

UTILIZACIÓN DEL PUERTO GPIO DE LA JETSON NANO PARA EL CONTROL DE MOTORES A PASOS: MÉTODOS Y APLICACIONES

Luis Alberto Tovar Ortiz¹, Alexis Fernando Cruz Baños¹, Fernando Iván Antonio Hernández Díaz¹, Juan Carlos Herrera Lozada, Dr.¹, Jacobo Sandoval Gutiérrez²

¹Instituto Politécnico Nacional

¹CIDETEC

²Universidad Autónoma Metropolitana

luistovar53@gmail.com, acruzba.2100@aulmno.ipn.mx, fherandezd0200@alumno.ipn.mx, jlozada@ipn.mx, j.sandoval@correo.ler.uam.mx

Boletín No. 104, 1o. de septiembre de 2024

Resumen

La Jetson Nano, una plataforma de desarrollo compacta y potente se está convirtiendo en una opción popular para proyectos de robótica y automatización debido a su capacidad de procesamiento y flexibilidad. Este estudio documenta los procedimientos para habilitar y utilizar las salidas digitales del puerto GPIO, y presenta varias metodologías para el control de motores a pasos, incluyendo el uso de buffers a las salidas GPIO, el uso de controladores directos como el ULN2003A y el L298N, así como módulos más avanzados como el EasyDriver, el A4988 y el DRV8825. Se analizan las conexiones eléctricas, el código necesario y las ventajas y desventajas de cada enfoque. Los resultados demuestran que la Jetson Nano es una plataforma versátil para el control preciso de motores a pasos, adecuada para una amplia gama de aplicaciones en robótica y sistemas embebidos.

1. Introducción

La NVIDIA Jetson Nano 4GB es una plataforma potente y accesible, diseñada para proyectos de inteligencia artificial, robótica y sistemas embebidos. Esta guía tiene como objetivo proporcionar a los principiantes la información necesaria para utilizar las salidas digitales de la Jetson Nano en sus proyectos. Se detallan los pasos para configurar los pines GPIO (General Purpose Input/Output, entradas/salidas de propósito general), la cantidad de salidas disponibles y cómo inicializarlas correctamente.

Además, se exploran diversas metodologías para el control de motores a pasos. Se incluyen métodos como el uso de buffers conversores de 3.3V (volts) a 5V, el controlador ULN2003 para motores unipolares, el controlador L298N y el Easydriver para motores bipolares, así como los módulos A4988 y DRV8825. También se presenta una alternativa para independizar el control del motor a pasos mediante el uso de un Arduino Nano en conjunto con el A4988. Esta guía práctica ofrece soluciones tanto eficaces como accesibles para desarrolladores como entusiastas, que no solo ofrece ejemplos de código y conexiones eléctricas, sino que también compara las ventajas y desventajas de cada método, proporcionando recomendaciones basadas en las necesidades específicas de diferentes proyectos.

El artículo está organizado de la siguiente manera: en la Sección 2 se detallan los fundamentos teóricos del control de motores a pasos y el uso de GPIO en la Jetson Nano. La Sección 3 describe el hardware y las configuraciones comparadas. En la Sección 4, se discuten los resultados, se comparan las metodologías y se proponen futuras líneas de investigación.

2. Fundamentos teóricos

2.1 Principios de funcionamiento y control de motores a pasos

Los motores a pasos son dispositivos electromecánicos que convierten pulsos eléctricos en movimientos mecánicos discretos. A diferencia de los motores convencionales que giran continuamente cuando se les

aplica energía, los motores a pasos giran en ángulos fijos llamados "pasos". Esta característica los hace ideales para aplicaciones que requieren control preciso de posición y velocidad.

Existen principalmente dos tipos de motores a pasos:

- **Unipolares:** Tienen una construcción más simple y suelen tener seis cables, con un punto común en el centro de cada bobina. Son más fáciles de controlar, pero su eficiencia es menor en comparación con los motores bipolares. Su funcionamiento es mediante la activación secuencial de sus bobinas en un orden específico, conmutando el voltaje a estas.
- **Bipolares:** Tienen una construcción más compleja con cuatro cables y no tienen un punto común. Requieren un controlador más complejo, pero ofrecen mayor torque y eficiencia. Para su funcionamiento además de activar las bobinas en un orden específico, también cambia la dirección de la corriente en estas.

2.2 Identificación y Uso de Pines GPIO en la Jetson Nano

La NVIDIA Jetson Nano incluye una serie de pines GPIO que permiten la interacción directa con hardware externo. Los pines GPIO (General Purpose Input/Output) son pines digitales que se pueden configurar como entradas o salidas, permitiendo al desarrollador controlar dispositivos externos como LEDs, sensores y motores.

La Jetson Nano cuenta con 40 pines, de los cuales varios son GPIO utilizables. La disposición es similar a la de la Raspberry Pi, lo que facilita la transición para aquellos familiarizados con esa plataforma.

Cada pin tiene una función específica que se puede configurar mediante software a través del script `/opt/nvidia/jetson`

Pin	Función	Color
GPIO1	3.3V	Naranja
GPIO2	5V	Rojo
GPIO3	5V	Rojo
GPIO4	GND	Verde
GPIO5	GND	Verde
GPIO6	GPIO	Púrpura
GPIO7	GPIO	Púrpura
GPIO8	GPIO	Púrpura
GPIO9	GPIO	Púrpura
GPIO10	GPIO	Púrpura
GPIO11	GPIO	Púrpura
GPIO12	GPIO	Púrpura
GPIO13	GPIO	Púrpura
GPIO14	GPIO	Púrpura
GPIO15	GPIO	Púrpura
GPIO16	GPIO	Púrpura
GPIO17	GPIO	Púrpura
GPIO18	GPIO	Púrpura
GPIO19	GPIO	Púrpura
GPIO20	GPIO	Púrpura
GPIO21	GPIO	Púrpura
GPIO22	GPIO	Púrpura
GPIO23	GPIO	Púrpura
GPIO24	GPIO	Púrpura
GPIO25	GPIO	Púrpura
GPIO26	GPIO	Púrpura
GPIO27	GPIO	Púrpura
GPIO28	GPIO	Púrpura
GPIO29	GPIO	Púrpura
GPIO30	GPIO	Púrpura
GPIO31	GPIO	Púrpura
GPIO32	GPIO	Púrpura
GPIO33	GPIO	Púrpura
GPIO34	GPIO	Púrpura
GPIO35	GPIO	Púrpura
GPIO36	GPIO	Púrpura
GPIO37	GPIO	Púrpura
GPIO38	GPIO	Púrpura
GPIO39	GPIO	Púrpura
GPIO40	GPIO	Púrpura

Figura 1 Asignación de pines del conector GPIO.

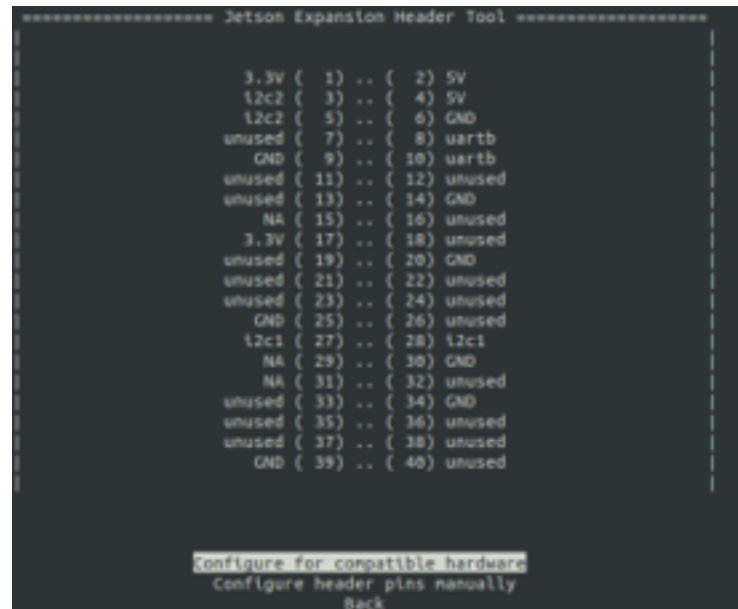


Figura 2 Configuración de pines GPIO (`/opt/nvidia/jetson-io/jetson-io.py`).

Los pines marcados como "unused", pueden ser usados como GPIO directamente, este script tiene la función de configurar las funciones secundarias de los pines cuando están disponibles, por ejemplo, habilitar los pines como PWM, o los pines para comunicación SPI o UART. Los pines más comúnmente utilizados son: 4, 17, 18, 22, 23, 24, 25, 27.

Para poder controlar en nuestros programas en Python los pines GPIO, deberemos cargar la biblioteca `Jetson.GPIO`, habilitar los permisos y reiniciar para que sean aplicados los cambios, como se muestra a continuación:

```

sudo pip3 install Jetson.GPIO
sudo groupadd -f -r gpio
sudo usermod -a -G gpio your-username
sudo reboot

```

Para utilizar un pin como salida digital se debe configurar como sigue:

```

import Jetson.GPIO as GPIO

pin = 18 # Usa el número de pin físico según el esquema de pines

GPIO.setmode(GPIO.BOARD) # O usa GPIO.BCM si prefieres la numeración BCM
GPIO.setup(pin, GPIO.OUT, initial=GPIO.LOW)

GPIO.output(pin, GPIO.HIGH) # Función para encender el pin
GPIO.cleanup()             # Limpiar la configuración al finalizar

```

3. Hardware y configuraciones

3.1 Buffers conversores de 3.3V a 5V para motores unipolares

Es importante considerar que los pines GPIO de la Jetson Nano operan a 3.3V. Para controlar dispositivos que operan a 5V, se requieren conversores de nivel lógico, estos aseguran que las señales de control sean compatibles con los controladores de motor a pasos de 5V. Para esto podemos auxiliarnos de un transistor [1] o de un conversor bidireccional como el circuito TXS0108E [2].

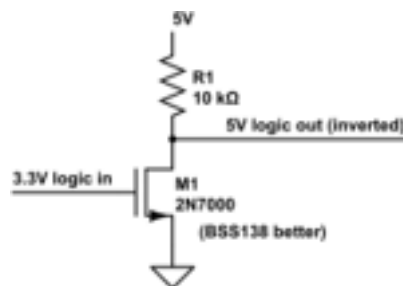


Figura 3 Transistor como conversor de nivel 3.3V a 5V [1].

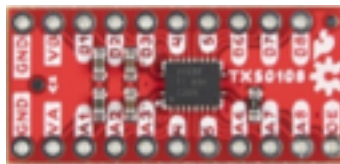


Figura 4 Conversor bidireccional TXS0108E [2].



Figura 5 Controlador ULN2003.

Aunque en la mayoría de los casos la lógica TTL de 3.3V alcanzará a activar el uno lógico en la lógica TTL de 5V [3].

3.2 Controlador ULN2003 para motores unipolares

El controlador ULN2003 es ampliamente utilizado para controlar motores a pasos unipolares. Este controlador es fácil de usar y económico. Es un controlador de alta tensión y corriente con una matriz de transistores Darlington que permite controlar hasta siete salidas. Este controlador proporcionará suficiente corriente para alimentar las bobinas del motor a pasos unipolar.

3.3 Controlador L298N para motores bipolares

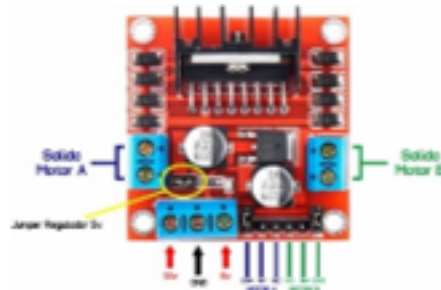


Figura 6 Controlador L298N [4].

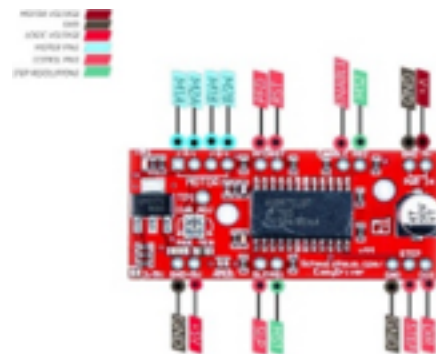


Figura 7 EasyDriver [5].

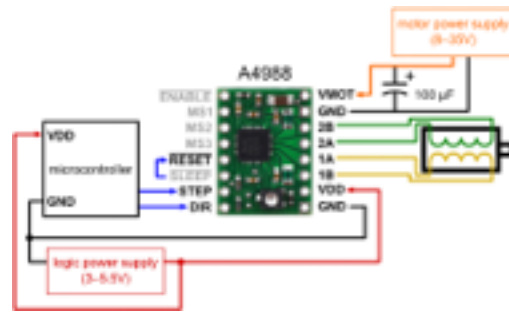


Figura 8 Controlador A4988 [6].

El controlador L298N es un módulo Puente-H que permite controlar motores a pasos bipolares y motores DC. Puede manejar cargas de alta corriente y tiene dos puentes H integrados, los cuales pueden manejar hasta dos motores de DC, o un motor a pasos bipolar. Es un módulo comercial bastante común y de un precio accesible, se recomienda tener especial cuidado en la programación para no mantener energizadas las bobinas del motor cuando no es necesario y optimizar el consumo de corriente para no sobrecalentarlo, ya que puede experimentar sobrecarga térmica muy fácilmente, hay que cuidar la correcta conexión de las bobinas del motor a pasos a las salidas del controlador también para evitar daños al circuito y las bobinas del motor a pasos.

3.4 Controlador EasyDriver, A4988 y DRV8825 para motores bipolares

El EasyDriver es un controlador de motor a pasos bipolar fácil de usar que puede controlar motores de hasta 30V y 750mA por fase. Proporciona una interfaz simple para el control de motores bipolares, con entrada de señal de pasos y dirección. Este tipo de controladores tienen la ventaja de tener protecciones térmicas y contra sobrecorrientes, tienen la opción de poder limitar la corriente que se suministra a las bobinas del motor, y puede seleccionarse el modo de trabajo en full, half step y las opciones de microstepping (1/4 y 1/8 step en el caso del EasyDriver, 1/16 step adicional para el controlador A4988 y 1/32 step adicional para el controlador DRV8825).

El controlador A4988 puede operar desde 8V hasta 35V y entregar aproximadamente 1A (ampere) por fase sin disipador de calor (puede llegar a los 2A con algún medio de disipación térmica), mientras que el DRV8825 puede operar desde 8.2V hasta 45V y entregar aproximadamente 1.5A por fase sin disipador (2.2A con disipación térmica). La limitación de corriente en el controlador A4988 se hace ajustando el potenciómetro hasta obtener el voltaje de referencia en el pin (sobre el potenciómetro) tomando en cuenta la siguiente expresión:

$$V_{REF} = 8 \cdot I_{MAX} \cdot R_{CS}$$

donde R_{CS} puede variar dependiendo la versión del controlador entre 0.05Ω a 0.10Ω .

Cuadro 1 Selección de resolución de paso EasyDriver.

MS1	MS2	Resolución
L	L	Full step
H	L	1/2 step
L	H	1/4 step
H	H	1/8 step

Cuadro 2 Selección de resolución de paso A4988.

MS1	MS2	MS3	Resolución
L	L	L	Full step
H	L	L	1/2 step
L	H	L	1/4 step
H	H	L	1/8 step
H	H	L	1/16 step

3.5 Arduino nano como controlador independiente

El uso de un Arduino Nano como controlador intermedio ofrece varias ventajas, incluyendo la independización del control de motores, el aislamiento eléctrico de la Jetson Nano con la etapa de potencia, facilidad de programación y flexibilidad. Esto permite que la Jetson Nano se concentre en tareas más complejas mientras el Arduino gestiona el control del motor.

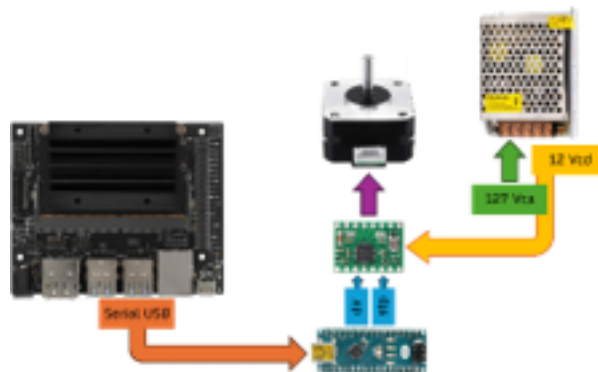


Figura 9 Configuración de Arduino como controlador independiente.

4. Resultados y conclusiones

En general cuando se requiera adaptar los niveles lógicos de las salidas de la Jetson Nano es conveniente el empleo de los buffers conversores, cuando se exceda la corriente requerida por nuestra aplicación, como es el caso de cuando queremos controlar motores a pasos, es más conveniente emplear algún controlador.

En el caso de los motores a pasos unipolares de poca potencia, pueden ser manejados directamente por el controlador ULN2003. Si requerimos un poco más de potencia, podemos emplear el L298N, que puede manejar corrientes de hasta 2 A, siempre y cuando nuestra aplicación pueda ser gestionada para ahorrar energía no mantener encendidas las bobinas constantemente para evitar sobrecalentamientos. Cuando requerimos mayor potencia y una mejor gestión de corriente y potencia, es más confiable el empleo de los controladores como el EasyDriver, el A4988 y el DRV8825, que nos ofrecen un mejor control del motor, mayor precisión en el movimiento usando los modos microstepping, y una mejor gestión de la potencia, velocidad y aceleración empleando las librerías "AccelStepper.h", por ejemplo, en Arduino.

Referencias

- [1] Arduino Forum, (s.f.). *How to control Stepper Driver Puls 5v and Dir 5v with 3.3V from newer controller*. [En línea]. Disponible en: <https://forum.arduino.cc/t/how-to-control-stepper-driver-puls-5v-and-dir-5v-with-3-3v-from>

477300/2

- [2] sparkfun, (s.f.). *Level Shifter - 8 Channel (TXS0108E) Hookup Guide*. [En línea]. Disponible en: <https://learn.sparkfun.com/tutorials/level-shifter---8-channel-txs0108e-hookup-guide/all#troubleshooting>. [Último acceso: Junio 2024].
- [3] sparkfun, (s.f.). *Logic Levels*. [En línea]. Disponible en: <https://learn.sparkfun.com/tutorials/logic-levels>. [Último acceso: Junio 2024].
- [4] Proyectos con Arduino, (s.f.). *Módulo Controlador de Motores L298N*. [En línea]. Disponible en: https://proyectosconarduino.com/modulos/motor-driver-1298n/?expand_article=1. [Último acceso: Junio 2024].
- [5] Electropeak, (s.f.). *Interfacing EasyDreiver - Stepper Motor Driver with Arduino*. [En línea]. Disponible en: <https://electropeak.com/learn/interfacing-a3967-easydriver-stepper-motor-driver-with-arduino/>. [Último acceso: Junio 2024].
- [6] Hardware Libre, (s.f.). *ULN2003: driver para motores eléctricos*. [En línea]. Disponible en: <https://www.hwlibre.com/uln2003/>. [Último acceso: Junio 2024].
- [7] Pololu, (s.f.). *A4988 Stepper Motor Driver Carrier*. [En línea]. Disponible en: <https://www.pololu.com/product/1182>. [Último acceso: Junio 2024].

Tovar Ortiz, L. A., Cruz Baños, A. F., Hernández Diaz, F. I. A., Herrera Lozada, J. C., Sandoval Gutiérrez, J. (2026). UTILIZACIÓN DEL PUERTO GPIO DE LA JETSON NANO PARA EL CONTROL DE MOTORES A PASOS: MÉTODOS Y APLICACIONES. *Boletín UPIITA*. año XX, (NÚM) 2026.