



# **PDN609 Stepper Motor Controller Data Sheet**

*Serial Interface Stepper Motor Sequencer IC*



# Functionality & Feature Set

---

- ***One Wire Interface***
- ***Step Interval Adjustable Down to 417 $\mu$ s***
- ***Clockwise and Counter-Clockwise Directional Control***
- ***Single Step, Counted Steps, or Stop-On-Home Modes***
- ***Count Function Automatically Incorporates Ramping***
- ***Home Input Automatically Stops Motor***
- ***Dual Coil Drive Modes: Auto Shut Off or Constant Drive***
- ***Automatic Increase/Decrease Ramping Functions***
- ***Wavedrive, Twophase, and Halfstep Stepping Sequences***
- ***Interfaces to any Microcontroller or Microprocessor***
- ***Companion Serial Router IC Allows Multi-Axis Control***
- ***Available in 8 pin DIP or SOIC packages***

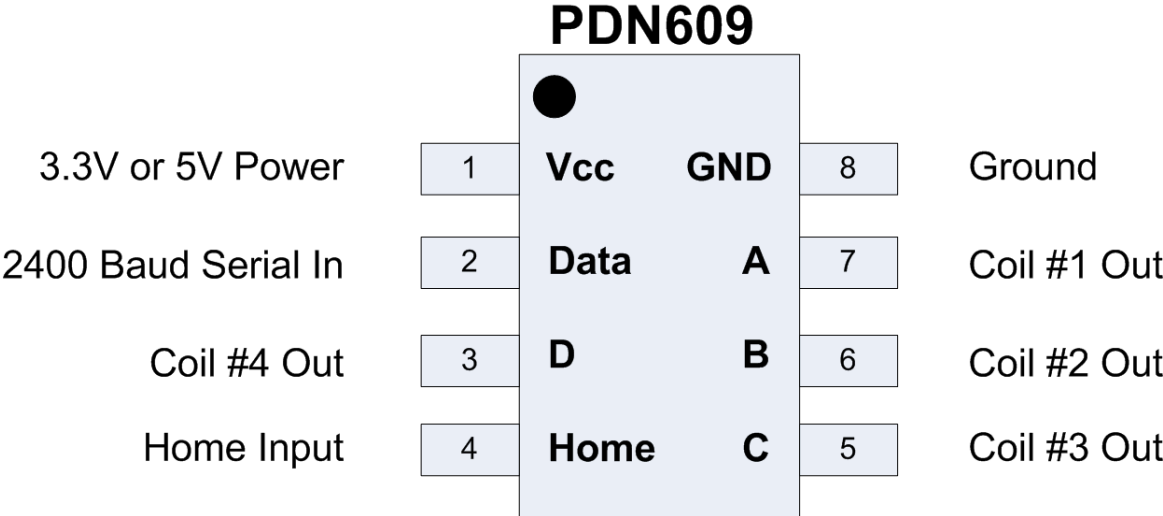
The PDN609 provides convenient control of unipolar and bi-polar stepper motors. Stepper motors provide precise positioning and control due to their unique geometry. The PDN609 offers full control over any stepper motor using only one serial output pin from your host microcontroller.

The PDN609 provides all logic control for stepper motor coil sequencing. The PDN609 is designed to work in conjunction with a drive IC or circuit capable of handling the current and voltage requirements of your particular motor. Several connection examples are offered in this datasheet illustrating connection of unipolar (five or six wire) or bi-polar (four wire) motors.

The PDN609 is designed to receive serial data commands from the host microcontroller. Motion control and mode of operation is determined by values loaded into various control registers. These registers determine basic motor operation such as clockwise or counter-clockwise rotation, full or half-stepping, as well as more advanced motor control in areas such as ramp up/ramp down and preloaded step count operation.

# Pin Definitions

---



**Control & Data Pins:**

- A (Pin 7).....First Coil Signal Output
- B (Pin 6).....Second Coil Signal Output
- C (Pin 5).....Third Coil Signal Output
- D (Pin 3).....Fourth Coil Signal Output
- Home (Pin 4).....Home Function Input
- Control (Pin 2).....Serial Data Input

**Power Pins:**

- V++ (Pin 1).....Connect to +3.3V or +5V DC
- GND (Pin 8).....Connect to 0V DC (Ground)

# Operational Overview

The PDN609 can operate in several modes depending upon the type of control required. The simplest, Single Step mode, steps once in the preset direction for each step command received. Alternately, Continuous Stepping mode steps the motor at a preset step interval until a stop command or home input is activated.

For more advanced applications, Count mode may be utilized. For this, a predetermined number of steps is loaded into the count register and the Go command issued. The motor will step at a predetermined step interval as specified by the StoreStepTime register. If ramping is active, the PDN609 will accelerate on motion start and decelerate on motion stop assuming that there are enough steps requested to fulfill the ramp parameters specified by the ramping registers.

To begin with, the user must first select WaveDrive, TwoPhase, or HalfStep stepping. This selection determines the manner in which the stepper coils are driven. WaveDrive, the simplest drive method, activates only a single coil at a time. This results in reduced power consumption at the expense of motor torque. TwoPhase, the most commonly used drive method, activates two coils simultaneously in a sequence that maximizes motor torque. In both of these modes the motor steps one arc angle ( $1.8^\circ$  for a  $1.8^\circ$  per step motor). The third mode, HalfStep, moves the motor one-half of the specified arc angle ( $0.9^\circ$  for a  $1.8^\circ$  per step motor). The tradeoff for this increased accuracy is reduced torque and, consequently, reduced maximum speed. The following tables illustrate the coil energizing sequences for WaveDrive, TwoPhase, and HalfStep modes.

	Coil A	Coil B	Coil C	Coil D
1	On	Off	Off	Off
2	Off	On	Off	Off
3	Off	Off	On	Off
4	Off	Off	Off	On

**Table One: WaveDrive Stepping Mode Coil Energization Sequence**

	Coil A	Coil B	Coil C	Coil D
1	On	On	Off	Off
2	Off	On	On	Off
3	Off	Off	On	On
4	On	Off	Off	On

**Table Two: TwoPhase Stepping Mode Coil Energization Sequence**

	Coil A	Coil B	Coil C	Coil D
1	On	Off	Off	Off
2	On	On	Off	Off
3	Off	On	Off	Off
4	Off	On	On	Off
5	Off	Off	On	Off
6	Off	Off	On	On
7	Off	Off	Off	On
8	On	Off	Off	On

**Table Three: HalfStep Stepping Mode Coil Energization Sequence**

Following stepping mode selection, the user must select whether or not the coils remain energized while the motor is not stepping. The KeepEnergized command will cause the most recently energized coils to remain active when the motor is not turning. This will hold the motor shaft in a locked position at the expense of added current draw and heat generation. The DeEnergize command will release all coils while the motor is not being driven. This reduces current draw, however, the motor shaft can free spin during this time. If DeEnergize is used, the mechanical assembly connected to the motor must be designed such that it cannot continue to rotate due to inertia, otherwise the shaft will be out of place and, consequently, future step commands may cause jerking on the first step as the motor shaft is realigned.

Direction of rotation must also be set prior to a motion command being sent. This is as simple as sending either the clockwise (CW) or counter-clockwise (CCW) command. The motor will rotate in the chosen direction until an alternate direction command is sent.

**Note: If using a 2544 driver chip in conjunction with a PDN609, you must first send the *USE2544* command in order to invert the B and C pins for correct operation.**

# Register Loading

The PDN609 receives 1 byte commands over a 2400 baud serial line connection. All of the commands accepted by the PDN609 are 1 byte in length. Commands that use a value, (such as the LoadCount, StoreStepTime, StoreRampInterval, etc.) use the Command Register to retrieve their values. Therefore, the Command Register must be loaded before these commands can be given.

The Command Register is a two-byte register whose value is broken into four 4-bit nibbles. The Command Register must be loaded one nibble at a time using the following four commands:

- LoadNibble1           (lowest 4 bits)
- LoadNibble2
- LoadNibble3
- LoadNibble4           (highest 4 bits)

xxxx	xxxx	xxxx	xxxx
4 <sup>th</sup> Nibble	3 <sup>rd</sup> Nibble	2 <sup>nd</sup> Nibble	1 <sup>st</sup> Nibble

**Table Four: Command Register Structure**

**Note:** The LoadNibble1 command clears nibbles 2 through 4 when loaded with a value. You must send the LoadNibble1 command before sending the others or *all* nibble values will be cleared.

The format for the four load Command Register commands are as follows:

Command	Value Loaded	Result
LoadNibble1	1000xxxx	Loads xxxx into nibble 1 of Command Register
LoadNibble2	1001xxxx	Loads xxxx into nibble 2 of Command Register
LoadNibble3	1010xxxx	Loads xxxx into nibble 3 of Command Register
LoadNibble4	1011xxxx	Loads xxxx into nibble 4 of Command Register

**Table Five: Loading The Command Register**

# Modes Of Operation

---

The PDN609 may be operates in several modes such at single stepping, continuous run, or variable run mode. These modes are described below:

## Single Stepping Mode

Single Stepping Mode is the simplest stepping mode available. It caused the motor to rotate one step in the preselected rotational direction. It is configured as follows:

1. Set the stepping sequence by sending the *Wavedrive*, *TwoPhase*, or *Halfstep* command.
2. Set the direction by sending the *CW* (clockwise) or *CCW* (counter clockwise) command.
3. Send the *StepOne* command to move the stepper motor one step.

## Continuous Stepping Mode

Continuous stepping mode causes the motor to rotate indefinitely at a preset speed and rotational direction. It is configured as follows:

1. Set the stepping sequence by sending the *Wavedrive*, *TwoPhase*, or *Halfstep* command.
2. Set the direction by sending the *CW* (clockwise) or *CCW* (counter clockwise) command.
3. Set the Stepping time by loading the Command Register using the *LoadNibble* commands. The stepping time is equal to the value loaded into the Command Register multiplied by .417 milliseconds.
4. Send the *StoreStepTime* command.
5. Start the stepper motor by sending the *GO* command.

## Variable Step Count Mode

Variable Step mode rotates the motor a specified number of steps in the preselected rotational direction.

1. Set the stepping sequence by sending the *Wavedrive*, *TwoPhase*, or *Halfstep* command.
2. Set the direction by sending the *CW* (clockwise) or *CCW* (counter clockwise) command.
3. Set the Stepping time by loading the Command Register using the *LoadNibble* commands. The stepping time is equal to the value loaded into the Command Register multiplied by .417 milliseconds.

4. Send the *StoreStepTime* command.
5. Set the number of steps by sending a *LoadCount* command.
6. Set the PDN609 to Count mode by sending the *CountOn* command.
7. Start the stepper motor by sending the *GO* command.

#### Home Function Operation

**Note:** *The Home function can be used in any operation mode.*

1. There are two ways to use the home function. The first is by sending the *HomeOn* command which will cause the motor to stop stepping when the Home pin is a logic high.
2. The second way to use the Home Function is by loading the number of steps desired *after* the Home pin goes high by using the *LoadCount* command and then sending the *HomeCountOn* command. If ramping functions are active it will ramp down if there are a sufficient number of steps loaded into *LoadCount* to allow for a proper ramp down.

### Stepping With Ramping Options

1. The four special ramping commands are *RampOn*, *StoreRampStartTime*, *StoreRampEndTime*, *StoreRampInterval*, and *StoreStepTime*. All these command take the value stored in the Command Register and transfers it to the appropriate variable. The time use for these variables is .417ms multiplied by the value in the Command Register.
2. *StoreRampStartTime* (1-65535) is the stepper motor speed. More concisely, it is the interval at which the stepper motor will step.
3. *StoreRampEndTime* (1-65535) is the interval at which the stepper motor will stop decreasing the stepping time. The stepper motor will continue to ramp until this value is reached. *StoreRampStartTime* must be greater than *StoreRampEndTime*.
4. *StoreRampInterval* (1-255) is the number of times the stepper motor will step before it adds/subtracts the *StoreRampStartTime* to the stepping time of the motor.
5. *StoreRampStepTime* (1-255) is the amount of time that will be added/subtracted to the step time while ramping.
6. The ramping function can be used with any stepping mode. Once the ramping variables are set and the *RampOn* command is given it will stay in ramping mode until the *RampOff* command is sent.
7. The ramping function will automatically ramp up and down when the stepper motor is only stepping for a set amount of time. When in continuous run mode the stepper will automatically ramp up but will not ramp down until the *LoadCount* and *CountOn* commands are sent. If the *Stop* command is sent no ramping will occur.
8. Once the ramping variables are set there is no need to reset them until after a power cycling has occurred.



# Command Set

The table below represent the entire command set of the PDN609 IC. Because the PDN609 has several modes of operation, they are divided into categories for clarity. The command codes are written in decimal with Hex beside them, with the exception of the command register loading which is written in Binary.

Binary	Name	Function
	<b>Special Commands</b>	
1000XXXX	LoadNibble1	Clear Nibbles 2-4 and load XXXX value into Nibble 1
1001XXXX	LoadNibble2	Load XXXX value into Nibble 2
1010XXXX	LoadNibble3	Load XXXX value into Nibble 3
1011XXXX	LoadNibble4	Load XXXX value into Nibble 4

**Table Six: Loading The Command Register**

Decimal	Name	Function
	<b>General Commands</b>	
193 (0xC1)	WaveDrive	Sets mode to Wave Drive
194 (0xC2)	TwoPhase	Sets mode to Two Phase
195 (0xC3)	HalfStep	Sets mode to Half Step
197 (0xC5)	CW	Sets motor direction to clockwise
196 (0xC4)	CCW	Sets motor direction to count-clockwise
218 (0xDA)	KeepEnergized	Keep motor energized when not stepping; Locks Motor
219 (0xDB)	DeEnergize	De-energizes coils when motor is not running; Unlocks
207 (0xCF)	StoreStepTime	Stores time between steps
206 (0xCE)	Use2544	Allows for use with 2544 Driver IC; Inverts out2 & out3

**Table Seven: General Commands**

Decimal	Name	Function
	<b>Counter Register Control</b>	
204 (0xCC)	CountOn	Steps the motor a specific # of steps when active
215 (0xD7)	CountOff	Turns off Count Mode
205 (0xCB)	LoadCount	Loads # of steps to travel when Count is on; Two bytes

**Table Eight: Counter-Specific Registers**

Decimal	Name	Function
	<b><i>Ramping Control</i></b>	
210 (0xD2)	StoreRampStartTime	Stores ramp start time; One byte
222 (0xDE)	StoreRampEndTime	Stores ramp end time
208 (0xD0)	StoreRampInterval	Stores ramp interval; One byte
209 (0xD1)	StoreRampStepTime	Stores ramp step time; One byte
213 (0xD5)	RampOn	Turns on Ramping mode
217 (0xD9)	RampOff	Turns off Ramping Mode

***Table Nine: Ramping-Specific Registers***

Decimal	Name	Function
	<b><i>Home Position Control</i></b>	
200 (0xC8)	HomeOn	Stops motor on home when active
214 (0xD6)	HomeOff	Turns home mode off
203 (0xCB)	HomeCountOn	Steps motor # steps after home is turned on

***Table Ten: Home-Specific Registers***

Decimal	Name	Function
	<b><i>Motion Control Commands</i></b>	
192 (0xC0)	Go	Starts motor in selected mode
198 (0xC6)	Stop	Stops motor; Does not clear mode
223 (0xDF)	Continue	Clear Stop command and restarts motor
199 (0xC7)	StepOne	Steps motor one step in selected direction

***Table Eleven: Motion Control Registers***

# Code Example

The following program, written for a BS2 microcontroller from Parallax Inc. ([www.parallax.com](http://www.parallax.com)), illustrates advanced setup of the PDN609 IC. This program causes the stepper motor to ramp up to maximum speed, sustain, and ramp down to a stop all in 500 steps.

```
{ $STAMP BS2 }
{ $PBASIC 2.5 }
{ $PORT COM1 }

-----
' Command Register Declarations for the PDN609
-----

' Loading the Command Register
LOADNIBBLE1      CON 128
LOADNIBBLE2      CON 144
LOADNIBBLE3      CON 160
LOADNIBBLE4      CON 176

' General Commands
WAVEDRIVE        CON 193
TWOPHASE         CON 194
HALFSTEP         CON 195
CW               CON 197
CCW              CON 196
KEEPENERGIZED   CON 218
DEENERGIZE       CON 219
STORESTEPTIME   CON 207
USE2544          CON 206

' Counter-Specific Registers
COUNTON         CON 204
COUNTOFF        CON 215
LOADCOUNT       CON 205

' Ramping-Specific Registers
STORERAMPSTARTTIME CON 210
STORERAMPENDTIME  CON 222
STORERAMPINTERVAL CON 208
STORERAMPSTEPTIME CON 209
RAMPON           CON 213
RAMPOFF          CON 217

' Home-Specific Registers
HOMEON           CON 200
HOMEOFF          CON 214
HOMECOUNTON     CON 203

' Motion Control Registers
GO               CON 192
STOPNOW         CON 198
CONTINUE        CON 223
STEPONE        CON 199

-----
' Variable Declaration
-----
Value           VAR Word
Command         VAR Byte
-----
```

```

'Serial Output Pin
'-----
PDN609PIN CON 0

'-----
'PDN609 Setup and Control
'-----
' Drive two phases at a time for increased torque
Command = TWOPHASE: GOSUB SendCommand

' Leave coils energized between steps
Command = KEEPENERGIZED: GOSUB SendCommand

' Select CW (clockwise) rotation
Command = CW: GOSUB SendCommand

' Set period of pulses at start of ramping
Value = 60: GOSUB LoadValue
Command = STORERAMPSTARTTIME: GOSUB SendCommand

' Set period of pulses at end of ramping
Value = 10: GOSUB LoadValue
Command = STORERAMPENDTIME: GOSUB SendCommand:

' Set the incremental period of ramping
Value = 1: GOSUB LoadValue
Command = STORERAMPSTEPTIME: GOSUB SendCommand

' Set time between incremental periods
Value = 2: GOSUB LoadValue
Command = STORERAMPINTERVAL: GOSUB SendCommand

' Turn Ramping Mode On
Command = RAMPON: GOSUB SendCommand

' Set Number of Steps to Take
Value = 500: GOSUB LoadValue
Command = LOADCOUNT: GOSUB SendCommand

' Turn on Counter Mode
Command = COUNTON: GOSUB SendCommand

' Execute Motion
Command = GO: GOSUB SendCommand

STOP ' End of Program

'-----
' LoadValue and SendCommand Subroutines
'-----

' LoadValue -----
' This will load the Value Register in the PDN609.
' with the value that is stored in the variable 'Value'
LoadValue:
SEROUT PDN609PIN,16780,[LOADNIBBLE1 + (Value // 16)]
SEROUT PDN609PIN,16780,[LOADNIBBLE2 + ((Value / 16) // 16)]
SEROUT PDN609PIN,16780,[LOADNIBBLE3 + ((Value / 256) // 16)]
SEROUT PDN609PIN,16780,[LOADNIBBLE4 + ((Value / 4096) // 16)]
RETURN

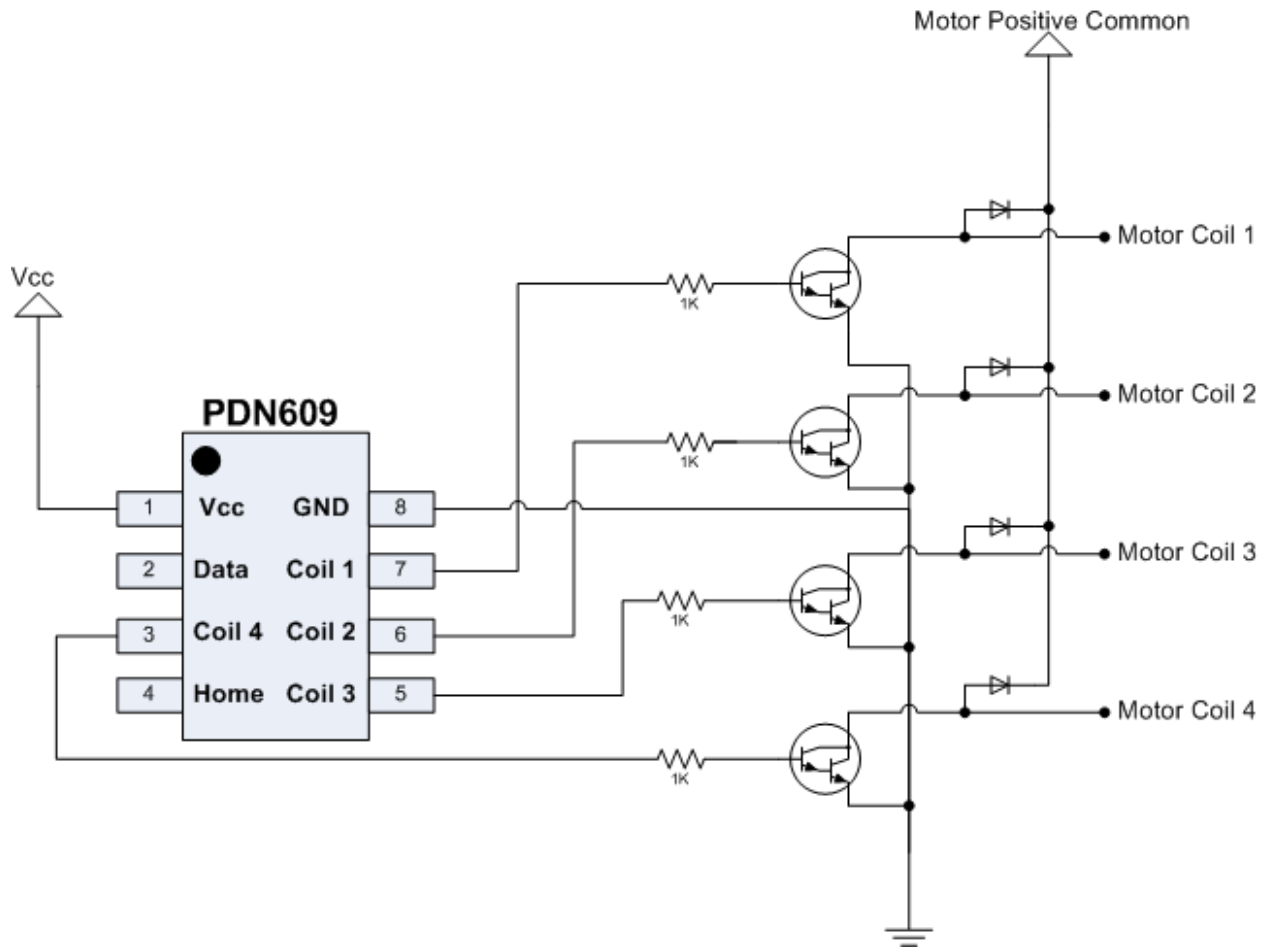
' SendCommand -----
' This will send a command to the PDN609.
' The command must be stored in the variable CommandValue.
SendCommand:
SEROUT PDN609PIN,16780,[Command]
RETURN

END

```

# Connection Schematics

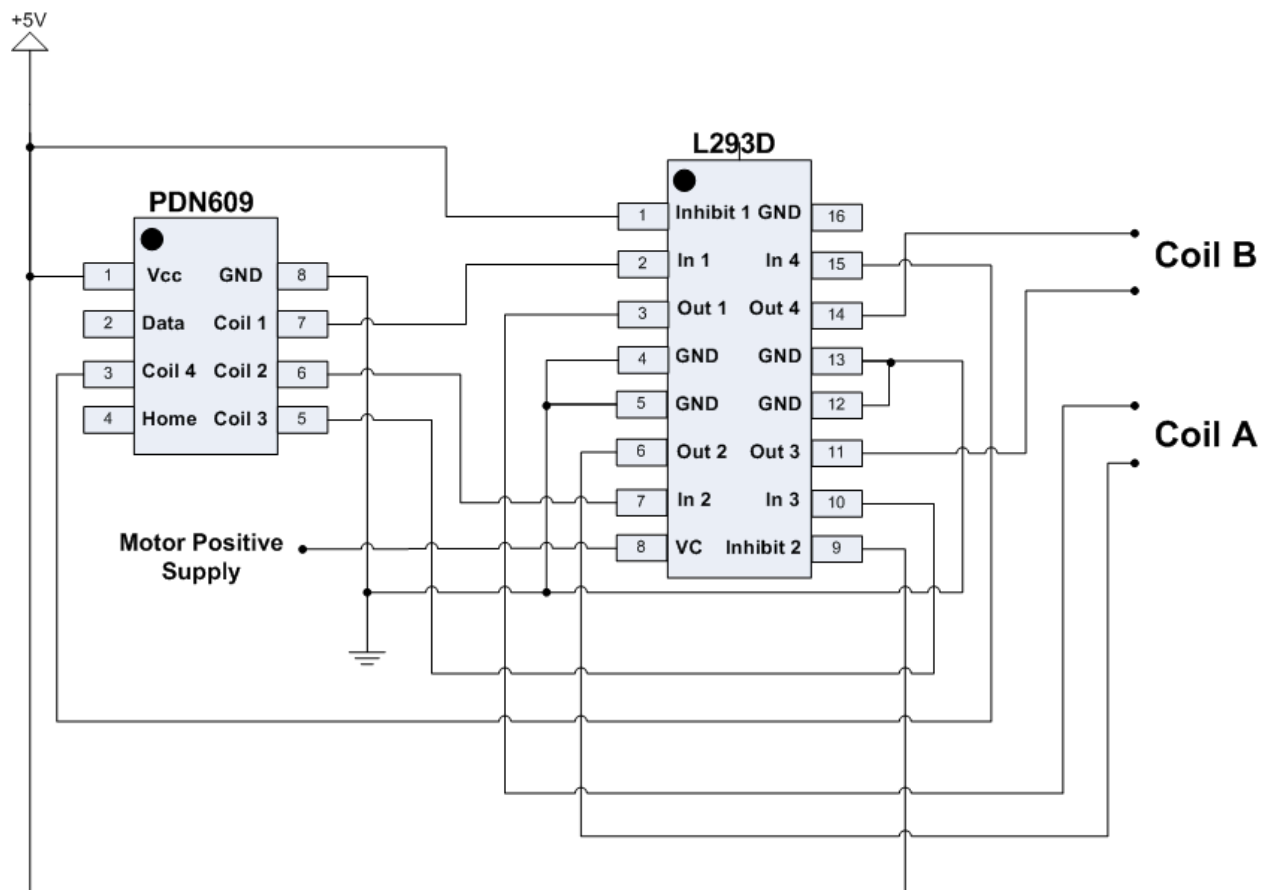
Connection of both unipolar (5 or 6 wire) and bi-polar (4 wire) motors may be made to the PDN609. The schematic below (Figure One) illustrates a typical connection of the PDN609 IC to a unipolar stepper motor. NPN Darlington transistors are used to ensure the current gain required to drive the motor coils. The diodes on the coil lines (typically 1N4004) are used to clamp spikes from the coils. Notice that the motor positive supply is connected to the coil common, and that the individual coils are grounded via the transistors to flow coil current. Serial control data from a microcontroller's TX (serial transmit) pin connects to the PDN609 Serial Data Input, pin 2.



**Figure One: Unipolar (5 or 6 Wire) Motor Connection**

The schematic below (Figure Two) illustrates a typical connection of the PDN609 IC to a bi-polar stepper motor. The L293D H-Bridge IC, with integral clamp diode array, is shown. If, instead, the L293B IC (or other H-Bridge without integral clamp diodes) is utilized, external clamp diodes must be added to the motor coil connections as shown in Figure Three.

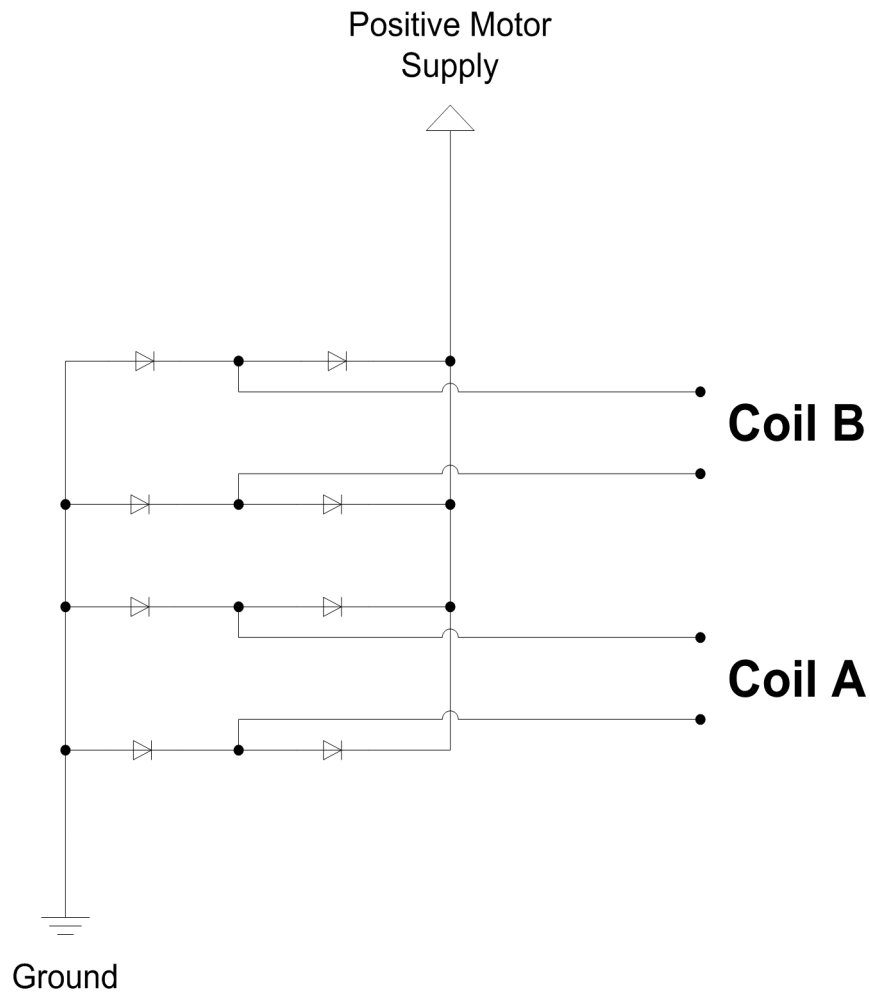
Numerous H-Bridge circuits are possible, including integrated circuit solutions as shown below as well as discrete transistor H-bridges. If constructing an H-Bridge circuit from scratch, care should be taken that the high and low drivers of each H-Bridge half not be turn on together, as this can create a short-circuit current path through the bridge driver.



**Figure Two: Bi-polar (4 Wire) Motor Connection**

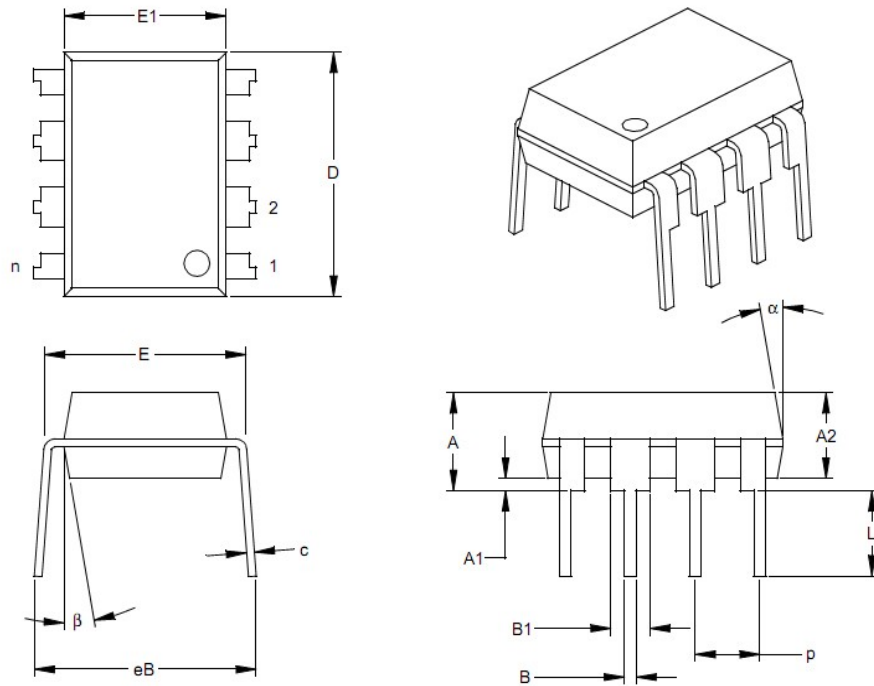
Due to the inductive nature of the coils inside a stepping motor, turning these coils on and off during stepping can lead to voltage spikes. If unchecked, these spikes can sometimes cause improper circuit operation. The diode array shown below, with each of the 4 coil wires having two diodes connected, works well for bipolar stepper motor driving. The diode connected from the coil wire to the motor positive supply clamps spikes greater than the positive power supply, and the diode connecting from the coil to ground clamps spikes on the coil line dipping below ground.

Diodes capable of handling the voltage and current levels generated by your particular motor should be chosen. For smaller motors, a 1N4004 diode is often sufficient.



**Figure Three: Spike-Clamping Diodes**

## 8-Lead Plastic Dual In-line (P) – 300 mil (PDIP)



Units		INCHES*			MILLIMETERS		
Dimension	Limits	MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		8			8	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.140	.155	.170	3.56	3.94	4.32
Molded Package Thickness	A2	.115	.130	.145	2.92	3.30	3.68
Base to Seating Plane	A1	.015			0.38		
Shoulder to Shoulder Width	E	.300	.313	.325	7.62	7.94	8.26
Molded Package Width	E1	.240	.250	.260	6.10	6.35	6.60
Overall Length	D	.360	.373	.385	9.14	9.46	9.78
Tip to Seating Plane	L	.125	.130	.135	3.18	3.30	3.43
Lead Thickness	c	.008	.012	.015	0.20	0.29	0.38
Upper Lead Width	B1	.045	.058	.070	1.14	1.46	1.78
Lower Lead Width	B	.014	.018	.022	0.36	0.46	0.56
Overall Row Spacing	§ eB	.310	.370	.430	7.87	9.40	10.92
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

\* Controlling Parameter

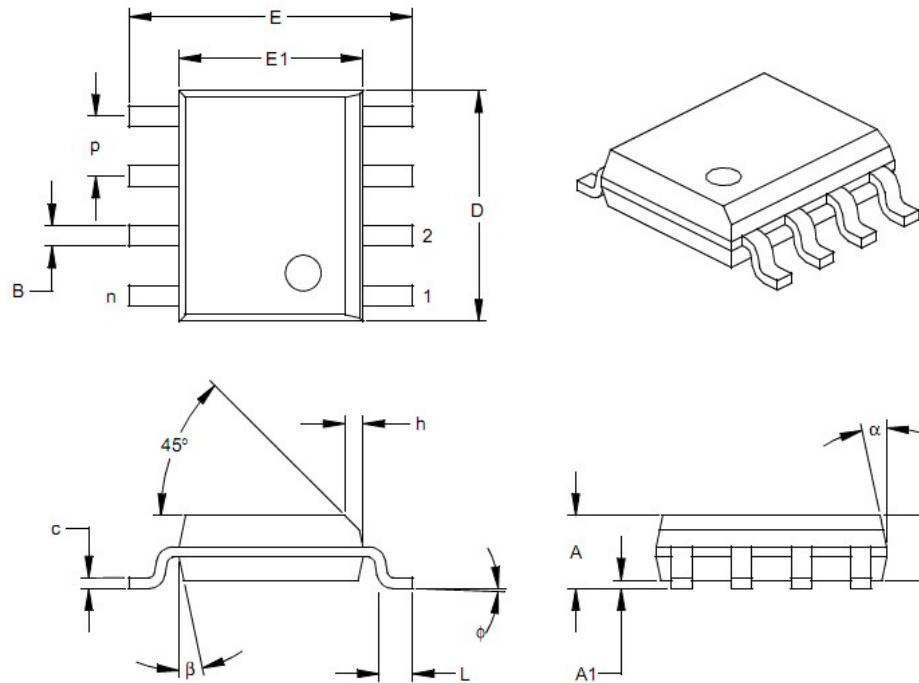
§ Significant Characteristic

**Notes:**

Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.



## 8-Lead Plastic Small Outline (SN) – Narrow, 150 mil (SOIC)



Units		INCHES*			MILLIMETERS		
Dimension	Limits	MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		8			8	
Pitch	p		.050			1.27	
Overall Height	A	.053	.061	.069	1.35	1.55	1.75
Molded Package Thickness	A2	.052	.056	.061	1.32	1.42	1.55
Standoff §	A1	.004	.007	.010	0.10	0.18	0.25
Overall Width	E	.228	.237	.244	5.79	6.02	6.20
Molded Package Width	E1	.146	.154	.157	3.71	3.91	3.99
Overall Length	D	.189	.193	.197	4.80	4.90	5.00
Chamfer Distance	h	.010	.015	.020	0.25	0.38	0.51
Foot Length	L	.019	.025	.030	0.48	0.62	0.76
Foot Angle	$\phi$	0	4	8	0	4	8
Lead Thickness	c	.008	.009	.010	0.20	0.23	0.25
Lead Width	B	.013	.017	.020	0.33	0.42	0.51
Mold Draft Angle Top	$\alpha$	0	12	15	0	12	15
Mold Draft Angle Bottom	$\beta$	0	12	15	0	12	15

\* Controlling Parameter

§ Significant Characteristic

Notes:

Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

## ABSOLUTE MAXIMUM RATINGS

Supply Voltage .....	7.5V
Max. current sunk by an output pin .....	25mA
Max. current sourced by an output pin.....	20mA
Max. current sunk by all inputs.....	50mA
Max. current sourced by all outputs.....	40mA

## STANDARD OPERATING CONDITIONS

Supply voltage .....	2.5V to 5.5V
Typical current draw.....	1.8mA
Operating temperature .....	0°C to +70°C
Oscillator frequency .....	4 MHz

## IMPORTANT NOTICE

Although we strive for a high degree of technical accuracy, information contained in this datasheet may contain errors. Paladin Semiconductor reserves the right to change products or specifications without notice. Customers are advised to obtain the latest versions of product specifications, which should be considered when evaluating a product's appropriateness for a particular use.

BY USING THIS PRODUCT, CUSTOMER AGREES THAT IN NO EVENT SHALL PALADIN SEMICONDUCTOR, ITS PARENT COMPANY OR SUBSIDIARIES BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES AS A RESULT OF THE PERFORMANCE, OR FAILURE TO PERFORM, OF THIS PRODUCT.

PALADIN SEMICONDUCTOR MAKES NO WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. PALADIN'S LIABILITY IS FOR A PERIOD NO GREATER THAN 90 DAYS FROM DATE OF SHIPMENT BY PALADIN SEMICONDUCTOR AND IS LIMITED TO REPLACEMENT OF DEFECTIVE PRODUCT. This warranty covers only defects arising under normal use and not malfunctions resulting from misuse, abuse, or modification.

PALADIN SEMICONDUCTOR'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS. Life support devices or systems are those which are intended to support or sustain life and whose failure to perform can be reasonably expected to result in a significant injury or death to the user. Critical components are those whose failure to perform can be reasonably expected to cause failure of a life support device or system or affect its safety or effectiveness.

The PDN609 IC is implemented as firmware on a PIC12F508 microcontroller, manufactured by Microchip Technology, Inc. For a more comprehensive technical summary of this device, please refer to the PIC12F508 datasheet ([www.microchip.com](http://www.microchip.com)).

## COPYRIGHT NOTICE

This product may not be duplicated. Paladin Semiconductor maintains firmware copyright, with all rights reserved. Unauthorized duplication of this device will be subject to penalty under state and/ or federal law.

PDN609 and the Paladin Semiconductor logo are trademarks of Paladin Semiconductor. All other trademarks and registered trademarks are property of their respective owners.

# Paladin Semiconductor

Carefree Industrial Park  
1600 N. State Rte. 291 Hwy. Suite 140E  
Independence, MO 64058

Telephone: (816) 257-9954  
FAX: (816) 257-9945

Online: [paladinsemi.com](http://paladinsemi.com)  
Email: [support@paladinsemi.com](mailto:support@paladinsemi.com)

*Although we strive to maintain a high degree of technical accuracy, this document may contain typographical errors or other inaccuracies. We reserve the right to correct any inaccuracy in future editions of this document without notice. This document and examples contained herein are provided 'As Is' without warranty of any kind expressed or implied. This document is to be used for informational purposes only and may not be duplicated in volume for commercial purposes other than by distributors authorized by Paladin Semiconductor.*