OPENCOCKPITS IOCard USBSTEPPER

MANUAL DE INSTALACION Y USO

Opencockpits

All for your cockpit

INTRODUCCION

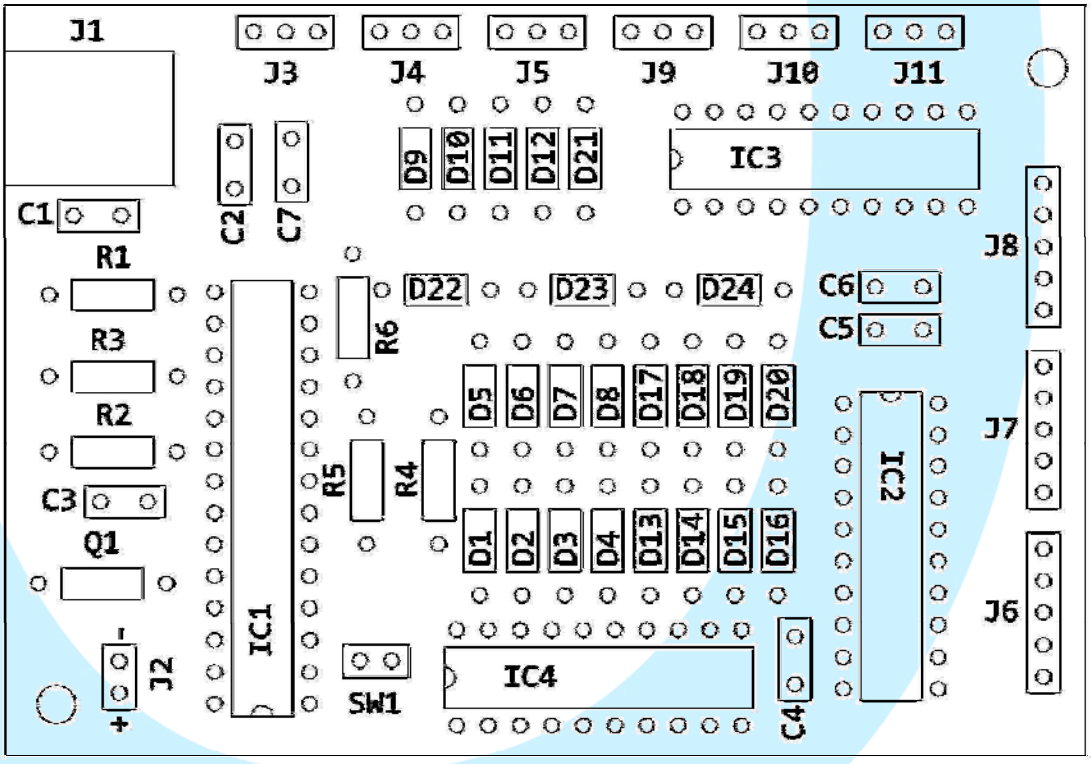
La placa USBStepper puede gestionar hasta 3 motores paso a paso, tanto bipolares como unipolares.

Esta placa permite controlar muy fácilmente los motores paso a paso que pueden servir como base, por ejemplo, a gauges que necesiten de un giro de mas de una vuelta, como en el caso de un altímetro y en los que un servo no sea suficiente.

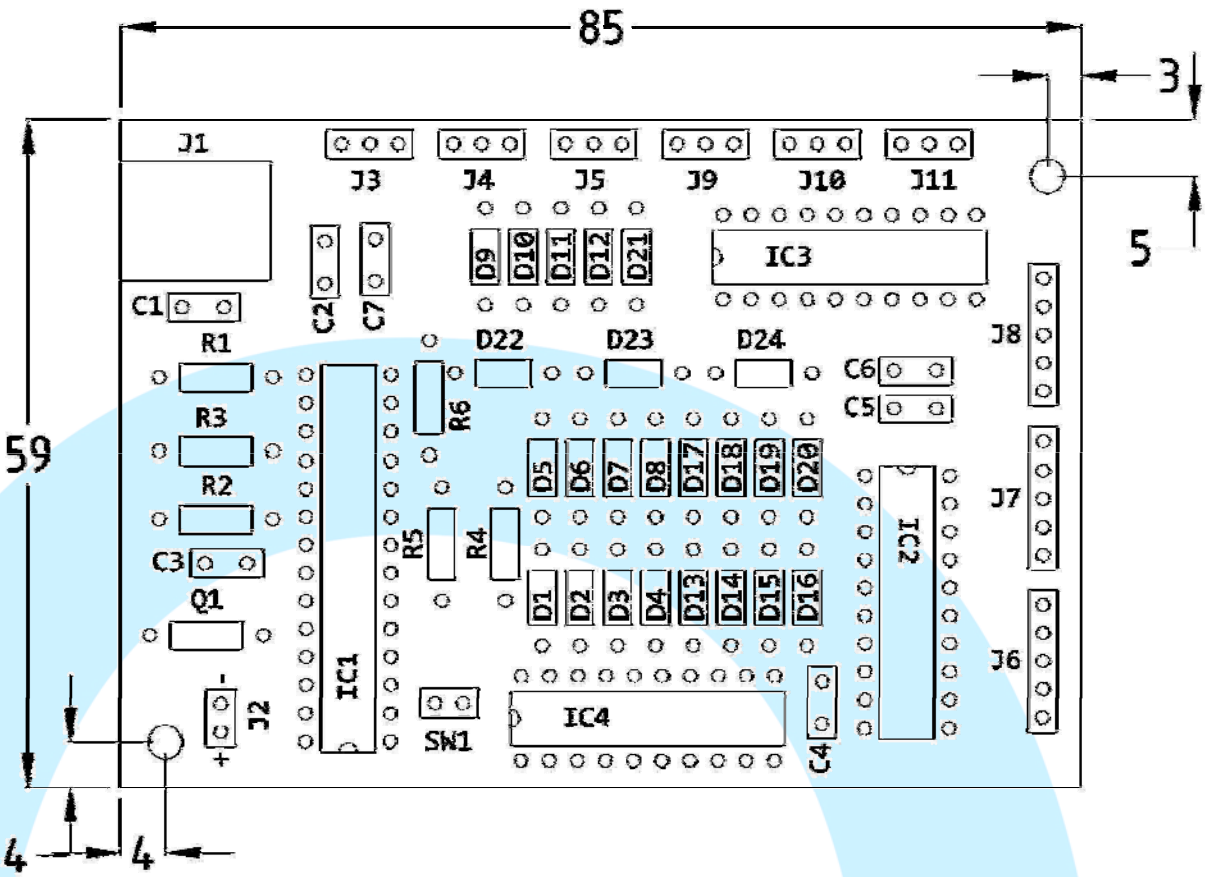
La conexión de la tarjeta al ordenador es mediante puerto USB y al conectarla es automáticamente detectada e instalada como dispositivo HID, asimismo para su gestión se usa el protocolo IOCP.

LISTADO DE COMPONENTES

- C1, C4, C5, C6, C7 = CONDENSADORES 0.1Mf
- C2, C3 = CONDENSADORES 22Pf
- D1 a D24= DIODOS 1N4007
- IC1 = MICROCONTROLADOR 16C745
- IC2, IC3, IC4 = CIRCUITOS INTEGRADOS L293E
- J1 = CONECTOR USB
- J2 = CONECTOR ALIMENTACION 2 PINES
- J3, J4, J5, J9, J10, J11 = CONECTORES DE 3 PINES
- J6, J7, J8 = CONECTORES 5 PINES
- Q1 = CRISTAL CUARZO 6MHZ
- R1 = RESISTENCIA 100R
- R2, R4, R5, R6 = RESISTENCIA 10K
- R3 = RESISTENCIA 1K5
- SW1= RESET 2 PINES



DIMENSIONES PRINCIPALES:



DESCRIPCION DE LOS CONECTORES:

- J1 = Permite la conexión al ordenador directamente a través del puerto USB. En el momento de conectarse el ordenador reconocerá la tarjeta y automáticamente instalará el driver para dispositivos HID.
- J2 = Conector de alimentación para los motores (la misma alimentación para los 3)
- J3 a J5 = Conectores para las entradas analógicas.
- J6 a J8 = Conectores para los motores (ver esquema de conexionado)
- J9 a J11 = Conectores de 3 pines para los sensores de posición

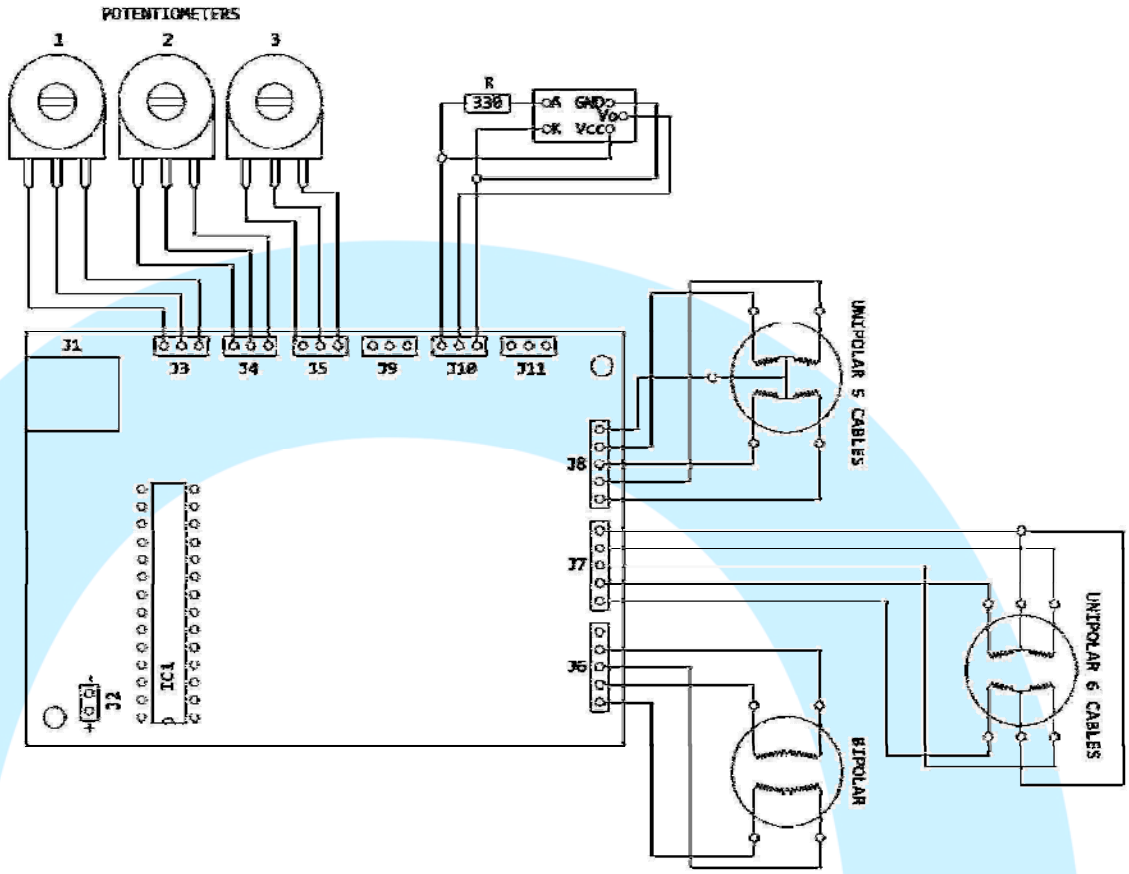
ESQUEMA DE CONEXIONADO

La conexión de la tarjeta es extremadamente simple, para los potenciómetros tenemos los conectores de 3 pines (J3 a J6) y van conectados como podemos ver en la imagen.

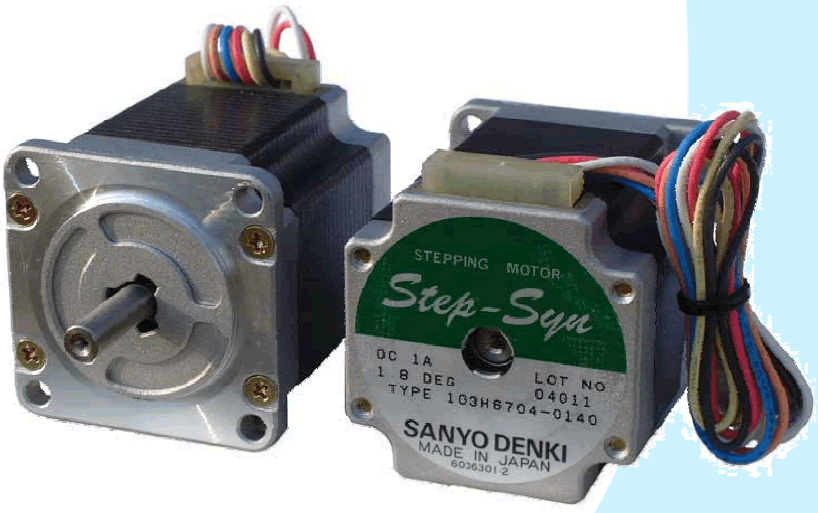
Para los motores tenemos los conectores J6 a J8, conectando los motores en función del tipo de motor que usemos (ver sección motores más abajo).

Para los motores bipolares, usaremos los pines 1 y 3 para una bobina y los pines 2 y 4 para la otra bobina, dejando el pin 5 libre. Para conectar los motores unipolares usaremos los mismos pines pero en este caso usaremos el pin 5 para conectar el común de las bobinas, en el caso de 5 cables solo tenemos un cable común, pero en el de 6 hilos debemos juntar los 2 comunes y conectarlos, como antes, en el pin 5.

Los sensores, asimismo van conectados en los conectores J9 a J11 en la forma que podemos ver en el esquema publicado más abajo, de esta forma podremos controlar el número de pasos de cada vuelta (ver sección sensores más abajo).



MOTORES PASO A PASO



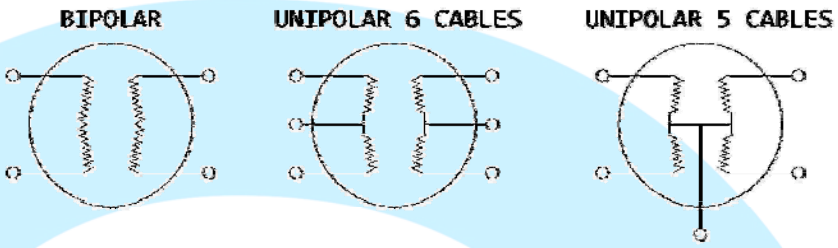
La característica principal de los motores paso a paso es el hecho de poder moverse un paso cada vez, por cada pulso que le enviemos.

Estos pasos pueden variar según los grados que se mueve en cada pulso, que van desde los 90° hasta los más pequeños, que se mueven 1.8 °, siendo estos últimos los que más precisión nos pueden dar, esto quiere decir que

para dar una vuelta completa (360 °), para el primer caso necesitaríamos tan solo 4 pasos (90 ° x 4 = 360 °), siendo necesarios para el segundo caso 200 pasos (1.8 ° x 200 = 360 °).

Estos motores están formados básicamente por un rotor sobre el que van aplicados diferentes imanes permanentes y por un cierto número de bobinas excitadoras, bobinadas en su estator. Las bobinas son parte del estator y el rotor es un imán permanente. Toda la excitación de las bobinas debe controlarse externamente, mediante el controlador adecuado.

Dependiendo de la configuración de las bobinas del estator tendremos básicamente dos tipos diferentes de motores paso a paso:



BIPOLARES:

Usualmente tienen cuatro cables que corresponden a los extremos de las dos bobinas que lo forman (ver dibujo explicativo arriba)

UNIPOLARES:

Estos suelen tener 5 o 6 cables, dependiendo del conexionado interno del cable común de las bobinas (ver dibujo más arriba), obsérvese que el común en el caso de los motores de 6 cables puede unirse asimismo externamente, para que el conexionado sea de 5 cables.

Para la tarjeta USBStepper no es problema controlar ninguno de los dos tipos, tanto bipolares como unipolares, simplemente deben configurarse adecuadamente los cables a conectar.

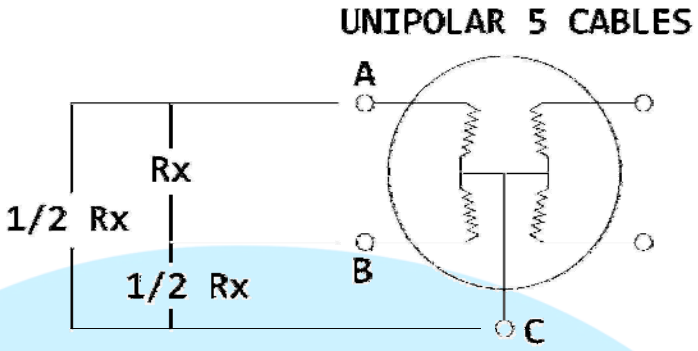
Debe tenerse en cuenta, que como dispositivos mecánicos que son, los motores paso a paso deben vencer ciertas inercias, por lo tanto el tiempo de duración y la frecuencia de los pulsos aplicados es un punto muy importante. Es este sentido el motor debe alcanzar el paso antes de enviarle el siguiente pulso, por este motivo, si la frecuencia de pulsos es muy elevada, el motor podría reaccionar de la siguiente forma:

- No realizar ningún movimiento en absoluto
- Vibra pero sin llegar a girar
- Giro errático
- O llegar a girar en sentido contrario al deseado

Si no disponemos de hojas de datos de los motores, bien porque sean reciclados o recuperados, o bien porque son nuevos pero no disponemos de esos datos, es posible averiguar la distribución o conexionado de los cables a los bobinados y el cable común, en el caso de unipolares de 5 o 6 cables, siguiendo unas pequeñas instrucciones detalladas a continuación:

En el caso de motores unipolares de 6 cables, es mejor unir los dos cables comunes (generalmente son del mismo color) antes de empezar a realizar los tests.

Use un tester para chequear la resistencia entre pares de cables, el cable común, será el único que tenga la mitad de la resistencia. Esto es debido a que entre el común y cualquier otro cable solo tiene una bobina, en cambio entre los pares de cables siempre existen dos bobinas (ver esquema de testeo)



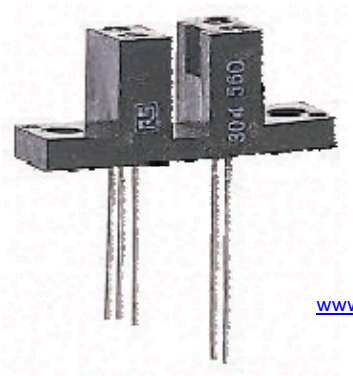
En este caso, si medimos con el tester la resistencia entre los puntos A y B, obtendremos una resistencia x, pero si por el contrario medimos entre los puntos A y C o B y C, solamente obtendremos la mitad de la resistencia, lo cual nos indicara que el cable C es el común.

Para los motores bipolares (generalmente de 4 cables), la identificación es más sencilla, simplemente midiendo la resistencia entre pares de cables, los que formaran parte de la misma bobina tendrán continuidad (resistencia muy baja), siendo los otros dos cables los extremos de la otra bobina. Para averiguar la polaridad de estas bobinas, simplemente lo haremos con el método prueba y error, invirtiendo los cables de posición si el sentido de giro no es el esperado.

Recuerde:

- Un motor con 5 cables es, casi seguro, UNIPOLAR.
- Un motor con 6 cables es, casi seguro asimismo, UNIPOLAR, pero con 2 cables comunes (pueden ser del mismo color)
- Un motor con solo 4 cables es comúnmente BIPOLAR

SENSORES DE POSICION



www.amidata.es refª 304-560

Estos sensores nos servirán para que la tarjeta USBStepper conozca siempre la posición del eje cuando el motor gira, así el eje, queda siempre en la misma posición.

La tarjeta, al conectarse al software, lo primero que hace es comenzar a girar el motor para que pase hasta 2 veces por el sensor óptico. ¿Por qué 2 veces?, pues la primera lo que hace es poner un contador interno a 0 y a partir de este momento empieza a contar el numero de pasos que da hasta completar la segunda vuelta y pasar por segunda

vez por el sensor óptico, así conoce exactamente el numero de pasos que da el motor y el tiempo que tarda en dar esa vuelta completa.

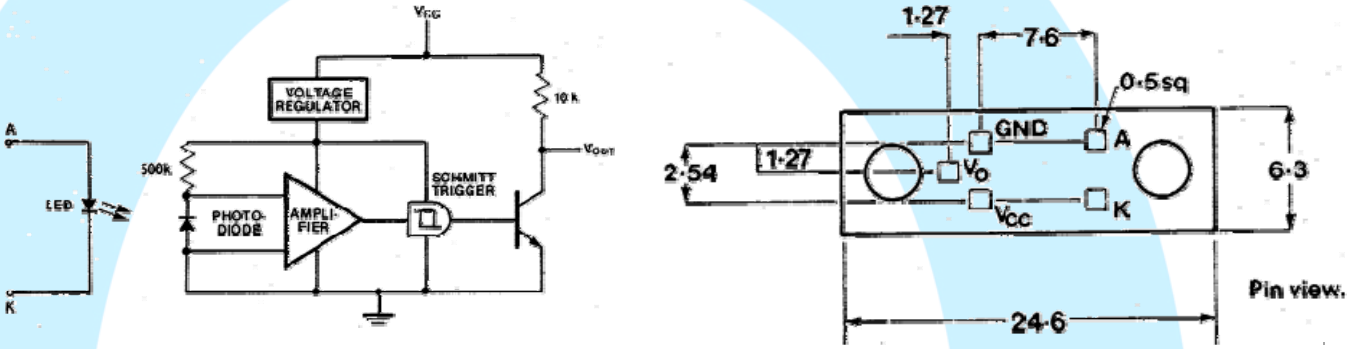
En este momento, ya conocemos el numero de pasos de una vuelta, asimismo conoce la posición inicial marcada por el sensor, entonces ya puede calcular cualquier posición relativa a partir de esa posición inicial, simplemente llevando la cuenta de los pasos que da hacia un lado y hacia el otro.

Los sensores ópticos que se utilicen deben ser capaces de suministrar tensiones TTL (o y +5V), por esto se recomiendan los que llevan incorporada la microelectrónica para entregar tensiones TTL.

Es tos sensores consisten en un emisor de luz y un receptor, de tal manera que la tensión pasa a ser 0 al ser interrumpido el rayo y de +5V cuando el sensor no está obturado.

Para el conexionado, vea el esquema más arriba.

Nosotros recomendamos por ejemplo el que se puede localizar en AMIDATA en www.amidata.es con referencia 304-560, este es el esquema de dimensiones y pines del citado sensor:



USANDO SIOC

Asegúrese de que tiene instalada la versión 3.46 o superior, si no la tiene puede bajarse la última versión aquí:

http://www.opencockpits.com/catalog/info/information.php?info_id=31&cPath=2

Una vez tenga instalada la versión adecuada, lo primero que haremos será configurar los parámetros del fichero sioc.ini, para asegurarnos de que la tarjeta este correctamente identificada con el numero de dispositivo que le corresponda.

Deberemos editar la entrada en el fichero sioc.ini, de tal manera que asignemos un índice de dispositivo a cada tarjeta que instalemos, creando una entrada en el fichero sioc.ini por cada tarjeta USBStepper conectada, y será en el siguiente formato:

USBStepper=XX,YY

Donde XX nos indica el número de índice, dentro de nuestro sistema de tarjetas e YY el número de Dispositivo del puerto USB al que está conectado.

Por ejemplo, si conectamos dos tarjetas USBStepper con los números de dispositivos 35 y 42 (estos números se pueden averiguar fácilmente con el propio programa SIOC.exe, ya que en él se nos suministra la información de la tarjeta) entonces las declararíamos en el sioc.ini de la siguiente manera:

USBStepper=1,35
USBStepper=2,42

No hay problema por tener más tarjetas IOCards conectadas en este ordenador, mientras estén correctamente definidas, como tampoco por tener módulos Opencockpits también conectados.

MOTORES PASO A PASO:

Para hacer referencia al número de servo de forma exacta, debemos tener en cuenta el número de índice que le hemos asignado a cada una de las tarjetas USBRelays.

Ahora en SIOC, debemos definir la salida en la forma estándar:

Var *VVVV*, **name** *NNNNNNNNNNN*, **Link USB_STEPPER, device** *DD*, **Output** *S*, **posL** *LLL*, **posC** *CCC*, **posR** *RRR*,**Type** *T*

- VVVV= número variable
- NNNNNNNNNNNNN = nombre variable *(opcional)*
- DD = numero de índice definido en el ini *(opcional, si la tenemos declarada como 0 no importa hacer referencia al número de Device)*
- S = numero de motor 1-3
- LLL = valor de la velocidad del motor 0-255 *(cuanto menor sea su valor, mayor velocidad, pero debemos tener cuidado de no sobrepasar la velocidad máxima del motor)*
- CCC = numero de pasos del motor 0-65535 *(si dejamos este número en 0, la calibración será automática)*
- RRR = número máximo de pasos por decima de segundo (0-255) *(este valor es delicado porque si ponemos un número muy alto, desbordaremos al buffer interno donde acumulamos las ordenes de posicionamiento y por lo tanto perdería la posición, si es bajo, por el contrario bajara la velocidad de posicionamiento, un numero entre 3 y 5 es perfecto)*
- T = si ponemos este parámetro como “H”, activaremos la opción de medios pasos, teniendo el doble de pasos totales

Ejemplo de definición:

Var 0001, name step_alt, Link USB_STEPPER, Device 1, Output 1, PosL 6, PosC 0, PosR 4, Type H

ENTRADAS ANALOGICAS:

Para la lectura de las entradas analógicas se deberá usar el formato siguiente:

Var *VVVV*, **name** *NNNNNNNNNNN*, **Link USB_ANALOGIC, Device** *DD*, **Input#** *EE*, **posL** *LLL*, **posC** *CCC*, **posR** *RRR*

- EE = numero de entrada analógica 1-5
- LLL = posición máxima del dispositivo a la izquierda
- CCC = posición central del dispositivo
- RRR = posición máxima del dispositivo a la derecha

El resto de parámetros de usaran como en la definición de los relés.

Ejemplo de definición de entrada analógica:

Var 1506, name pot_flaps, Link USB_ANALOGIC, Device 1, Input# 2, posL 1, posC 128, posR 255

EJEMPLO SCRIPT DE SIOC

Fabricaremos un gauge para la altitud (altímetro) y le montaremos un motor a paso en la aguja de cientos, con la reducción de engranajes que consideremos adecuada, para obtener la suficiente precisión. Conectaremos el motor y el sensor a la tarjeta y la alimentaremos con el voltaje preciso para el motor.

Ejecutaremos SIOC y dejaremos que el motor se auto-calibre, si tenemos ejecutado el FS la aguja después del calibrado debería situarse en la posición adecuada respecto a la altura del avión y seguir las acciones del avión, para ello podemos realizar algunos ascensos y descensos.

```
Var 0001, name alt_fs_h, Link FSUIPC_INOUT, Offset $3324, Length 4
{
  L0 = MOD &alt_fs_h ,1000
  &step_alt = L0 * 0.36
}
Var 0002, name step_alt, Link USB_STEPPER, Device 1, Output 1, PosL 6, PosC 0, PosR 4,
Type H
```

La única modificación que deberíamos hacer, es la propia corrección de la aguja, si al montar el instrumento la alineación no ha sido perfecta, para ello sumariamos los grados de diferencia al resultado que enviamos al motor:

```
  L0 = MOD &alt_fs_h ,1000
  L1 = L0 * 0.36
  &step_alt = L1 + XXX // XXX = grados de diferencia
```