vez de en pulsos del encóder por el puerto serie? ¿Cómo?

Trabajando en simulación

Se puede simular un encoder en wokwi utilizando el componente Rotatory Encoder. El diagrama en simulación quedaría de la siguiente forma:



PREGUNTA: ¿Qué diferencias hay entre el funcionamiento de este encóder con el del motor de las prácticas?