

AN56551

PSoC[®] 3 Stepper Motor Control with Precision Microstepping

Author: Isaac Sever

Associated Project: Yes

Associated Part Family: CY8C3866AXI-040

Software Version: : PSoC[®] Creator™ 2.0

Related Application Notes: None

Abstract

AN56551 details the implementation of stepper motor control with microstepping (up to 128 microstepping divisions) on the PSoC[®] 3 CY8C3866AXI-040 device. Hardware current limiting and microstepping control of the motor are also explained.

Contents

Stepper Motor Introduction	1
Types of Stepping Motors	2
Variable Reluctance Motors	2
Permanent Magnet Stepping Motor	3
Hybrid Stepping Motors	4
Stepping Motor Control	5
Rotation Control	5
PSoC 3 Introduction	8
Digital Subsystem	8
Analog Subsystem	8
Stepper Motor Control Based on PSoC 3	9
Microstepping and Current Protection Implementation	11
Rotation Control	13
Speed Control	14
Position Control	14
Firmware Architecture	15
PSoC Resource Utilization	16
Summary	16
Related Application Notes	16

Stepper Motor Introduction

Stepper motors convert electrical energy into discrete mechanical rotation. They are ideally suited for many measurement and control applications where positional accuracy is important. Stepping motors have the following advantages:

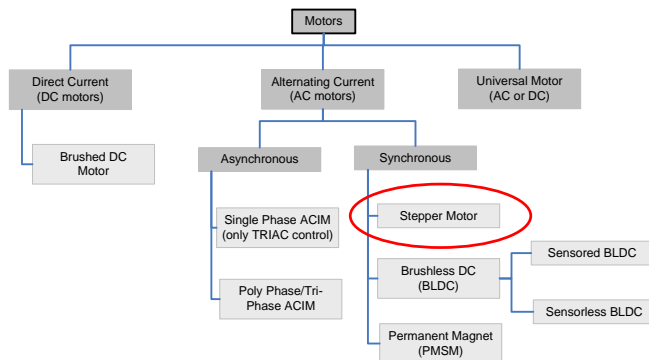
- Full torque when rotation is stopped. This is in contrast to brushed and brushless DC motors, which cannot provide full torque continuously when the rotor is stopped. This aids in maintaining the current position.
- Precise open-loop positioning and repetition. Stepper motors move in discrete steps as long as the motor stays under the maximum torque and current limits. This allows the rotor position to be determined by the control sequence without additional tracking or feedback. High quality stepping motors have three to five percent precision within a single step.
- Quick start, stop, and reverse capability.
- High reliability because there is no brush or physical contact required for commutation. The life span of a stepping motor is dependent on the performance of the bearings.
- Microstepping mode can be used allowing direct connection to a load without intermediate gearing.
- A wide speed range can be controlled by varying the drive signal timing.

Stepping motors also have some drawbacks:

- Inherent resonance can cause noise, jerky rotation and at extreme levels, loss of position.
- It is possible to lose position control in some situations, because no feedback is natively provided.
- Power consumption does not decrease to zero even if load is absent.
- Stepping motors have low power density and lower maximum speed compared to brushed and brushless DC motors. Typical loaded maximum operating speeds for stepper motors are around 1000 RPM.
- Complex electronic controls are required.

Stepper motors are synchronous AC motors, and in the same family as brushless DC (BLDC).

Figure 1. Structure of Motors



Types of Stepping Motors

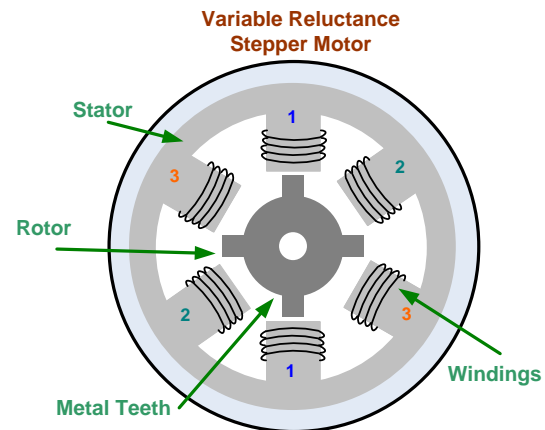
There are several basic types of stepping motors:

- Variable reluctance motors with metal teeth
- Permanent magnet motors
- Hybrid motors with both permanent magnets and metal teeth

Variable Reluctance Motors

Variable reluctance stepping motors have three to five windings and a common terminal connection, creating several phases on the stator. The rotor is toothed and made of metal, but is not permanently magnetized. A simplified variable reluctance stepping motor is shown in Figure 2. In this figure, the rotor has four teeth and the stator has three independent windings (six phases) creating 30-degree steps.

Figure 2. Simple Variable Reluctance Stepping Motor



The rotation of a variable reluctance stepping motor is produced by energizing individual windings. When a winding is energized, current flows and magnetic poles are created, which attracts the metal teeth of the rotor. The rotor moves one step to align the offset teeth to the energized winding. At this position, the next adjacent windings can be energized to continue rotation to another step, or the current winding can remain energized to hold the motor at its current position. When the phases are turned on sequentially, the rotor rotates continuously.

The described rotation is identical to a typical brushless DC motor. The fundamental difference between a stepper and brushless DC motor is that the stepper is designed to operate continuously stalled without overheating or damage.

Rotation for variable reluctance stepping motor with three windings and four rotor teeth is illustrated in Figure 3.

Winding sequence:

1, 2, 3, 1 → 3 steps → quarter turn

12 steps per rotation

As shown in Figure 3, energizing each of the windings in sequence moves the rotor a quarter turn, 12 steps are required for a full rotation.

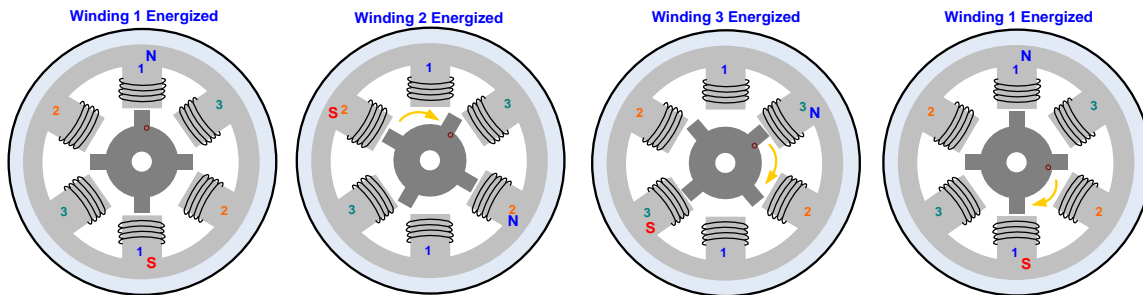
Table 1. Variable Reluctance Stepper Motor in Figure 3

Winding 1	0	+	0	0
Winding 2	0	0	+	0
Winding 3	+	0	0	+

The three steps shown in the following figure move the rotor a quarter turn. A full rotation requires 12 steps for a variable reluctance stepper motor.

Typical variable reluctance motors have more teeth and use a tooth pole along with a toothed rotor to produce step angles near one degree.

Figure 3. Rotation Control of Variable Reluctance Stepper Motor

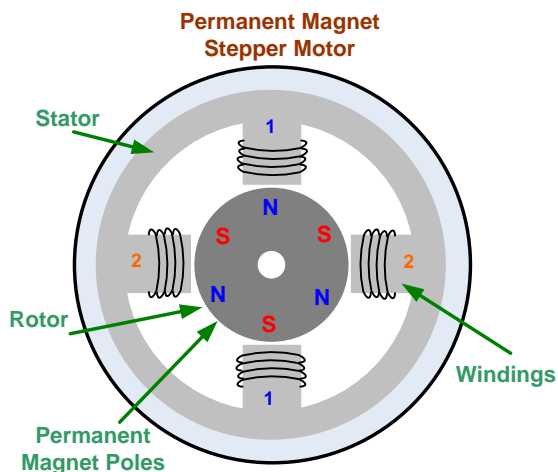


Permanent Magnet Stepper Motor

A permanent magnet stepping motor consists of a stator with windings and a rotor with permanent magnet poles. Alternate rotor poles have rectilinear forms parallel to the motor axis. Stepping motors with magnetized rotors provide greater flux and torque than motors with variable reluctance. The motor, shown in Figure 4, has three rotor pole pairs and two independent stator windings, creating 30-degree steps.

Motors with permanent magnets are subjected to influence from the back-EMF of the rotor, which limits the maximum speed. Therefore, when high speeds are required, motors with variable reluctance are preferred over motors with permanent magnets.

Figure 4. Permanent Magnet Stepper Motor



Rotation of a permanent magnet stepping motor is produced by energizing individual windings in a positive or negative direction. When a winding is energized, a north and south pole are created, depending on the polarity of the current flowing. These generated poles attract the permanent poles of the rotor. The rotor moves one step to align the offset permanent poles to the corresponding energized windings. At this position, the next adjacent windings can be energized to continue rotation to another step, or the current winding can remain energized to hold the motor at its current position. When the phases are turned on sequentially the rotor is continuously rotated.

Rotation for a permanent magnet stepping motor with two windings and three pairs of permanent rotor poles (six poles) is shown in Figure 5.

Winding in sequence:

1 +/-, 2 +/-, 1 -/+, 2 -/+ → 3 steps → quarter turn

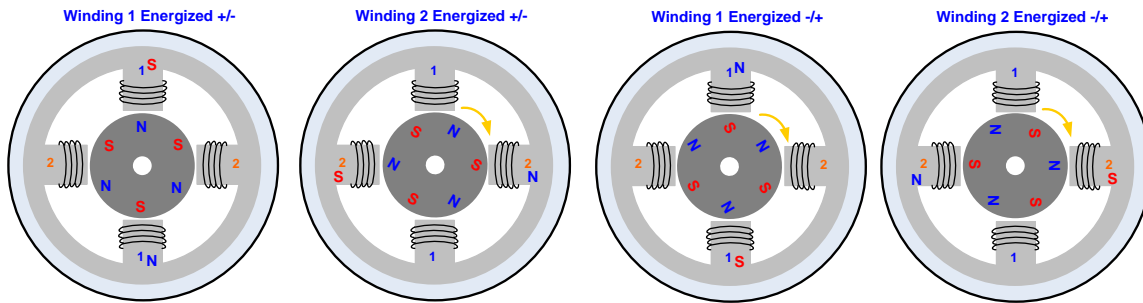
12 steps per rotation

Table 2. Permanent Magnet Stepper Motor in Figure 5

Winding 1	+	0	-	0
Winding 2	0	+	0	-

With one winding energized, the three steps move the rotor a quarter turn. A full rotation requires 12 steps for a permanent magnet stepper motor (bipolar) with both windings energized in each step. As shown in Figure 5, energizing each winding in sequence through each polarity moves the rotor a quarter turn. As before, 12 steps are required for a full rotation.

Figure 5. Rotation Control of Permanent Magnet Stepping Motor, Sequencing Individual Windings



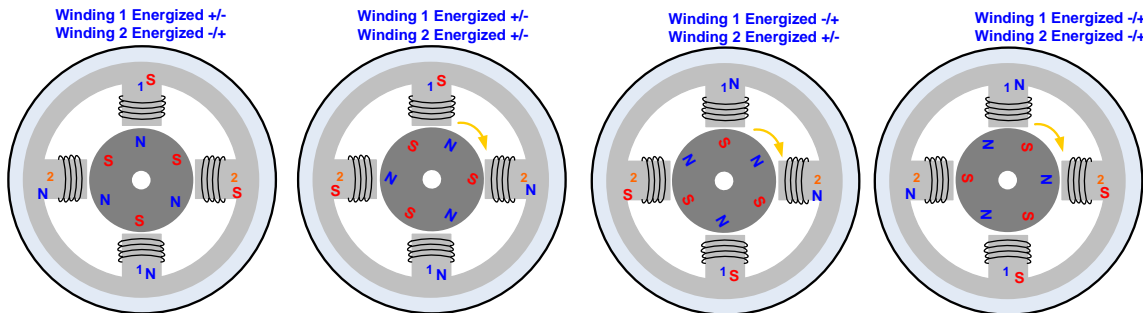
Another alternative to rotate a permanent magnet rotor is to energize both windings in each step. The vector torque generated by each of the coils add. This doubles the current flowing in the motor, and increases the torque. More complex control is also required to sequence the turning on and off of both windings.

As shown Figure 6, energizing two windings in each step, sequencing through each combination of polarities moves the rotor a quarter turn. As before, 12 steps are required for a full rotation.

Table 3. Permanent Magnet Stepping Motor in Figure 6

Winding 1	+	+	-	-
Winding 2	-	+	+	-

Figure 6. Rotation Control of Permanent Magnet Stepping Motor Using Both Windings Together



Hybrid Stepping Motors

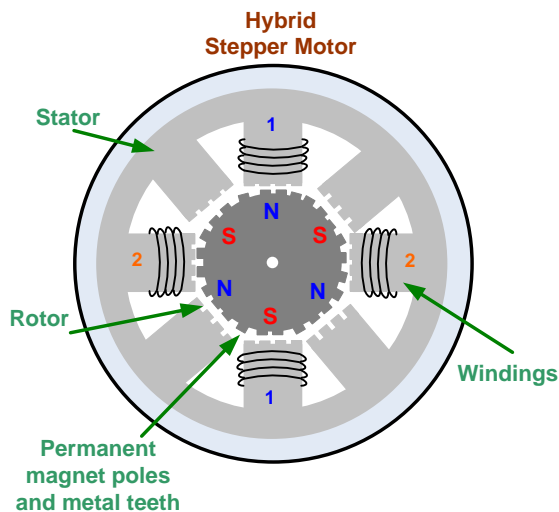
Hybrid stepping motors combine a permanent magnet and a rotor with metal teeth to provide features of the variable reluctance and permanent magnet motors. Hybrid motors are more expensive than motors with permanent magnets, but they use smaller steps, have greater torque, and have greater maximum speeds.

A hybrid motor rotor has teeth, placed on the directional axes. The rotor is divided into parts between a constant magnet pole. The number of rotor pole pairs is equal to the number of teeth on one of the rotor's parts. The hybrid motor stator has teeth creating more poles than just the main poles containing windings. The rotor teeth provide a

smaller magnetic circuit resistance in some rotor positions, which improves static and dynamic torque. This is provided by corresponding teeth positioning; some parts of the rotor teeth are placed opposite the stator teeth and the remaining rotor teeth are placed between the stator teeth. Dependence between the number of rotor poles, the stator equivalent poles, and the phase number define step angle:

$$\begin{aligned} \text{Step Angle} &= \frac{360}{\text{Poles} / \text{phase} * \text{phases}} \\ &= \frac{360}{\text{total poles}} \end{aligned}$$

Figure 7. Hybrid Stepping Motor



Rotation of a hybrid stepping motor is produced with the same control method as a permanent magnet stepping motor, by energizing individual windings in a positive or negative direction. When a winding is energized, a north and south pole are created, depending on the polarity of the current flowing. These generated poles attract the permanent poles of the rotor and the finer metal rotor teeth. The rotor moves one step to align the offset magnetized rotor teeth to the corresponding energized windings.

Stepping Motor Control

A step motor is a synchronous electric motor. Its fixed rotor equilibrium position occurs when aligned with the stator magnetic field. When the stator changes position, the rotor rotates to occupy a new equilibrium position.

There are several stepper motor drive modes:

- Full Step Mode
- Double Step Mode
- Half Step Mode
- Microstep Mode

Each of these modes controls stepping motor phases in different ways.

Rotation Control

Stepping motors can be controlled in a variety of ways, trading off implementation requirements with greater accuracy and smoother transitions. Rotation control with full steps, half steps, and microsteps is described as follows:

Full Step Mode

Full step mode for a permanent magnet and hybrid stepping motor is detailed in the Stepper Motor Introduction. Figure 5 illustrates one-phase full step mode in which only one winding is turned on at a time. In this mode, the rotor's balanced position for each step is in line with the stator poles. With only half of the motor coils used at a given time, the full torque obtained is limited.

Double Step Mode

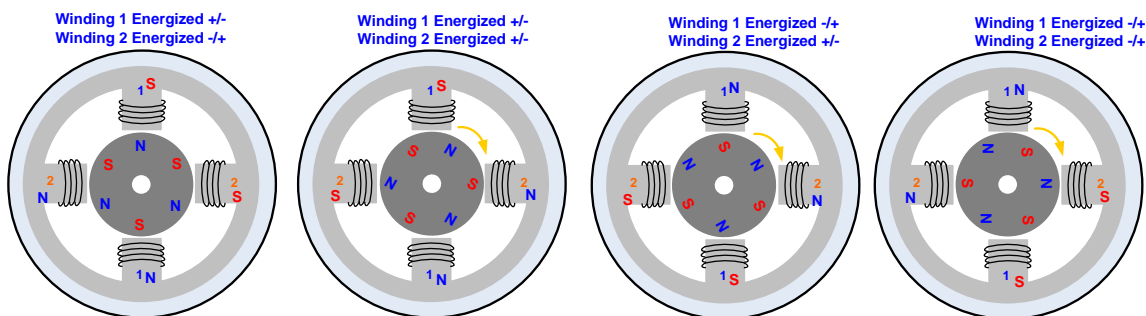
Two-phase, full step mode shown in Figure 6 uses both windings energized in each step. This doubles the current through the motor and provides 40 percent more torque than when only one phase is used at a time. With two windings energized, the rotor's balanced position for each step is halfway between the two energized stator poles.

Half Step Mode

The full step and double step drive modes can be combined to generate half steps of rotation for half step mode. First one winding is turned on, and then the second winding is energized, moving the rotor half a step towards the second, as shown in Figure 8.

A half step with the combination of one and two windings energized in full step mode produces higher resolution, but does not provide constant torque throughout rotation.

Figure 8. Three Half Steps, 1/8 of a Rotation

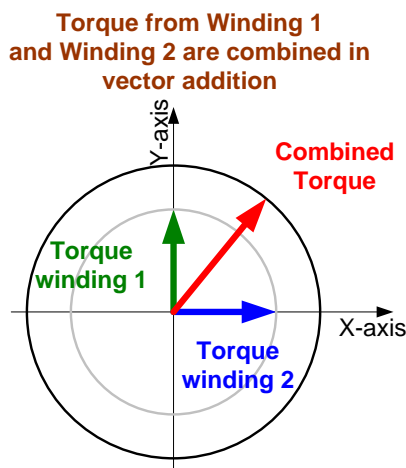


Microstep Mode

Microstepping mode is an extension of the half step drive mode. Instead of switching the current in a winding from on to off, the current is scaled up and down in smaller steps. When two phases are turned on and the current on each phase is not equal, the rotor position is determined by the current phase ratio. This changing current ratio creates discrete steps in the torque exerted on the rotor and results in smaller fractional steps of rotation between each full step. Microstep mode reduces torque ripple and low speed resonance present in the other modes and is required in many situations.

Microstepping creates rotation of the rotor by scaling the contributions of the two additive torque vectors from the two stepper motor windings.

Figure 9. Torque in Microstepping Control Mode



The total torque exerted on the rotor is the vector addition of the torque from the two rotors. Each of the torques is proportional to the position of the rotor and the sine/cosine of the step angle.

$$T_{total} = T_1 + T_2$$

$$Torque_1 = -T_{keeping} \cdot \sin\left(\frac{\pi/2}{step\ angle}\right) \cdot \phi_{rotor}$$

$$Torque_2 = -T_{keeping} \cdot \cos\left(\frac{\pi/2}{step\ angle}\right) \cdot \phi_{rotor}$$

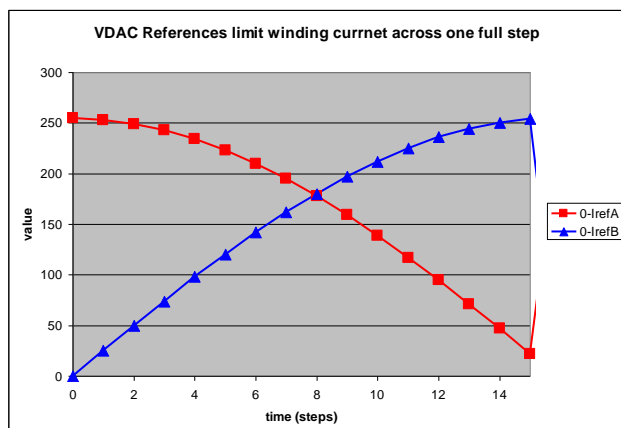
These equations can be combined and solved for the position of the rotor.

$$position = \phi_{rotor} = \frac{2 \cdot step\ angle}{\pi} \arctan\left(\frac{T_1}{T_2}\right)$$

Fractional steps are created by scaling torque contributions between windings. Because torque is proportional to magnetic flux that is proportional to the current in the winding, the position of the rotor can be controlled by controlling the current flowing in each winding. To create smooth microsteps between full steps, the current is varied sinusoidally with a 90-degrees phase shift between the two windings as shown in Figure 10.

The current is scaled by controlling the RMS current using a current mode buck converter, commonly called a chopper drive when used with stepper motors. The phase current is converted into a voltage using a sense resistor in each phase ground path. This voltage is routed to a comparator that disables the output whenever the phase current rises above a reference. The comparator reference is provided by a voltage Digital to Analog Converter (VDAC). By changing the VDAC supplied current limit for each microstep the total motor torque remains approximately constant for each step of the sinusoidal current waveform.

Figure 10. VDAC Current Limit for Microstep Mode

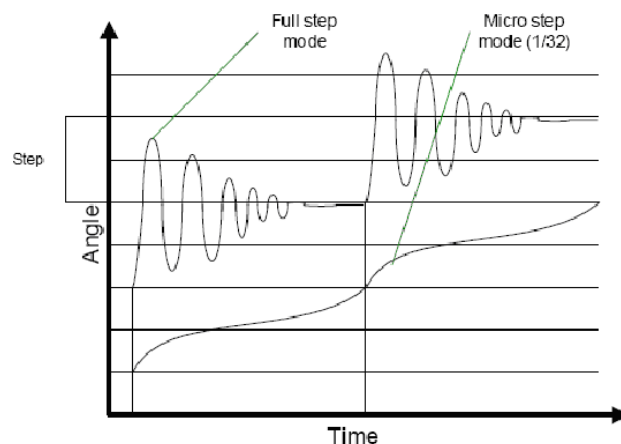


Microstepping allows the rotor position to be controlled with more accuracy and also has advantages in the rotation. Advantages of microstepping are:

- Position is controlled with more accuracy
- Rotation can be stopped at specific fraction of a step
- Transitions are smoother
- Damp resonance creates fewer oscillations as motor steps (especially at startup and slowdown)

The smoother transitions are shown in Figure 11. The position and transitions vary from ideal calculations due to non idealities in the system such as resistance losses, static friction, and the approximation of the sinusoidal current limits. Microstep resolutions in excess of 16 may not be noticeable due to motor friction and the effects of mechanical error.

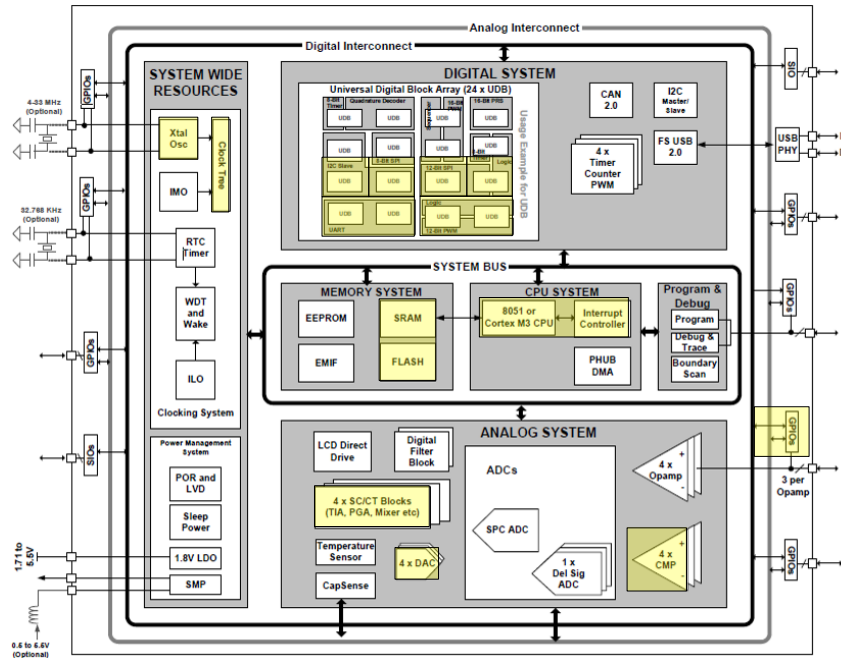
Figure 11. Smooth Transitions Between Steps and Limited Oscillations and Settling in Microstep Mode



PSoC 3 Introduction

The CY8C3866AXI device is in the PSoC 3 architecture. A block diagram of the device is shown in Figure 12 with the blocks used in the stepper application highlighted.

Figure 12. PSoC 3 (CY8C3866AXI) Block Diagram



Digital Subsystem

The PSoC 3 digital subsystem provides unique configurability of functions and interconnects. The stepper motor control uses these digital resources to implement timers, pulse width modulator (PWM) blocks, control registers, and a hardware lookup table (LUT).

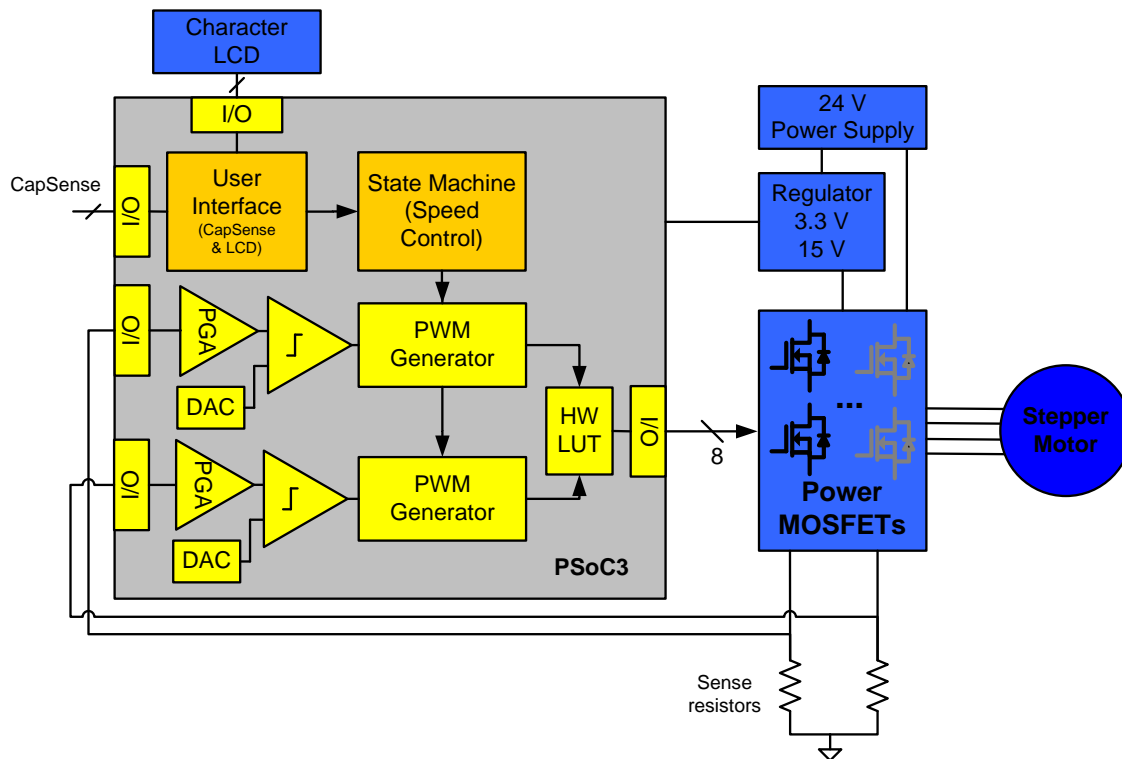
Analog Subsystem

The PSoC 3 analog subsystem provides the device the second half of its unique configurability. The stepper motor uses dedicated comparators, voltage DACs, and programmable gain amplifiers (PGA).

Stepper Motor Control Based on PSoC 3

The block diagram of the stepper motor control based on the CY8C3866AXI is shown in Figure 13. The PSoC Creator™ schematic is shown in Figure 14.

Figure 13. Block Diagram of PSoC 3 Stepper Motor Controller



Input control signals to the PSoC 3 device are:

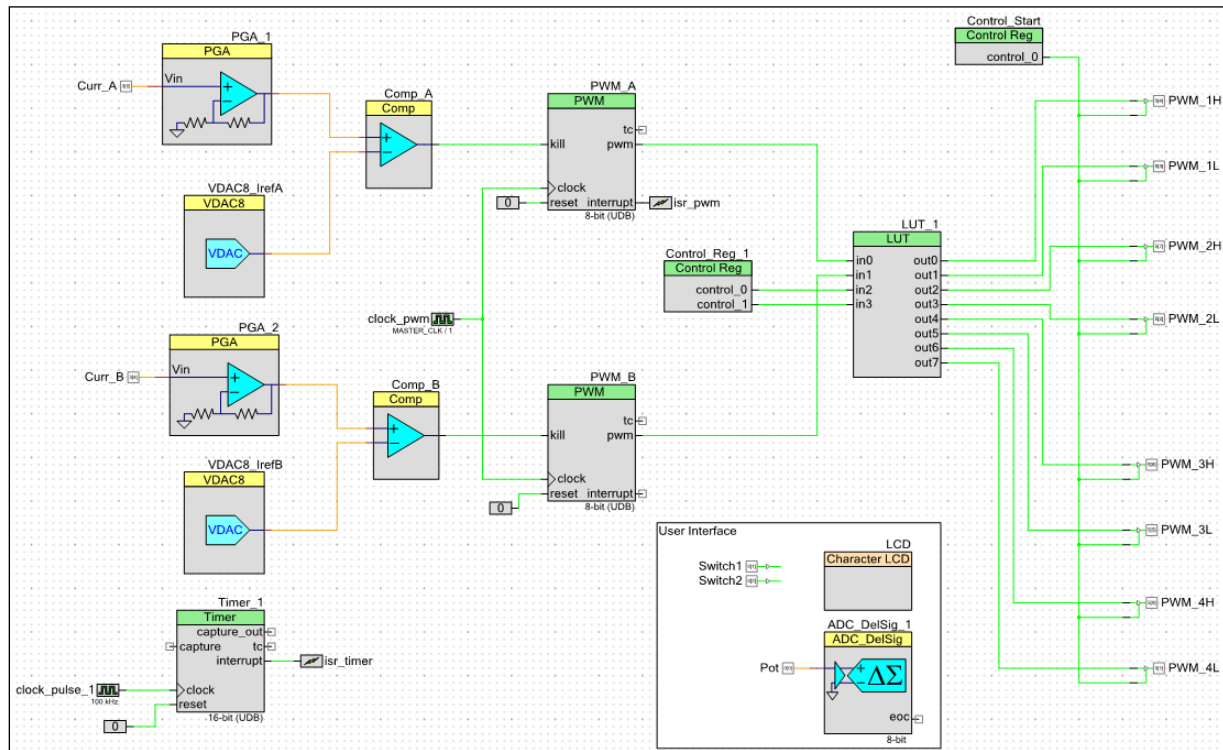
1. **Motor Current Sensing:** Analog input pins to detect motor phase current on shunt resistor. Used to limit current of the motor phases. See details in the following section.
2. **User Interface Pins**
 - ❑ **User Input:** Analog pin to read potentiometer for parameter input. Two digital pins for Menu control buttons.
 - ❑ **Character LCD:** Digital output port (7 pins) to drive the character LCD on the DVK for menu options and user feedback.

Outputs from the PSoC 3 device are power device driver signals:

1. PWM signals to the high side drivers (four digital output pins).
2. PWM signals to the low side drivers (four digital output pins).

There are eight total digital output pins used for motor drive.

Figure 14. PSoC Creator Schematic for Stepper Motor Control



The PWMs are not used to produce the typical pulse width modulation output used with other motors. Instead, the PWMs act more as a timer to ensure a maximum chopping frequency to avoid overheating the drivers. Additionally the PWM kill circuit natively includes the cycle kill mode that implements the chopper drive method by disabling the drive outputs for the remainder of the current PWM cycle after the comparator trips.

The PWM signals are routed to a Look Up Table logic block, along with the current stepping stage index. This logic block implements a look up table (LUT) using the PLD capabilities of a Universal Digital Block (UDB) and routes the PWM signals to the eight legal output control combinations based on the current polarity of each phase. These control signals are routed through GPIOs to the external power driver circuits that drive the stepper motor. In the demonstrated chopper drive topology Transistors or MOSFETs are typically used to switch the high voltages and currents used to drive the stepper motors. The sequencing of the PWM control signals on the external power drivers produces the step by step rotation of the motor.

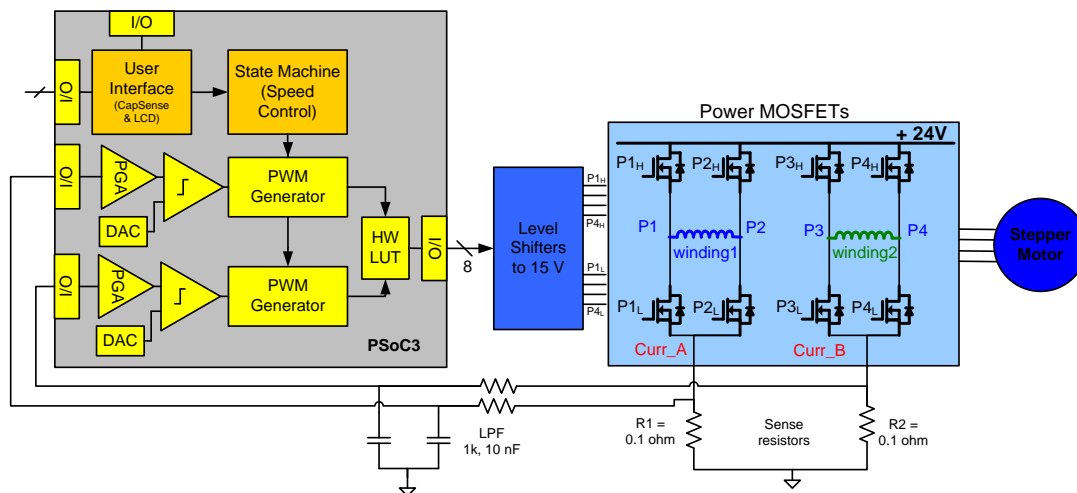
A timer generates periodic interrupts that generate each step (or microstep) of the motor. This timer can be used to run the motor at a specific speed, or to a specific position (exact number of steps). To set the speed of the motor, the interrupt period of the timer is updated by firmware.

PSoC 3 also implements current limiting for motor over current protection and microstepping in hardware. This is described in the following section.

Microstepping and Current Protection Implementation

Microstepping limits the current flowing in the motor windings to create smooth and well controlled transitions between full steps. This functionality also builds protection in hardware for over current that shields the motor from damage. The block diagram of the system with the current feedback sensing paths is shown in Figure 15.

Figure 15. Over-Current Protection Block Diagram

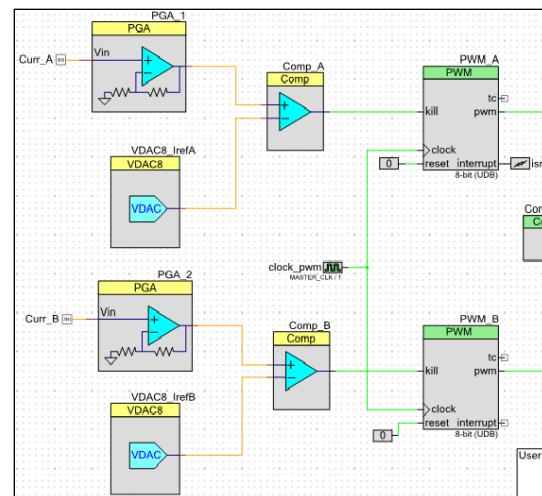


Motor current is measured with two shunt resistors in the ground paths of the power driver MOSFETs (R1 and R2 in Figure 15). This voltage is low pass filtered on the board and connected to two analog pins on PSoC 3 (labeled Curr_A and Curr_B).

The input voltages are fed into programmable gain amplifiers (PGA) implemented with the analog continuous time (CT) blocks. The PGA buffers the input voltage and drives it to a continuous time comparator. This voltage level from the sense resistor is compared to the current limit, set by an 8-bit voltage DAC. For microstepping the DACs' output, sine and cosine waveforms are generated from a software lookup table. This limits the motor current sinusoidally for smooth microstepping.

The output of the comparator is connected to the PWM block and kills the PWM output when the current limit threshold is exceeded. This provides cycle-by-cycle current limiting to the motor and creates smooth microstepping transitions. The implementation of the current limiting protection in PSoC Creator is shown in Figure 16.

Figure 16. PSoC Creator Schematic Implementation of Current Limiting Block for Microstepping



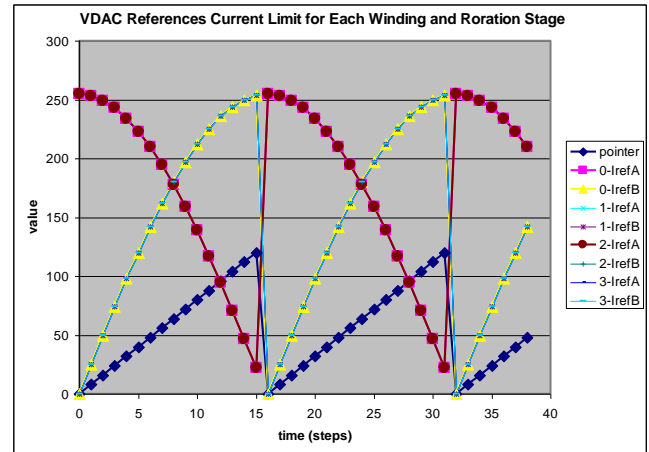
The PSoC 3 resources used in current limiting are:

- Two Continuous Time (CT) blocks implement the PGAs
- Two fixed analog comparators are dedicated analog resources and do not use any SC/CT blocks.
- Two 8-bit PWMs implemented in UDBs. (The same PWMs used to control the power device drivers.) The output of the comparator triggers the kill input to the PWM when a current limiting condition is detected.
- Two 8-bit VDACS. These built in 8-bit voltage DACs are used to set the threshold for the comparator current limit.

The current limits are set to limit the torque generated by each winding and to control the microstep position of the rotor. For each microstep, the current limit for each winding is updated from a sine/cosine LUT, keeping the 90 degree shift between the two (as shown in Figure 17). These current limits also provide protection to the motor, limiting the instantaneous current flowing in the motor and preventing overheating, magnetic flux saturation, or loss of step position.

Shown in Figure 17 are the settings for each DAC and rotation index, and the microstep pointer (ramping 1-128).

Figure 17. Current Limit Versus Time Step for 128-Step Microstepping



The currents flowing in the two windings are measured with small sense resistors between the power devices and ground. The value of the current detection shunt resistor is a trade-off between power efficiency and robustness of the detection blocks. For a given current limit, enough change in voltage must be generated by the motor current to accurately detect the change with the comparator, but increasing the resistor increases heat and reduces efficiency.

The current limiting protection mechanism implemented in PSoC 3 hardware is an on-chip low cost solution.

Rotation Control

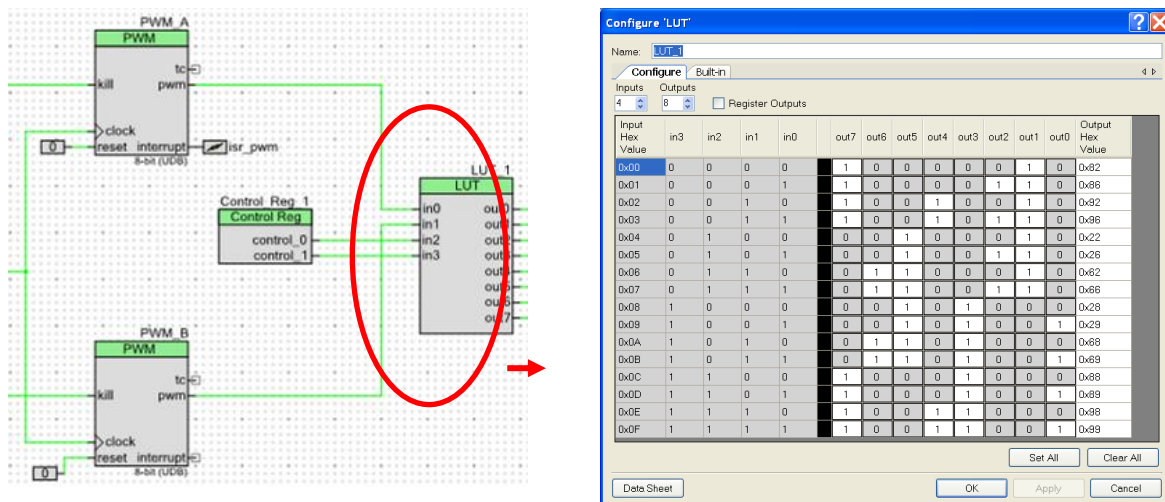
The output PWM drivers are controlled by a hardware lookup table (LUT). The table takes inputs from the two PWM blocks and a control register that holds the rotation index (as shown in Figure 18).

In Table 4, the PWM control hardware LUT receives the stage index and PWM signals as inputs and outputs the eight PWM driver signals.

Table 4. PWM Control Hardware LUT

Index	PWM A	PWM B	MPhase Output Levels (High 1, Low -1, Floating 0)			
			MPhase 1	MPhase 2	MPhase 3	MPhase 4
0	0	0	0	0	0	0
0	0	1	0	0	-1	1
0	1	0	1	-1	0	0
0	1	1	1	-1	-1	1
1	0	0	0	0	0	0
1	0	1	0	0	-1	1
1	1	0	-1	1	0	0
1	1	1	-1	1	-1	1
2	0	0	0	0	0	0
2	0	1	0	0	1	-1
2	1	0	-1	1	0	0
2	1	1	-1	1	1	-1
3	0	0	0	0	0	0
3	0	1	0	0	1	-1
3	1	0	1	-1	0	0
3	1	1	1	-1	1	-1

Figure 18. PSoC Creator Schematic LUT Implementation of PWM Output Control



When operating with microstep drive mode, the PWM outputs for PWM_A and PWM_B cycle through 01, 10, and 11. When the stepper motor operates under full step mode, both PWMs are on (11). In this case, the LUT simplifies to the following table the rotation sequence described in the full step descriptions earlier.

In Table 5, the simplified MPhase output control hardware LUT receives the stage index and PWM signals as inputs and outputs the eight PWM driver signals.

Table 5. MPhase Output Control Hardware LUT

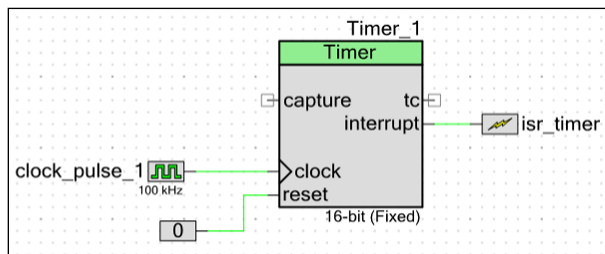
Index	MPhase Output Levels (High 1, Low -1, Floating 0)				Winding Current	
	MPhase 1	MPhase 2	MPhase 3	MPhase 4	Winding 1	Winding 2
0	1	-1	-1	1	+	-
1	-1	1	-1	1	-	-
2	-1	1	1	-1	-	+
3	1	-1	1	-1	+	+

Speed Control

The stepper motor can be run at a fixed speed or to a desired position. To run at a fixed speed, the timer period that triggers each step (or microstep) is adjusted. The 16-bit timer terminal count triggers an interrupt that is used to initiate each step. Input frequency of the timer is 100 kHz to ensure precision speed control. PSoC 3 is also able to receive step pulse commands from an external controller, such as a PLC although this feature is not provided in this application note project.

In Figure 19, the timer terminal count triggers an interrupt that initiates each step.

Figure 19. PSoC Creator Schematic Implementation of Speed Control Timer



Position Control

To run in position control mode, the stepper motor turns a specific number of steps and then stops. Position control mode is not supported with the user interface in the included application note project. An internal counter is used to count the desired steps. When the desired position is reached, the step control from the timer interrupt is masked until the user requests another action.

When the motor stops, phase current is lowered automatically to save power and reduce heating.

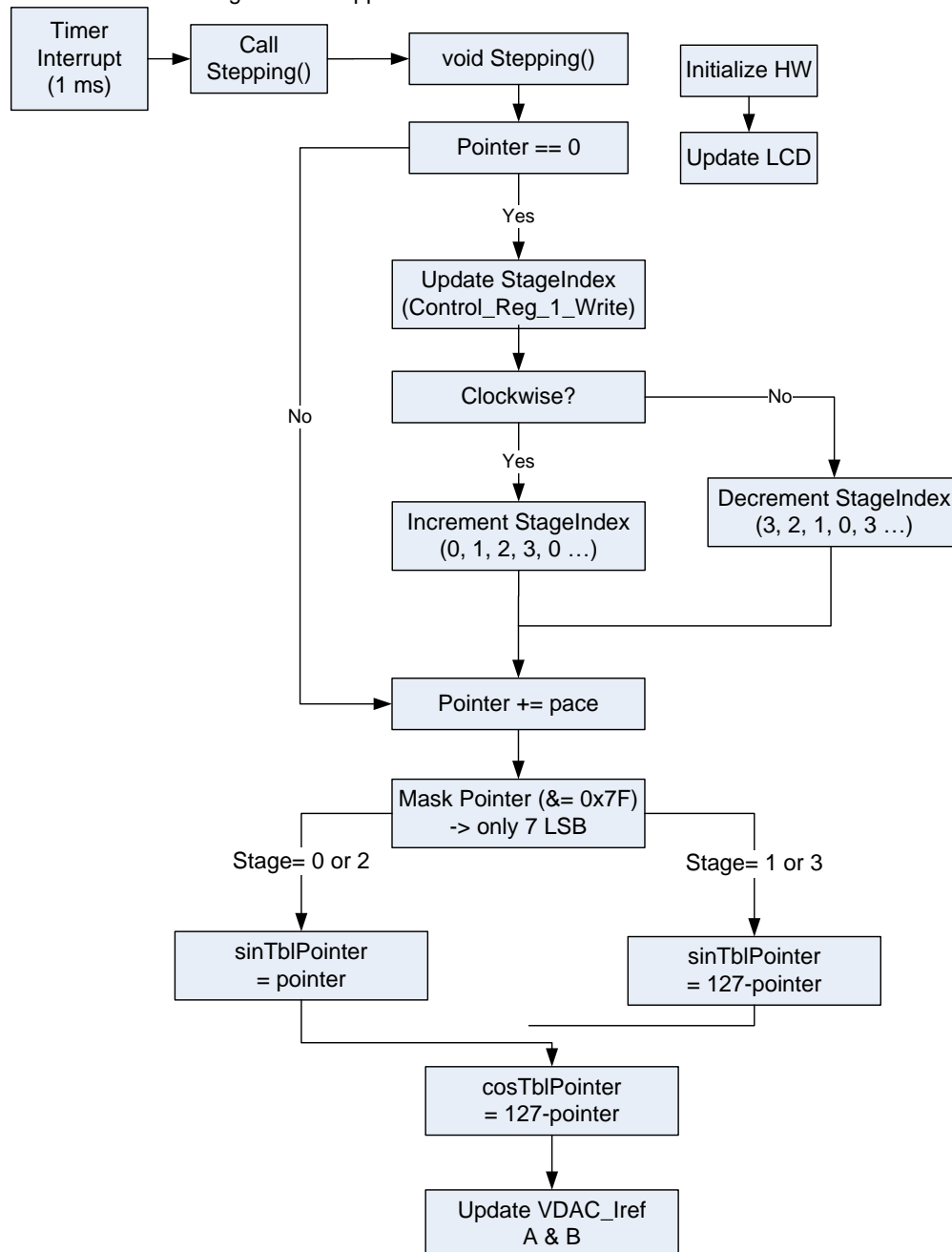
The ability to control the position in an open loop configuration by counting steps (or microsteps) is dependent on the stepper motor operating within the torque and motor load limits. If the torque/load limits are exceeded, the motor can miss steps and the absolute rotational position information is lost.

Firmware Architecture

There is one main loop and one interrupt service routine (ISR) for control of the motor, the timer ISR. The timer ISR generates an interrupt that triggers the step control function (Figure 19). Each time the step function is called, the motor takes one step (or microstep). The step function

looks up the sinusoidal values from a table and sets the DAC output voltage to control the phase currents. A flow chart of the firmware operation is shown in Figure 20. An ISR for the ADC is also used for the example project UI.

Figure 20. Stepper Motor Control Firmware Flow Chart



PSoC Resource Utilization

The stepper motor uses resources from the digital and analog portions of the PSoC 3 device. The highest use of resources are the VIDACs and comparators. Two VDACs and two comparators are used for the stepper motor microstepping control. This constraint limits the CY8C3866AXI-040 device to a maximum of two stepper motor controllers.

Table 6. Stepper Motor Demo CY8C3866AXI-040
Resource Utilization (Blocks with 0 Used are Not Shown)

Resource Type	Used	Free	Max	% Used
Macrocells	27	165	192	14.06%
Unique Pterms	32	352	384	8.33%
Total Pterms	91			
I/O Cells	23	49	72	31.94%
Datapath Cells	4	20	24	16.67%
Control Cells	5	19	24	20.83%
Interrupts	3	29	32	9.38%

Table 7. Stepper Motor Demo on CY8C3866AXI-040
Memory Utilization (Keil™ Compiler, Level-5 Optimization)

			Stepper Motor Usage (bytes)
CY8C3866AXI-40	PSoC 3	RAM	488
CY8C3866AXI-40	PSoC 3	Flash	12,004

Summary

Cypress stepper motor control with PSoC 3 incorporates current limiting and microstepping control for an optimized solution. Up to 128 microsteps is suitable for precision position control. The PSoC 3 stepper motor control solution has low total system cost and leaves significant PSoC 3 resources available for additional system functions.

Related Application Notes

[AN2229](#) “Motor Control - Multi-Functional Stepping Motor Driver” by Victor Kremin and Ruslan Bachinsky.

Document History

Document Title: PSoC® 3 Stepper Motor Control with Precision Microstepping

Document Number: 001-56551

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	2782790	IQS/HMT	10/09/09	New application note.
*A	2896476	GJV	3/19/10	Minor edits to clarify operation of motor drive.
*B	3009045	IQS	8/16/10	Update screenshots and resources to Beta5.
*C	3153420	IQS	01/25/11	Updated screenshots and resources for Creator 1.0.
*D	3450441	RLRM	11/29/11	Updated document to remove mention of the UART in the design. Updates project files.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at [Cypress Locations](#).

Products

Automotive	cypress.com/go/automotive
Clocks & Buffers	cypress.com/go/clocks
Interface	cypress.com/go/interface
Lighting & Power Control	cypress.com/go/powerpsoc cypress.com/go/plc cypress.com/go/memory

PSoC® Solutions

psoc.cypress.com/solutions

[PSoC 1](#) | [PSoC 3](#) | [PSoC 5](#)

[Cypress Developer Community](#)
[Community](#) | [Forums](#) | [Blogs](#) | [Video](#) | [Training](#)

Memory

Optical Navigation Sensors	cypress.com/go/ons
PSoC	cypress.com/go/psoc
Touch Sensing	cypress.com/go/touch
USB Controllers	cypress.com/go/usb
Wireless/Rf	cypress.com/go/wireless

PSoC is the registered trademarks of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are the property of their respective owners.



Cypress Semiconductor Phone : 408-943-2600
 198 Champion Court Fax : 408-943-4730
 San Jose, CA 95134-1709 Website : www.cypress.com

© Cypress Semiconductor Corporation, 2009-2011. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.