



UNIVERSIDAD TECNICA  
FEDERICO SANTA MARIA



DEPARTAMENTO DE  
ELECTRONICA

# Documento de Diseño de < Control PID de un Motor DC >

**Fecha** [06/10/2023]  
**Versión** [1]

**Integrantes  
del grupo** Nicolás Mansilla Mansilla  
Ítalo Muñoz Lira  
Gustavo Fernández  
Joel Rivera Fernández

**Auspiciador** Álvaro Aguirre Mattig

## Historia del documento

Versión	Fecha	Explicación del cambio	Autor
1.0	10/09/2023	Primer borrador	Grupo 3

# Índice de Materias

Historia del documento	II
Lista de Figuras	IV
Lista de Tablas	IV
1. Introducción	V
2. Funcionalidades identificadas	V
3. Elementos identificados	V
4. Diseño de hardware propuesto	VI
4.1. Diagrama de bloques	VI
4.2. Detalle por bloque	VI
4.2.1. Motor DC	VI
4.2.2. Encoder	VI
4.2.3. Microcontrolador STM32F	VI
4.2.4. Driver Motor	VI
4.2.5. Fuente 12 [V]	VII
4.2.6. Potenciómetro	VII
4.2.7. Conector Phoenix	VII
4.2.8. Conector Pin Socket	VII
4.2.9. Conector Molex	VII
5. Diseño de firmware propuesto	VII
5.1. Detalle de drivers	VII
5.1.1. Driver de GPIO	VII
5.1.2. Driver de ADC	VII
5.1.3. Driver de Temporizador	VII
5.1.4. Driver de PWM	VII
5.1.5. Driver de UART	VII
5.2. Diagrama de flujo	VIII
5.3. Detalle por módulo	IX
5.3.1. Módulo 5V	IX
5.3.2. Módulo ADC	IX
5.3.3. Módulo Encoder	IX
5.3.4. Módulo PWM	IX
5.3.5. Módulo Control	IX
5.3.6. Módulo Transmisión	IX
6. Pruebas	X
6.1. Prueba sobre elementos	X
6.1.1. Prueba Driver	X
6.1.2. Prueba Potenciómetro	X
6.1.3. Prueba de PCB	X
6.1.4. Prueba Encoder	X
6.2. Prueba sobre funcionalidades	X
6.2.1. Prueba Referencia con potenciómetro	X
6.2.2. Prueba envío de datos a consola	X
6.2.3. Prueba Creación PWM	X
6.2.4. Prueba control Motor lazo abierto	XI
6.2.5. Prueba control Motor lazo cerrado	XI

<b>7. Cronograma</b>	<b>XI</b>
<b>8. Costos</b>	<b>XII</b>

## Lista de Figuras

1. Diagrama de bloques . . . . .	VI
2. Diagrama de flujo firmware . . . . .	VIII
3. Diagrama módulos firmware . . . . .	IX

## Lista de Tablas

1. Cronograma . . . . .	XI
2. Costos Unitarios . . . . .	XII
3. Costos Totales . . . . .	XII

## 1. Introducción

Este proyecto se enfoca en el diseño y desarrollo de un sistema de control PID para un Motor DC. El sistema consta de un motor DC y un shield específicamente diseñado para ser utilizado con una tarjeta de desarrollo STM32 Nucleo-F401. El objetivo principal es permitir un control preciso y eficiente de la posición angular del motor, utilizando un encoder para capturar su posición en tiempo real.

En el presente documento se detallan los componentes del sistema, tanto sus elementos físicos como los módulos de firmware que lo integran, las pruebas necesarias para comprobar el correcto funcionamiento, la planeación del proyecto y los costos asociados a la fabricación del equipo.

## 2. Funcionalidades identificadas

El sistema se compone de un motor DC y un shield para una tarjeta de desarrollo stm32 Nucleo-F401. El shield debe operar de una manera plug and play controlando la posición angular del motor, captada por un encoder, en base a la referencia generada por la posición de un potenciómetro ajustable manualmente. Paralelamente se imprime por consola la posición del motor cada cierto tiempo por conexión serial.

Por otro lado, se requiere que el equipo se encuentre protegido contra excesos de corriente y/o voltaje.

## 3. Elementos identificados

- Motor DC: Globe Motors 415A832
- Printed Circuit Board(PCB)
- STM32F401 nucleo
- Encoder: HEDS-5505#A04
- Driver: DRV8220DRL
- Potenciómetro: PTV09A-4220F-B103
- Pin socket 2.54mm: SSQ-119-01-T-S
- Conector Molex macho: 22-23-2021
- Conector Molex hembra: 22-01-3057
- Phoenix Terminal Block (5.08mm en pitch): 5452257
- Componentes pasivos (resistencias, inductores, capacitores, diodos)

Se debe destacar que se utiliza un módulo que integra el motor y el encoder en una pieza, que lleva como part number Globe Motors 125052. Estos, junto con el STM32F401, serán entregados por el cliente, por lo que no se considerarán a la hora de comprar componentes.

## 4. Diseño de hardware propuesto

### 4.1. Diagrama de bloques

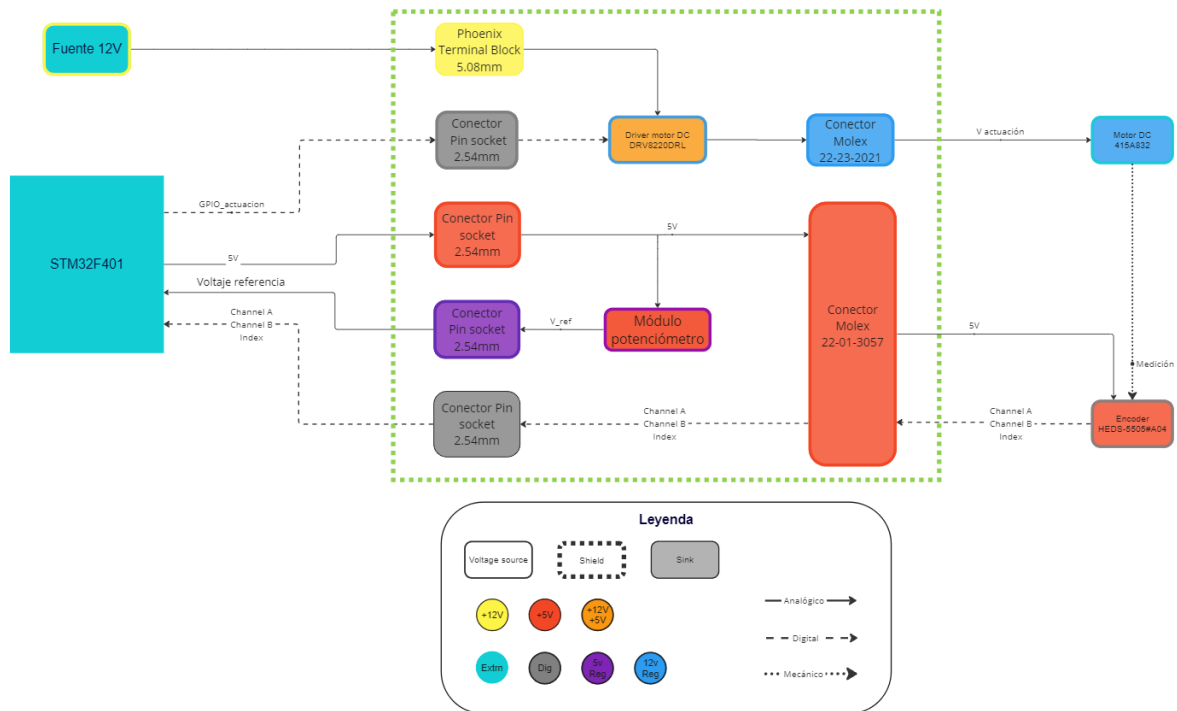


Figura 1: Diagrama de bloques

### 4.2. Detalle por bloque

A continuación se detalla las características de cada bloque del diagrama.

#### 4.2.1. Motor DC

Motor DC de marca Globe Motors el cual es conectado al driver para generar su giro. Este motor es alimentado desde el driver para hacerlo girar en sentido reloj o anti-reloj.

#### 4.2.2. Encoder

Elemento capaz de traducir la rotación del motor a señales eléctricas que pueden ser leídas por el microcontrolador. El encoder va acoplado al motor de forma que componen 1 unidad. Se conecta a la alimentación proveniente del microcontrolador.

#### 4.2.3. Microcontrolador STM32F

Elemento que realiza la toma de datos proveniente del encoder y que envía la actuación hacia el driver. Además provee de 5 [V] al shield para los elementos que lo necesiten. Y envía la posición medida por el encoder por USB.

#### 4.2.4. Driver Motor

Chip que permite la rotación del motor dado una entrada de voltaje que define si se gira en sentido reloj o anti-reloj. Este elemento es alimentado con los 5 [V] provenientes del microcontrolador.

#### 4.2.5. Fuente 12 [V]

Fuente de voltaje de 12 [V] externa para alimentar el driver del motor. Esta fuente corresponde a una fuente de laboratorio.

#### 4.2.6. Potenciómetro

Potenciómetro lineal de 10[k $\Omega$ ] que se utiliza para generar un potencial eléctrico a partir de la posición rotacional del eje. Posee un rango de rotación de  $280 \pm 10$  deg.

#### 4.2.7. Conector Phoenix

Conector que permite la conexión de cables a la PCB. Consta de un bloque de plástico, pines de montura y un receptáculo que se atornilla para fijar el cable a los terminales. El bloque cuenta con 2 terminales para proveer voltaje y tierra eléctrica.

#### 4.2.8. Conector Pin Socket

Corresponde a un conector compatible con los zócalos disponibles en la tarjeta de desarrollo STM32F. Permiten la conexión con los pines internos del microcontrolador, permitiendo comunicar el procesador con los elementos externos, como el motor y el encoder.

#### 4.2.9. Conector Molex

Este elemento conecta los terminales de alimentación y tierra provenientes del motor DC a la placa. Esto permite que el encoder pueda funcionar.

### 5. Diseño de firmware propuesto

Dado que el firmware de este proyecto solo corresponde a un

#### 5.1. Detalle de drivers

##### 5.1.1. Driver de GPIO

Se utiliza el controlador de GPIO HAL (HAL\_GPIO) proporcionado por STMicroelectronics.

##### 5.1.2. Driver de ADC

Se utiliza el controlador de ADC HAL (HAL\_ADC) proporcionado por STMicroelectronics.

##### 5.1.3. Driver de Temporizador

Se utiliza el controlador de temporizador HAL (HAL\_TIM) proporcionado por STMicroelectronics.

##### 5.1.4. Driver de PWM

Para la generación de señales PWM, se utiliza HAL\_TIM (parte de la biblioteca de temporizador).

##### 5.1.5. Driver de UART

Se utiliza el controlador de UART HAL (HAL\_UART) proporcionado por STMicroelectronics.

## 5.2. Diagrama de flujo

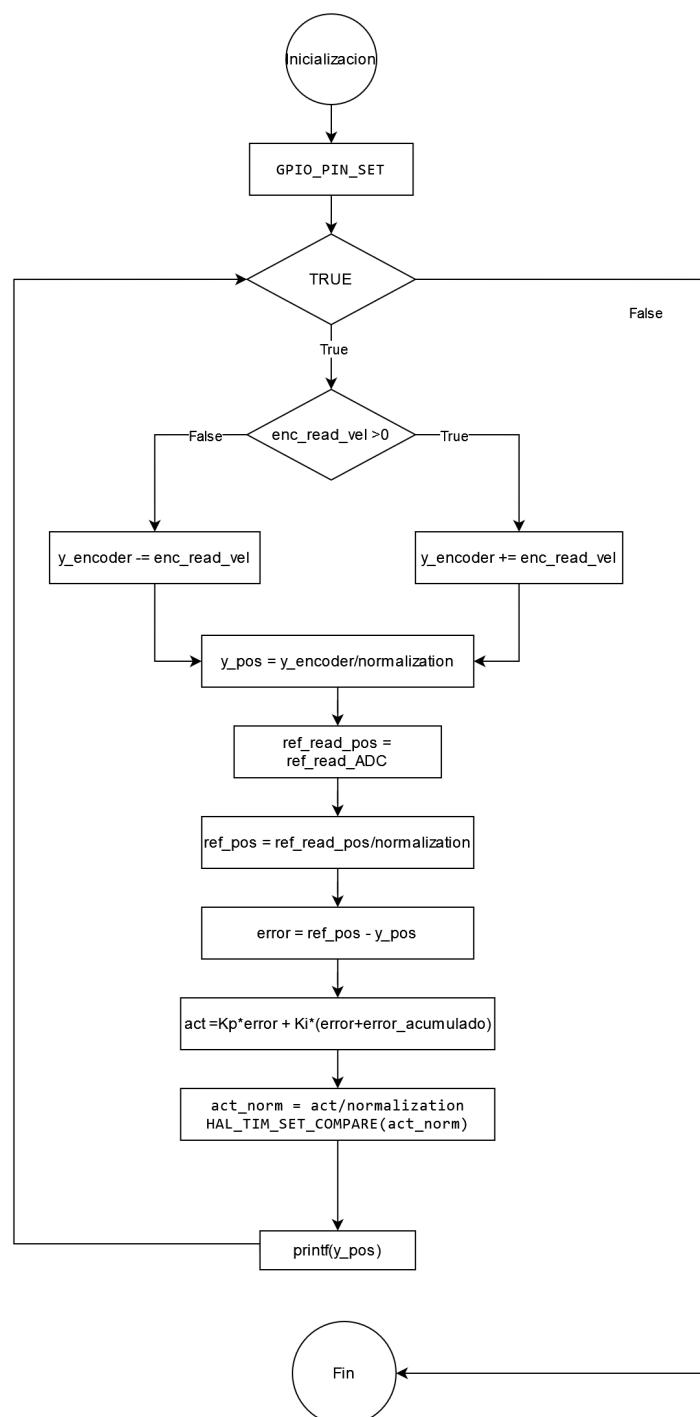


Figura 2: Diagrama de flujo firmware



### 5.3. Detalle por módulo

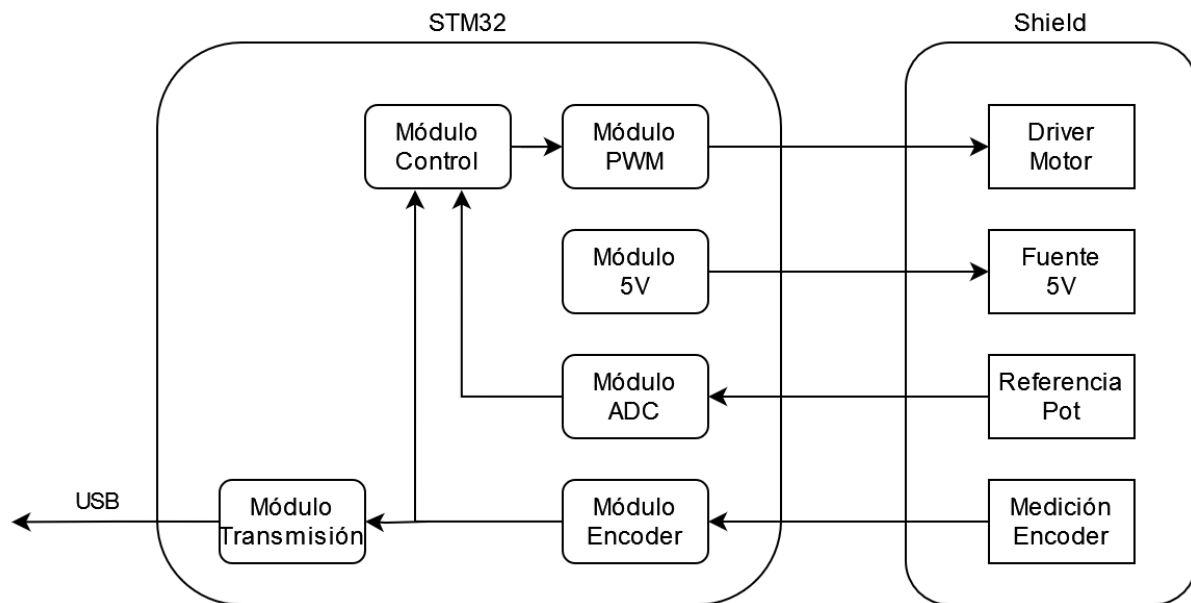


Figura 3: Diagrama módulos firmware

#### 5.3.1. Módulo 5V

Este módulo se encarga de proporcionar una fuente de voltaje constante, generando 5V, que permita a todas las componentes del shield funcionar correctamente.

#### 5.3.2. Módulo ADC

Este módulo se encarga de recibir la señal analógica correspondiente al voltaje de referencia para el control, primero convirtiéndolo a un número digital, para luego generar una equivalencia en grados del potenciómetro que pueda ser comparable con la medición proveniente del módulo encoder, obteniéndose el error en el módulo de control.

#### 5.3.3. Módulo Encoder

Este módulo se encarga de recibir la señal proveniente del canal A y B del encoder, determinando un número correspondiente a la posición angular del motor, que pueda ser comparada con la referencia del potenciómetro en el módulo de control.

#### 5.3.4. Módulo PWM

Este módulo se encarga de convertir la señal de actuación proveniente del controlador, en una señal PWM equivalente que pueda ser recibida por el driver del motor.

#### 5.3.5. Módulo Control

Este módulo se encarga de comparar la referencia de posición angular dada por el potenciómetro, con la medición de posición angular del motor, para luego sumar la integral del error con un factor proporcional, que dan lugar a una señal de actuación que posteriormente es manejada por el módulo PWM.

#### 5.3.6. Módulo Transmisión

Este módulo se encarga de enviar por USB con protocolo UART la medición de posición angular del motor determinada por el módulo encoder.

## **6. Pruebas**

### **6.1. Prueba sobre elementos**

#### **6.1.1. Prueba Driver**

- Se energiza el driver con 12[V]
- Se inyecta una señal pwm de 5[V] a la entrada 1, se conecta la segunda entrada a tierra y se observa la salida.
- Se inyecta una señal pwm de 5[V] a la entrada 2, se conecta la primera entrada a tierra y se observa la salida.
- Se inyectan dos señales pwm alternando la señal de giro del motor y se observa la salida.

#### **6.1.2. Prueba Potenciómetro**

- Se mide la resistencia exacta del potenciómetro
- Se mide el rango completo de giro del potenciómetro

#### **6.1.3. Prueba de PCB**

- Se realizan las pruebas de continuidad de todas las conexiones de la PCB

#### **6.1.4. Prueba Encoder**

- Se energiza el encoder
- Se girar el motor manualmente en ambos sentidos
- Se leen las señales de salida del encoder

### **6.2. Prueba sobre funcionalidades**

#### **6.2.1. Prueba Referencia con potenciómetro**

- Se energiza la placa de desarrollo stm32
- Se carga un script para leer el voltaje del potenciómetro
- Se conecta el potenciómetro a la stm32 y se varía el voltaje imprimiendo el resultado por consola.

#### **6.2.2. Prueba envío de datos a consola**

- Se energiza la placa de desarrollo stm32
- Se carga un script para leer los datos del encoder
- Se ajusta manualmente el ángulo del rotor y se imprimen los datos por consola.

#### **6.2.3. Prueba Creación PWM**

- Se energiza la placa de desarrollo stm32
- Se carga un script para generar un pwm de ciclo de trabajo variable controlado por una constante.
- Se mide la salida de la pwm a distintos ciclos de trabajo

#### 6.2.4. Prueba control Motor lazo abierto

- Se conecta el motor al terminal block
- Se conecta la alimentación al terminal block para el driver
- Se conecta la stm32 al driver
- Se energizan los ambos sistemas
- Se genera una señal pwm desde la stm32 hacia el driver
- Se observa el giro del motor

#### 6.2.5. Prueba control Motor lazo cerrado

- Se conecta el motor al terminal block
- Se conecta la alimentación al terminal block para el driver
- Se conecta la stm32 al driver
- Se conecta el potenciómetro a la stm32
- Se carga el script de control
- Se energizan los sistemas
- Se inicializa el control
- Se gira la perilla del potenciómetro
- Se miden las señales de entrada al motor
- Se observa el cambio en el ángulo del rotor

## 7. Cronograma

	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5
Diseño Industrial	x				
Diseño esquemático	x	x			
Diseño PCB		x	x		
Compra de componentes			x		
Compra y envío de PCB				x	x
Integración					x
Desarrollo firmware			x		
Pruebas					x

Tabla 1: Cronograma

---

Semana 1: 16 Oct - 22 Oct  
Semana 2: 23 Oct - 29 Oct  
Semana 3: 30 Oct - 5 Nov  
Semana 4: 6 Nov - 12 Nov  
Semana 5: 13 Nov - 19 Nov

---

## 8. Costos

	Descripción	Costo unitario en dolares	
		Por 1 unidad	Por 100 unidades
Envío Componentes	Desde Mouser	28.08	-
PCB		1	
Ensamblado	In-House Assembly	-	-
Pruebas	In-House testing	-	-
Envío PCB	Se envían 5 PCB	48.48	-
Resistor 3,09k $\Omega$	PN: RK73H1ETTP3091F	0.12	0.0082
Conector Molex macho	PN: 22-23-2021	0.25	0.11
Potenciómetro	PN: PTV09A-4220F-B103	1.11	0.79
Conector Molex hembra	PN: 22-01-3057	0.31	0.18
Phoenix Terminal Block 5.08mm	PN: 5452257	0.58	0.53
Driver	PN: DRV8220DRL	1.04	0.68
Condensador 100nF	PN: C0603C104K1RACAUTO	0.32	0.123
Pin Socket	PN: SSQ-119-01-T-S	2.70	1.80

Tabla 2: Costos Unitarios

	Cantidad	Costo
Envío Componentes	1	28.08
PCB	5	5
Envío PCB	5	48.48
Resistor 3,09k $\Omega$	2	0.24
Conector Molex macho	1	0.25
Potenciómetro	1	1.11
Conector Molex hembra	1	0.31
Phoenix Terminal Block 5.08mm	1	0.58
Driver	1	1.04
Condensador 100nF	1	0.32
Pin Socket	2	5.4
<b>Total</b>	-	90.81

Tabla 3: Costos Totales