Dual Gauge Driver with Configurable Response Time

33976

The 33976 is a single-packaged, Serial Peripheral Interface (SPI) controlled, dual step motor gauge driver integrated circuit (IC). This monolithic IC consists of four dual output H-Bridge coil drivers and the associated control logic. Each pair of H-Bridge drivers is used to automatically control the speed, direction, and magnitude of current through the two coils of a two-phase instrumentation step motor, similar to an MMT-licensed AFIC 6405 or Switec MS-X15.xxx motor.
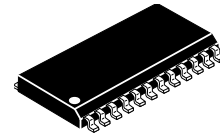
The 33976 is ideal for use in automotive instrumentation systems requiring distributed and flexible step motor gauge driving. The device also eases the transition to step motors from air core motors by emulating the air core pointer movement with little additional processor bandwidth utilization.

CONFIGURABLE DUAL GAUGE DRIVER

Features

- MMT-Licensed Two-Phase Step Motor Compatible
- Switec MS-X15.xxx Step Motor Compatible
- Minimal Processor Overhead Required
- Fully Integrated Pointer Movement and Position State Machine with Channel-Independent Configurable Pointer Movement
- 4096 Possible Steady State Pointer Positions
- 340° Maximum Pointer Sweep
- Maximum Acceleration of 4500°/s^2
- Maximum Pointer Velocity of 400°/s
- Analog Microstepping (12 Steps/Degree of Pointer Movement)
- Pointer Calibration and Return to Zero (RTZ)
- SPI-Controlled 16-Bit Word
- Calibratable Internal Clock
- Low Sleep Mode Current
- Pb-Free Packaging Designated by Suffix Code EG

DW SUFFIX
EG SUFFIX (PB-FREE)
98ASB42344B
24-PIN SOICW

ORDERING INFORMATION

Device	Temperature Range (T$_A$)	Package
MC33976DW/R2	-40°C to 125°C	24 SOICW
MCZ33976EG/R2		
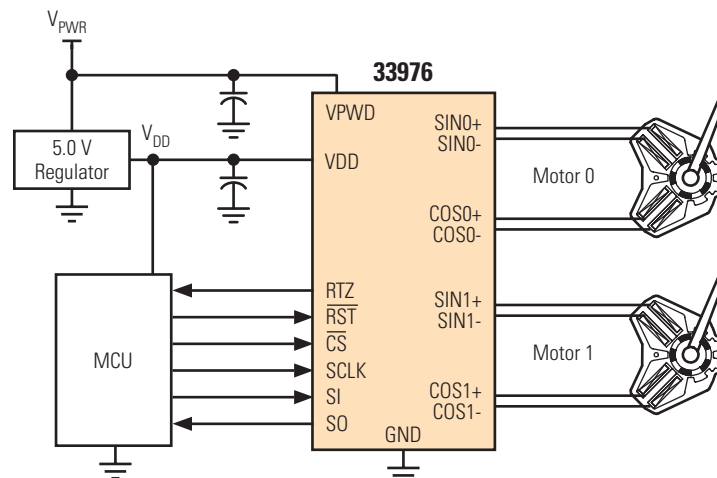
Figure 1. 33976 Simplified Application Diagram

freescale™
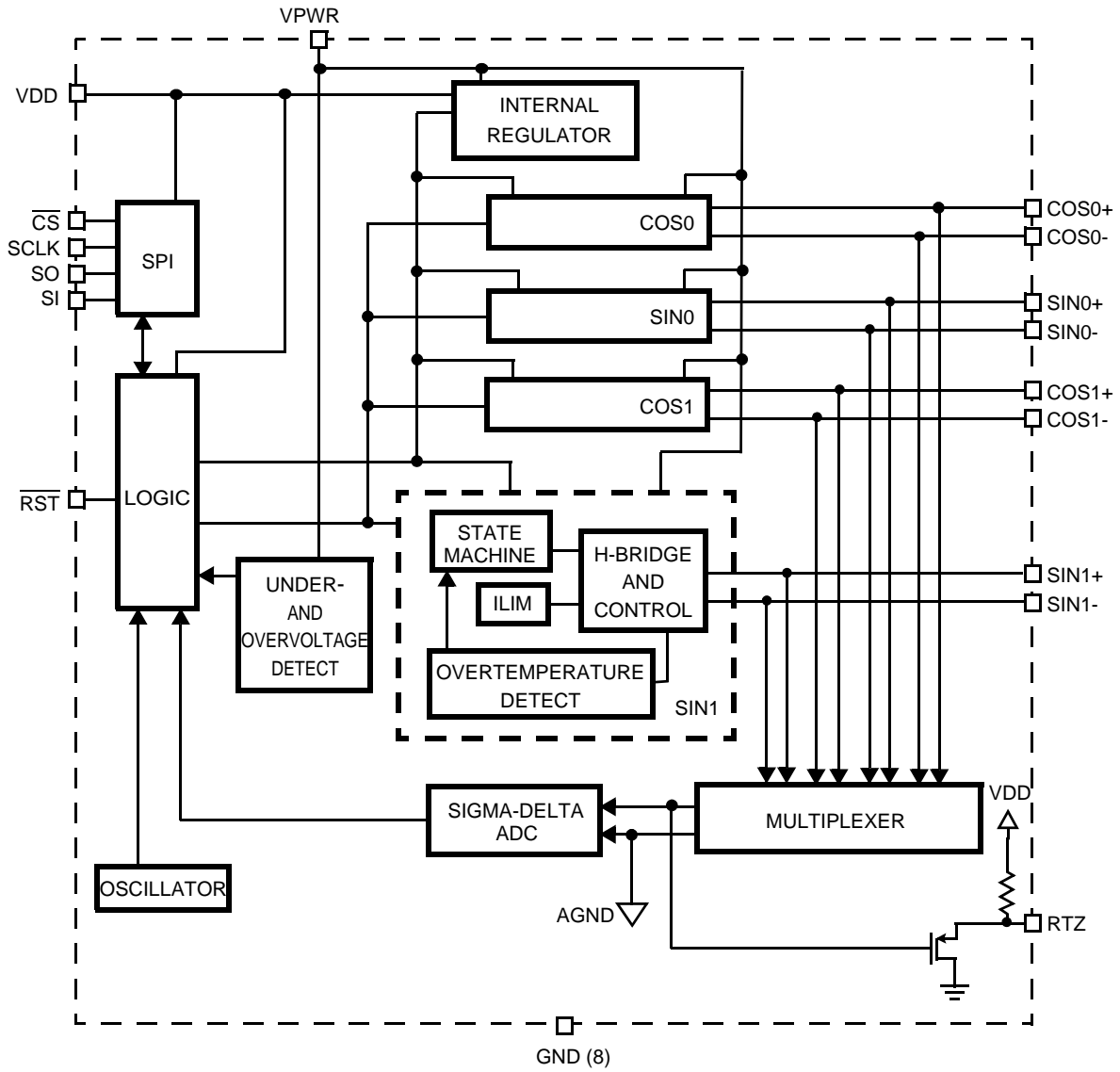semiconductor

INTERNAL BLOCK DIAGRAM

Figure 2. 33976 Simplified Internal Block Diagram

PIN CONNECTIONS
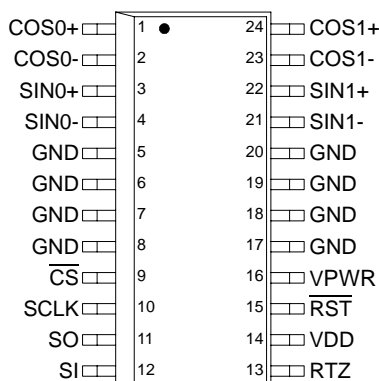
Figure 3. 33976 Pin Connections

Table 1. 33976 Pin Definitions

Pin Number	Pin Name	Pin Function	Formal Name	Definition
1 2 3 4	(MS Motor Pin #) COS0+ (MS #4) COS0− (MS #3) SIN0+(MS #1) SIN0−(MS #2)	Output	H-Bridge Outputs 0	Each pin is the output pin of a half bridge, designed to source or sink current.
5−8, 17−20	GND	Ground	Ground	These pins serve as the ground for the source of the low-side output transistors as well as the logic portion of the device.
9	CS	Input	Chip Select	This pin is connected to a chip select output of a LSI IC.
10	SCLK	Input	Serial Clock	This pin is connected to the SCLK pin of the master device and acts as a bit clock for the SPI port.
11	SO	Output	Serial Output	This pin is connected to the SPI Serial Data Input pin of the master device or to the SI pin of the next device in a daisy chain.
12	SI	Input	Serial Input	This pin is connected to the SPI Serial Data Output pin of the master device from which it receives output command data.
13	RTZ	Output	Multiplexed Output	This is a multiplexed output pin for the non-driven coil, during a Return to Zero (RTZ) event.
14	VDD	Input	Voltage	This SPI and logic power supply input will work with 5.0 V supplies.
15	RST	Input	Reset	This input has an internal active pull-up.
16	VPWR	Input	Battery Voltage	Power supply.
21 22 23 24	(MS Motor Pin #) SIN1−(MS #2) SIN1+(MS #1) COS1−(MS #3) COS1+(MS #4)	Output	H-Bridge Outputs 1	Each of these pins is the output pin of a half bridge, designed to source or sink current.

ELECTRICAL CHARACTERISTICS

MAXIMUM RATINGS

Table 2. Maximum Ratings

All voltages are with respect to ground unless otherwise noted. Exceeding these ratings may cause a malfunction or permanent damage to the device.

Rating	Symbol	Value	Unit
Power Supply Voltage Steady State	$V_{PWR(SUS)}$	-0.3 to 41	V
Input Pin Voltage [1]	V_{IN}	-0.3 to 7.0	V
SIN+/- COS+/- Continuous Per Output Current [2]	I_{OUTMAX}	40	mA
Storage Temperature	T_{STG}	-55 to 150	°C
Operating Junction Temperature	T_J	-40 to 150	°C

Thermal Resistance
   Junction to Ambient
   Junction to Lead
	$R_{\theta JA}$
	$R_{\theta JL}$
		60
		20
			°C/W

ESD Voltage [3]
   Human Body Model
   Machine Model
	V_{ESD1}
	V_{ESD2}
		±2000
		±200
			V

Peak Package Reflow Temperature During Reflow [4], [5]	T_{PPRT}	Note 5	°C

Notes
1. Exceeding voltage limits on Input pins may cause permanent damage to the device.
2. Output continuous output rating so long as maximum junction temperature is not exceeded. Operation at 125°C ambient temperature will require maximum output current computation using package thermal resistances.
3. ESD1 testing is performed in accordance with the Human Body Model (C_{ZAP} = 100 pF, R_{ZAP} = 1500 Ω), ESD2 testing is performed in accordance with the Machine Model (C_{ZAP} = 200 pF, R_{ZAP} = 0 Ω).
4. Pin soldering temperature limit is for 10 seconds maximum duration. Not designed for immersion soldering. Exceeding these limits may cause malfunction or permanent damage to the device.
5. Freescale's Package Reflow capability meets Pb-free requirements for JEDEC standard J-STD-020C. For Peak Package Reflow Temperature and Moisture Sensitivity Levels (MSL),
Go to www.freescale.com, search by part number [e.g. remove prefixes/suffixes and enter the core ID to view all orderable parts. (i.e. MC33xxxD enter 33xxx), and review parametrics.

STATIC ELECTRICAL CHARACTERISTICS

Table 3. Static Electrical Characteristics

Characteristics noted under conditions 4.75 V $\leq V_{DD} \leq$ 5.25 V, -40°C $\leq T_J \leq$ 150°C, GND = 0 V unless otherwise noted.
Typical values noted reflect the approximate parameter means at T_A = 25°C under nominal conditions unless otherwise noted.

Characteristic	Symbol	Min	Typ	Max	Unit
POWER INPUT					
Supply Voltage Range	V_{PWR}				V
Fully Operational		6.5	–	26	
Limited Operational [6], [7]		4.0	–	26	
V_{PWR} Supply Current	$I_{PWR(ON)}$				mA
Gauge 1 and 2 Outputs ON, No Output Loads		–	4.0	6.0	
V_{PWR} Supply Current (All Outputs Disabled)					µA
Reset = Logic [0], V_{DD} = 5.0 V	I_{PWSLP1}	–	42	60	
Reset = Logic [0], V_{DD} = 0 V	$I_{PWRSLP2}$	–	15	25	
Overvoltage Detection Level [8]	V_{PWROV}	26	32	38	V
Undervoltage Detection Level [9]	V_{PWRUV}	5.0	5.6	6.2	V
Logic Supply Voltage Range (5.0 V Nominal Supply)	V_{DD}	4.5	5.0	5.5	V
Under V_{DD} Logic Reset	V_{DDUV}	–	–	4.5	V
V_{DD} Supply Current					
Sleep: Reset Logic [0]	$I_{DD(OFF)}$	–	40	65	µA
Outputs Enabled	$I_{DD(ON)}$	–	1.0	1.8	mA
POWER OUTPUTS					
Microstep Output (Measured Across Coil Outputs) SIN0,1, ± (COS0,1, ±) (refer to **Table 1**) R_{OUT} = 200 Ω, PE6 = 0					V
Steps 6, 18 (0, 12)	V_{ST6}	4.82	5.3	6.0	
Steps 5, 7, 17, 19 (1, 11, 13, 23)	V_{ST5}	0.94 V_{ST6}	0.97 V_{ST6}	1.0 V_{ST6}	
Steps 4, 8, 16, 20 (2, 10, 14, 22)	V_{ST4}	0.84 V_{ST6}	0.87 V_{ST6}	0.96 V_{ST6}	
Steps 3, 9, 15, 21 (3, 9, 15, 21)	V_{ST3}	0.68 V_{ST6}	0.71 V_{ST6}	0.8 V_{ST6}	
Steps 2, 10, 14, 22 (4, 8,16, 20)	V_{ST2}	0.47 V_{ST6}	0.50 V_{ST6}	0.57 V_{ST6}	
Steps 1, 11, 13, 23 (5, 7, 17, 19)	V_{ST1}	0.23 V_{ST6}	0.26 V_{ST6}	0.31 V_{ST6}	
Steps 0, 12 (6, 18)	V_{ST0}	-0.1	0	0.1	

Notes
6. Outputs and logic remain active; however, the larger coil voltage levels may be clipped. The reduction in drive voltage may result in a loss of position control.
7. The logic will reset at some level below the specified Limited Operational minimum.
8. Outputs will disable and must be re-enabled via the PECCR command.
9. Outputs remain active; however, the reduction in drive voltage may result in a loss of position control.

33976

Table 3. Static Electrical Characteristics (continued)

Characteristics noted under conditions 4.75 V $\leq$ V_{DD} $\leq$ 5.25 V, -40°C $\leq$ T_J $\leq$ 150°C, GND = 0 V unless otherwise noted.
Typical values noted reflect the approximate parameter means at T_A = 25°C under nominal conditions unless otherwise noted.

Characteristic	Symbol	Min	Typ	Max	Unit
POWER OUTPUTS (CONTINUED)					
Full Step Active Output (Measured Across Coil Outputs) SIN0, 1, ± (COS0, 1, ±) (see Figure 9, page 26) Steps 1, 3 (0, 2)	V_{FS}	4.9	5.3	6.0	V
Microstep, Full Step Output (Measured from Coil Low Side to Ground) SIN0, 1, ± (COS0, 1, ±), I_{OUT} = 30 mA	V_{LS}	0	0.1	0.3	V
Output Flyback Clamp [11]	V_{FB}	–	V_{ST6} + 0.5	V_{ST6}+ 1.0	V
Output Current Limit (Output = Vst6)	I_{LIM}	40	100	170	mA
Overtemperature Shutdown [10]	OT_{SD}	155	–	180	°C
Overtemperature Hysteresis [11]	OT_{HYST}	8.0	–	16	°C
CONTROL I/O					
Input Logic High Voltage [12]	V_{IH}	2.0	–	–	V
Input Logic Low Voltage [12]	V_{IL}	–	–	0.8	V
Input Logic Voltage Hysteresis [10]	$V_{IN(HYST)}$	–	100	–	mV
Input Logic Pull Down Current (SI, SCLK)	I_{DWN}	3.0	–	20	µA
Input Logic Pull-Up Current ($\overline{CS}$, $\overline{RST}$)	I_{UP}	5.0	–	20	µA
SO High-State Output Voltage (I_{OH} = 1.0 mA)	V_{SOH}	0.8 V_{DD}	–	–	V
SO Low-State Output Voltage (I_{OL} = -1.6 mA)	V_{SOL}	–	0.2	0.4	V
SO Tri-State Leakage Current ($\overline{CS}$ $\geq$ 3.5 V)	I_{SOLK}	-5.0	0	5.0	µA
Input Capacitance [13]	C_{IN}	–	4.0	12	pF
SO Tri-State Capacitance [13]	C_{SO}	–	–	20	pF
ANALOG TO DIGITAL CONVERTER (RTZ ACCUMULATOR COUNT)					
ADC Gain [10], [14]	G_{ADC}	100	188	270	Counts/ V/ms

Notes
10. This parameter is guaranteed by design, but it is not production tested.
11. Not 100 percent tested.
12. V_{DD} = 5.0 V.
13. Capacitance not measured. This parameter is guaranteed by design, but it is not production tested.
14. Reference RTZ Accumulator (Typical) on page 23

DYNAMIC ELECTRICAL CHARACTERISTICS

Table 4. Dynamic Electrical Characteristics

Characteristics noted under conditions 4.75 V $\leq$ V$_{DD}$ $\leq$ 5.25 V, -40°C $\leq$ T$_J$ $\leq$ 150°C, GND = 0 V unless otherwise noted. Typical values noted reflect the approximate parameter means at T$_A$ = 25°C under nominal conditions unless otherwise noted.

Characteristic	Symbol	Min	Typ	Max	Units
POWER OUTPUT AND CLOCK TIMINGS					
SIN0,1, ± (COS0,1, ±) Output Turn ON Delay Time (Time from Rising $\overline{CS}$ Enabling Outputs to Steady State Coil Voltages and Currents) [15]	t$_{DLY(ON)}$	–	–	1.0	ms
SIN0,1, ± (COS0,1, ±) Output Turn OFF Delay Time (Time from Rising $\overline{CS}$ Disables Outputs to Steady State Coil Voltages and Currents) [15]	t$_{DLY(OFF)}$	–	–	1.0	ms
Uncalibrated Oscillator Cycle Time	t$_{CLU}$	0.65	1.0	1.7	μs
Calibrated Oscillator Cycle Time Calibration Pulse = 8.0 μs, PECCR D4 = Logic [0] Calibration Pulse = 8.0 μs, PECCR D4 = Logic [1]	t$_{CLC}$	1.0 0.9	1.1 1.0	1.2 1.1	μs
Maximum Pointer Speed [16]	V$_{MAX}$	–	–	400	°/s
Maximum Pointer Acceleration [16]	A$_{MAX}$	–	–	4500	°/s^2

Notes

15. Maximum specified time for the 33976 is the minimum guaranteed time needed from the microcontroller.
16. The minimum and maximum value will vary proportionally to the internal clock tolerance. These numbers are based on an ideally calibrated clock frequency of 1.0 MHz. These are not 100 percent tested.

33976

Table 4. Dynamic Electrical Characteristics (continued)

Characteristics noted under conditions 4.75 V $\leq$ V$_{DD}$ $\leq$ 5.25 V, -40°C $\leq$ T$_J$ $\leq$ 150°C, GND = 0 V unless otherwise noted. Typical values noted reflect the approximate parameter means at T$_A$ = 25°C under nominal conditions unless otherwise noted.

| Characteristic	Symbol	Min	Typ	Max	Units
SPI INTERFACE TIMING [17]					
Recommended Frequency of SPI Operation	f$_{SPI}$	–	1.0	2.0	MHz
Falling Edge of $\overline{CS}$ to Rising Edge of SCLK (Required Setup Time) [18]	t$_{LEAD}$	–	50	167	ns
Falling Edge of SCLK to Rising Edge of $\overline{CS}$ (Required Setup Time) [18]	t$_{LAG}$	–	50	167	ns
SI to Falling Edge of SCLK (Required Setup Time) [18]	t$_{SISU}$	–	25	83	ns
Required High State Duration of SCLK (Required Setup Time [18]	t$_{WSCLKH}$	–	–	167	ns
Required Low State Duration of SCLK (Required Setup Time [18]	t$_{WSCLKL}$	–	–	167	ns
Falling Edge of SCLK to SI (Required Hold Time) [18]	t$_{SI(HOLD)}$	–	25	83	ns
SO Rise Time C$_L$ = 200 pF	t$_{RSO}$	–	25	50	ns
SO Fall Time C$_L$ = 200 pF	t$_{FSO}$	–	25	50	ns
SI, $\overline{CS}$, SCLK, Incoming Signal Rise Time [19]	t$_{RSI}$	–	–	50	ns
SI, $\overline{CS}$, SCLK, Incoming Signal Fall Time [19]	t$_{FSI}$	–	–	50	ns
Falling Edge of $\overline{RST}$ to Rising Edge of $\overline{RST}$ (Required Setup Time) [18]	t$_{W\overline{RST}}$	–	–	3.0	μs
Rising Edge of $\overline{CS}$ to Falling Edge of $\overline{CS}$ (Required Setup Time) [18], [20]	t$_{\overline{CS}}$	–	–	5.0	μs
Rising Edge of $\overline{RST}$ to Falling Edge of $\overline{CS}$ (Required Setup Time) [18]	t$_{EN}$	–	–	5.0	μs
Time from Falling Edge of $\overline{CS}$ to SO Low Impedance [21]	t$_{SO(EN)}$	–	–	145	ns
Time from Rising Edge of $\overline{CS}$ to SO High Impedance [22]	t$_{SO(DIS)}$	–	1.3	4.0	μs
Time from Rising Edge of SCLK to SO Data Valid [23] 0.2 V$_{DD}$ $\leq$ SO $\geq$ 0.8 V$_{DD}$, C$_L$ = 200 pF	t$_{VALID}$	–	90	150	ns

Notes
17. The 33976 shall meet all SPI interface timing requirements specified in the SPI Interface Timing section of this table, over the specified temperature range. Digital interface timing is based on a symmetrical 50 percent duty cycle SCLK Clock Period of 333 ns. The device shall be fully functional for slower clock speeds. Reference Figure 4 and 5.
18. The maximum setup time specified for the 33976 is the minimum time needed from the microcontroller to guarantee correct operation.
19. Rise and Fall time of incoming SI, $\overline{CS}$, and SCLK signals suggested for design consideration to prevent the occurrence of double pulsing.
20. The value is for a 1.0 MHz calibrated internal clock. The value will change proportionally as the internal clock frequency changes.
21. Time required for output status data to be terminated at SO. 1.0 kΩ load on SO
22. Time required for output status data to be available for use at SO. 1.0 kΩ load on SO.
23. Time required to obtain valid data out from SO following the rise of SCLK.

TIMING DIAGRAMS
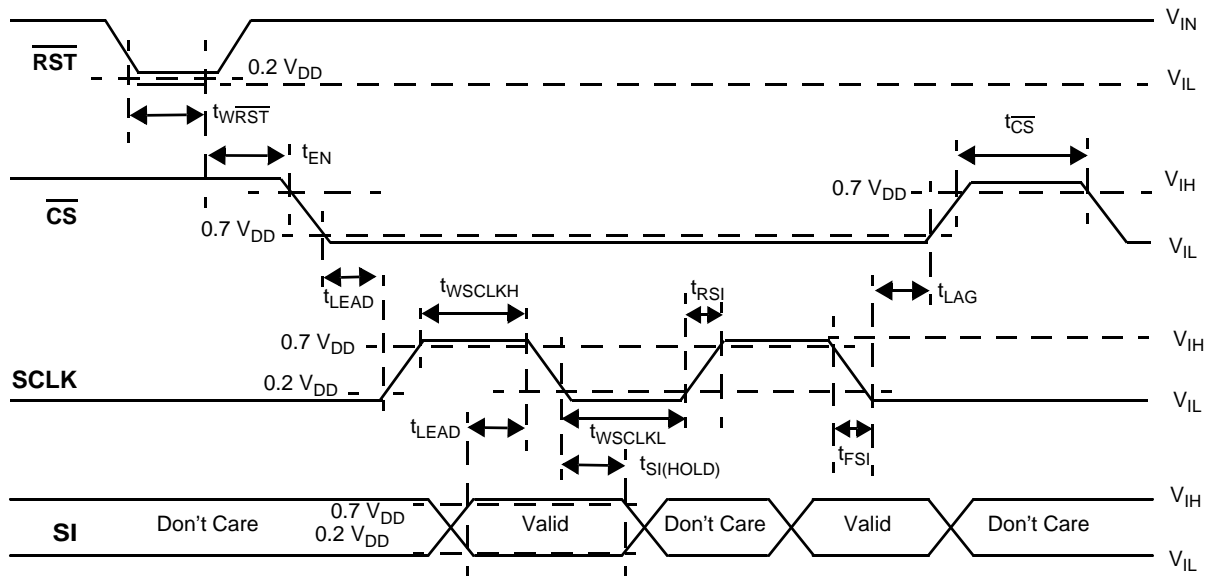
Figure 4. Input Timing Switching Characteristics
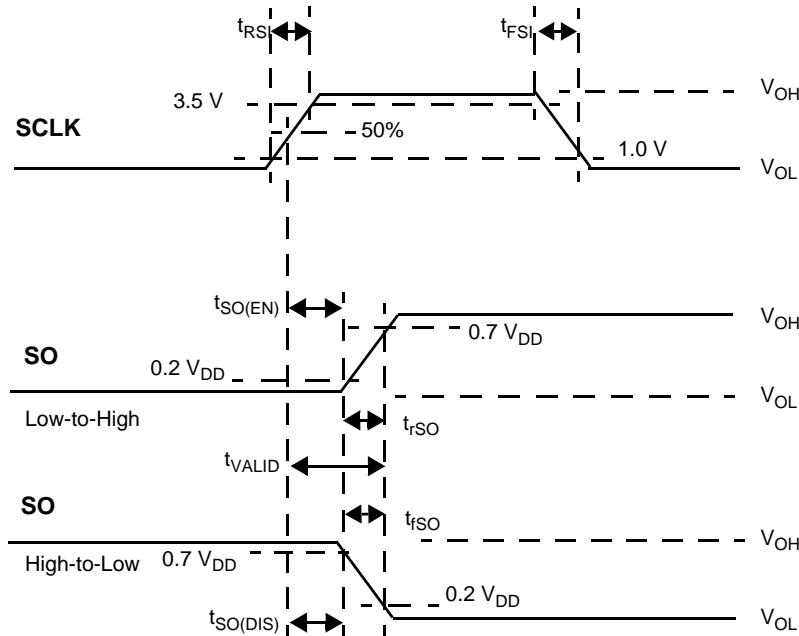
Figure 5. Valid Data Delay Time and Valid Time Waveforms

33976

FUNCTIONAL DESCRIPTION

INTRODUCTION

This 33976 is a single-packaged, Serial Peripheral Interface (SPI) controlled, dual step motor gauge driver integrated circuit (IC). This monolithic IC consists of four dual output H-Bridge coil drivers and the associated control logic. Each pair of H-Bridge drivers is used to automatically control the speed, direction, and magnitude of current through the two coils of a two-phase instrumentation step motor, similar to an MMT-licensed AFIC 6405 or a Switec MS-X15.xxx motor.

The 33976 is ideal for use in automotive instrumentation systems requiring distributed and flexible step motor gauge driving. The device also eases the transition to step motors from air core motors by emulating the air core pointer movement with little additional processor bandwidth utilization.

FUNCTIONAL PIN DESCRIPTION

H-BRIDGE OUTPUTS 0 (COS0+, COS0-, SIN0+, SIN0-)

Each pin is the output pin of a half bridge, designed to source or sink current. The H-Bridge pins linearly drive the sine and cosine coils of two separate step motors to provide four-quadrant operation.

GROUND (GND)

These pins serve as the ground for the source of the low-side output transistors as well as the logic portion of the device. They also help dissipate heat from the device.

CHIP SELECT ($\overline{CS}$)

The $\overline{CS}$ pin enables communication with the master device. When this pin is in a logic [0] state, the 33976 is capable of transferring information to, and receiving information from, the master. The 33976 latches data in from the Input Shift registers to the addressed registers on the rising edge of $\overline{CS}$. The output driver on the SO pin is enabled when $\overline{CS}$ is logic [0]. When $\overline{CS}$ is logic high, signals at the SCLK and SI pins are ignored and the SO pin is tri-stated (high impedance). $\overline{CS}$ will only be transitioned from a logic [1] state to a logic [0] state when SCLK is a logic [0]. $\overline{CS}$ has an internal pull-up (I_{UP}) connected to the pin, as specified in the section of the Static Electrical Characteristics table entitled CONTROL I/O, which is found on page 6.

SERIAL CLOCK (SCLK)

SCLK clocks the Internal Shift registers of the 33976 device. The SI pin accepts data into the Input Shift register on the falling edge of the SCLK signal, while the Serial Output pin (SO) shifts data information out of the SO Line Driver on the rising edge of the SCLK signal. It is important that the SCLK pin be in a logic [0] state whenever the $\overline{CS}$ makes any transition. SCLK has an internal pull down (I_{DWN}), as specified in the section of the Static Electrical Characteristics table entitled CONTROL I/O, which is found on page 6. When $\overline{CS}$ is logic [1], signals at the SCLK and SI pins are ignored and SO is tri-stated (high impedance). Refer to the data transfer timing diagrams in Figure 6 and Figure 7 on page 12.

SERIAL OUTPUT (SO)

The SO data pin is a tri-stateable output from the Shift register. The Status register bits are the first 16 bits shifted out. Those bits are followed by the message bits clocked in FIFO, when the device is in a daisy chain connection or being sent words that are multiples of 16 bits. Data is shifted on the rising edge of the SCLK signal. The SO pin will remain in a high impedance state until the $\overline{CS}$ pin is put into a logic low state.

SERIAL INPUT (SI)

The SI pin is the input of the SPI. Serial input information is read on the falling edge of SCLK. A 16-bit stream of serial data is required on the SI pin, beginning with the most significant bit (MSB). Messages that are not multiples of 16 bits (e.g., daisy chained device messages) are ignored. After transmitting a 16-bit word, the $\overline{CS}$ pin must be de-asserted (logic [1]) before transmitting a new word. SI information is ignored when $\overline{CS}$ is in a logic high state.

MULTIPLEXED OUTPUT (RTZ)

This is a multiplexed output pin for the non-driven coil, during a Return to Zero (RTZ) event.

VOLTAGE (V_{DD})

This SPI and logic power supply input will work with 5.0 V supplies.

RESET (RST)

If the master decides to reset the device or place it into a sleep state, the RST pin is driven to a logic [0]. A logic [0] on the $\overline{RST}$ pin will force all internal logic to the known default state. This input has an internal active pull-up.

33976

BATTERY VOLTAGE (V_{PWR})

Power supply.

H-BRIDGE OUTPUTS 1 (SIN1-, SIN1+, COS1-, COS1+)

Each of these pins is the output pin of a half bridge, designed to source or sink current. The H-Bridge pins linearly drive the sine and cosine coils of two separate step motors to provide four-quadrant operation.

33976

FUNCTIONAL DEVICE OPERATION

OPERATIONAL MODES

SPI PROTOCOL DESCRIPTION

The SPI interface has a full-duplex, three-wire synchronous, 16-bit serial synchronous interface data transfer and four I/O lines associated with it: Chip Select ($\overline{CS}$), Serial Clock (SCLK), Serial Input (SI), and Serial Output

(SO). The SI/SO pins of the 33976 follow a first in/first out (D15/D0) protocol with both input and output words transferring the most significant bit first. All inputs are compatible with 5.0 V CMOS logic levels.

TIMING DESCRIPTION

This section provides a description of the 33976 SPI behavior. To follow the explanations below, refer to **Table 5** and to the timing diagrams shown in Figure 6 and Figure 7.

Table 5. Data Transfer Timing

Pin	Description
$\overline{CS}$ (1-to-0)	SO pin is enabled.
$\overline{CS}$ (0-to-1)	33976 configuration and desired output states are transferred and executed according to the data in the Shift registers.
SO	Will change state on the rising edge of the SCLK pin signal.
SI	Will accept data on the falling edge of the SCLK pin signal.

Output shift register is loaded here.
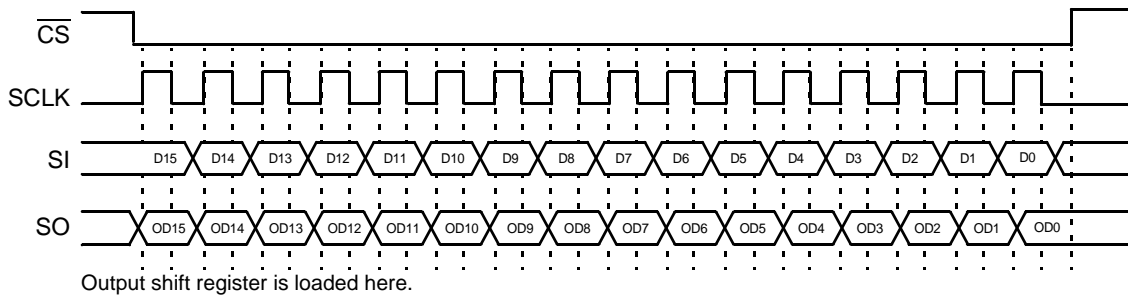
Note SO is tri-stated when $\overline{CS}$ is logic [1].

Figure 6. Single 16-Bit Word SPI Communication
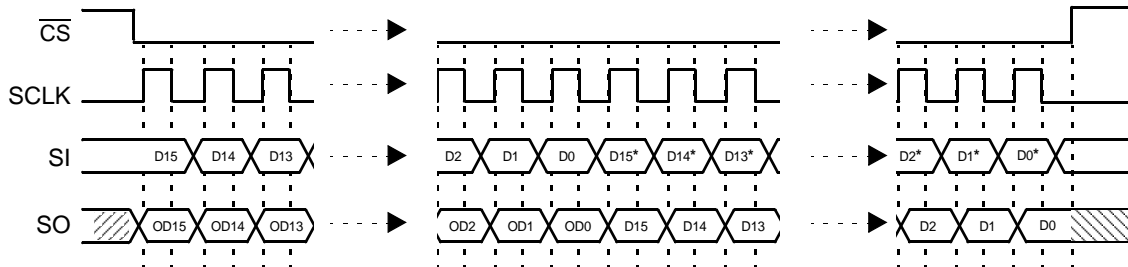
Notes 1. SO is tri-stated when $\overline{CS}$ is logic [1].
2. D15, D14, D13, ..., and D0 refer to the first 16 bits of data into the 33976.
3. D15*****, D14*****, D13*****, ..., and D0***** refer to the most recent entry of program data into the 33976.
4. OD15, OD14, OD13, ..., and OD0 refer to the first 16 bits of fault and status data out of the 33976.

Figure 7. Multiple 16-Bit Word SPI Communication

33976

DATA INPUT

The Input Shift register captures data at the falling edge of the SCLK clock. The SCLK clock pulses exactly 16 times only inside the transmission windows ($\overline{CS}$ in a logic [0] state). By the time the $\overline{CS}$ signal goes to logic [1] again, the contents of the Input Shift register are transferred to the appropriate internal register addressed in bits 15:13. The minimum time $\overline{CS}$ should be kept high depends on the internal clock speed, specified in the SPI INTERFACE TIMING [(17)] section of the Static Electrical Characteristics, found on page 6. It must be long enough so the internal clock is able to capture the data

from the Input Shift register and transfer it to the internal registers.

DATA OUTPUT

At the first rising edge of the SCLK clock, with $\overline{CS}$ at logic [0], the contents of the selected Status Word register are transferred to the Output Shift register. The first 16 bits clocked out are the status bits. If data continues to clock in before the $\overline{CS}$ transitions to a logic [1], the device begins to shift out the data previously clocked in FIFO after the $\overline{CS}$ first transitioned to logic [0].

LOGIC COMMANDS AND REGISTERS

COMMUNICATION MEMORY MAPS AND REGISTER DESCRIPTIONS

The 33976 device is capable of interfacing directly with a microcontroller via the 16-bit SPI protocol specified below. The device is controlled by the microprocessor and reports back status information via the SPI. This section provides a detailed description of all registers accessible via serial interface. The various registers control the behavior of this device.

A message is transmitted by the master beginning with the MSB (D15) and ending with the LSB (D0). Multiple messages can be transmitted in succession to accommodate those applications where daisy chaining is desirable, or to confirm transmitted data, as long as the messages are all multiples of 16 bits. Data is transferred through daisy-chained devices, as illustrated in Figure 7, page 12. If an attempt is made to latch in a message smaller than 16 bits wide, it is ignored.

Table 6 lists the seven registers the 33976 uses to configure the device, control the state of the four H-bridge outputs, and determine the type of status information that is clocked back to the master. The registers are addressed via D15:D13 of the incoming SPI word.

Table 6. Module Memory Map

Address [15:13]	Register	Name	See Page
000	Power, Enable, Calibration, and Configuration Register	PECCR	Page 13
001	Maximum Velocity Register	VELR	Page 15
010	Gauge 0 Position Register	POS0R	Page 16
011	Gauge 1 Position Register	POS1R	Page 16
100	Return to 0 Register	RTZR	Page 16
101	Return to 0 Configuration Register	RTZCR	Page 17
110	Ramp Selection Register	RMPSELR	Page 19
111	Reserved for Test	–	–

MODULE MEMORY MAP

Various registers of the 33976 SPI module are addressed by the three MSBs of the 16-bit word received serially. Functions to be controlled include:

- Individual gauge drive enabling
- Power-up/down
- Internal clock calibration
- Gauge pointer position and velocity
- Gauge pointer zeroing
- Air core motor movement emulation
- Status information

Status reporting includes:

- Individual gauge overtemperature condition
- Battery overvoltage
- Battery undervoltage
- Pointer zeroing status
- Internal clock status
- Confirmation of coil output changes that should result in pointer movement
- Real time pointer position information
- Real time pointer velocity step information
- Pointer movement direction
- Command pointer position status
- RTZ accumulator value

REGISTER DESCRIPTIONS

The following section describes the registers, their addresses, and their impact on device operation.

Address 000 — Power, Enable, Calibration, and Configuration Register (PECCR)

The Power, Enable, Calibration, and Configuration Register is illustrated in **Table 7**, page 14. A write to the 33976 using this register allows the master to (1) independently enable or disable the output drivers of the two-gauge controllers, (2) calibrate the internal clock, (3) disable the air core emulation, (4) select the direction of the pointer movement during pointer positioning and zeroing, (5) configure the device for the desired status information to

33976

be clocked out into the SO pin, or (6) send a null command for the purpose of reading the status bits. This register is also used to place the 33976 into a low current consumption mode.

Each of the gauge drivers can be enabled by writing a logic [1] to their assigned address bits, PE0 and PE1 respectively. This feature could be used to disable a driver if it is failing or is not being used. The device can be placed into a standby current mode by writing a logic [0] to both PE0 and PE1. During this state, most current consuming circuits are biased off. When in the Standby mode, the internal clock will remain ON.

The internal state machine utilizes a ROM table of step times defining the duration that the motor will spend at each microstep as it accelerates or decelerates to a commanded position. The accuracy of the acceleration and velocity of the motor is directly related to the accuracy of the internal clock. Although the accuracy of the internal clock is temperature independent, the non-calibrated tolerance is +70% to -35%. The 33976 was designed with a feature allowing the internal clock to be software calibrated to a tighter tolerance of ±10%, using the $\overline{CS}$ pin and a reference time pulse provided by the microcontroller.

Calibration of the internal clock is initiated by writing a logic [1] to PE3. The calibration pulse, which must be 8.0 μs for an internal clock speed of 1.0 MHz, will be sent on the $\overline{CS}$ pin immediately after the SPI word is sent. No other SPI lines will be toggled. A clock calibration will be allowed only if the gauges are disabled or the pointers are not moving, as indicated by status bits MOV0 and MOV1. Additional details are provided in the Internal Clock Calibration section, beginning on page 30.

Some applications may require a guaranteed maximum pointer velocity and acceleration. Guaranteeing these maximums requires that the nominal internal clock frequency fall below 1.0 MHz. The frequency range of the calibrated clock will always be below 1.0 MHz if bit PE4 is logic [0] when initiating a calibration command, followed by an 8.0 μs reference pulse. The frequency will be centered at 1.0 MHz if bit PE4 is logic [1].

Some applications may require a slower calibrated clock due to a lower motor gear reduction ratio. Writing a logic [1] to bit PE2 will slow the internal oscillator by one-third. Slowing the clock accommodates a longer calibration pulse without overrunning the internal counter—a condition designed to

generate a CAL fault indication. For example, calibration for a clock frequency of 667 kHz would require a calibration pulse of 12 μs. Unless the internal oscillator is slowed by writing PE2 to logic [1], a 12 μs calibration pulse may overrun the counter and generate a CAL fault indication.

Some applications may require faster pointer positioning than is provided with the air core motor emulation feature. Writing logic [1] to bit PE5 will disable the air core emulation for both gauges and provide an acceleration and deceleration at the maximum that the velocity position ramp can provide. If the Hold Counts need to be enabled and disabled dynamically, then the POSxR commands could also be used.

Bit PE6 must always be written as a logic [0] during all PECCR writes if the device is being used to drive an MMT style motor. Similarly, this bit must always be written as a logic [1] when being used to control Switec style motors.

The default Pointer Position 0 (PE7 = 0) will be the farthest counter-clockwise position. A logic [1] written to bit PE7 will change the location of the position 0, for the gauge selected by bit PE8, to the farthest clockwise position. A change in position 0 of only one, or both, of the two coils can be accomplished by using bits PE8 and PE7. Performing an RTZ will always move the pointer to position 0. Exercise care when writing to PECCR bits PE8 and PE7 in order to prevent accidental changes of the position 0 locations.

Bits PE11:PE8 determine the content of the bits clocked out of the SO pin. When bit PE11 is at logic [0], the clocked out bits will provide device status. If a logic [1] is written to bit PE11, the bits clocked out of the SO pin, depending upon the state of bits PE10:PE8, provides either:

- Accumulator information and detection status during the RTZ (PE10 logic [0])
- Real time pointer position location at the time $\overline{CS}$ goes low (PE10 logic [1] and PE9 logic [0]), or
- The real time step position of the pointer as described in the velocity **Table 30**, page 28 (PE10, PE9, and PE8 logic [1]).

Additional details are provided in the SO Communication section beginning on page 21.

If bit PE12 is logic [1] during a PECCR command, the state of PE11:PE0 is ignored. This is referred to as the null command and can be used to read device status without affecting device operation.

Table 7. Power, Enable, Calibration, and Configuration Register (PECCR)

Address 000												
Bits	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Read	–	–	–	–	–	–	–	–	–	–	–	–	–
Write	PE12	PE11	PE10	PE9	PE8	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0

The bits in **Table 7** are *write-only*.

PE12 (D12)—Null Command for Status Read

- 0 = Disable
- 1 = Enable

33976

PE11 (D11)—Status Select bit. This bit selects the information clocked out of the SO pin.

- 0 = Device Status (the logic states of PE10, PE9, and PE8 don't cares)
- 1 = RTZ Accumulator Value, Gauge 0 or 1 Pointer position, or Gauge 0 and 1 Velocity ramp position (depending upon the logic states of PE10, PE9, and PE8)

PE10 (D10)—RTZ Accumulator or Pointer Status Select bit. This bit is recognized only when PE11 = 1.

- 0 = RTZ Accumulator Value and status
- 1 = Pointer Position or Speed

PE9 (D9)—Pointer Position or Pointer Speed Select bit. This bit is recognized only if PE11 and PE10 = 1.

- 0 = Gauge 0 or Gauge 1 Pointer Position
- 1 = Gauge 0 and Gauge 1 Pointer Speed

PE8 (D8)—Pointer Position Gauge Select bit. Also the Position 0 of the selected gauge is determined by the PE7 selection. This bit is recognized only if PE11 and PE10 = 1 and PE9 = 0.

- 0 = Gauge 0 position
- 1 = Gauge 1 position

PE7 (D7)—Position 0 Location Select bit. This bit determines the Position 0 of the gauge selected by PE8. RTZ direction will always be to the position 0.

- 0 = Position 0 is the most CCW (counterclockwise) position
- 1 = Position 0 is the most CW (clockwise) position

PE6 (D6)—Motor Type Selection bit.

- 0 = MMT Style (coil phase difference = 90°)
- 1 = Switec Style (coil phase difference = 60°)

PE5 (D5)—Air Core Motor Emulation bit. This bit is enabled or disabled (acceleration and deceleration is constant if disabled).

- 0 = Enable
- 1 = Disable

PE4 (D4)—Clock Calibration Frequency Selector

- 0 = Maximum f =1.0 MHz (for 8.0 μs calibration pulse)
- 1 = Nominal f =1.0 MHz (for 8.0 μs calibration pulse)

PE3 (D3)—Clock Calibration Enable bit. This bit enables or disables the clock calibration.

- 0 = Disable
- 1 = Enable

PE2 (D2)—Oscillator Adjustment

- 0 = t_{CLU}
- 1 = 0.66 x t_{CLU}

PE1 (D1)—Gauge 1 Enable bit. This bit enables or disables the output driver of Gauge 1.

- 0 = Disable
- 1 = Enable

PE0 (D0)—Gauge 0 Enable bit. This bit enables or disables the output driver of Gauge 0.

- 0 = Disable
- 1 = Enable

Address 001—Maximum Velocity Register (VELR)

The Gauge Maximum Velocity Register is used to set a maximum velocity for each gauge (refer to **Table 8**). Bits V7:V0 contain a position value from 1–225 that is representative of the velocity position value described in Table 30, Velocity Table, page 28. The table value becomes the maximum velocity until it is changed to another value. If a maximum value is chosen greater than the maximum velocity in the acceleration table, the maximum table value becomes the maximum velocity. If the motor is turning at a speed greater than the new maximum, the motor immediately moves down the velocity ramp until the speed falls equal to or below it. Velocity for each motor can be changed simultaneously or independently by writing V8 and/or V9 to a logic [1]. Bits V12:V10 must be at logic [0] for valid VELR commands.

Table 8. Maximum Velocity Register (VELR)

Address 001												
Bits	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Read	–	–	–	–	–	–	–	–	–	–	–	–	–
Write	0	0	0	V9	V8	V7	V6	V5	V4	V3	V2	V1	V0

The bits in **Table 8** are *write-only*.

V12:V10 (D12:D10)—These bits must be transmitted as logic [0] for valid VELR commands

V9 (D9)—Gauge 1 Velocity. Specifies whether the maximum velocity determined in the V7: V0 field will apply to Gauge 1.

- 0 = Velocity does not apply to Gauge 1
- 1 = Velocity applies to Gauge 1

V8 (D8)—Gauge 0 Velocity. Specifies whether the maximum velocity specified in the V7: V0 field will apply to Gauge 0.

- 0 = Velocity does not apply to Gauge 0
- 1 = Velocity applies to Gauge 0

V7:V0 (D7:D0)—Maximum Velocity. Specifies the maximum velocity position from **Table 30**, page 28. This velocity will remain the maximum of the intended gauge until

33976

changed by command. Velocities can range from position 1 (00000001) to position 225 (11111111).

Addresses 010 and 011—Gauge 0/1 Position Registers (POS0R, POS1R)

SI Address 010 (Gauge 0 Position Register) and SI Address 011 (Gauge 1 Position Register) Register bits PO 11:PO0 are written to when communicating the desired pointer positions.

Commanded positions can range from 0 to 4095. The D12 bit is used to disable the damping (i.e., hold counts) for each respective gauge. This feature allows the user to easily turn on and off the damping that was configured with the RMPSELR. Disabling the hold counts allows the pointer to decelerate to the commanded position, as fast as possible down the velocity ramp. When disabled, the acceleration and deceleration of the pointer are symmetrical.

Table 9. Gauge 0 Position Register (POS0R)

Address 010												
Bits	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Read	–	–	–	–	–	–	–	–	–	–	–	–	–
Write	HE012	P011	P010	P09	P08	P07	P06	P05	P04	P03	P02	P01	P00

The bits in **Table 9** are *write-only*.

HE0 12 (D12)—This bit is used to disable the damping (i.e., hold counts) for Gauge 0 (1 = Damping disabled; 0 = Damping enabled).

P011:P00 (D11:D0)—Desired pointer position of Gauge 0. Pointer positions can range from 0 (000000000000) to position 4095 (111111111111). For a step motor requiring 12 microsteps per degree of pointer movement, the maximum pointer sweep is 341.25°.

.
Table 10. Gauge 1 Position Register (POS1R)

Address 011												
Bits	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Read	–	–	–	–	–	–	–	–	–	–	–	–	–
Write	HE112	P111	P110	P19	P18	P17	P16	P15	P14	P13	P12	P11	P10

The bits in **Table 10** are *write-only*.

HE1 12 (D12)—This bit is used to disable the damping (i.e., hold counts) for Gauge 1 (1=Damping disabled; 0=Damping enabled).

P1 11:P1 0 (D11:D0)—Desired pointer position of Gauge 1. Pointer positions can range from 0 (000000000000) to position 4095 (111111111111). For a step motor requiring 12 microsteps per degree of pointer movement, the maximum pointer sweep is 341.25° (4095 ÷ 12).

Address 100—Gauge Return to Zero Register (RTZR)

Gauge Return to Zero Register (RTZR) (refer to **Table 11**, page) is written to return the gauge pointers to the zero position. During an RTZ event, the pointer is returned to zero using full steps, where only one coil is driven at any point in time. The back electromotive force (EMF) signal present on

the non-driven coil is integrated and its results are stored in an accumulator.

A logic [1] written to bit RZ1 enables a Return to Zero for Gauge 0 if RZ0 is logic [0], and Gauge 1 if RZ0 is logic [1], respectively. Similarly, a logic [0] written to bit RZ1 disables a Return to Zero for Gauge 0 when RZ0 is logic [0], and Gauge 1 when RZ0 is logic [1], respectively.

Bits D12:D5 and D3:D2 must be at logic [0] for valid RTZR commands.

Bit RZ4 is used to enable an unconditional RTZ event. A logic [0] results in a typical RTZ event, automatically providing a Stop when a stall condition is detected. A logic [1] will result in RTZ movement, causing a Stop if a logic [0] is written to bit RZ0. This feature is useful during development and characterization of RTZ requirements.

33976

Table 11. Return to Zero Register (RTZR)

Bits	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
					Address 100								
Read	–	–	–	–	–	–	–	–	–	–	–	–	–
Write	0	0	0	0	0	0	0	0	RZ4	0	RZ2	RZ1	RZ0

The register bits in **Table 11** are *write-only*.

RZ12:RZ5 (D12:D5)—These bits must be transmitted as logic [0] for valid commands.

RZ4 (D4)—This bit is used to enable an unconditional RTZ event.

- 0 = Automatic Return to Zero
- 1 = Unconditional Return to Zero

RZ3 (D3)—This bit must be transmitted as logic [0] for valid commands.

RZ2 (D2)—Return to Zero Direction bit. This bit is used to properly sequence the integrator, depending upon the desired zeroing direction.

- 0 = Return to Zero will occur in the CCW direction (PE7 = 0)
- 1 = Return to Zero will occur in the CW direction (PE7 = 1)

RZ1 (D1)—Return to Zero Direction. This bit commands the selected gauge to return the pointer to zero position.

- 0 = Return to Zero Disabled
- 1 = Return to Zero Enabled

RZ0 (D0)—Gauge Select: Gauge 0/Gauge 1. This bit selects the gauge to be commanded.

- 0 = Selects Gauge 0
- 1 = Selects Gauge 1

Address 101—Gauge Return to Zero Configuration Register

Gauge Return to Zero Configuration Register (RTZCR) is used to configure the Return to Zero Event (refer to **Table 12**, page 18). It is written to modify (1) the step time, or rate at which the pointer moves during an RTZ event, (2) the integration blanking time, which is the time immediately following the transition of a coil from a driven state to an open state in the RTZ mode, and (3) the threshold of the RTZ integration register.

The values used for this register should be selected during development to optimize the RTZ for each application. Selecting an RTZ step rate resulting in consistently successful zero detections depends on a clear understanding of the motor characteristics. Specifically, resonant frequencies exist due to the interaction between the motor and the pointer. This command allows movement of the RTZ pointer speed away from these frequencies. Also, some motors require a significant amount of time for the pointer to settle to a steady state position when moving from one full

step position to the next. Consistent and accurate integration values require the pointer be stationary at the end of the full step time.

Bits RC3:RC0, RC12:RC11, and RC4 determine the time spent at each full step during an RTZ event. Bits RC3:RC0 are used to select a Δt ranging from 0 ms (0000) to 61.44 ms (1111) in increments of 4.096 ms (refer to **Table 13**, page 18). The Δt is multiplied by the factor M, which is defined by bits RC12:RC11. The product is then added to the blanking time, selected using bit RC4, to generate the full step time. The multiplier selected with RC12:RC11 will be 1 (00), 2 (01), or 4 (10) as illustrated in the equations below. The multiplier selected with RC12:RC11 will be 1 (00), 2 (01), or 4 (10) as illustrated in the equations below. Note that the RC12:RC11 value of 8 (11) is not recommended for use in a product design application, because of the potential for an RTZ accumulator internal overflow, due to the long time step. The blanking time that is selected with bit RC4 determines the time that is provided immediately following a full step change, before enabling the integration of the non-driven coil signal. The blanking time is either 512 μs when RC4 is logic [0], or 768 μs when it is logic [1].The full step time is generated using the following equations:

When D3:D0 (RC3:RC0) $\neq$ 0000

Full Step (t) = Δt x M + blanking (t)   (1)

When D3:D0 (RC3:RC0) = 0000

Full Step (t) = blanking (t) + 2.048 ms (2)

Note In equation (2), a 2.048 ms offset is added to the full step time when the RC3:RC0 = 0000. The full step time default value after a logic reset is 12.80 ms (RC12:RC11 = 00, RC4 = 0, and RC3:RC0 = 0011).

If there are two full steps per degree of pointer movement, the pointer speed is 1/(FullStep x 2) deg/s.

Detecting pointer movement is accomplished by integrating the EMF present in the non-driven coil during the RTZ event. The integration circuitry is implemented using a Sigma-Delta converter resulting in the placement of a value in the 15-bit RTZ accumulator at the end of each full step. The value in the RTZ accumulator represents the change in flux and is compared to a threshold. Values above the threshold indicate a pointer is moving. Values below the threshold indicate a stalled pointer, thereby resulting in the cessation of the RTZ event.

The RTZ accumulator bits are signed and represented in two's complement. After a full step of integration, a sign bit of

33976

0 is the indicator of an accumulator exceeding the decision threshold of 0, and the pointer is assumed to still be moving. Similarly, if the sign bit is logic [1] after a full step of integration, the accumulator value is negative and the pointer is assumed to be stopped. The integrator and accumulator are initialized after each full step. If the PECCR command is written to clock out the RTZ accumulator values via the SO, the OD14 bit corresponds to the sign bit of the RTZ accumulator.

Accurate pointer stall detection depends on a correctly preloaded accumulator for specific gauge, pointer, and full step combinations. Bits RC10:RC5 are used to offset the initial RTZ accumulator value, properly detecting a stalled

motor. The initial accumulator value at the start of a full step of integration is negative. If the accumulator was correctly preloaded, a free-moving pointer will result in a positive value at the end of the integration time, and a stalled pointer will result in a negative value. The preloaded values associated with each combination of bits RC10:RC5 are illustrated in **Table 14**, page 19. The accumulator should be loaded with a value resulting in an accumulator MSB to a logic [1] when the motor is stalled. For the default mode, after a power-up or any reset, the 33976 device sets the accumulator value to -1, resulting in an unconditional RTZ pointer movement until it is increased.

.
Table 12. RTZCR SI Register Assignment

Address 101													
Bits	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Read	–	–	–	–	–	–	–	–	–	–	–	–	–
Write	RC12	RC11	RC10	RC9	RC8	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0

The bits in **Table 12** are *write-only*.

RC12:RC11 (D12:D11) — These bits, along with RC3:RC0 (D3:D0) and RC4 (D4), determine the full step time and, therefore, the rate at which the pointer will move during an RTZ event. The values of D12:D11 determine the multiplier (M) used in equation (1) (refer to page 17).

RC12:RC11 = M; default value = 00

- 00 = 1
- 01 = 2
- 10 = 4
- 11 = 8 (Not to be used for design)

RC10:RC5 (D10:D5) — These bits determine the value preloaded into the RTZ integration accumulator to adjust the detection threshold. Values range from -1 (00000000) to -1009 (11111111) as shown in **Table 14**, the default value = 000000.

RC4 (D4) — This bit determines the RTZ blanking time (*blanking* (*t*)). The default value = 0

- 0 = 512 μs
- 1 = 768 μs

RC3:RC0 (D3:D0) — These bits, along with RC12:RC11 (D12:D11) and RC4 (D4), determine the time variables used to calculate the full step times with equations (1) or (2) illustrated above. RC3:RC0 determines the Δt time. The Δt values range from 0 (0000) to 61.440 ms (1111) and are shown in **Table 13**. The default Δt is 0 (0011).

Note Equation (2) (refer to page 17) is only used to calculate the full step time if RC3:RC0 = 0000. Use equation (1) for all other combinations of RC3:RC0.

Table 13. RTZCR Full Step Time

RC3	RC2	RC1	RC0	Δt (ms)
0	0	0	0	0
0	0	0	1	4.096
0	0	1	0	8.192
0	0	1	1	12.288
0	1	0	0	16.384
0	1	0	1	20.480
0	1	1	0	24.576
0	1	1	1	28.672
1	0	0	0	32.768
1	0	0	1	36.864
1	0	1	0	40.960
1	0	1	1	45.056
1	1	0	0	49.152
1	1	0	1	53.248
1	1	1	0	57.344
1	1	1	1	61.440

33976

Table 14. RTZCR Accumulator Offset

RC10	RC9	RC8	RC7	RC6	RC5	Preload Value (PV)	Initial Accumulator Value = (-16 x PV) -1
0	0	0	0	0	0	0	-1
0	0	0	0	0	1	1	-17
0	0	0	0	1	0	2	-33
0	0	0	0	1	1	3	-49
0	0	0	1	0	0	4	-65
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
1	1	1	1	1	1	63	-1009

Address 110 — Ramp Selection Register (RMPSELR)

SI Address 110 Ramp Selection Register (RMPSELR) (refer to **Table 15**, page 20). A write to the 33976 using this register allows the master to independently modify the pointer movement response characteristics of each gauge driver. The user has three variables that can be configured, during the initialization of the device, to provide quick and responsive pointers (e.g., tachometer applications) or soft landing and less responsive pointers (e.g., speedometer or fuel indicators). These three variables are (1) the ramp zero selection RS (RS3:RS0), (2) the hold count cut-in location offset variable HCP (HCP2: HCP0), and (3) the hold count value HC, (HC3:HC0). Each of these variables is described below and an implementation example is shown in Figure 11, page 31.

The state machine uses the velocity ramp (refer to **Table 30**, page 28) to control the acceleration, deceleration and speed of the pointer movement. During an acceleration from a stopped position, the state machine will microstep the pointer at each velocity step, starting with step 0, in succession until the desired pointer speed is reached. Similarly, as the pointer approaches the commanded position, the state machine will microstep the motor at successive velocity steps down the velocity ramp until reaching step 0. The fastest that a pointer can accelerate, decelerate or change directions is limited by the velocity ramp.

For example, if a pointer is moving in the clockwise direction and is commanded to a position that is counter clockwise from the current pointer location, then the state machine must first decelerate the pointer down the ramp to the step 0 location, change directions and then accelerate up the ramp towards the commanded location. In this situation, the state machine will force movement down and then up the ramp as fast as possible by stepping at each Velocity Position only once for a direction change. The low velocity steps (e.g., Velocity Position 1 is 27 ms) are significant in that they can limit the speed with which a pointer can change direction.

Bits RS3:RS0 of the RMPSELR are used to truncate as many as 15 velocity steps off of the bottom of the velocity ramp. The value of RS determines the Initial Velocity Ramp Position:

Initial Velocity Position = RS + 1

For example, writing a value of 4 to these bits truncates the velocity ramp by 4 and would result in a first and last velocity step of 5.86ms (Velocity Position 5). A pointer will change directions much faster with this abbreviated ramp than it would if using the default ramp with a Velocity Position 1 of 27 ms

Most applications require a smooth dynamic pointer as the commanded position is constantly updated. Movement along the ramp at the maximum acceleration and deceleration (only one step at each velocity position) results in a choppy movement because the movement velocity range is large for small changes in position as the pointer quickly reaches commanded locations from command to command. Configuring the state machine to repeat velocity steps at several of the last few step locations, when the pointer decelerates to the commanded location, can eliminate this choppy movement. These repetitive steps are referred to as hold counts.

Bits HCP2: HCP0 of the RMPSELR determine the velocity step location at which the hold counts begin during a deceleration to the commanded position. The value written to HCP2: HCP0 (HCP) is multiplied by 8 and added to the RS value. The result is the first velocity position, or the Hold Count Cut-In Point, to which the hold counts will apply during a deceleration.

First Velocity Position w/ Hold Counts = HCP x 8 + RS

The exception to this is when the HCP2: HCP0 value is 000. In this case, HCP=8 and the cut-in point will be 64 steps above the RS value. The default value of the HCP=2 or a hold count cut-in point of 16 velocity steps above the RS value.

33976

The value of RS also determines the last velocity position step for which the Hold Counts are applied:

Last Velocity Position w/ Hold Counts = RS + 2

The number of hold counts per applicable velocity step is determined by the value written to HC3:HC0 (HC) and can range from 0 to 15 steps. This number of hold counts will be applied to each step below the Hold Count Cut-In as determined by HCP and RS. The default value of HC is 5.

Note: the following relationship between the variables must be adhered to for the state machine to work properly:

HC x (HCP x 8 - 1) + (225 - RS) < 512

Therefore, if RS = 0 and the Hold Count Cut-In point is 64, the largest value of Hold Counts you can choose is 4.

4 * (64 - 1) + (225 - 0) = 477

The GSEL bit determines which of the two gauges the rest of the RMPSELR bits are applied to. A GSEL bit set to logic 1 will apply the RMPSELR data to Gauge 1 and, Logic 0 to Gauge 0, respectively. Configuring both gauges requires two writes to this register.

Table 15. Ramp Selection Register (RMPSELR)

Address 101												
Bits	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Read	–	–	–	–	–	–	–	–	–	–	–	–	–
Write	GSEL12	GSEL11	HCP2	HCP1	HCP0	HC3	HC2	HC1	HC0	RS3	RS2	RS1	RS0

The bits in **Table 15** are *write-only*.

GSEL12 (D12)— Gauge Select bit. The value of this bit determines the gauge for which the settings apply (refer to page 17):

- 1 = Gauge 1
- 0 = Gauge 0

GSEL11 (D11)—This bit must be transmitted as Logic 0 for valid commands.

HCP2:HPC0 (D10:D8)—Hold Count Cut-in Point variable. These bits determine HCP, which is then multiplied by 8, and added to the RS number, to determine the actual Hold Count Cut-In Step value. The values of HCP range from 1 to 8 as shown in **Table 16**. The default value is 2.

.
Table 16. First Hold Count Velocity Position

HCP2	HCP1	HCP0	Velocity Step (HCP x 8 + RS)
0	0	0	64 + RS
0	0	1	8 + RS
0	1	0	16 + RS
0	1	1	24 + RS
1	0	0	32 + RS
1	0	1	40 + RS
1	1	0	48 + RS
1	1	1	56 + RS

HC3:HC0 (D7:D4)—These bits determine the number of Hold Counts that will be applied to the steps that are determined by the HCP2:HCP0 and RS3:RS0 bits. The HC values range from 0 to 15 and are shown in **Table 17**. The default value is 5.

Table 17. Hold Counts Per Step

HC3	HC2	HC1	HC0	Hold Counts / Step (HC)
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

RS3:RS0 (D3:D0)—These bits determine the number of velocity steps that are truncated from the Velocity Position ramp. The values range from 0 to 15 and are shown in **Table 18**. The default value is 0.

33976

Table 18. Truncated Velocity Steps

RS3	RS2	RS1	RS0	Zero Velocity Position # (RS)
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

SO Communication

When the $\overline{CS}$ pin is pulled low, the internal status register, as configured with the PECCR command bits PE11:PE8, is loaded into the output register and the data is clocked out MSB (OD15) first. Following a $\overline{CS}$ transition 0 to 1, the device determines if the shifted-in message was of a valid length (a valid message length is one that is greater than 0 bits and a multiple of 16 bits) and, if so, latches the incoming data into the appropriate registers.

At this time, the SO pin is tri-stated and the status register is now able to accept new status information. Fault status information will be latched and held until the Device Status

Output register is selected and it is clocked out via the SO. If the message length was determined to be invalid, the fault information will not be cleared and will be transmitted again during the next valid SPI message. Pointer status information bits (e.g., pointer position, velocity, and commanded position status) will always reflect the real time state of the pointer.

Any bits clocked out of the SO pin after the first 16 are representative of the initial message bits clocked into the SI pin since the $\overline{CS}$ pin first transitioned to a logic [0]. This feature is useful for daisy-chaining devices as well as message verification.

As described above, the last valid write to bits PE11:PE8 of the PECCR command determines the nature of the status data that is clocked out of the SO pin.

There are five different types of status information available:

1. Device Status (refer to Table 20, page 21)
2. RTZ Accumulator Status (refer to Table 22, page 23)
3. Gauge 0 Pointer Position Status (refer to Table 24, page 24)
4. Gauge 1 Pointer Position Status (refer to Table 26, page 24)
5. Gauge 1 and 2 Pointer Velocity Status (refer to Table 28, page 24)

Once a specific status type is selected, it will not change until either the PECCR command bits PE11:PE8 (D11:D8) are written to select another or the device is reset. Each of the Status types and the PECCR bit necessary to select them are described in the following paragraphs.

Device Status Information

Most recent valid PECCR command resulting in the Device Status output:

Table 19.

D11	D10	D9	D8
0	x	x	x

x = Don't care.

Table 20. Device Status Output Register

Bits	OD15	OD14	OD13	OD12	OD11	OD10	OD9	OD8	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0
Read	DIR1	DIR0	0POS1	0POS0	CMD1	CMD0	OV	UV	CAL	OVUV	MOV1	MOV0	RTZ1	RTZ0	OT1	OT0
Write	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

The bits in Table 20 are *read-only* bits.

DIR1 (OD15)—This bit indicates the direction Gauge 1 pointer is moving.

- 0 = Toward position 0
- 1 = Away from position 0

DIR0 (OD14)—This bit indicates the direction Gauge 0 pointer is moving.

- 0 = Toward position 0
- 1 = Away from position 0

33976

0POS1 (OD13)—This bit indicates the configured Position 0 for Gauge 1.

- •0 = Farthest CCW
- •1 = Farthest CW

0POS0 (OD12)—This bit indicates the configured Position 0 for Gauge 0.

- 0 = Farthest CCW
- 1 = Farthest CW

CMD1 (OD11)—This bit indicates whether Gauge 1 is at the most recently commanded position.

- 0 = At commanded position
- 1 = Not at commanded position

CMD0 (OD10)—This bit indicates whether Gauge 0 is at the most recently commanded position.

- 0 = At commanded position
- 1 = Not at commanded position

OV (OD9)—Overvoltage Indication. A logic [1] on this bit indicates V_{PWR} voltage exceeded the upper limit of V_{PWROV} since the last SPI communication (refer to the Static Electrical Characteristics table under POWER INPUT, page 5). An overvoltage event will automatically disable the driver outputs. Because the pointer may not be in the expected position, the master may want to re-calibrate the pointer position with an RTZ command after the voltage returns to a normal level. For an overvoltage event, both gauges must be re-enabled as quickly as this flag returns to logic [0]. The state machine will continue to operate properly as long as V_{DD} is within the normal range.

- 0 = Normal range
- 1 = Battery voltage exceeded V_{PWROV}

UV (OD8)—Undervoltage Indication. A logic [1] on this bit indicates the V_{PWR} voltage fell below V_{PWRUV} since the last SPI communication (refer to the Static Electrical Characteristics table under POWER INPUT, page 5). An undervoltage event is just flagged; however, at some voltage level below 4.0 V, the outputs turn OFF and the state machine resets. Because the pointer may not be in the expected position, the master may want to re-calibrate the pointer position with an RTZ command after the voltage returns to a normal level. For an undervoltage event, both gauges may need to be re-enabled as quickly as this flag returns to logic [0]. The state machine will continue to operate properly as long as V_{DD} is within the normal range.

- 0 = Normal range
- 1 = Battery voltage fell below V_{PWRUV}

CAL (OD7)—Calibrated Clock out of Specification. A logic [1] on this bit indicates the clock count calibrated to a value outside the expected range given the tolerance specified by t_{CLC} in the Dynamic Electrical Characteristics table under POWER OUTPUT AND CLOCK TIMINGS, page 7.

- 0 = Clock within spec
- 1 = Clock out of spec

OVUV (OD6)—Undervoltage or Overvoltage Indication. A logic [1] on this bit indicates the V_{PWR} voltage fell to a level below the V_{PWRUV} since the last SPI communication (refer to the Static Electrical Characteristics table under POWER INPUT, page 5). An undervoltage event is just flagged, while an overvoltage event automatically disables the drive outputs. Because the pointer may not be in the expected position, the master may want to re-calibrate the pointer with an RTZ command after the voltage returns to normal level. For an overvoltage event, both gauges must be re-enabled as soon as this flag returns to logic [0]. The state machine will continue to operate properly as long as V_{DD} is within the normal range.

- 0 = Normal range
- 1 = Battery voltage fell below V_{PWRUV} or exceeded V_{PWROV}

MOV1 (OD5)—This bit identifies Gauge 1 movement since last SPI communication. A logic [1] on this bit indicates the Gauge 1 pointer position changed since the last SPI command. This information allows the master to confirm the pointer is moving as commanded. This bit may also be used to determine if Gauge 1 is enabled or disabled.

- 0 = Gauge 1 position has not changed since the last SPI command
- 1 = Gauge 1 pointer position has changed since the last SPI command

MOV0 (OD4)—Gauge 0 Movement Since last SPI Communication. A logic [1] on this bit indicates the Gauge 0 pointer position has changed since the last SPI command. This information allows the master to confirm the pointer is moving as commanded. This bit may also be used to determine if Gauge 0 is enabled or disabled.

- 0 = Gauge 0 position has not changed since the last SPI command
- 1 = Gauge 0 pointer position has changed since the last SPI command

RTZ1 (OD3)—RTZ1 Is Enabled or Disabled. A logic [1] on this bit indicates Gauge 1 is in the process of returning to the zero position as requested with the RTZ command. This bit will continue to indicate a logic [1] until the SPI message following a detection of the zero position, or the RTZ feature is commanded OFF using the RTZ message.

- 0 = Return to Zero disabled
- 1 = Return to Zero enabled successfully

RTZ0 (OD2)—RTZ0 Is Enabled or Disabled. A logic [1] on this bit indicates Gauge 0 is in the process of returning to the zero position as requested with the RTZ command. This bit continues to indicate a logic [1] until the SPI message following a detection of the zero position, or the RTZ feature is commanded OFF using the RTZ message.

- 0 = Return to Zero disabled
- 1 = Return to Zero enabled successfully

OT1 (OD1)—Gauge 1 Junction Overtemperature. A logic [1] on this bit indicates that the coil drive circuitry dedicated to drive Gauge 1 has exceeded the maximum allowable junction temperature since the last SPI communication and that Gauge 1 has been disabled. It is recommended that the pointer be re-calibrated using the RTZ

33976

command after re-enabling the gauge using the PECCR command. This bit remains logic [1] until the gauge is enabled.

- 0 = Temperature within range
- 1 = Gauge 1 maximum allowable junction temperature condition has been reached

OT0 (OD0)—Gauge 0 Junction Overtemperature. A logic [1] on this bit indicates that the coil drive circuitry dedicated to drive Gauge 0 has exceeded the maximum allowable junction temperature since the last SPI communication and that Gauge 0 has been disabled. It is recommended that the pointer be re-calibrated using the RTZ command after re-enabling the gauge using the PECCR

command. This bit remains logic [1] until the gauge is re-enabled.

- •0 = Temperature within range1 = Gauge 0 maximum allowable junction temperature condition is reached
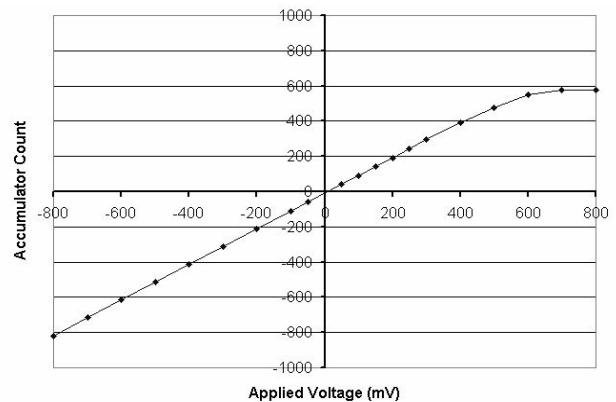
RTZ Accumulator Status Information

Most recent valid PECCR command resulting in the RTZ Accumulator status output:

Table 21.

D11	D10	D9	D8
1	0	x	x

x = Don't care.

Table 22. RTZ Accumulator Status Output Register

Bits	OD15	OD14	OD13	OD12	OD11	OD10	OD9	OD8	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0
Read	RTZ	ACC14	ACC13	ACC12	ACC11	ACC10	ACC9	ACC8	ACC7	ACC6	ACC5	ACC4	ACC3	AC2C	ACC1	ACC0
Write	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

The bits in Table 22 are *read-only* bits.

RTZ (OD15)—RTZ Bit Is Enabled or Disabled. A logic [1] on this bit indicates that the Gauge is in the process of returning to the zero position as requested with the RTZ command. This bit will continue to indicate a logic [1] until the SPI message following a detection of the zero position, or the RTZ feature is commanded OFF using the RTZ message.

- 0 = Return to Zero disabled
- 1 = Return to Zero enabled successfully

ACC14:ACC0 (OD14:OD0)—These 15 bits are from the RTZ accumulator. They represent the integrated signal present on the non-driven coil during an RTZ event. These bits are logic [0] after power-on reset, or after the RST pin transitions from logic [0] to [1]. After an RTZ event, they will represent the last RTZ accumulator result before the RTZ was stopped. ACC14 is the MSB and is the sign bit used for zero detection. Negative numbers have MSB logic [1] and are coded in two's complement.

Figure 8. RTZ Accumulator (Typical)

The analog-to-digital converter's linear input range covers the expected magnitude of motor back e.m.f. signals, which is usually less than 500mV. Input signals greater than this will not cause any damage (the circuit is connected to the motor H-Bridge drivers, and thus is exposed to the full magnitude of the drive voltages), but may cause some small loss of linearity. A typical plot of output vs. input is shown in Figure 8 for 4ms step times.

Gauge 0 Pointer Position Status Information

Most recent valid PECCR command resulting in the Gauge 0 Pointer Position status output:

Table 23.

D11	D10	D9	D8
1	1	0	0

33976

Table 24. Gauge 0 Pointer Position Status Output Register

Bits	OD15	OD14	OD13	OD12	OD11	OD10	OD9	OD8	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0
Read	ENB0	DIR0	DIRC0	CMD0	POS11	POS10	POS9	POS8	POS7	POS6	POS5	POS4	POS3	POS2	POS1	POS0
Write	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

The bits in Table 24 are *read-only* bits.

ENB0 (OD15)—This bit indicates whether Gauge 0 is enabled.

- 0 = Disabled
- 1 = Enabled

DIR0 (OD14)—This bit indicates the direction Gauge 0 is moving.

- 0 = Toward position 0
- 1 = Away from position 0

DIRC0 (OD13)—This bit is used to determine whether the direction of the most recent pointer movement is toward the last commanded position or away from it.

- 0 = Direction of the pointer movement is toward the commanded position
- 1 = Direction of the pointer movement is away from the commanded position

CMD0 (OD12)—This bit indicates whether Gauge 0 is at the most recently commanded position.

- 0 = At commanded position
- 1 = Not at commanded position

POS11:POS0 (OD11:OD0)—These 12 bits represent the actual position of the pointer at the time $\overline{CS}$ transitions to a logic [0].

Gauge 1 Pointer Position Status Information

Most recent valid PECCR command resulting in the Gauge 1 Pointer Velocity status output:

Table 25.

D11	D10	D9	D8
1	1	0	1

Table 26. Gauge 1 Pointer Position Status Output Register

Bits	OD15	OD14	OD13	OD12	OD11	OD10	OD9	OD8	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0
Read	ENB1	DIR1	DIRC1	CMD1	POS11	POS10	POS9	POS8	POS7	POS6	POS5	POS4	POS3	POS2	POS1	POS0
Write	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

The bits in Table 26 are *read-only* bits.

ENB1 (OD15)—This bit indicates if Gauge 1 is enabled.

- 0 = Disabled
- 1 = Enabled

DIR1 (OD14)—This bit indicates the direction Gauge 1 pointer is moving.

- 0 = Toward position 0
- 1 = Away from position 0

DIRC1 (OD13)—This bit determines if the direction of the most recent pointer movement is toward, or away from, the last commanded position.

- 0 = Direction of the pointer movement is toward the commanded position
- 1 = Direction of the pointer movement is away from the commanded position

CMD1 (OD12)—This bit indicates if Gauge 1 is at the most recently commanded position.

- 0 = At commanded position
- 1 = Not at commanded position

POS11:POS0 (OD11:OD0)—These 12 bits represent the actual position of the pointer at the time $\overline{CS}$ transitions to a logic [0].

Gauge 0 and 1 Pointer Velocity Status Information

Most recent valid PECCR command resulting in the Gauge 0 and 1 Pointer Velocity status output:

Table 27.

D11	D10	D9	D8
1	1	1	x

x = Don't care.

33976

Table 28. Gauge 0 and 1 Pointer Velocity Status Output Register

Bits	OD15	OD14	OD13	OD12	OD11	OD10	OD9	OD8	OD7	OD6	OD5	OD4	OD3	OD21	OD1	OD0
Read	1V7	1V6	1V5	1V4	1V3	1V2	1V1	1V0	0V7	0V6	0V5	0V4	0V3	0V2	0V1	0V0
Write	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

The bits in Table 28 are *read-only* bits.

1V7:1V0 (OD15:OD8)—These 8 bits represent the step table value that indicates the actual velocity step location (refer to **Table 30**, page 28) of the Gauge 1 pointer at the time that the $\overline{CS}$ transitions to a logic [0].

Note For both sets of bits,1V7:1V0 and 0V7:0V0, if the ramp is truncated with the RMPSELR, the velocity position step that will be read when the pointer is no longer moving will be the velocity position that identifies it in the untruncated ramp (e.g., if RS = 2, then the velocity step location will be 3 when the pointer is at the commanded position).

0V7:0V0 (OD7:OD0)—These 8 bits represent the step table value that indicates the actual velocity step location (refer to **Table 30**) of the Gauge 0 pointer at the time that the $\overline{CS}$ transitions to a logic [0].

STATE MACHINE OPERATION

The two-phase step motor has maximum allowable velocities and acceleration and deceleration.The purpose of the step motor state machine is to drive the motor with maximum performance while remaining within the motor's voltage, velocity, and acceleration constraints.

A requirement of the state machine is to ensure the deceleration phase begins at the correct time and pointer position. When commanded, the motor will accelerate constantly to the maximum velocity, then move toward the commanded position. Eventually, the pointer will reach the calculated location where the movement has to decelerate, slowing safely to a stop at the desired position. During the deceleration phase, the motor will not exceed the maximum deceleration.

During normal operation, both step motor rotors are microstepped with 24 steps per electrical revolution (see Figure 9). A complete electrical revolution results in two degrees of pointer movement. There is a second (smaller) state machine in the IC controlling these microsteps. This state machine receives *clockwise* or *counter-clockwise* index commands at intervals, stepping the motor in the appropriate direction by adjusting the current in each coil. Normalized values are provided in **Table 29**, page 26.
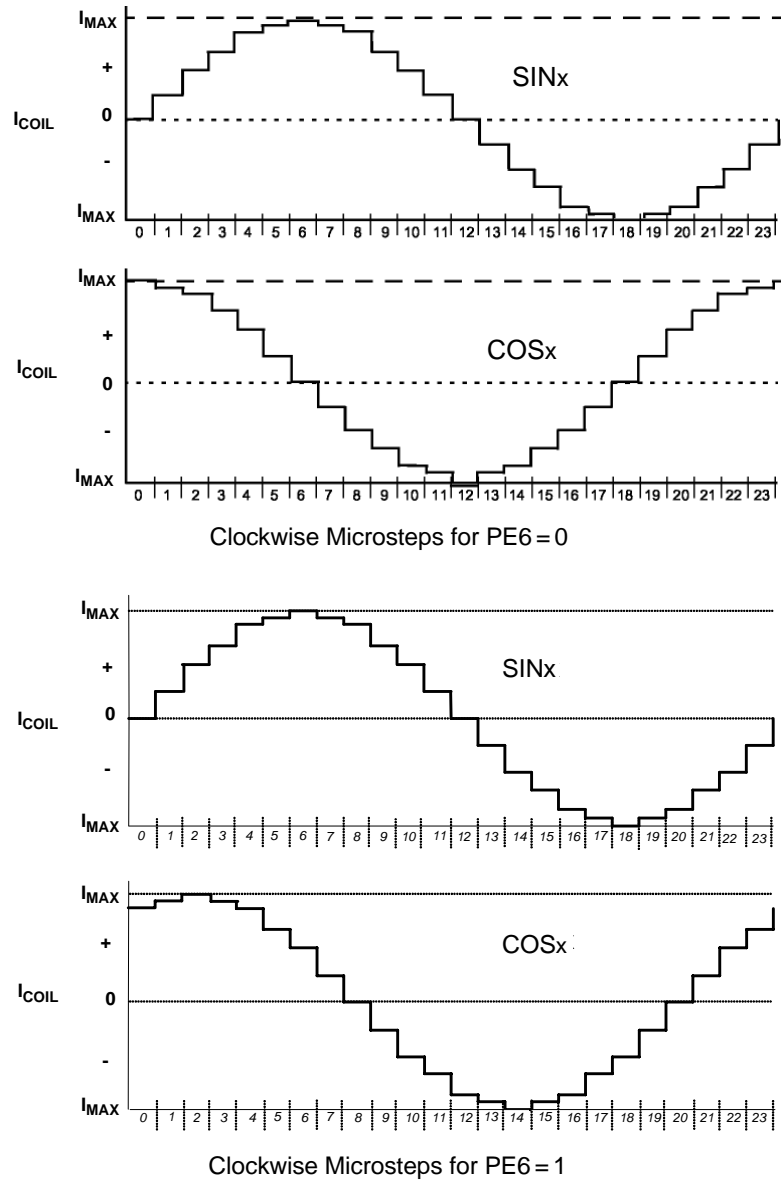
33976

Figure 9. Clockwise Microsteps

Table 29. Coil Step Value

Step	Angle	SINE (Angle)*	COS (Angle)* PE6=0	COS (Angle - 30)* PE6=1
0	0	0	1	0.866
1	15	0.259	0.965	0.966
2	30	0.5	0.866	1
3	45	0.707	0.707	0.966
4	60	0.866	0.5	0.866
5	75	0.966	0.259	0.707

Table 29. Coil Step Value

6	90	1	0	0.500
7	105	0.966	-0.259	0.259
8	120	0.866	-0.5	0
9	135	0.707	-0.707	-0.259
10	150	0.5	-0.866	-0.500
11	165	0.259	-0.966	-0.707
12	180	0	-1	-0.866
13	195	-0.259	-0.966	-0.966

33976

Table 29. Coil Step Value

14	210	-0.5	-0.867	-1
15	225	-0.707	-0.707	-0.966
16	240	-0.866	-0.5	-0.866
17	255	-0.966	-0.259	-0.707
18	270	-1	0	-0.500
19	285	-0.966	0.259	-0.259
20	300	-0.866	0.5	0
21	315	-0.707	0.707	0.259
22	330	-0.5	0.866	0.500
23	345	-0.259	0.966	0.707

* Denotes normalized values.

The motor is stepped by providing index commands at intervals. The time between steps defines the motor velocity, and the changing time defines the motor acceleration.

The state machine uses a table to define the allowed time and also the maximum velocity. A useful side effect of the table is that it also allows the direct determination of the position at which the velocity should reduce to stop the motor at the desired position.

The motor equations of motion are generated as follows. (The units of position are steps, and velocity and acceleration are in steps/second and steps/second².)

From an initial position of 0 with an initial velocity (u), the motor position (s) at a time (t) is:

$$s = ut + \tfrac{1}{2}at^2$$

For unit steps, the time between steps is:

$$\Rightarrow t = \frac{-u + \sqrt{u^2 + 2a}}{a}$$

This defines the time increment between steps when the motor is initially travelling at a velocity u. In the ROM, this time is quantized to multiples of the system clock by rounding upwards, ensuring acceleration never exceeds the allowed value. The actual velocity and acceleration is calculated from the time step actually used.

Using|

$$v^2 = u^2 + 2as$$
and
$$v = u + at$$

and solving for v in terms of u, s, and t gives:

$$v = \frac{2}{t} - u$$

The correct value of t to use in this equation is the quantized value obtained above.

From these equations a set of recursive equations can be generated to give the allowed time step between motor indexes when the motor is accelerating from a stop to its maximum velocity.

Starting from a position p of 0 and a velocity v of 0, these equations define the time interval between steps at each position. To drive the motor at maximum performance, index commands are given to the motor at these intervals. A table is generated giving the time step Δt at an index position n.

$$p_0 = 0$$
$$v_0$$

where $\lceil \; \rceil$ indicates rounding up.

$$v_n = \frac{2}{\Delta t_n} - v_{n-1}$$

$$P_n = n$$

Note $P_n = n$. This means on the nth step the motor has indexed by n positions and has been accelerating steadily at the maximum allowed rate. This is critical because it also indicates the minimum distance the motor must travel while decelerating to a stop. For example, the *stopping distance* is also equal to the current value of n.

The algorithm to drive the motor is similar to:

1. While the motor is stopped, wait until a command is received.

2. Send index pulses to the motor at an ever-increasing rate, according to the time steps in **Table 30** until:

   a. The maximum velocity is reached, at which point the time intervals stop decreasing, or

   b. The distance remaining to travel is less than the current index in the table. At this point, the stopping distance is equal to the remaining distance, and to ensure it will stop at the required position, the motor must begin decelerating.

An example of the velocity table for a particular motor is provided in **Table 30**. This motor's maximum speed is 4800 microsteps/s (at 12 microsteps/degrees), and its maximum acceleration is 54000 microsteps/s² . The table is quantized to a 1.0 MHz clock.

33976

Table 30. Velocity Table

Velocity Position	Time Between Steps (μs)	Velocity (μSteps/s)	Velocity Position	Time Between Steps (μs)	Velocity (μSteps/s)	Velocity Position	Time Between Steps (μs)	Velocity (μSteps/s)
0	0	0.00	76	380	2631.6	152	257	3891.1
1	27217	36.7	77	377	2652.5	153	256	3906.3
2	13607	73.5	78	374	2673.8	154	255	3921.6
3	11271	88.7	79	372	2688.2	155	254	3937.0
4	7970	125.5	80	369	2710.0	156	254	3937.0
5	5858	170.7	81	366	2732.2	157	253	3952.6
6	4564	219.1	82	364	2747.3	158	252	3968.3
7	3720	268.8	83	361	2770.1	159	251	3984.1
8	3132	319.3	84	358	2793.3	160	250	4000.0
9	2701	370.2	85	356	2809.0	161	249	4016.1
10	2373	421.4	86	354	2824.9	162	248	4032.3
11	2115	472.8	87	351	2849.0	163	248	4032.3
12	1908	524.1	88	349	2865.3	164	247	4048.6
13	1737	575.7	89	347	2881.8	165	246	4065.0
14	1594	627.4	90	344	2907.0	166	245	4081.6
15	1473	678.9	91	342	2924.0	167	244	4098.4
16	1369	730.5	92	340	2941.2	168	244	4098.4
17	1278	782.5	93	338	2958.6	169	243	4115.2
18	1199	834.0	94	336	2976.2	170	242	4132.2
19	1129	885.7	95	334	2994.0	171	241	4149.4
20	1066	938.1	96	332	3012.0	172	241	4149.4
21	1010	990.1	97	330	3030.3	173	240	4166.7
22	960	1041.7	98	328	3048.8	174	239	4184.1
23	916	1091.7	99	326	3067.5	175	238	4201.7
24	877	1140.3	100	324	3086.4	176	238	4201.7
25	842	1187.6	101	322	3105.6	177	237	4219.4
26	812	1231.5	102	321	3115.3	178	236	4237.3
27	784	1275.5	103	319	3134.8	179	235	4255.3
28	760	1315.8	104	317	3154.6	180	235	4255.3
29	737	1356.9	105	315	3174.6	181	234	4273.5
30	716	1396.6	106	314	3184.7	182	233	4291.8
31	697	1434.7	107	312	3205.1	183	233	4291.8
32	680	1470.6	108	310	3225.8	184	232	4310.3

33976

Table 30. Velocity Table (continued)

Velocity Position	Time Between Steps (μs)	Velocity (μSteps/s)	Velocity Position	Time Between Steps (μs)	Velocity (μSteps/s)	Velocity Position	Time Between Steps (μs)	Velocity (μSteps/s)
33	663	1508.3	109	309	3236.2	185	231	4329.0
34	648	1543.2	110	307	3257.3	186	231	4329.0
35	634	1577.3	111	306	3268.0	187	230	4347.8
36	621	1610.3	112	304	3289.5	188	229	4366.8
37	608	1644.7	113	303	3300.3	189	229	4366.8
38	596	1677.9	114	301	3322.3	190	228	4386.0
39	585	1709.4	115	300	3333.3	191	227	4405.3
40	575	1739.1	116	298	3355.7	192	227	4405.3
41	565	1769.9	117	297	3367.0	193	226	4424.8
42	555	1801.8	118	295	3389.8	194	226	4424.8
43	546	1831.5	119	294	3401.4	195	225	4444.4
44	538	1858.7	120	293	3413.0	196	224	4464.3
45	529	1890.4	121	291	3436.4	197	224	4464.3
46	521	1919.4	122	290	3448.3	198	223	4484.3
47	514	1945.5	123	289	3460.2	199	222	4504.5
48	507	1972.4	124	287	3484.3	200	222	4504.5
49	500	2000.0	125	286	3496.5	201	221	4524.9
50	493	2028.4	126	285	3508.8	202	221	4524.9
51	487	2053.4	127	284	3521.1	203	220	4545.5
52	481	2079.0	128	282	3546.1	204	220	4545.5
53	475	2105.3	129	281	3558.7	205	219	4566.2
54	469	2132.2	130	280	3571.4	206	218	4587.2
55	464	2155.2	131	279	3584.2	207	218	4587.2
56	458	2183.4	132	278	3597.1	208	217	4608.3
57	453	2207.5	133	277	3610.1	209	217	4608.3
58	448	2232.1	134	275	3636.4	210	216	4629.6
59	444	2252.3	135	274	3649.6	211	216	4629.6
60	439	2277.9	136	273	3663.0	212	215	4651.2
61	434	2304.1	137	272	3676.5	213	215	4651.2
62	430	2325.6	138	271	3690.0	214	214	4672.9
63	426	2347.4	139	270	3703.7	215	214	4672.9
64	422	2369.7	140	269	3717.5	216	213	4694.8
65	418	2392.3	141	268	3731.3	217	212	4717.0

33976

Table 30. Velocity Table (continued)

Velocity Position	Time Between Steps (μs)	Velocity (μSteps/s)	Velocity Position	Time Between Steps (μs)	Velocity (μSteps/s)	Velocity Position	Time Between Steps (μs)	Velocity (μSteps/s)
66	414	2415.5	142	267	3745.3	218	212	4717.0
67	410	2439.0	143	266	3759.4	219	211	4739.3
68	406	2463.1	144	265	3773.6	220	211	4739.3
69	403	2481.4	145	264	3787.9	221	210	4761.9
70	399	2506.3	146	263	3802.3	222	210	4761.9
71	396	2525.3	147	262	3816.8	223	209	4784.7
72	393	2544.5	148	261	3831.4	224	209	4784.7
73	389	2570.7	149	260	3846.2	225	208	4807.7
74	386	2590.7	150	259	3861.0			
75	383	2611.0	151	258	3876.0			

Internal Clock Calibration

Timing-related functions on the 33976 (e.g., pointer velocities, acceleration, and Return To Zero Pointer speeds) depend upon a precise, consistent time reference to control the pointer accurately and reliably. Generating accurate time references on an integrated circuit can be accomplished. For example trimming can be used however, it tends to be costly due to the large amount of die area required for trim pads. Another possibility is an externally generated clock signal; however, this requires a dedicated pin on the device and controller. An alternate approach would require the use of an additional crystal or resonator, which is expensive.

The internal clock in the 33976 is temperature independent and area efficient; however, it can vary by as much as ±35 percent due to process variation. Using the existing SPI inputs and the precision timing reference already available to the microcontroller, the 33976 allows more accurate clock calibration to within ±10 percent.

Calibrating the internal 1.0 MHz clock is initiated by writing a logic [1] to PECCR bit PE3 (see Figure 10, page 30). The 8.0 μs calibration pulse is then provided by the controller to ideally result in an internal 33976 clock speed of 1.0 MHz. The pulse is sent on the $\overline{CS}$ pin immediately after the SPI word is sent. During the calibration, no other SPI lines should be toggled. At the moment the $\overline{CS}$ pin transitions from logic [1] to logic [0], an internal 7-bit counter counts the number of cycles of an internal, 8.0 MHz clock. The counter stops when the $\overline{CS}$ pin transitions from logic [0] to logic [1]. The value in the counter represents the number of cycles of the 8.0 MHz clock occurring in the 8.0 μs window; it should range from 32 to 119. An offset is added to this number to help center or skew the calibrated result to generate a desired maximum or nominal frequency. The modified counter value is truncated by 4 bits to generate the calibration divisor, which should range from 4 to 15. The 8.0 MHz clock is divided by the calibration divisor, resulting in a calibrated 1.0 MHz clock. If the calibration divisor lies outside the range of 4 to 15, the 33976 flags the CAL bit in the device status register, indicating the calibration procedure was not successful. A clock calibration is allowed only if the gauges are disabled or the pointers are not moving, as indicated by status bits MOV1 and MOV0 (**Table 20**, page 21).
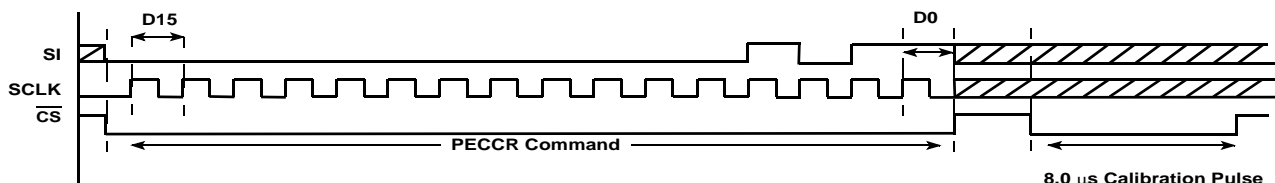
Figure 10. Gauge Enable and Clock Calibration Example

33976

Some applications may require a guaranteed maximum pointer velocity and acceleration. Guaranteeing these maximums requires that the nominal internal clock frequency falls below 1.0 MHz. The frequency range of the calibrated clock will always be below 1.0 MHz if PECCR bit PE4 is logic [0] prior to initiating a calibration command, followed by an 8.0 µs reference pulse. The frequency will be centered at 1.0 MHz if bit D4 is logic [1].

The 33976 can be fooled into calibrating faster or slower than the optimal frequency by sending a calibration pulse longer or shorter than the intended 8.0 µs. As long as the calibration divisor remains between 4 and 15 there will be no clock calibration flag. For applications requiring a slower calibrated clock—e.g., a motor designed with a gear ratio of 120:1 (8 microsteps/deg)—the user will have to provide a longer calibration pulse. The device allows a SPI-selectable slowing of the internal oscillator, using the PECCR command, so that the calibration divisor safely falls within the 4-to-15 range when calibrating with a longer time reference. For example, for the 120:1 motor, the pulse would be 12 µs instead of 8.0 µs. The result of this slower calibration results in the longer step times necessary to generate pointer movements meeting acceleration and velocity requirements. The resolution of the pointer positioning decreases from 0.083 deg/microstep (180:1) to 0.125 deg/microstep (120:1)

while the pointer sweep range increases from approximately 340 degrees to over 500 degrees.

Note Be aware that a fast calibration could result in violations of the motor acceleration and velocity maximums, resulting in missed steps.

Pointer Deceleration

Constant acceleration and deceleration of the pointer produces relatively choppy movements when compared to those of an air core gauge. Modifying the velocity position ramp during deceleration can create the desired damped movement. This modification is accomplished by adding repetitive steps at several of the last velocity position step values as the pointer decelerates. The 33976 allows the user to tailor the response characteristics to the application with three independent ramp characteristic variables. The RS, HCP and HC variables can be used to change the slowest velocity position steps, the number of Hold Counts, and the number of ramp positions to which the Hold Counts apply. More information is available in the RMPSEL description and in the example shown in Figure 11. If the maximum acceleration and deceleration of the pointer is desired, the Hold Counts can be disabled dynamically by either writing a logic [1] to the global Hold Count Disable bit, PECCR bit PE5, or to the HE0 or HE1 bits of the POS0R or POS1R, respectively.

First Velocity w/ Hold Counts = $HCP \times 8 + RS = 8 + 0 = 8$

Initial Velocity Position = $RS + 1 = 0 + 1 = 1$

Last Velocity w/ Hold Counts = $RS + 2 = 0 + 2 = 2$

Hold Counts per Step = $HC = 3$

For this example:
- RS = 0
- HC = 3
- HCP = 1

VELOCITY POSITION

MICROSTEPS
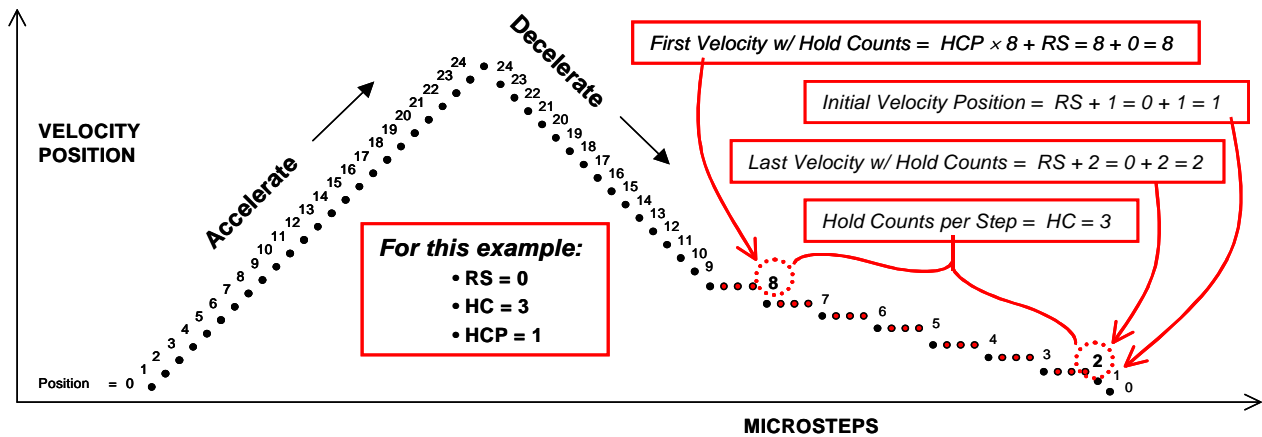
Figure 11. Deceleration Ramp

Return to Zero Calibration

Many step motor applications require that the IC detect when the motor is stalled after commanded to return to the zero position for calibration purposes. The stalling occurs when the pointer hits the end stop on the gauge bezel, which is usually at the *zero* position. It is important that when the pointer reaches the end stop it immediately stops without bouncing away.

The 33976 device provides the ability to automatically and independently return each of the two pointers to the zero position via the RTZR and RTZCR SPI commands. An automatic RTZ is initiated using the RZ0, RZ1, and RZ2 bits

provided the RZ4 bit is a logic [0]. Unconditional RTZ movement is initiated using the RZ0, RZ1, and RZ2 bits provided the RZ4 bit is a logic [1]. During an RTZ event, all commands related to the gauge being returned are ignored until the pointer has successfully zeroed or the RTZR bit RZ1 is written to disable the event. Once an RTZ event is initiated, the device reports back via the SO pin that an RTZ is underway.

The RTZCR command is used to set the RTZ pointer speed, choose an appropriate blanking time, and preload the integration accumulator with an appropriate offset. On reaching the end stop, the device reports back to the

33976

microcontroller via the status message that the RTZ was successful. The RTZ automatically disables, allowing other commands to be valid. In the event the master determines an RTZ sequence is not working properly (i.e., the RTZ taking too long), it can disable the command via the RTZR bit RZ1.

RTZCR bits RC10:RC5 are written to preload the accumulator with a predetermined value that will assure an accurate pointer stall detection. This preloaded value is determined during application development by disabling the automatic shutdown feature of the device with the RTZR bit RZ4. This operating mode allows the master to monitor the RTZ event, using the accumulator information available via the SO if the device is configured to provide the RTZ

Accumulator Status. The unconditional RTZ event can be turned OFF using the RTZR bit RZ1.

If the Position 0 location bit is in the default logic [0] mode, then during an RTZ event the pointer is returned counter-clockwise (CCW) using full steps at a constant speed determined by the RTZCR RC3:RC0 and RC12:RC11 bits during RTZ configuration (see [Figure 12]). Full steps are used during an RTZ so that only one coil of the motor is being driven at any time. The coil not being driven is used to determine if the pointer is moving. If the pointer is moving, the EMF signal that is present in the non-driven coil is processed by integrating the signal present on the opened pin of the coil while essentially grounding the other end.
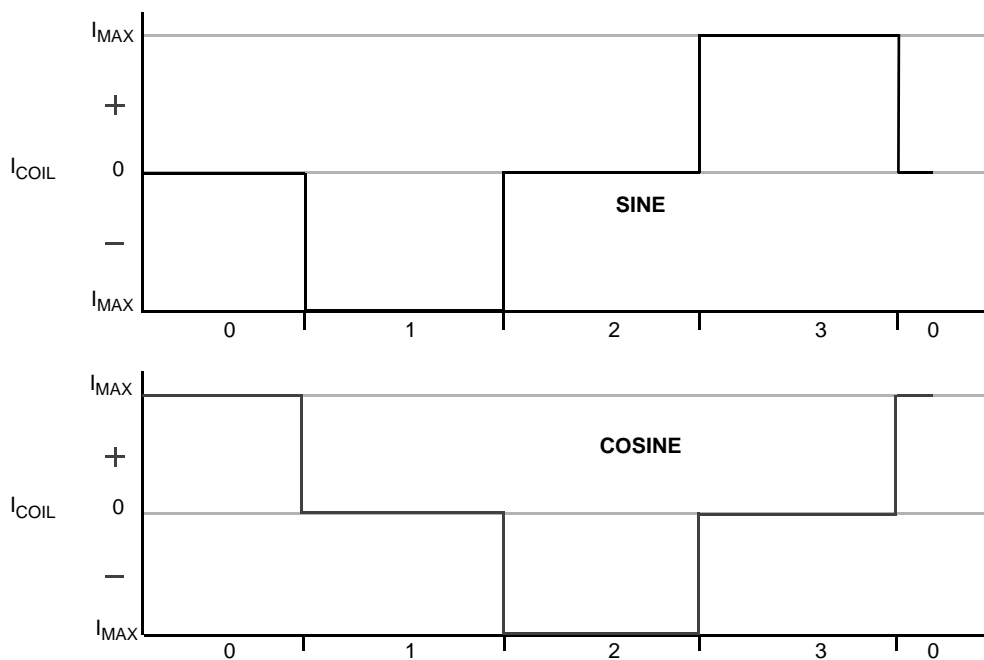
Figure 12. Full Steps Counterclockwise

The IC automatically prepares the non-driven coil at each step, waits for a predetermined blanking time, then processes the signal for the duration of the full step. When the pointer reaches the stop and no longer moves, the dissipating flux is detected. The processed results are placed in the RTZ accumulator, then compared to a decision threshold. If the signal exceeds the decision threshold, the pointer is assumed to be moving. If the threshold value is not exceeded, the drive sequence is stopped if RTZR bit RZ4 is logic [0]. If bit RZ4 is logic [1], the RTZ movement will continue indefinitely until the RTZR bit RZ1 is used to stop the RTZ event.

A pointer that is not on a full step location or that is in magnetic alignment prior to the RTZ event may cause a false RTZ detection. More specifically, an RTZ event beginning from a non-full step position may result in an abbreviated integration value potentially interpreted as a stalled pointer. Advancing the pointer by at least 12 microsteps clockwise (if

PE7 = 0) to the nearest full step position (e.g., 0, 6, 12, 18, 24, etc.) prior to initiating an RTZ ensures the magnetic fields line up and increases the chances of a successful pointer stall detection. It is important that the pointer be in a static, or commanded, position before starting the RTZ event. Because the time duration and the number of steps the pointer moves prior to reaching the commanded position can vary depending upon its status at the time a position change is communicated, the master should assure sufficient elapsed time prior to starting an RTZ. If an RTZ is desired after first enabling the outputs or after forcing a reset of the device, the pointer should first be commanded to move 12 microsteps clockwise to the nearest full step location. Because the pointer was in a static position at default, the master could determine the number of microsteps the device has taken by monitoring and counting the MOV0. MOV1 device status bit transitions to confirm the pointer is again in a static position. Alternatively, the user could monitor the device status bits CMD1 and CMD2.

It should be pointed out that the flux value, for an ideal motor with the coils perfectly aligned at 90°, will vary little from full step to full step if all other variables (e.g., temperature) are held constant. The full steps are evenly spaced which results in equidistant movement as the motor is full stepped.

In comparison, motors that have coils aligned at a 60° angle will result in two distinct flux values as the coils are driven in the same full step fashion. This lack of symmetry in the measured flux is due to the difference in the electrical angles between full steps. In other words, the distance that the rotor moves changes from full step to full step. This difference can be observed in Figure 9 and Table 29. In Figure 9, where PE6=0, the difference in microsteps between alternating full steps (one coil at maximum current while the other is at zero) is always six. In contrast, as seen in Figure 9, where PE6=1, the difference in microsteps between full steps of the 60° coils alternates between four and eight. These expected differences need to be taken into account when setting the RTZ threshold.

Only one gauge at a time can be returned to the zero position. The gauge not returning to zero can continue to be controlled. An RTZ should not begin until the gauge to be calibrated is at a static position and its pointer is at a full step position. An attempt to calibrate a gauge while the other is in the process of an RTZ event is ignored by the device. In most applications of the RTZR command, it is possible to avoid a visually obvious sequential calibration by first bringing the pointers back close to their previous zero positions, then re-calibrating them sequentially.

After completion of an RTZ, the 33976 automatically assigns the zero-step position to the full step position at the end-stop location. Because the actual zero position could lie anywhere within the full step where the zero was detected, the assigned zero position could be within a window of ±0.5 degree. An RTZ can be used to detect stall, even if the pointer already rests on the end stop when an RTZ sequence is initiated. However, it is recommended the pointer be advanced by at least 12 microsteps to the nearest full step prior to initiating the RTZ.

RTZ OUTPUT

During an RTZ event the non-driven coil is analyzed to determine the state of the motor. The 33976 multiplexes the coil voltages and provides the signal from the non-driven coil to the RTZ pin.

DEFAULT MODE

Default mode refers to the state of the 33976 after an internal or external reset prior to SPI communication. An internal reset occurs during V_{DD} power-up or if V_{PWR} falls below 4.0 V. An external reset is initiated by the $\overline{RST}$ pin driven to a logic [0]. With the exception of the RTZCR full step time and the RMPSEL Register values, all of the specific pin functions and internal registers will operate as though all of the addressable configuration register bits were set to logic [0]. This means, for example, all of the outputs will be disabled after a power-up or external reset, and SO flag OD6 and OD8 are set, indicating an undervoltage event. Anytime an external reset is exerted and the default is restored, all configuration parameters (e.g., clock calibration, maximum speed, RTZ parameters, etc.) are lost and must be reloaded.

FAULT LOGIC REQUIREMENTS

The 33976 device indicates each of the following faults as they occur:

- •Overtemperature fault
- •Undervoltage V_{PWR}
- •Overvoltage V_{PWR}
- •Clock out of spec

These fault bits remain enabled until they are clocked out of the SO pin with a valid SPI message.

Overcurrent faults are not reported directly; however, it is likely an overcurrent condition will become a thermal issue and be reported.

Overtemperature Fault Requirements

The 33976 incorporates overtemperature protection circuitry, which shuts off the affected gauge driver when excessive temperatures are detected. In the event of a thermal overload, the affected gauge driver is automatically disabled. The overtemperature fault is flagged via the OT0 and/or OT1 device status bits. The indicating flag continues to be set until the affected gauge is successfully re-enabled, provided the junction temperature has fallen to a temperature below the hysteresis level.

OVERVOLTAGE FAULT REQUIREMENTS

The device is capable of surviving V_{PWR} voltages within the maximum specified in Maximum Ratings table, page 4. V_{PWR} levels resulting in an Overvoltage Shutdown condition can result in uncertain pointer positions. Therefore, the pointer position should be re-calibrated. The master will be notified of an overvoltage event via the SO pin if the device status is selected. Overvoltage detection and notification occurs regardless of whether the gauge(s) are enabled or disabled.

Overcurrent Fault Requirements

Output currents are limited to safe levels allowing the device to rely on thermal shutdown to protect itself.

Undervoltage Fault Requirements

Undervoltage V_{PWR} conditions may result in uncertain pointer positions. Therefore, the internal clock and the pointer position may require re-calibration. The state machine continues with V_{PWR} voltage levels as low as 4.0 V; however, the coil voltages may be clipped. The master can be notified of an undervoltage event via the SO pin.

RESET (SLEEP MODE)

The device can reset internally or externally. If the V_{DD} level falls below the V_{DDUV} level (refer to the Static Electrical

33976

Characteristics table under POWER INPUT, page 5), the device resets and powers up in the Default mode. Similarly, If the $\overline{RST}$ pin is driven to a logic [0], the device resets to its default state. The device consumes the least amount of current (I_{DD} and I_{PWR}) when the $\overline{RST}$ pin is logic [0]. This is also referred to as the Sleep mode.

TYPICAL APPLICATIONS

The 33976 is an extremely versatile device that can be used in a variety of applications. The flexibility has been significantly improved, over that of the older MC33991 and MC33970 devices, with the addition of the Velocity Ramp configuration features that are available in the RMPSEL register (see **Table 31**). Some applications (e.g., high performance tachometers) require responsive pointers that change directions quickly. Figure 10 shows some characteristics of the ramp modifications that are possible with the RMPSELR RS bit.
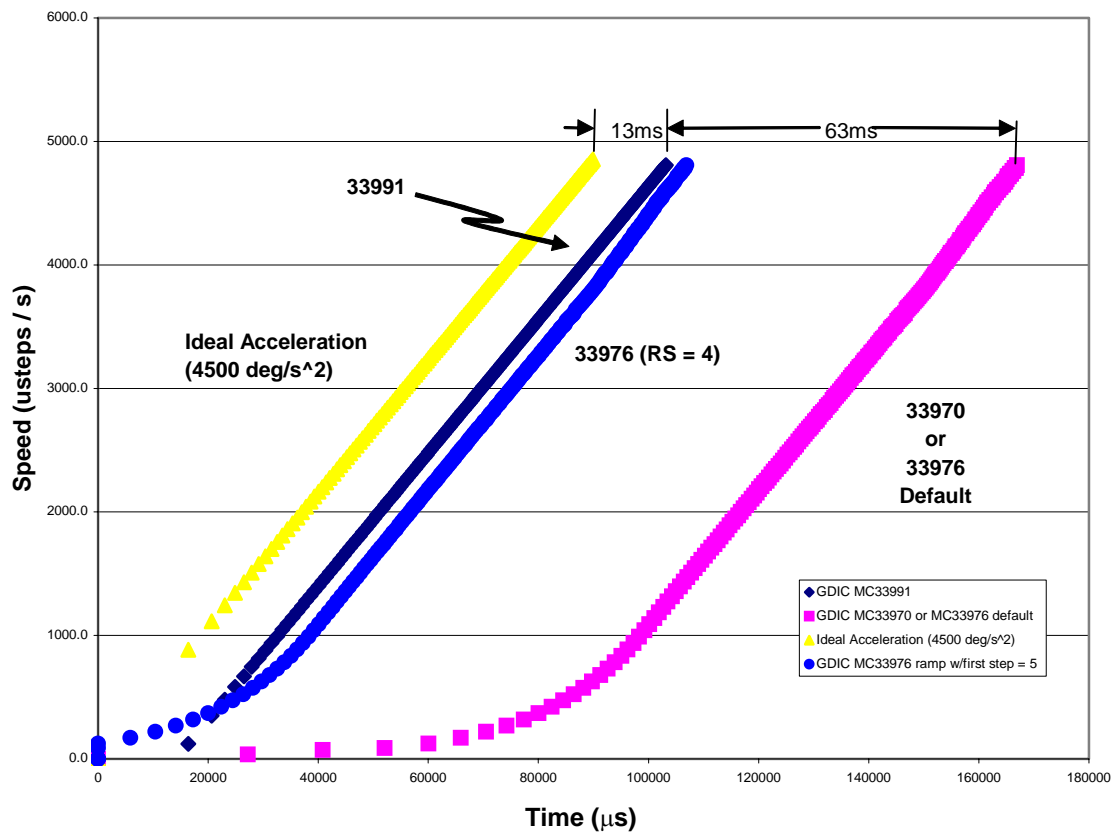
Figure 13. Start/Stop Response Characteristics

Other applications (e.g., speedometers and fuel indicators) require smooth, low speed movement. For these applications, the damping of the pointer can be optimized with the HC and HCP bits of the RMPSELR. Figure 14, page 36, gives several examples of different damping characteristics that are possible with the device. Once configured, the damping can be dynamically enabled and disabled using the HE bits of the Position registers.
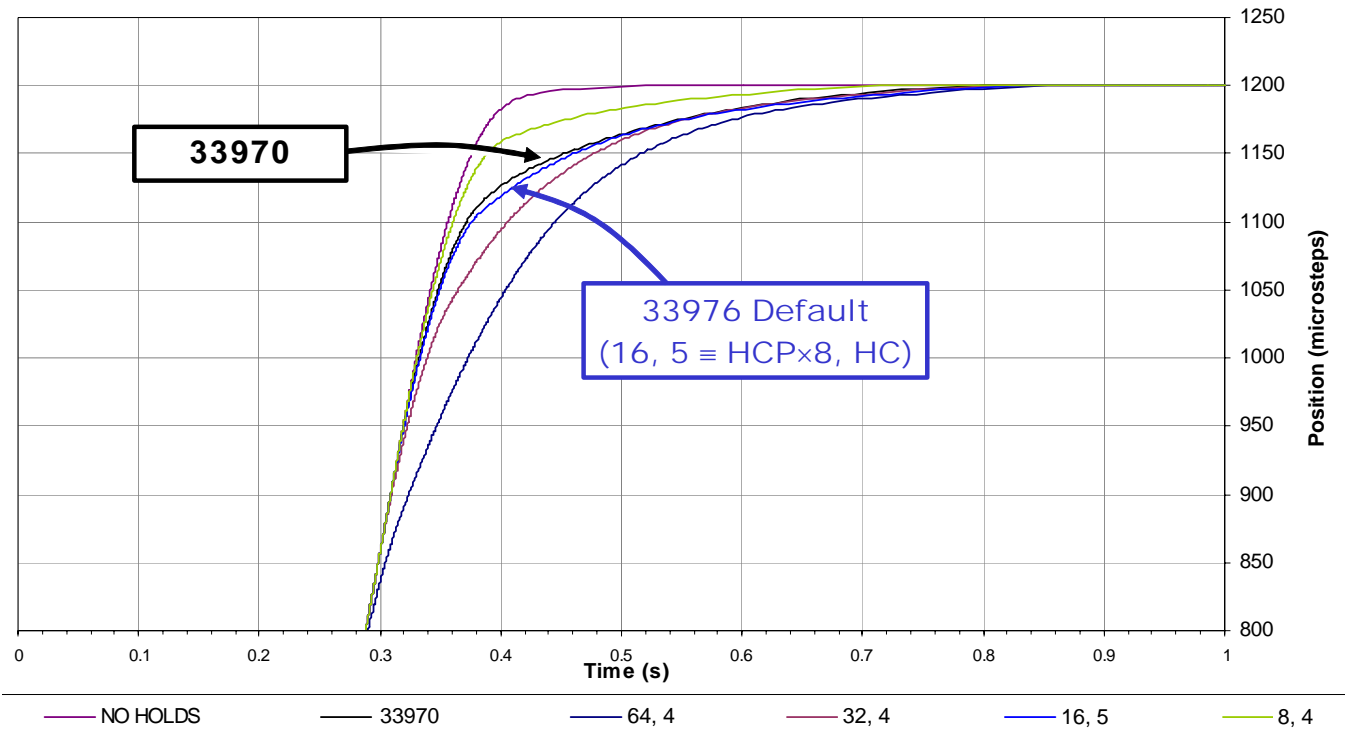
33976

Figure 14. 33976 Damping Response Examples

Table 31 provides a step-by-step example of configuring and using many of the features designed into the IC. This example is intended to give a generic overview how the device could be used. Further, it is intended to familiarize users with some of its capabilities.

Table 31. 33976 Setup, Configuration, and Usage Example

| Step	Command	Description	Reference Table and/or Figure
1	PECCR	Enable Gauges Bit PE0: Gauge 0 Bit PE1: Gauge 1	Table 7 (page 14), Figure 10 (page 30)
		Clock Calibration Bit PE3: Enables Calibration Procedure Bit PE4: Set clock f = 1.0 MHz maximum or nominal	
		Send 8.0 μs pulse on $\overline{CS}$ to calibrate 1.0 MHz clock	
2	RTZCR	Set RTZ Full Step Time Bits RC3:RC0	Table 12 (page 18), Table 13 (page 18)
		Set RTZ Blanking Time Bit RC4	
		Preload RTZ Accumulator Bits RC12:RC11 and RC10:RC5	Table 14 (page 19)
		Check SO for an Out-of-Range Clock Calibration Is bit CAL logic [1]? If so, then repeat Steps 1 and 2	Table 7 (page 14), Table 20 (page 21)

Table 31. 33976 Setup, Configuration, and Usage Example (continued)

Step	Command	Description	Reference Table and/or Figure
3	POS0R	Move pointer to position 12 prior to RTZ	Table 9 (page 16)
4	POS1R	Move pointer to position 12 prior to RTZ	Table 10 (page 16)
		Check SO to see if Gauge 0 has moved Is bit MOV0 (OD4) logic [1]? If so, then the Gauge 0 has moved to the first microstep	Table 7 (page 14), Table 20 (page 21)
5	PECCR	Send null command to see if gauges have moved Bit PE12	Table 7 (page 14)
		Check SO to see if Gauge 0 (Gauge 1) has moved Is bit MOV0 (OD4) (MOV1 (OD5)) logic [1]? If so, then Gauge 0 (Gauge 1) moved another microstep. Keep track of movement and if 12 steps are finished and both gauges are at a static position, then RTZ. Otherwise, repeat steps (a) and (b) Bit CMD0 (OD10) (CMD1 (OD11)) could also be monitored to determine that the pointer is static	Table 7 (page 14), Table 20 (page 21)
6	RTZ	Return one gauge at a time to the zero stop using RTZ command Bit RZ0 selects the gauge Bit RZ1 is used to enable or disable an RTZ Bits RZ2 is used to select the direction (along with PE7)	Table 11 (page 17)
		Select the RTZ accumulator bits to clock out on the SO bits using bits PE11:PE10. These will be used if characterizing the RTZ.	Table 7 (page 14), Table 22 (page 23)
7	PECCR	Check the Status of the RTZ by sending the null command to monitor SO bit RTZ0, RTZ1 of Device Status SO Bit PE12 is the null command	Table 7 (page 14), Table 20 (page 21)
		Is RTZ0 (OD2) logic [0]? If not, Gauge 0 still returning and null command should be resent	
8	RTZ	Return the other gauge to the zero stop. If the second gauge is driving a different pointer than the first, a new RTZCR command may be required to change the Full Step time	Table 11 (page 17)
9	PECCR	Check the Status of the RTZ by sending the null command to monitor SO, bit RTZ1 (OD3) Bit PE12 is the null command	Table 7 (page 14), Table 20 (page 21)
		Is RTZ1 (OD3) logic [0]? If not, Gauge 1 still returning and null command should be resent	
10	VELR	Change the maximum velocity of the gauge Bits V8:V9 determine which gauge(s) will change the maximum velocity Bits V7:V0 determine the maximum velocity position from Table 30, Velocity Table	Table 7 (page 15), Table 30 (page 28)
11	POS0R	Position Gauge 0 pointer Bits P011:P00: Desired Pointer Position	Table 9 (page 16), Table 30 (page 28)
		Check SO for Out-of-Range V_{PWR} Bit OVUV (OD6) logic [1]? If so, use UV (OD8) and OV (OD9) to decide whether to RTZ after valid V_{PWR}	
		Check SO for overtemperature Bit OT0 logic [1]? If so, enable driver again. If OT0 continues to indicate overtemperature, shut down Gauge 0 If RTZ0 returns to normal, re-establish the zero reference by RTZ command	

33976

Table 31. 33976 Setup, Configuration, and Usage Example (continued)

Step	Command	Description	Reference Table and/or Figure
12	POS1R	Position Gauge 1 pointer Bits P1 11:P1 0: Desired Pointer Position	Table 10 (page 16), Table 30 (page 28)
		Check SO for Out-of-Range V$_{PWR}$ Bit OVUV logic [1]? If so, use UV (OD8) and OV (OD9) to decided whether to RTZ after valid V$_{PWR}$	
		Check SO for overtemperature Bit OT1 logic [1]? If so, enable driver again. If OT1 continues to indicate overtemperature, shut down Gauge 1. If OT1 returns to normal, re-establish the zero reference by RTZ command	
13	POS0R	Return the pointers close to zero position using POS0R	Table 9 (page 16)
		Move pointer position at least 12 microsteps CW to the nearest full step prior to RTZ	
14	POS1R	Return the pointers close to zero position using POS1R	Table 10 (page 16)
		Move pointer position at least 12 microsteps CW to the nearest full step position prior to RTZ	
		Check SO to see if Gauge 0 has moved Bit MOV0 logic [1]? If so, Gauge 0 moved to the first microstep	Table 10 (page 16), Table 20 (page 21)
15	PECCR	Send null command to see if gauges have moved Bits PE12	Table 7 (page 14), Table 20 (page 21)
		Check SO to see if Gauge 0 (Gauge 1) moved Bit MOV0 (MOV1) logic [1]? If so, Gauge 0 (Gauge 1) moved another microstep. Keep track of movement. If 12 steps are finished, and both gauges are at a static position, then RTZ. Otherwise repeat steps (a) and (b) Bit CMD0 (OD10) (CMD1 (OD1)) could also be monitored to determine that the pointer is static	
16	RTZ	Return one gauge at a time to the zero stop using RTZ command Bit RZ0 selects the gauge Bit RZ1 is used to enable or disable an RTZ Bit RZ2 is used to select the direction (along with PE7)	Table 7 (page 14), Table 11 (page 17), Table 22 (page 23)
		Select the RTZ accumulator bits clocking out on the SO bits using bits PE11:PE10. These will be used if characterizing the RTZ	
17	PECCR	Check the status of the RTZ by sending the null command to monitor SO bit RTZ0 Bit PE12 is the null command	Table 7 (page 14), Table 20 (page 21)
		Is RTZ0 logic [0]? If not, Gauge 0 still returning and null command should be resent	
18	RTZ	Return the other gauge to the zero stop. If the second gauge is driving a different pointer than the first, a new RTZCR command may be required to change the Full Step time	Table 11 (page 17), Table 14 (page 19)
19	PECCR	Check the status of the RTZ by sending the null command to monitor SO bit RTZ1 Bit PE12 is the null command	Table 7 (page 14), Table 20 (page 21)
		Is RTZ1 logic [0]? If not, Gauge 1 still returning and null command should be resent	Table 11 (page 17)
20	PECCR	Disable both gauges and go to standby Bit PE0:PE1 are used to disable the gauges	Table 7 (page 14)
		Put the device to sleep $\overline{RST}$ pin is pulled to logic [0]	

33976

PACKAGING

PACKAGE DIMENSIONS

For the most current package revision, visit www.freescale.com and perform a keyword search using the "98A" listed below.
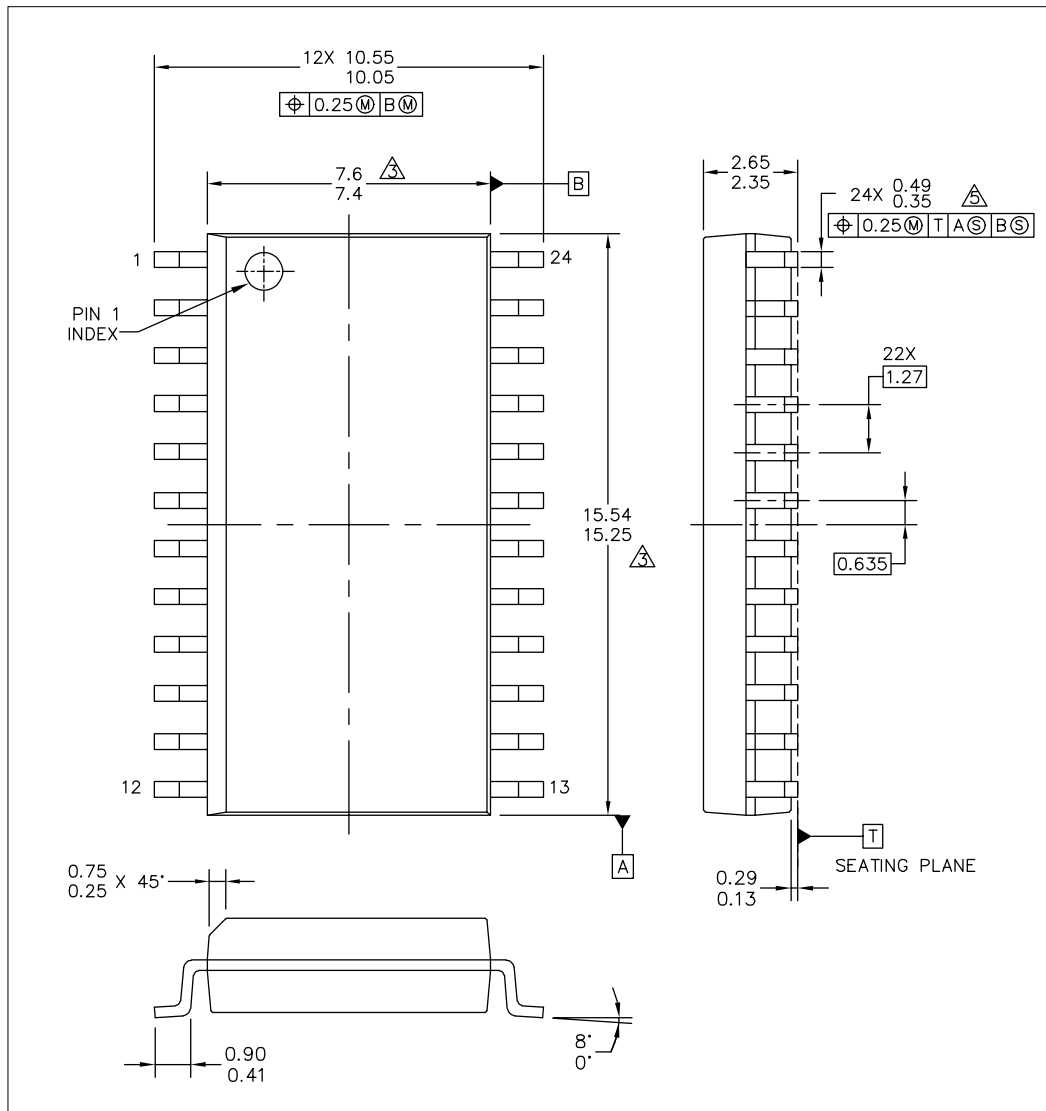
© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE	
TITLE: 24LD SOIC W/B, 1.27 PITCH 7.5 X 15.4 CASE-OUTLINE	DOCUMENT NO: 98ASB42344B	REV: F
	CASE NUMBER: 751E-04	26 APR 2005
	STANDARD: JEDEC MS-013 AD	

DW SUFFIX
EG SUFFIX (PB-FREE)
24-LEAD SOICW
PLASTIC PACKAGE
98ASB42344B
ISSUE F

33976

REVISION HISTORY

REVISION	DATE	DESCRIPTION OF CHANGES
3.0	8/2006	• Implemented Revision History page • Converted to Freescale format • Updated package drawing • Corrected symbol labels on Microstep Output (Measured Across Coil Outputs) and Output Flyback Clamp [11] • Added maximum pointer calculation on page 16 • Corrected detect threshold upper range from 4081 to 1009 • Changed internal clock variation from 35% to 70% • Changed EMF to flux on page 31 • Added MCZ33976EG/R2 to the Ordering Information block.
4.0	1/2007	• Revised Internal Block Diagram to enhance readability • Added parameter Peak Package Reflow Temperature During Reflow [4], [5] on page 4 and notes [4] and [5] • Added ADC Gain [10], [14] to Static Electrical Characteristics table • Made wording additions to Address 101—Gauge Return to Zero Configuration Register on page 17 and RC12:RC11 = M; default value = 00 on page 18 • Added RTZ Accumulator (Typical) on page 23 and accompanying text

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

MC33976
Rev 4.0
1/2007

freescale™
semiconductor