Guía de uso:

DESARROLLO DE UN SISTEMA DE CONTROL EN LAZO CERRADO PARA LOS MOVIMIENTOS DEL EXTRUSOR Y LA PLATAFORMA DEL SISTEMA DE IMPRESIÓN UAO 3DP

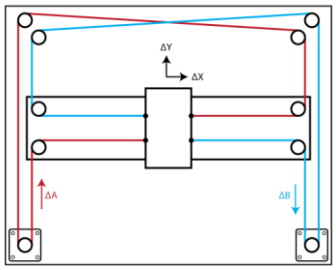
Manuela Cerón Viveros

Tabla de contenido

Guía de uso

Impresora UAO 3DP

La impresora UAO 3DP es una impresora cartesiana desarrollada en la Universidad Autónoma de Occidente, cuenta con la RAMPS 1.4 como controlador principal y tiene configuración CORE XY, lo que significa que se requiere dos motores para permitir el movimiento en los ejes X y Y.



Equations of Motion:

$\Delta X = {}^1/_2(\Delta A + \Delta B), \quad \Delta Y = {}^1/_2(\Delta A - \Delta B)$

$\Delta A = \Delta X + \Delta Y, \quad \Delta B = \Delta X - \Delta Y$

Figura 1 Configuración y ecuaciones sistema Core XY

Componentes

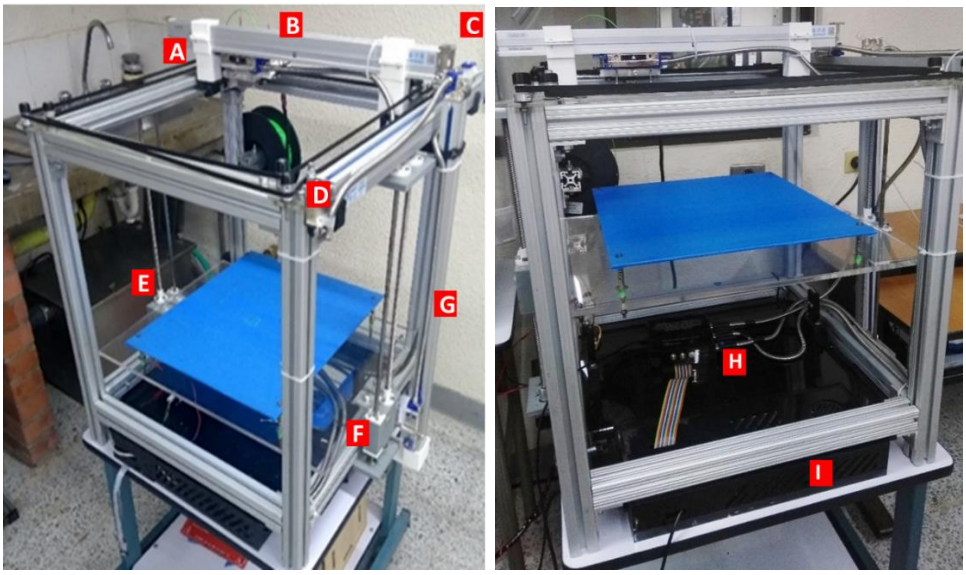


Figura 2 Componentes impresora UAO 3DP

A	Motor 1
B	Encoder óptico lineal eje X
C	Motor 2
D	Encoder eje Y
E	Motor 3
F	Motor 4
G	Encoder eje Z
H	Sistema de control
I	Electrónica impresora UAO 3DP

Ubicación Encoders

Figura 3Ubicación encoder eje X

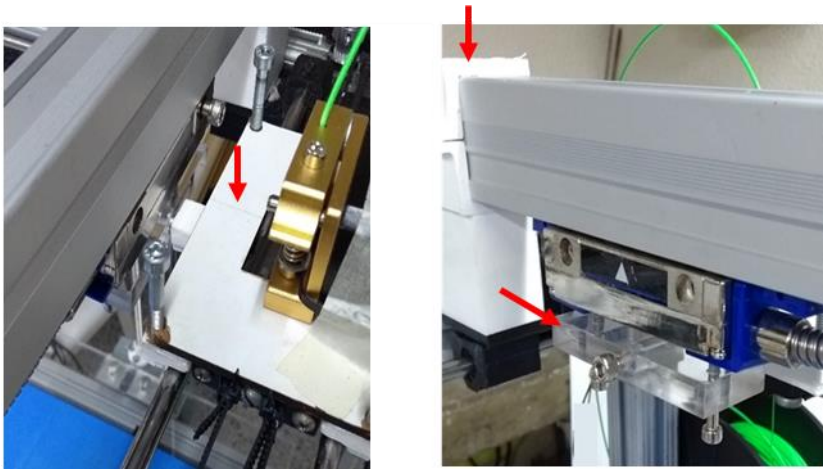


Figura 4 Ubicación encoder eje Y

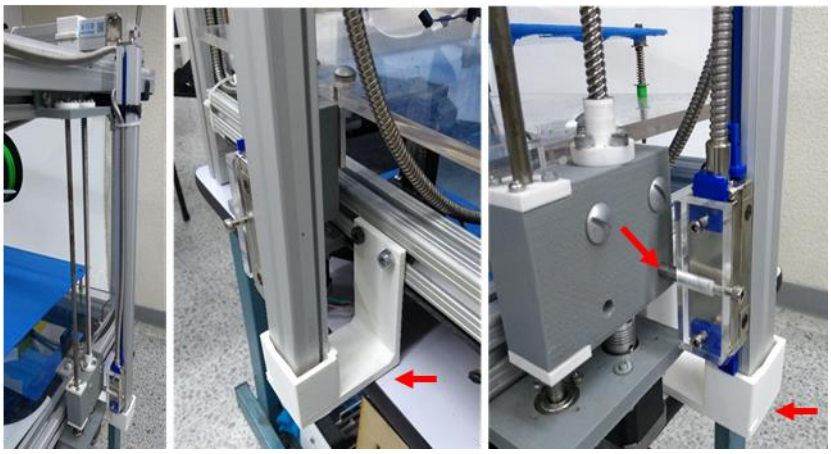


Figura 5 Ubicación encoder eje Z

La posición en los ejes X y Y depende de la resolución de la impresora y de los pulsos enviados por los controladores 1 y 2, mientras que para el eje Z, la posición depende solo de la resolución y los pulsos enviados por el controlador 3. Las siguientes ecuaciones describen el movimiento de la impresora UAO 3DP:

$$1 = (\Delta X - \Delta Y) * Fr \quad (1)$$

$$D2 = (-\Delta X - \Delta Y) * Fr \quad (2)$$

$$D3 = \Delta Z * Fr \quad (3)$$

$$PosX = \frac{D1 - D2}{2Fr} \quad (4)$$

$$PosY = \frac{-D1 - D2}{2Fr} \quad (5)$$

$$PosZ = \frac{D3}{Frz} \quad (6)$$

D1= Pulsos driver 1

D2 Pulsos driver 2

D3= Pulsos driver 3

Fr= Factor de resolución X y Y

Frz= Factor de resolución Z

El factor de resolución depende de los componentes mecánicos de la impresora y corresponden a Fr= 80.5 y Frz= 252.

Sistema de control de posición

El control de posición en lazo cerrado de la impresora se realiza con un microcontrolador Arduino Mega. La posición deseada proviene de los pulsos enviados desde la RAMPS, mientras que la posición actual proviene de la lectura que entregan los encoders lineales. Todas estas señales se conectan a una placa adicional para ubicar sobre la RAMPS, este circuito actúa como un puente, conecta las señales de la RAMPS hacia los drivers, aislando las señales de control PASO y DIR, y las señales que se conectan a las bobinas de los motores de los tres drivers, para evitar enviar que las señales de control se envíen directamente a los drivers del motor, sino que se envían primero al microcontrolador Arduino del sistema de control de posición, que las procesa, corrige si es necesario y envía la secuencia a los drivers del motor.

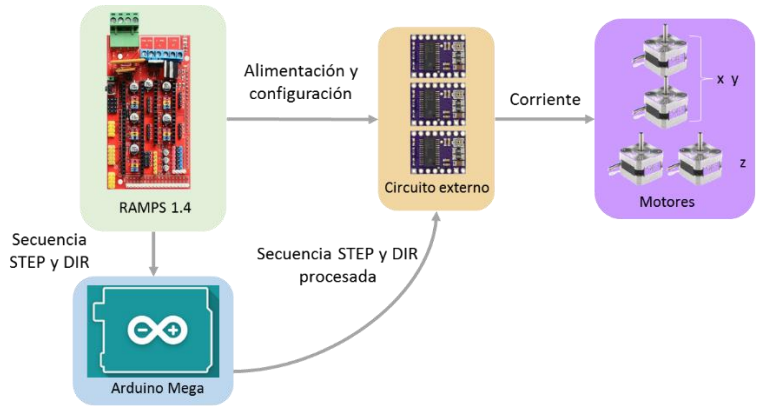


Figura 6 Esquema del sistema de control de posición

Lógica del control de posición

El sistema de control se basa en la siguiente estrategia, en la cual, una vez se reciben los pulsos de los encoders, estos se traducen a la forma de posición en milímetros, luego, utilizando las ecuaciones de la impresora, la posición se traduce en una cantidad de pulsos por cada controlador. Los pulsos deseados se comparan con los pulsos obtenidos, y se calcula un error. Si el error en cada controlador está dentro de un rango establecido, la impresora continúa enviando pulsos sin ninguna intervención. De lo contrario, si el error está fuera de los límites, identifica si el error es positivo o

negativo para generar una secuencia de pulsos y mover el motor en la dirección opuesta y corregir el error; estos movimientos pueden ser en el sentido de las manecillas del reloj (CW) o en contra (CCW). A continuación, se describe la lógica del sistema.

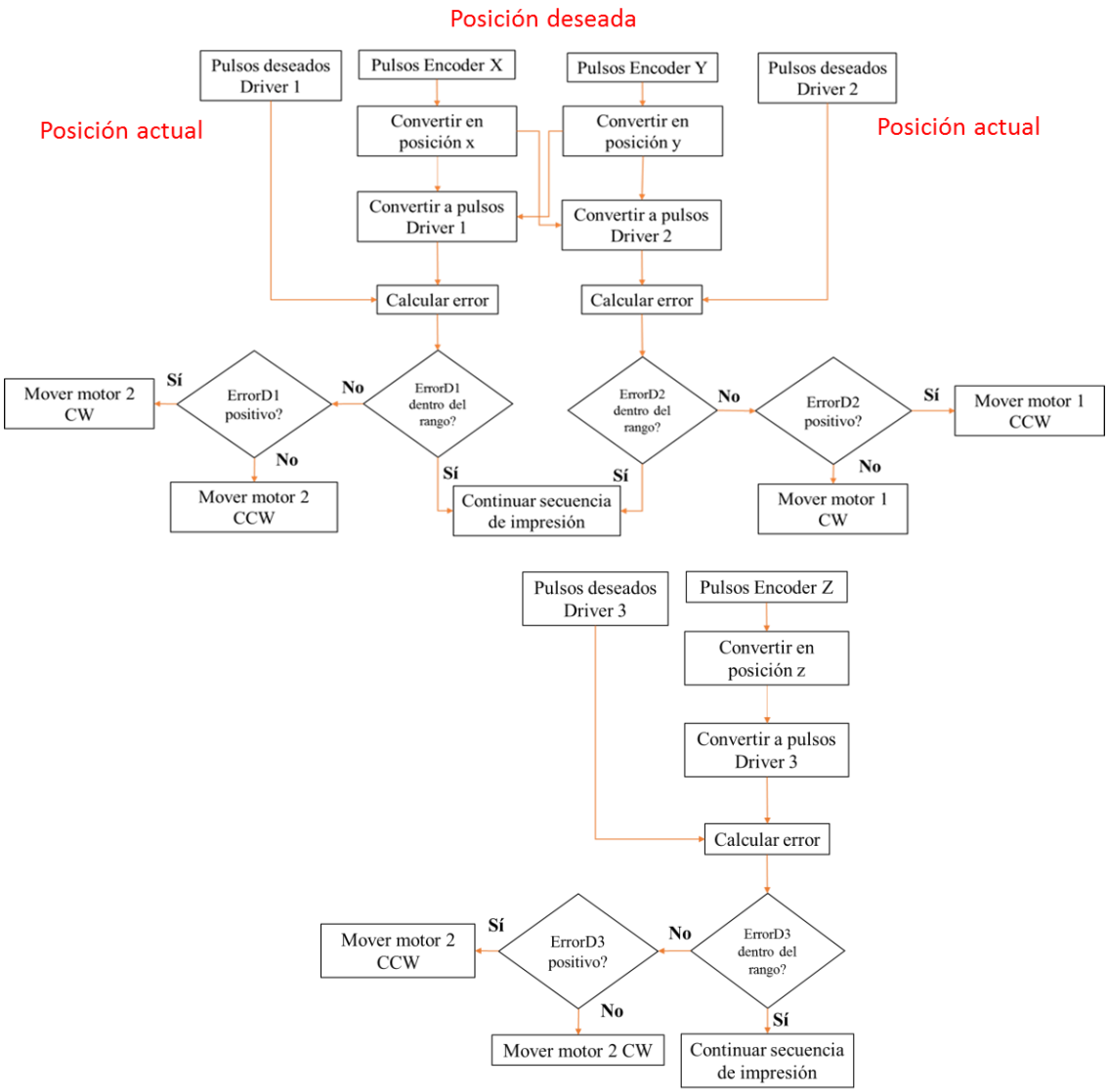


Figura 7 Diagrama de flujo del sistema de control de posición

Programa en Arduino

Variables, contantes y conexiones físicas

Descripción	Nombre	Pin Arduino
Encoder X canal A	encoderXCA	2
Encoder X canal B	encoderXCB	8
Encoder Y canal A	encoderYCA	19
Encoder Y canal B	encoderYCB	11
Encoder Z canal A	encoderZCA	20
Encoder Z canal B	encoderZCB	12
Señal STEP para driver 1 desde Ramps	stepD1Ramps	21
Señal DIR para driver 1 desde Ramps	dirD1Ramps	9
Señal STEP para driver 2 desde Ramps	stepD2Ramps	18
Señal DIR para driver 2 desde Ramps	dirD2Ramps	13
Señal STEP para driver 3 desde Ramps	stepD3Ramps 3	3
Señal DIR para driver 3 desde Ramps	dirD3Ramps	10
Señal STEP hacia motor 1 desde SDC	stepM1	22
Señal DIR hacia motor 1 desde SDC	dirM1	23
Señal STEP hacia motor 2 desde SDC	stepM2	28
Señal DIR hacia motor 2 desde SDC	dirM2	29
Señal STEP hacia motor 3 desde SDC	stepM3	24
Señal DIR hacia motor 3 desde SDC	dirM3	25
Lectura encoder x	PosEncoderx	-
Lectura encoder y	PosEncodery	-
Lectura encoder z	PosEncoderz	-
Posición actual encoder x (mm)	x	-
Posición actual encoder y (mm)	y	-
Posición actual encoder z (mm)	z	-
Pulsos actuales driver 1	PosEnPulD1	-
Pulsos actuales driver 2	PosEnPulD2	-
Pulsos actuales driver 3	PosEnPulD3	-
Pulsos deseados enviados de RAMPS a driver 1	pasosD1	-

Pulsos deseados enviados de RAMPS a driver 2	pasosD2	-
Pulsos deseados enviados de RAMPS a driver 3	pasosD3	-
Posición deseada eje x	Posx	-
Posición deseada eje y	Posy	-
Posición deseada eje z	Posz	-
Error driver 1	errorD1	-
Error driver 2	errorD2	-
Error driver 3	errorD2	-
Error permitido (pulsos)	ep	-
Factor resolución eje x y y	fr	-
Factor resolución eje z	frz	-
Delay en envío de pulsos a motores	dl	-

*SDC: Sistema de control

Estructura del programa

Sección 1: **Definición de librerías y variables**

Set Up

Sección 2: **Definición de modo E/S de las señales que se reciben desde RAMPS y se envían hacia motores**

Sección 3: **Definición de las interrupciones**

Loop

Sección 4: **Selección del lazo abierto o cerrado**

Sección 5: **Posición actual: Convertir lectura encoders en posición (mm)**

Sección 6: **Posición actual: Convertir posición (mm) en pulsos por drivers**

Sección 7: **Posición deseada: Convertir los pulsos por drivers en posición deseada utilizando las ecuaciones de movimiento**

Sección 8: **Cálculo del error - Se resta los pulsos que la RAMPS envía a cada driver, menos la posición actual de cada motor transformada en pulsos por driver**

Sección 9: **Control de posición - Se valida si el error de cada driver está dentro de un rango permitido de error, si es así, se envía una secuencia para contrarrestar el error**

Sección 10: **Impresión de variables**

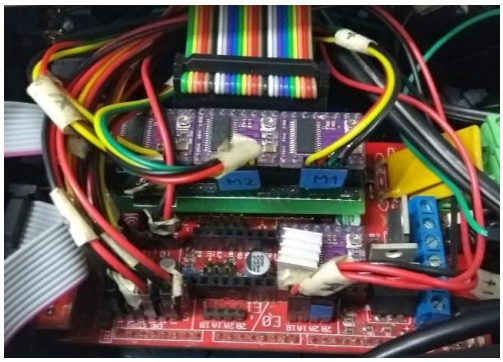
Sección 11: **Recepción de posición actual (Rutinas de atención a la interrupción)**
Sección 12: **Recepción de posición deseada (Rutinas de atención a la interrupción)**
Sección 13: **Métodos para contrarrestar el error en cada driver**

Electrónica y conexiones

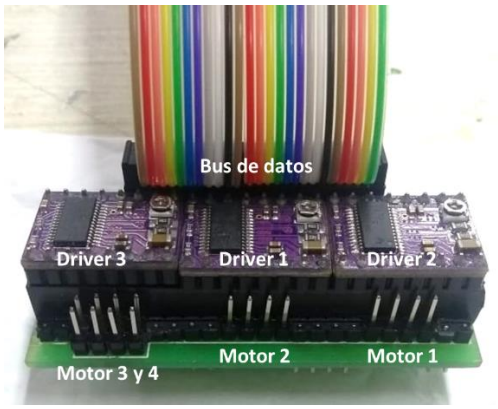
Conexiones generales

RAMPS

La placa se conecta sobre la RAMPS



Sobre la placa se conectan los drivers, los motores y el bus de datos



En la caja donde se protege el microcontrolador del sistema de control, se contectan los encoders, se ubica el extremo opuesto del bus de datos, y se encuentran los interruptores para definir si el sistema trabaja en lazo abierto o cerrado

Nota: La caja donde se protege el microcontrolador del sistema de control cuenta con tres interruptores que permiten definir si la impresora imprime en lazo abierto o cerrado. La posición de estos interruptores debe ser consistente con el modo seleccionado desde la interfaz de Arduino.

Circuito de control de posición

El circuito tiene las siguientes secciones:

- Filas negras: Se conecta el bus de datos hacia el microcontrolador Arduino
- Filas rojas: Se conectan los drivers
- Filas amarillas: Se conecta hacia la RAMPS

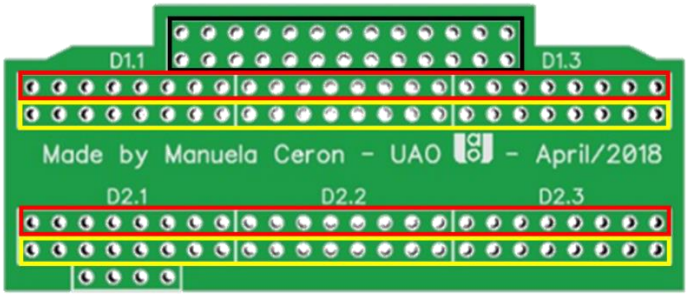


Figura 8 Circuito del sistema de control

Cada driver se conecta a un interruptor de la siguiente manera:

- A través de los pines de la fila amarilla, el circuito recibe las señales DIR y STEP de cada driver enviados desde la RAMPS.
- A través del bus de datos, estas señales deben conectarse al punto común de cada interruptor.
- **Lazo abierto:** Las señales DIR y STEP llegan al punto común de cada interruptor, se conectan nuevamente al circuito y se envían a cada uno de los drivers.
- **Lazo cerrado:** Las señales DIR y STEP se conectan desde el interruptor hacia el microcontrolador Arduino, donde se procesan, y se envían las nuevas señales DIR y STEP hacia los drivers, conectándose mediante el bus de datos hacia el circuito.

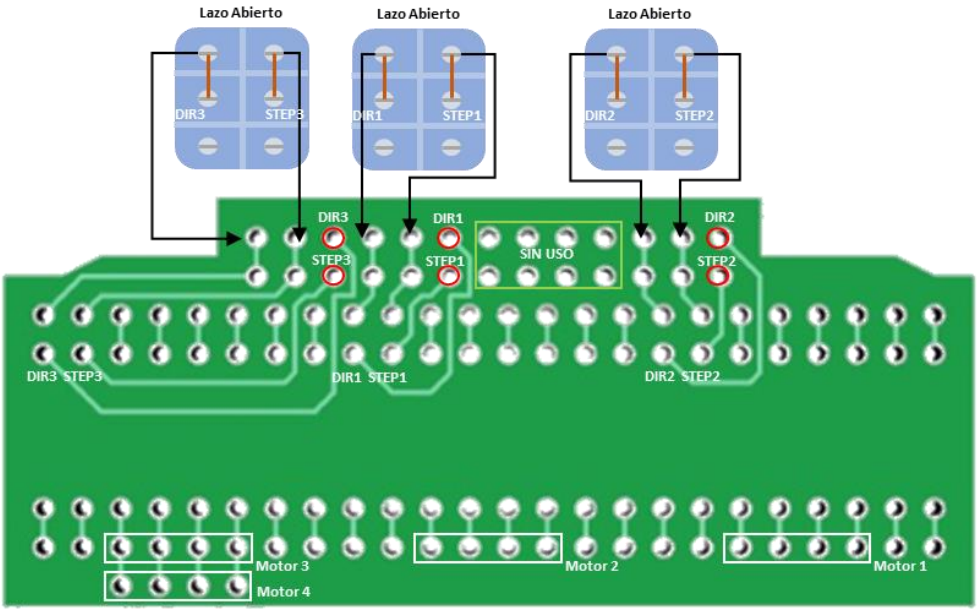


Figura 9 Circuito en lazo abierto

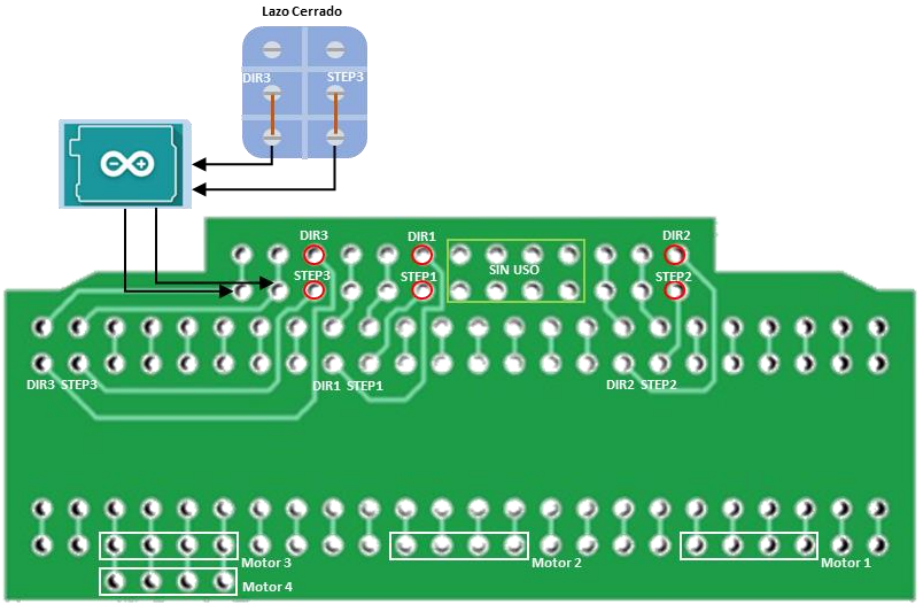


Figura 10 Circuito en lazo cerrado

Conexión bus de datos y Arduino

El extremo opuesto del bus de datos se ubica en la cara frontal de la caja que almacena el microcontrolador. Las señales que provienen del bus deben conectarse con los interruptores y con el microcontrolador Arduino como se muestra a continuación:

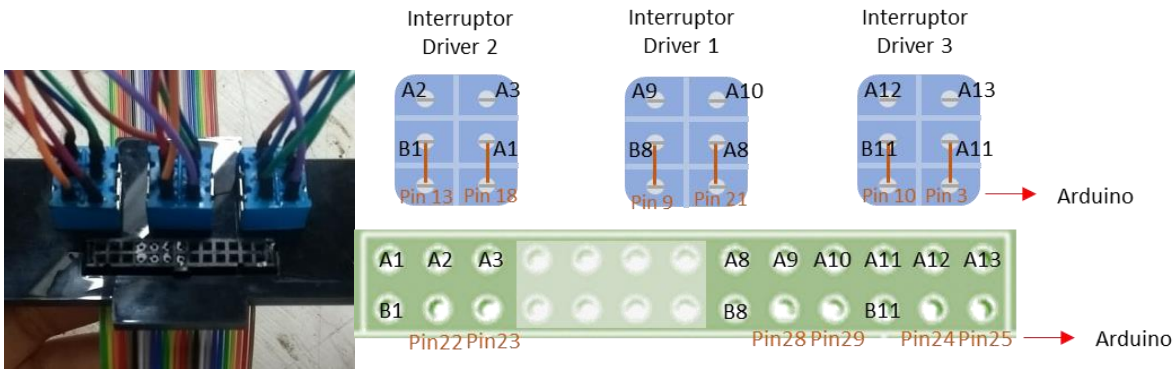
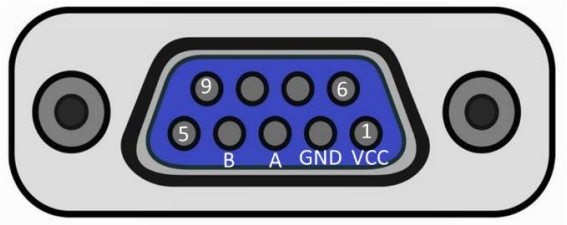


Figura 11 Conexiones entre el bus de datos y el microcontrolador Arduino

Conexión encoders y Arduino

Las señales de cada encoder se identifican y conectan de la siguiente manera:



Señal	Pin Arduino		
	Eje x	**Eje y**	**Eje z**
Canal A	2	19	20
Canal B	8	11	12
GND	GND		
VCC	5V		

Figura 12 Conexiones entre encoders y Arduino

Consideraciones adicionales

- La posición de los tres interruptores debe coincidir con el modo de operación seleccionado desde la interfaz de Arduino.
- Los encoders son de tipo incremental, por tal razón si el microcontrolador Arduino se reinicia o no está energizado, ninguna información será almacenada.
- Cada vez que cambie el modo de operación a través de los interruptores (lazo abierto o lazo cerrado), se debe reiniciar el microcontrolador Arduino, indicando de nuevo el modo de operación para que sea consistente con la configuración física.