

Preliminary Information

Gauge Driver Integrated Circuit

The 33991 device is a single packaged, Serial Peripheral Interface (SPI) controlled, dual stepper 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 2-phase instrumentation stepper motor, similar to an MMT licensed AFIC 6405.

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

The device has many attractive features including:

- MMT-licensed two-phase stepper motor compatible
- Minimal processor overhead required
- Fully integrated pointer movement and position state machine with air core movement emulation
- 4096 possible steady state pointer positions
- 340° maximum pointer sweep
- Linear 4500°²
- Max pointer velocity of 400°
- Analog micro stepping (12 steps/degree of pointer movement)
- Pointer calibration and return to zero
- SPI controlled 16-bit word
- Calibratable Internal Clock
- Low Sleep mode current

33991

GAUGE DRIVER
INTEGRATED CIRCUIT

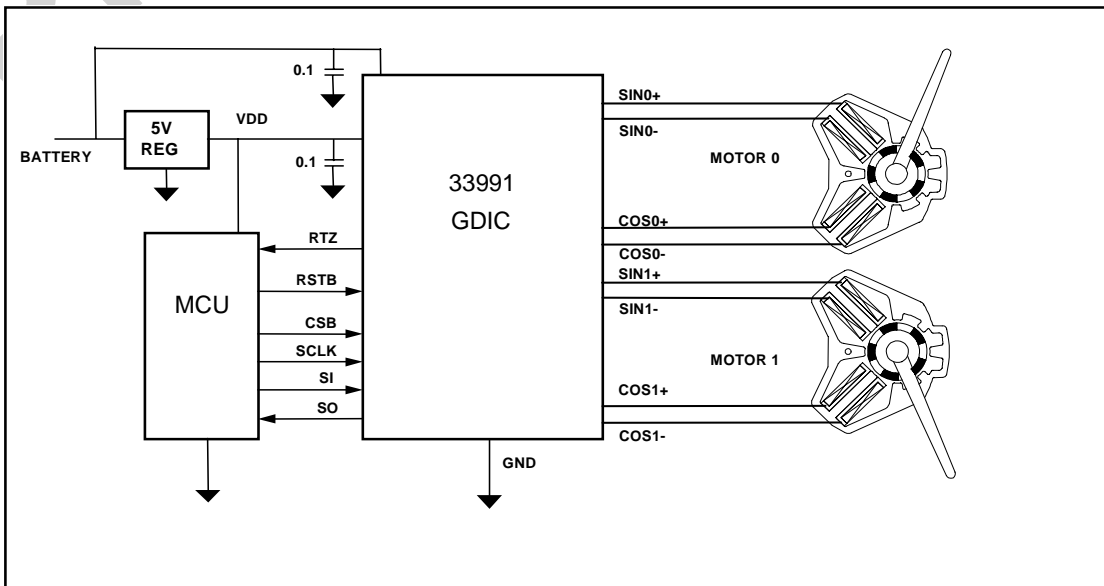


ORDERING INFORMATION

Device	Temperature Range (T _A)	Package
PC33991DH/R2	-40 to 125°C	SOICW

Freescale Semiconductor, Inc.

33991 Simplified Application Schematic



This document contains information on a product under development. Motorola reserves the right to change or discontinue this product without notice.

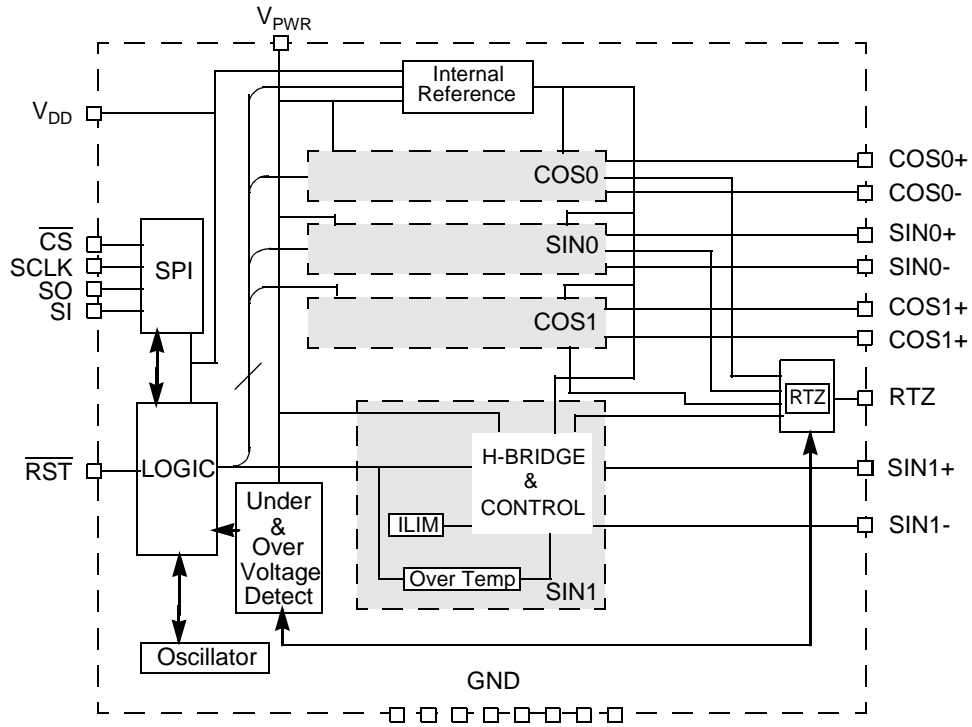
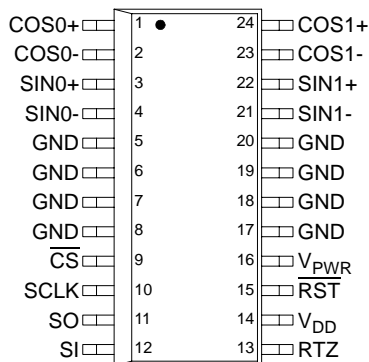


Figure 1. 33991 Block Diagram



24 Wide Body SOIC
Thermally Enhanced Lead Frame
R_{J-LEAD} = 15 C/W

Pin Functions

Pin Number	Pin Name	Description
1	COS0+	H-Bridge Output. This 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 stepper motors to provide four-quadrant operation.
2	COS0-	H-Bridge Output. This 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 stepper motors to provide four-quadrant operation.
3	SIN0+	H-Bridge Output. This 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 stepper motors to provide four-quadrant operation.
4	SIN0-	H-Bridge Output. This 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 stepper motors to provide four-quadrant operation.
5-8	GND	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. They also help dissipate heat from the device.
9	\overline{CS}	Chip Select. This pin is connected to a chip select output of a LSI IC. This IC controls which device is addressed by pulling the \overline{CS} pin of the desired device low, enabling the SPI communication with the device, while other devices on the serial link keep their serial outputs tri-stated. This input has an internal active pull-up and requires CMOS logic levels. This pin is also used to calibrate the internal clock.
10	SCLK	Serial Clock. This pin is connected to the SCLK pin of the master device and acts as a bit clock for the SPI port. It transitions one time per bit transferred at an operating frequency, f _{SPI} , defined in the Coil Output Timing Table. It is idle between command transfers. The pin is 50 percent duty cycle, with CMOS logic levels. This signal is used to shift data to and from the device.
11	SO	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. This output will remain tri-stated unless the device is selected by a low \overline{CS} signal. The output signal generated will have CMOS logic levels and the output data will transition on the rising edges of SCLK. The serial output data provides status feedback and fault information for each output and is returned MSB first when the device is addressed.
12	SI	Serial Input. This pin is connected to the SPI Serial Data Output pin of the master device from which it receives output command data. This input has an internal active pull-down requiring CMOS logic levels. The serial data transmitted on this line is a 16-bit control command sent MSB first, controlling the gauge functions. The master ensures data is available on the falling edge of SCLK.
13	RTZ	Multiplexed Output. This multiplexed output pin of the non-driven coil during an RTZ event.
14	V _{DD}	Voltage. This SPI and logic power supply input will work with 5.0 V supplies.

Freescale Semiconductor, Inc.

Pin Functions

Pin Number	Pin Name	Description
15	$\overline{\text{RST}}$	Reset. If the master decides to reset the device, or place it into a sleep state, the $\overline{\text{RST}}$ pin is driven to a logic 0. A logic 0 on the $\overline{\text{RST}}$ pin will force all internal logic to the known default state. This input has an internal active pull-up.
16	V _{PWR}	Battery Voltage. Power supply.
17-20	GND	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. They also help dissipate heat from the device.
21	SIN1-	H-Bridge Output. This pin is the output of a half bridge, designed to source or sink current. The H-Bridge pins linearly drive the sine and cosine coils of two separate stepper motors to provide four-quadrant operation.
22	SIN1+	H-Bridge Output. This pin is the output of a half bridge, designed to source or sink current. The H-Bridge pins linearly drive the sine and cosine coils of two separate stepper motors to provide four-quadrant operation.
23	COS1-	H-Bridge Output. This pin is the output of a half bridge, designed to source or sink current. The H-Bridge pins linearly drive the sine and cosine coils of two separate stepper motors to provide four-quadrant operation.
24	COS1+	H-Bridge Output. This pin is the output of a half bridge, designed to source or sink current. The H-Bridge pins linearly drive the sine and cosine coils of two separate stepper motors to provide four-quadrant operation.

STATIC ELECTRICAL CHARACTERISTICS

(Characteristics noted under conditions $4.75\text{ V} \leq V_{DD} \leq 5.25\text{ V}$, $-40^\circ\text{ C} \leq T_J \leq 150^\circ\text{ C}$, unless otherwise noted)

Characteristic	Symbol	Min	Nom	Max	Unit
Power Input					
Supply Voltage Range Fully Operational	V_{PWR}	6.5	—	26.0	V
V_{PWR} Supply Current (Gauge 1 & 2 outputs On, no output loads)	$I_{PWR(on)}$	—	4.0	6.0	mA
V_{PWR} Supply Current (all Outputs Disabled) (Reset = logic 0, $V_{DD} = 5\text{ V}$) (Reset = logic 0, $V_{DD} = 0\text{ V}$)	$I_{PWRslp1}$	—	42	60	μA
	$I_{PWRslp2}$	—	15	25	μA
Over voltage Detection Level (Note1)	V_{PWROV}	26	32	38	V
Under voltage Detection Level (Note2)	V_{PWRUV}	5.0	5.6	6.2	V
Logic Supply Voltage Range (5 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
V_{DD} Supply Current (Outputs Enabled)	$I_{DD(on)}$	—	1	1.8	mA

Notes:

1. Outputs will disable and must be re-enabled via the PECCR command.
2. Outputs remain active; however, the reduction in drive voltage may result in a loss of position control.

Freescale Semiconductor, Inc.

STATIC ELECTRICAL CHARACTERISTICS

(Characteristics noted under conditions $4.75\text{ V} \leq V_{DD} \leq 5.25\text{ V}$, $-40^\circ\text{ C} \leq T_J \leq 150^\circ\text{ C}$, unless otherwise noted)

Characteristic	Symbol	Min	Norm	Max	Unit
Power Outputs					
Microstep Output (measured across coil outputs) Sin0,1,+/- (Cos0,1,+/-) (see Pin Functions) Rout = 200 Ω Steps 6,18 (0,12) Steps 5,7,17,19 (1,11,13,23) Steps 4,8,16,20 (2,10,14,22) Steps 3, 9,15,21 (3,9,15,21) Steps 2,10,14,22 (4,8,16,20) Steps 1,11,13,23 (5,7,17,19) Steps 0,12 (6,18)	Vst6 Vst5 Vst4 Vst3 Vst2 Vst1 Vst0	4.9 0.94xVst6 0.84xVst6 0.69xVst6 0.47xVst6 0.23xVst6 -0.1	5.3 0.97xVst6 0.87xVst6 0.71xVst6 0.50xVst6 0.26xVst6 0	6.0 1.00xVst6 0.94xVst6 0.79xVst6 0.57xVst6 0.31xVst6 0.1	V
Full step Active Output (measured across coil outputs) Sin0,1, \pm (Cos0,1, \pm) (see 5-4) Steps 1,3 (0,2)	VFS	4.9	5.3	6.0	V
Microstep, Full Step Output (measured from coil low side to ground) Sin0,1, \pm (Cos0,1, \pm) $I_{OUT} = 30\text{mA}$	VLS	0	0.1	0.3	V
Output Flyback Clamp (Note3)	VFB	—	Vst1+0.5	Vst1+1.0	V
Output Current Limit (Out=Vstp6)	I_{LIM}	40	100	170	mA
Over temperature Shutdown	OTSD	155	—	180	$^\circ\text{C}$
Over temperature Hysteresis (Note3)	OT _{HYS}	8	—	16	$^\circ\text{C}$

Notes:

- Not 100 percent tested.

Freescale Semiconductor, Inc.

STATIC ELECTRICAL CHARACTERISTICS

(Characteristics noted under conditions $4.75\text{ V} \leq V_{DD} \leq 5.25\text{ V}$, $-40^\circ\text{ C} \leq T_J \leq 150^\circ\text{ C}$, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Control I/O					
Input Logic High Voltage (Note4)	V_{IH}	2.0	—	—	V
Input Logic Low Voltage (Note4)	V_{IL}	—	—	0.8	V
Input Logic Voltage Hysteresis (Note5)	$V_{in(hyst)}$	—	100	—	mV
Input Logic Pull Down Current (SI, SCLK)	I_{dwn}	3	—	20	μA
Input Logic Pull-Up Current (\overline{CS} , \overline{RST})	I_{up}	5	—	20	μA
SO High State Output Voltage (IOH = 1.0 mA)	V_{SOH}	0.8VDD	—	—	V
SO Low State Output Voltage (IOL = -1.6 mA)	V_{SOL}	—	0.2	0.4	V
SO Tri-State Leakage Current ($\overline{CS} \geq 3.5\text{ V}$)	S_{OLK}	-5	0	5	μA
Input Capacitance (Note6)	C_{in}	—	4	12	pF
SO Tri-State Capacitance (Note6)	C_{SO}	—	—	20	pF

Notes:

4. $V_{DD} = 5\text{ V}$
5. Not Production Tested; guaranteed by design.
6. Capacitance not measured; guaranteed by design.

Freescale Semiconductor, Inc.

STATIC ELECTRICAL CHARACTERISTICS

(Characteristics noted under conditions $4.75\text{ V} \leq V_{DD} \leq 5.25\text{ V}$, $-40^\circ\text{ C} \leq T_J \leq 150^\circ\text{ C}$, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
----------------	--------	-----	-----	-----	------

Power Output and Clock Timings

SIN, COS Output Turn ON delay Time (time from rising $\overline{\text{CS}}$ enabling outputs to steady state coil voltages and currents) (Note7)	$T_{\text{DHY(ON)}}$	—	—	1	ms
SIN, COS Output Turn OFF delay Time (time from rising $\overline{\text{CS}}$ disables outputs to steady state coil voltages and currents) (Note7)	$T_{\text{DHY(OFF)}}$	—	—	1	ms
Uncalibrated Oscillator Cycle Time	T_{CLU}	0.65	1.0	1.7	μs
Calibrated Oscillator Cycle Time (Cal pulse = 8 μs , PE CR D4 is logic 0)	T_{CLO}	1.0	1.1	1.2	μs
Calibrated Oscillator Cycle Time (Cal pulse = 8 μs , PE CR D4 is logic 1)	T_{CLO}	0.9	1.0	1.1	μs
Maximum Pointer Speed (Note8)	V_{MAX}	—	—	400	deg/s
Maximum Pointer Acceleration (Note8)	A_{MAX}	—	—	4500	deg/s ²

Notes:

- Maximum specified time for the 33991 is the minimum guaranteed time needed from the micro.
- The minimum and maximum value will vary proportionally to the internal clock tolerance. These are not 100 percent tested.

STATIC ELECTRICAL CHARACTERISTICS

(Characteristics noted under conditions $4.75\text{ V} \leq V_{DD} \leq 5.25\text{ V}$, $-40^\circ\text{ C} \leq T_J \leq 150^\circ\text{ C}$, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
SPI Timing Interface					
Recommended Frequency of SPI Operation	f_{SPI}	—	1	3	MHz
Falling edge of $\overline{\text{CS}}$ to Rising Edge of SCLK Required Setup Time (Note9)	T_{LEAD}	—	50	167	ns
Falling edge of SCLK to Rising Edge of $\overline{\text{CS}}$ (Required Setup Time)(Note9)	T_{LAG}	—	50	167	ns
SI to Falling Edge of SCLK (Required Setup Time) (Note9)	T_{LSU}	—	25	83	ns
Falling Edge of SCLK to SI (Required Hold Time) (Note9)	$T_{\text{SI(HOLD)}}$	—	25	83	ns
SO Rise Time (CL = 200 pF)	T_{rSO}	—	25	50	ns
SO Fall Time (CL = 200 pF)	T_{fSO}	—	25	50	ns
SI, $\overline{\text{CS}}$, SCLK, Incoming Signal Rise Time (Note10)	T_{rSI}	—	—	50	ns
SI, $\overline{\text{CS}}$, SCLK, Incoming Signal Fall Time (Note10)	T_{fSI}	—	—	50	ns
Falling Edge of $\overline{\text{RST}}$ to Rising Edge of $\overline{\text{RST}}$ (Required Setup Time) (Note9)	T_{WRST}	—	—	3	μs
Rising Edge of $\overline{\text{CS}}$ to Falling Edge of $\overline{\text{CS}}$ (Required Setup Time) (Note9) (Note11)	$T_{\overline{\text{CS}}}$	—	—	5	μs
Rising Edge of $\overline{\text{RST}}$ to Falling Edge of $\overline{\text{CS}}$ (Required Setup Time) (Note9)	T_{EN}	—	—	5	μs
Time from Falling Edge of $\overline{\text{CS}}$ to SO Low Impedance (Note12)	$T_{\text{SO(EN)}}$	—	—	145	ns
Time from Rising Edge of $\overline{\text{CS}}$ to SO High Impedance (Note13)	$T_{\text{SO(DIS)}}$	—	1.3	4	μs
Time from Rising Edge of SCLK to SO Data Valid (Note14) $0.2 V_{DD} <= SO >= 0.8 V_{DD}$, CL = 200 pF	T_{VALID}	—	65	105	ns

Notes:

9. The maximum setup times specified for the 33991 is the minimum time needed from the microcontroller to guarantee correct operation.
10. Rise and Fall time of incoming SI, $\overline{\text{CS}}$, and SCLK signals suggested for design consideration to prevent the occurrence of double pulsing.
11. This value is for a 1 MHz calibrated internal clock; it will change proportionally as the internal clock frequency changes.
12. Time required for output status data to be available for use at SO. 1 K Ω load on SO.
13. Time required for output status data to be terminated at SO. 1 K Ω load on SO.
14. Time required to obtain valid data out from SO following the rise of SCLK.

The device shall meet all SPI interface-timing requirements specified in the *SPI Interface Timing*, over the temperature range specified in the environmental requirements section. 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.

MAXIMUM RATING (All voltages are with respect to ground unless otherwise noted)			
Rating	Symbol	Value	Limit
Power Supply Voltage Steady State	$V_{PWR(sus)}$	-0.3 to 41	V
Input Pin Voltage (Note15)	V_{IN}	-0.3 to 7.0	V
SIN+/- COS +/- Continuous Per Output Current (Note16)	I_{OUTMAX}	40	mA
Storage Temperature	T_{stg}	-55 to 150	°C
Operating Junction Temperature	T_{Junc}	-40 to 150	°C
Thermal Resistance (C/W)	q_{JA}	60	°C/W
Ambient Junction to Lead	q_{JL}	20	°C/W
ESD Voltage	V_{ESD1}	2000	V
Human Body Model (Note17)	V_{ESD2}	200	V
Machine Model (Note18)			

Notes:

15. Exceeding voltage limits on Input pins may cause permanent damage to the device.
16. 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
17. V_{ESD1} testing is performed in accordance with the Human Body Model (Czap = 100pF, Rzap = 1500 Ω)
18. V_{ESD2} testing is performed in accordance with the Machine Model (Czap = 100pF, Rzap = 0 Ω)

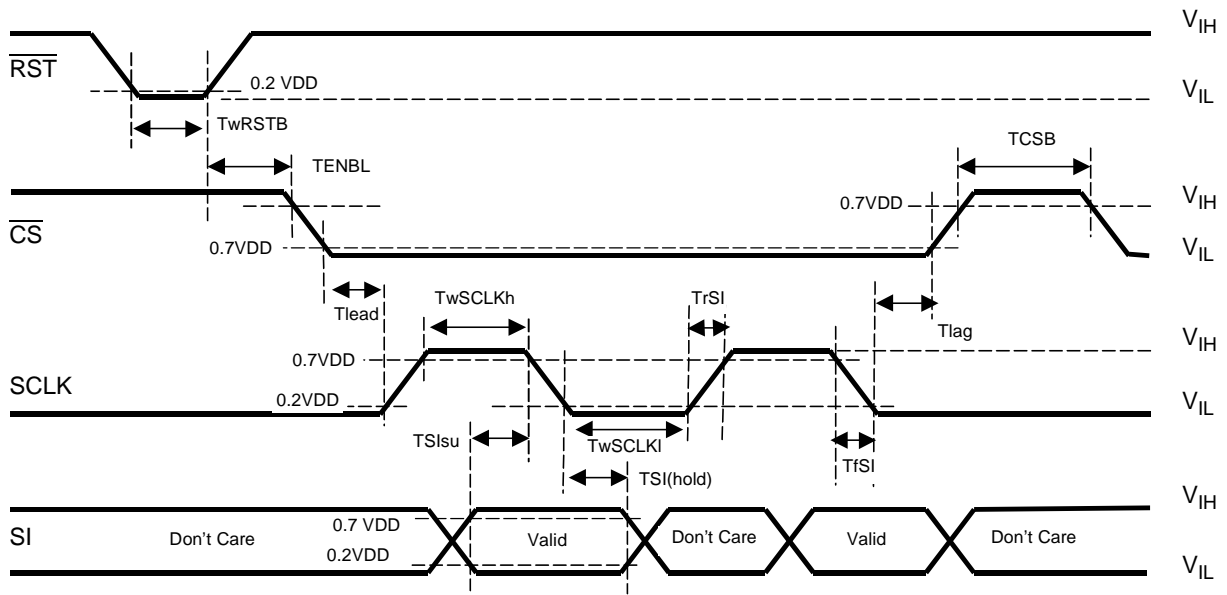


Figure 2. Input Timing Switching Characteristics

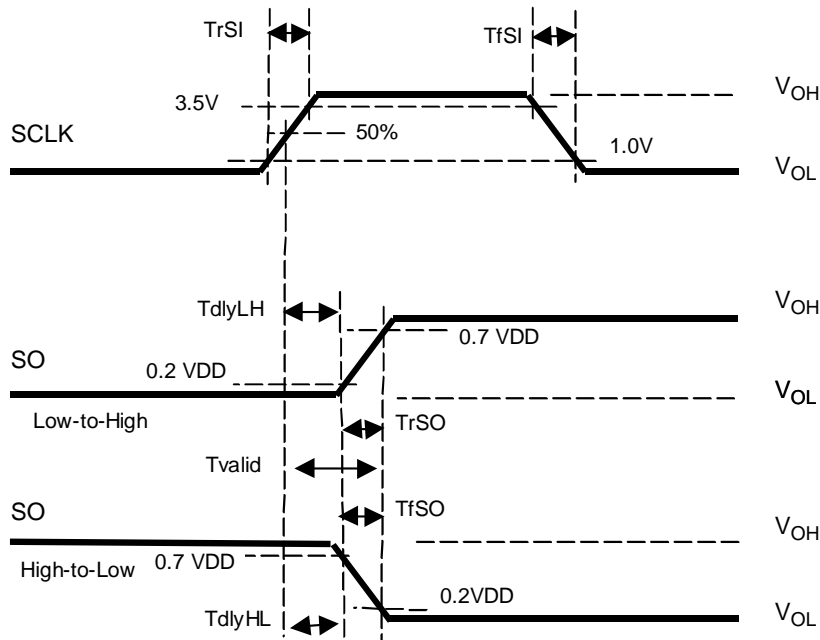


Figure 3. Valid Data Delay Time and Valid Time Waveforms

PC33991 SPI INTERFACE AND PROTOCOL DESCRIPTION

INTRODUCTION

The SPI interface has full duplex, three wire synchronous, 16-bit serial synchronous interface data transfer and four I/O lines associated with it: (SI, SO, SCLK, and \overline{CS}). The SI/SO pins of the 33991 follows 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.

DETAILED SIGNAL DESCRIPTIONS

Chip Select (\overline{CS})

The Chip Select (\overline{CS}) pin enables communication with the Master device. When this pin is in a logic [0] state, the GDIC is capable of transferring information to, and receiving information from, the Master. The 33991 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; 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 (*Iup*) connected to the pin as specified in the Control I/O table.

ignored; SO is tri-stated (high impedance). See the Data Transfer Timing diagrams in s 2 and 3.

Serial Input (SI)

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

Serial Clock (SCLK)

SCLK clocks the internal shift registers of the 33991 device. The serial input (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 the SCLK pin be in a logic [0] state whenever the \overline{CS} makes any transition. SCLK has an internal pull down (*I_{dwn}*), specified in the Control I/O table. When \overline{CS} is logic [1], signals at the SCLK and SI pins are

Serial Output (SO)

The Serial Output (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 multiples of 16 bits. Data is shifted on the rising edge of the SCLK signal. The SO pin remains in a high impedance state until the \overline{CS} pin is put into a logic low state.

SYSTEM APPLICATION INFORMATION

This section provides a description of the 33991 device SPI behavior. To follow the explanations below, please refer to the

timing diagrams illustrated in Figures 4 and 5.

Table 1. Data Transfer Timing

Pin	Description
\overline{CS} (1-to-0)	SO pin is enabled
\overline{CS} (0-to-1)	33991 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

Freescale Semiconductor, Inc.

TIMING DESCRIPTIONS AND DIAGRAMS

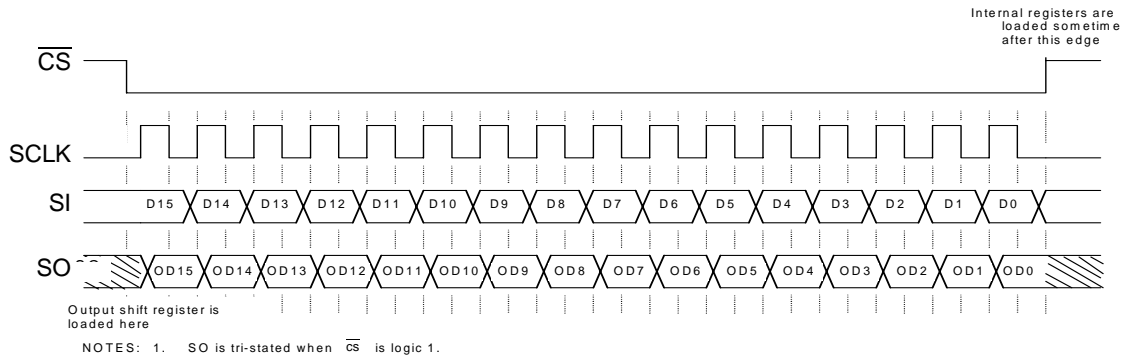


Figure 4. Single 16-bit Word SPI Communication

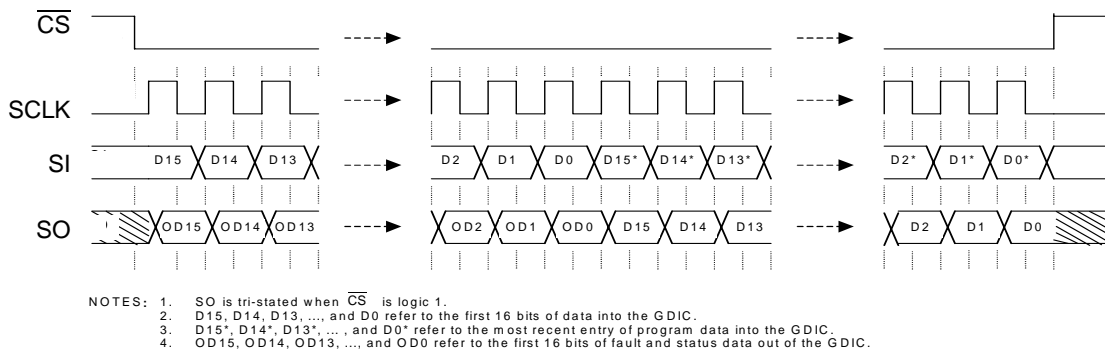


Figure 5. Multiple 16-bit Word SPI Communication

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, according to the address contained in bits 15-13. The minimum time \overline{CS} should be kept high depends on the internal clock speed. That data is specified in the SPI Interface Timing table. 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 the \overline{CS} at logic [0], the contents of the 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].

COMMUNICATION MEMORY MAPS

The 33991 device is capable of interfacing directly with a microcontroller, via the 16-bit SPI protocol described and 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 starting 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 will transfer through daisy chained devices, illustrated in Figure 5. If an attempt is made to latch in a message smaller than 16-bits wide, it is ignored.

The 33991 uses six registers to control the device and control the state of the four H-bridge outputs. Registers are addressed via D15-D13 of the incoming SPI word. Refer to Table 2.

Module Memory Map

Various registers of the 33991 SPI module are addressed by the three MSB 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

Status reporting includes:

- Individual gauge over temperature condition
- Battery out of range condition
- Pointer zeroing status
- Internal clock status
- Confirmation of coil output changes should result in pointer movement

Table 2 provides the register available to control the above functions.

Table 2. Module Memory Map

Address [15:13]	Use	Name
000	Power, Enable, and Calibration Register	PECR
001	Maximum Velocity Register	VELR
010	Gauge 0 Position Register	POS0R
011	Gauge 1 Position Register	POS1R
100	Return to Zero Register	RTZR
101	Return to Zero Configuration Register	RTZCR
110	Not Used	
111	Reserved for Test	

Register Descriptions

Power, Enable, and Calibration Register (PECR)

This register allows the master to independently enable or disable the output drivers of the two gauge controllers.

SI Address 000—Power, Enable, and Calibration Register is illustrated in Table 3. A write to the 33991 using this register allows the master to independently enable or disable the output drivers of the two gauge controllers as well as to calibrate the internal clock, or send a null command for the purpose of reading the status bits. This register is also used to place the 33991 device into a low current consumption mode.

Each of the gauge drivers can be enabled by writing a logic [1] to their assigned address bits, D0 and D1 respectively. This feature could be useful to disable a driver if it is failing or not being used. The device can be placed into a standby current mode by writing a logic [0] to both D0 and D1. 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 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 percent to -35 percent. The 33991 device was designed with a feature allowing the internal clock to be software calibrated to a tighter tolerance of ± 10 percent, using the CS pin and a reference time pulse provided by the microcontroller.

Calibration of the internal clock is initiated by writing a logic [1] to D3. The calibration pulse, must be $8 \mu\text{s}$ for an internal clock speed of 1 MHz, will be sent on the CS pin immediately after the SPI word is sent. No other SPI lines will be toggled. A clock calibration is allowed only if the gauges are disabled or the pointers are not moving, indicated by status bits ST4 and ST5.

Some applications may require a guaranteed maximum pointer velocity and acceleration. Guaranteeing these maximums require the nominal internal clock frequency fall below 1 MHz. The frequency range of the calibrated clock is always below 1 MHz if bit D4 is logic [0] when initiating a calibration command followed by an $8 \mu\text{s}$ reference pulse. The frequency is centered at 1MHz if bit D4 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 D2 will slow the internal oscillator by one-third, leading to a situation where it is possible to calibrate at maximum 667 kHz or centered at 667 kHz. In these cases, it may be necessary to provide a longer calibration pulse of exactly $12 \mu\text{s}$ without any indication of a calibration fault at status bit ST7. The preceding description should be the case for 1 MHz if D2 is left logic [0].

If bit D12 is logic [1] during a PECR command, the state of D11: D0 is ignored. This is referred to as the null command and can be used to read device status without affecting device operation.

Table 3. Power, Enable and Calibration Register (PECR)

Address 000													
	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Write	PE12	0	0	0	0	0	0	0	PE4	PE3	PE2	PE1	PE0

These bits are *write-only*.

PE12–Null Command for Status Read

- 0 = Disable
- 1 = Enable

PE11–PE5—These bits must be transmitted as logic [0] for valid PECR commands.

PE4–Clock Calibration Frequency Selector

- 0 = Maximum f = 1 MHz (for 8 μs calibration pulse)
- 1 = Nominal f = 1 MHz (for 8 μs calibration pulse)

PE3–Clock Calibration Enable—This bit enables or disables the clock calibration.

- 0 = Disable
- 1 = Enable

PE2–Oscillator Adjustment

- 0 = Tosc
- 1 = 0.66 x Tosc

PE1–Gauge 1 Enable—This bit enables or disables the output driver of Gauge 1.

- 0 = Disable
- 1 = Enable

PE0–Gauge 0 Enable—This bit enables or disables the output driver of Gauge 0.

- 0 = Disable
- 1 = Enable

Maximum Velocity Register (VELR)

SI Address 001—Gauge Maximum Velocity Register, is used to set a maximum velocity for each gauge. See Table 4.

Bits D7-D0 contain a position value from 1-255 representative of the table position value. 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 will become the maximum velocity. If the motor is turning at a value greater than the new maximum, the motor will ignore the new value until the speed falls equal to, or below it. Velocity for each motor can be changed simultaneously, or independently, by writing D8 and/or D9 to a logic [1]. Bits D10 –D12 must be at logic [0] for valid VELR commands.

Table 4. Maximum Velocity Register (VELR)

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

These bits are *write-only*.

V12–V10—These bits must be transmitted as logic [0] for valid VELR commands.

V9–Gauge 1 Velocity—Specifies whether the maximum velocity determined in the V7-V0 field applies to Gauge 1

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

V8–Gauge 0 Velocity—Specifies whether the maximum velocity specified in the V7-V0 field applies to Gauge 0

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

V7–V0 Maximum Velocity—Specifies the maximum velocity position from the acceleration table. This velocity will remain the

maximum of the intended gauge until changed by command. Velocities can range from position 1 (00000001) to position 255 (11111111).

Gauge 0/1 Position Register (POS0R, POS1R)

SI Addresses 010—Gauge 0 Position Register receives writing when communicating the desired pointer positions.

SI Address 011—Gauge 1 Position Register receives writing when communicating the desired pointer positions.

Register bits D11–D0 receives writing when communicating the desired pointer positions.

Commanded positions can range from 0 to 4095. The D12 bit must be at logic [0] for valid POS0R and POS1R commands.

Table 5. Gauge 0 Position Register (POS0R)

Address 010													
	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Write	0	P011	P010	P09	P08	P07	P06	P05	P04	P03	P02	P01	P00

These bits are *write-only*.

P012—This bit must be transmitted as logic [0] for valid commands.

P011—P0 0 Desired pointer position of Gauge 0.

Pointer positions can range from 0 (000000000000) to position 4095 (111111111111). For a stepper motor requiring 12 microsteps per degree of pointer movement, the maximum pointer sweep is 341.25°.

Table 6. Gauge 1 Position Register (POS1R)

Address 011													
	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Write	0	P011	P010	P09	P08	P07	P06	P05	P04	P03	P02	P01	P00

These bits are *write-only*.

P012—This bit must be transmitted as logic [0] for valid commands.

P011—P0 0 Desired pointer position of Gauge 1.

Pointer positions can range from 0 (000000000000) to position 4095 (111111111111). For a stepper motor requiring 12 microsteps per degree of pointer movement, the maximum pointer sweep is 341.25°.

the 8 MSBs of the SO word. See Table 12. This feature provides the flexibility to look at 15 bits of content with eight bits of the SO word. This 8-bit window can be dynamically changed while in the RTZ mode.

A logic [00], written to bits D3:D2, results in the RTZ accumulator bits 7:0 clocked out as SO bits D15:D8, respectively. Similarly, a logic [01] results in RTZ counter bits 11:4 clocked out, and logic [10] delivers counter bits 14:8 as SO bits D14:D8, respectively. A logic [11] clocks out the same information as logic [10]. This feature allows the master to monitor the RTZ information regardless the size of the signal. Further, this feature is very useful during the determination of the accumulator offset to be loaded in for a motor and pointer combination. It should be noted, RTZ accumulator contents will reflect the data from the previous step. The first accumulator results to be read back during the first step will be 1111111111111111.

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

Bit D4 is used to enable an unconditional RTZ event. A logic [0] results in a typical RTZ event automatically stopping when a stall condition is detected. A logic [1] results in RTZ movement, stopping only if a logic [0] is written to bit D0. This feature is useful during development and characterization of RTZ requirements.

Gauge Return to Zero Register (RTZR)

SI Address 100—Gauge Return to Zero Register (RTZR), provided in Table 7, 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, its results stored in an accumulator. Contents of this register's 15-bit RTZ accumulator can be read eight bits at a time.

A logic [1] written to bit D1 enables a return to zero for Gauge 0 when D0 is logic [0], and Gauge 1 when D0 is 1, respectively. Similarly, a logic [0] written to bit D1 disables a return to zero for Gauge 0 when D0 is logic [0], and Gauge 1 when D0 is 1, respectively.

Bits D3 and D2 are used to determine which eight bits of the 15-bit RTZ accumulator are clocked out of the SO register as

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

Table 7. Return to Zero Register (RTZR)

Address 100													
	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Write	0	0	0	0	0	0	0	0	RZ4	RZ3	RZ2	RZ1	RZ0

Table 8. RTZ Accumulator Bit Select

D3	D2	RTZ Accumulator Bits to SO Bits ST15:ST8
0	0	[7:0]
0	1	[11:4]
1	0	[14:8]
1	1	[14:8]

RZ12–RZ5—These bits must be transmitted as logic [0] for valid commands.

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

- 0 = Automatic return to zero
- 1 = Unconditional return to zero

RZ3–RZ2—These bits are used to determine which eight bits of the RTZ accumulator will be clocked out via the SO pin. See Table 8.

RZ1—Return to zero—Commands the selected gauge to return the pointer to zero position.

- 1 = Return to zero enabled
- 0 = Return to zero disabled

RZ0—Gauge Select: Gauge 0/Gauge 1—Selects the gauge to be commanded.

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

Gauge Return to Zero Configuration Register

SI Address 101—Gauge Return to Zero Configuration Register (RTZCR) is used to con the Return to Zero Event. See Table 9. It is written to modify the step time, or rate, the pointer moves during an RTZ event. Also, the integration blanking time is adjustable with this command. Integration blanking time is the time immediately following the transition of a coil from a driven state to an open state in the RTZ mode. Finally, this command is used to adjust the threshold of the RTZ integration register.

The values used for this register are chosen during development to optimize the RTZ for each application. Various resonance frequencies can occur due to the interaction between the motor and the pointer. This command permits moving the RTZ pointer speed away from these frequencies.

Bits D3:D0 determine the time spent at each full step during an RTZ event. The step time associated with each bit combination is illustrated in Table 10. The default full step time is 21.25 ms (0101). If there are 2 full steps per degree of pointer movement, the pointer speed is: $1/(FS \times 2)$ degrees.

Bit D4 determines the provided blanking time immediately following a full step change, and before enabling the integration of the non-driven coil signal. The blanking time is either 512 μ s, when D4 is logic [0], or 768 μ s when D4 is logic [1].

Detecting pointer movement is accomplished by integrating the back EMF present in the non-driven coil during the RTZ event. The integration circuitry is implemented using a Sigma-Delta converter resulting in a representative 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 moving pointer. 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. If the RTZR D3:D2 bits were written as 10 or 11, the ST14 bit corresponds to bit D14 of the RTZ accumulator, the sign bit. After a full step of integration, a sign bit of 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.

Accurate pointer stall detection depends on a correctly preloaded accumulator for specific gauge, pointer, and full step combinations. Bits D12:D5 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. A stalled pointer results in a negative value. The preloaded values associated with each combination of bits D12:D5 are illustrated in Table 11. The accumulator should be loaded with a negative value resulting in a transition of the accumulator MSB to a logic [1] when the motor is stalled. After a power-up, or any reset in the Default mode, the 33991 device sets the accumulator value to -1, resulting in an unconditional RTZ pointer movement.

Table 9. RTZCR SI Register Assignment

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

These bits are *write-only*.

RC12–RC5—These bits (D12:D5) determine the preloaded value into the RTZ integration accumulator to adjust the detection threshold. Values range from -1 (00000000) to -4081 (11111111) provided in Table 11.

RC4—This bit determines the RTZ blanking time

- 0 = 512 μ s
- 1 = 768 μ s

RC3–RC0—These bits (D3:D0) determine the full step time during an RTZ event, determining the pointer moving rate. Step times range from 4.86 ms (0000) to 62.21 ms (1111). Those are illustrated in Table 10. The default time is 21.25 ms (0101).

Table 10. RTZCR Full Step Time

RC3	RC2	RC1	RC0	Full Step Time (ms)
0	0	0	0	4.86
0	0	0	1	4.86
0	0	1	0	8.96
0	0	1	1	13.06
0	1	0	0	17.15
0	1	0	1	21.25
0	1	1	0	25.34
0	1	1	1	29.44
1	0	0	0	33.54
1	0	0	1	37.63
1	0	1	0	41.73
1	0	1	1	45.82
1	1	0	0	49.92
1	1	0	1	54.02
1	1	1	0	58.11
1	1	1	1	62.21

Table 11. RTZCR Accumulator Offset

RC 12	RC 11	RC 10	RC 9	RC 8	RC 7	RC 6	RC 5	Preload Value (PV)	Initial Accumulator Value = (-16×PV)-1
0	0	0	0	0	0	0	0	0	-1
0	0	0	0	0	0	0	1	1	-17
0	0	0	0	0	0	1	0	2	-33
0	0	0	0	0	0	1	1	3	-49
0	0	0	0	0	1	0	0	4	-65
"	"	"	"	"	"	"	"	"	"
"	"	"	"	"	"	"	"	"	"
"	"	"	"	"	"	"	"	"	"
1	1	1	1	1	1	1	1	255	-4081

Freescale Semiconductor, Inc.

SO Communication

When the \overline{CS} pin is pulled low, the internal status word register is loaded into the output register and the fault data is clocked out MSB (OD15) first. Following a \overline{CS} transition 0 to 1, the device determines if the message shift was a valid length and if so, latches the data into the appropriate registers. A valid message length is greater than 0 bits and a multiple of 16 bits. At this time, the SO pin is tri-stated and the fault status register is now able to accept new fault status information. If the

message length is determined to be invalid, the status information is not cleared. It is transmitted again during the next SPI message.

Any bits clocked out of the SO pin after the first sixteen, is representative of the initial message bits clocked into the SI pin. That is due to the \overline{CS} pin first being transitioned to a logic [0]. This feature is useful for daisy chaining devices as well as message verification.

Table 12. Status Output Register

	OD15	OD14	OD13	OD12	OD11	OD10	OD9	OD8	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0
Read	ST15	ST14	ST13	ST12	ST11	ST10	ST9	ST8	ST7	ST6	ST5	ST4	ST3	ST2	ST1	ST0

These are *read-only* bits.

ST15–ST8—Bits representing the eight bits from the RTZ accumulator as determined by the status of bits RZ2 and RZ3 of the RTZR, defined in Table 8. These bits represent the integrated signal present on the non-driven coil during an RTZ event. These bits will be logic [0] after power-on reset, or after the \overline{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.

ST7—Calibrated Clock out of Specification—A logic [1] on this bit indicates the clock count calibrated to a value outside of the expected range, and given the tolerance specified by T_{CLC} in the *SPI Interface Timing* table.

- 0 = Clock with in spec
- 1 = Clock out of spec

ST6—Under voltage or over voltage Indication—A logic [1] on this bit indicates the V_{PWR} voltage fell to a level below the V_{PWRUV} , or it exceeded an upper limit of V_{PWROV} , as specified in the *Static Electrical Characteristics* table, since the last SPI communication. An under voltage event is just flagged, while an

over voltage event automatically disables the driver outputs. Because the pointer may not be in the expected position, the master may want to re-calibrate the pointer position with a RTZ command after the voltage returns to a normal level. For an over voltage event, both gauges must be re-enabled as soon as this flag returns to logic [0]. The state machine continues 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}

ST5—Gauge 1—Movement since last SPI communication. A logic [1] on this bit indicates the Gauge 1 pointer position has changed since the last SPI command. This allows the master to confirm the pointer is moving as commanded.

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

- 1 = Gauge 1 pointer position has changed since the last SPI command

ST4—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. The master confirms the pointer is moving as commanded.

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

ST3—RTZ1—Enabled successful 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 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 successful

ST2—RTZ0—Enabled successful 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 indicating 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 successful

ST1—Gauge 1—Junction over temperature. A logic [1] on this bit indicates coil drive circuitry dedicated to drive Gauge 1 has exceeded the maximum allowable junction temperature since the last SPI communication. Additionally, the same indication signals the circuitry Gauge 1 is disabled. It is recommended the pointer be re-calibrated using the RTZ command after re-enabling the gauge using the PECR 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

ST0—Gauge 0—Junction over temperature. A logic [1] on this bit indicates coil drive circuitry dedicated to drive Gauge 0 has exceeded the maximum allowable junction temperature since the last SPI communication. Additionally, the same indication signals the circuitry Gauge 0 is disabled. It is recommended the pointer be re-calibrated using the RTZ command after re-

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

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

DEVICE FUNCTIONAL DESCRIPTION

State Machine Operation

The two-phase stepper motor is defined as maximum velocity and acceleration, and deceleration. It is the purpose of the stepper motor state machine to drive the motor with maximum performance while remaining within the motor's velocity and acceleration constraints.

When commanded, the motor should accelerate constantly to its maximum velocity then decelerate and stop at the desired position. During the deceleration phase, the motor should not exceed the maximum deceleration. A required function of the state

machine is to ensure the deceleration phase begins at the correct time, or position.

During normal operation, both stepper motor rotors are microstepped with 24 steps per electrical revolution. See Figure 6. A complete electrical revolution results in 2 degrees of pointer movement. There is a second, and 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 can be seen in Table 13.

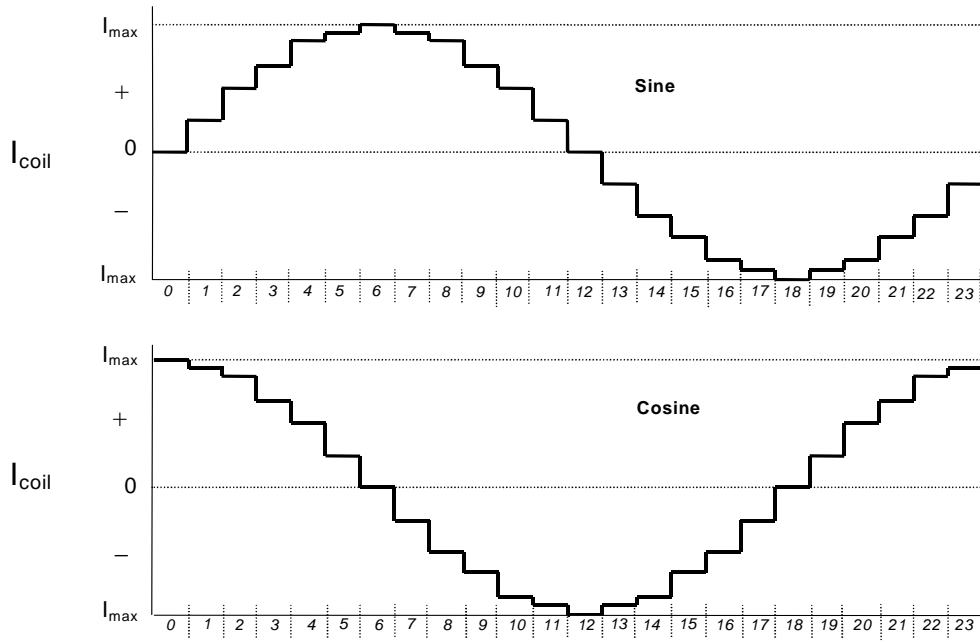


Figure 6. Microstepping

Table 13. Coil Step Value

STEP#	ANGLE	SINE Angle*	SINE Current Flow	8-Bit Value (DEC)	8-Bit Value (HEX)	COS Angle*	COS Current Flow	8-Bit Value (DEC)	8-Bit Value (HEX)
0	0	0	+	0	0	1	+	255	FF
1	15	0.259	+	66	42	0.965	+	247	F7
2	30	0.5	+	128	80	0.866	+	222	DE
3	45	0.707	+	181	B5	0.707	+	181	B5
4	60	0.866	+	222	DE	0.5	+	128	80
5	75	0.966	+	247	F7	0.259	+	66	42
6	90	1	+	255	FF	0	+	0	0
7	105	0.966	+	247	F7	-0.259	-	66	42
8	120	0.866	+	222	DE	-0.5	-	128	80
9	135	0.707	+	181	B5	-0.707	-	181	B5
10	150	0.5	+	128	80	-0.866	-	222	DE
11	165	0.259	+	66	42	-0.966	-	247	F7
12	180	0	+	0	0	-1	-	255	FF
13	195	-0.259	-	66	42	-0.966	-	247	F7
14	210	-0.5	-	128	80	-0.867	-	222	DE
15	225	-0.707	-	181	B5	-0.707	-	181	B5
16	240	-0.866	-	222	DE	-0.5	-	128	80
17	255	-0.966	-	247	F7	-0.259	-	66	42
18	270	-1	-	255	FF	0	+	0	0
19	285	-0.966	-	247	F7	0.259	+	66	42
20	300	-0.866	-	222	DE	0.5	+	128	80
21	315	-0.707	-	181	B5	0.707	+	181	B5
22	330	-0.5	-	128	80	0.866	+	222	DE
23	345	-0.259	-	66	42	0.966	+	247	F7

Notes: * 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.

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

The state machine uses a table defining the allowed time steps, including the maximum velocity. A useful side effect of the table is, it also allows the direct determination of the position the velocity should reduce to allow the motor to stop at the desired position.

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

The motor equations of motion are generated as follows:

Using

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

The units of position are steps, and velocity and acceleration are in steps/second, and steps/second².

and

$$v = u + at$$

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

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

For unit steps, the time between steps is:

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

$$v = \frac{2s}{t} - u \qquad p_n = n$$

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 .

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

$$\Delta t_n = \left\lceil \frac{-v_{n-1} + \sqrt{v_{n-1}^2 + 2a}}{a} \right\rceil, \text{ where } \lceil \rceil \text{ indicates}$$

rounding up.

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

Note: $p_n = n$

This means: on the n th 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. In other words, the *stopping distance* is also equal to the current value of n .

The algorithm to drive the motor is similar to:

- While the motor is stopped, wait until a command is received.
- Send index pulses to the motor at an ever-increasing rate, according to the time steps in the table above until:
 - The maximum velocity is reached, at this point the time intervals stop decreasing

or,

- 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, ensuring it will stop at the required position, the motor must begin decelerating.

An example of the table for a particular motor is provided in Table 14. This motor's maximum speed is 4800 microsteps/s (at 12 microsteps/degree), and its maximum acceleration is 54000 microsteps/s². The table is quantized to a 1 MHz clock.

Table 14. Velocity Ramp

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	72	363	2771.81	144	255	3931.78
1	16383	122.08	73	360	2791.22	145	255	3945.49
2	6086	350.58	74	358	2810.50	146	254	3959.15
3	2521	480.52	75	355	2829.65	147	253	3972.77
4	1935	582.15	76	353	2848.67	148	252	3986.34
5	1631	668.51	77	351	2867.56	149	251	3999.86
6	1437	744.92	78	348	2886.33	150	250	4013.34
7	1299	814.19	79	346	2904.98	151	249	4026.77
8	1195	878.01	80	344	2923.51	152	249	4040.16
9	1112	937.50	81	342	2941.92	153	248	4053.51
10	1045	993.43	82	340	2960.22	154	247	4066.81
11	988	1046.38	83	338	2978.41	155	246	4080.06
12	940	1096.77	84	336	2996.48	156	245	4093.28
13	898	1144.95	85	334	3014.45	157	245	4106.45
14	861	1191.18	86	332	3032.31	158	244	4119.58

Table 14. Velocity Ramp

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)
15	829	1235.68	87	330	3050.07	159	243	4132.66
16	800	1278.63	88	328	3067.72	160	242	4145.71
17	773	1320.19	89	326	3085.27	161	241	4158.71
18	750	1360.48	90	324	3102.73	162	241	4171.68
19	728	1399.61	91	322	3120.08	163	240	4184.60
20	708	1437.67	92	320	3137.34	164	239	4197.49
21	690	1474.76	93	319	3154.51	165	238	4210.33
22	673	1510.93	94	317	3171.58	166	238	4223.14
23	657	1546.25	95	315	3188.56	167	237	4235.91
24	642	1580.79	96	314	3205.45	168	236	4248.64
25	628	1614.59	97	312	3222.25	169	236	4261.33
26	615	1647.70	98	310	3238.97	170	235	4273.98
27	603	1680.15	99	309	3255.60	171	234	4286.60
28	592	1711.99	100	307	3272.14	172	234	4299.17
29	581	1743.24	101	306	3288.60	173	233	4311.72
30	571	1773.95	102	304	3304.98	174	232	4324.22
31	561	1804.13	103	303	3321.28	175	232	4336.69
32	552	1833.82	104	301	3337.50	176	231	4349.13
33	543	1863.04	105	300	3353.64	177	230	4361.53
34	534	1891.80	106	298	3369.70	178	230	4373.89
35	526	1920.13	107	297	3385.69	179	229	4386.22
36	519	1948.05	108	295	3401.60	180	228	4398.51
37	511	1975.58	109	294	3417.44	181	228	4410.77
38	504	2002.72	110	293	3433.21	182	227	4423.00
39	497	2029.51	111	291	3448.90	183	226	4435.19
40	491	2055.94	112	290	3464.52	184	226	4447.35
41	485	2082.04	113	289	3480.07	185	225	4459.47
42	479	2107.82	114	287	3495.55	186	225	4471.57
43	473	2133.28	115	286	3510.97	187	224	4483.63
44	467	2158.45	116	285	3526.32	188	223	4495.65
45	462	2183.32	117	284	3541.60	189	223	4507.65
46	457	2207.92	118	282	3556.81	190	222	4519.61
47	452	2232.24	119	281	3571.96	191	222	4531.55

Table 14. Velocity Ramp

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)
48	447	2256.30	120	280	3587.05	192	221	4543.45
49	442	2280.11	121	279	3602.07	193	220	4555.32
50	437	2303.67	122	278	3617.03	194	220	4567.15
51	433	2326.99	123	277	3631.93	195	219	4578.96
52	429	2350.09	124	275	3646.77	196	219	4590.74
53	425	2372.95	125	274	3661.54	197	218	4602.49
54	420	2395.60	126	273	3676.26	198	218	4614.21
55	417	2418.04	127	272	3690.92	199	217	4625.89
56	413	2440.27	128	271	3705.52	200	216	4637.55
57	409	2462.30	129	270	3720.07	201	216	4649.18
58	405	2484.13	130	269	3734.56	202	215	4660.78
59	402	2505.77	131	268	3748.99	203	215	4672.36
60	398	2527.23	132	267	3763.36	204	214	4683.90
61	395	2548.51	133	266	3777.68	205	214	4695.41
62	392	2569.61	134	265	3791.95	206	213	4706.90
63	389	2590.54	135	264	3806.17	207	213	4718.36
64	385	2611.30	136	263	3820.33	208	212	4729.79
65	382	2631.90	137	262	3834.44	209	212	4741.19
66	379	2652.34	138	261	3848.49	210	211	4752.57
67	376	2672.62	139	260	3862.50	211	211	4763.92
68	374	2692.75	140	259	3876.45	212	210	4775.24
69	371	2712.73	141	258	3890.36	213	210	4786.53
70	368	2732.56	142	257	3904.22	214	209	4797.80
71	366	2752.25	143	256	3918.02	215	209	4800.00

Internal Clock Calibration

Timing related functions on the 33991 device (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; however, they tend to be costly due to the large amount of die area required for trim pads and the associated trim procedure. One possibility to reduce cost is an externally generated clock signal. An external generated clock requires a dedicated pin on the device and on the controller. Another inexpensive approach would be to require the use of an additional crystal or resonator.

The internal clock in the 33991 is temperature independent and area efficient; however, it can vary by as much as + 70 percent to - 35 percent due to process variation. Using the existing SPI inputs and the precision timing reference already available to the controller, the 33991 allows clock calibration to within ± 10 percent.

Calibrating the internal 1 MHz clock is initiated by writing a logic [1] to PECR bit D3. See Figure 7. The 8 μs calibration pulse is provided by the controller. It ideally results in an internal 33991 clock speed of 1 MHz. The pulse is sent on the CS pin immediately after the SPI word is launched. No other SPI lines must be toggled. At the moment the CS pin transitions from logic [1] to [0], an internal 7-bit counter counts the number of cycles of an internal, non-calibrated, and temperature independent, 8 MHz clock. The counter stops when the CS pin transitions from logic [0] to logic [1]. The value in the counter represents the number of cycles of the 8 MHz clock occurring in the 8 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 four bits, generating the calibration divisor, ranging from four to 15. The 8 MHz clock is divided by the calibration divisor, resulting in a calibrated 1 MHz clock. If the calibration divisor lies outside the range of four to 15, then the 33991 device flags the ST7 bit, indicating the

calibration procedure was not successful. A clock calibration is allowed only if the gauges are disabled or the pointers are not moving, indicated by status bits ST4 and ST5.

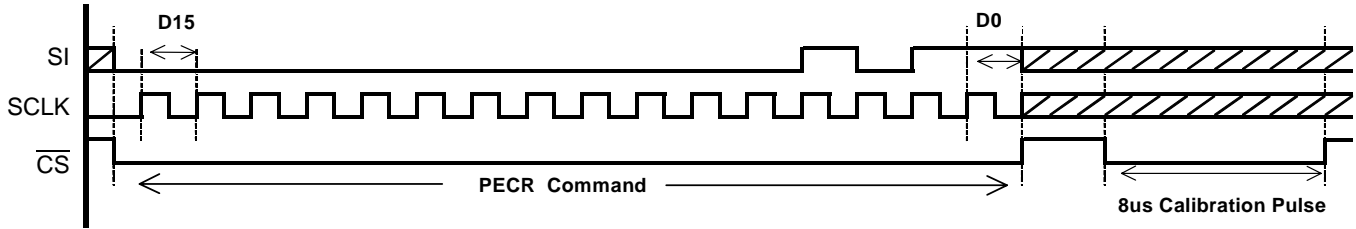


Figure 7. Gauge Enable and Clock Calibration Example

Some applications may require a guaranteed maximum pointer velocity and acceleration. Guaranteeing these maximums requires the nominal internal clock frequency fall below 1 MHz. The frequency range of the calibrated clock is always below 1 MHz if PECR bit D4 is logic [0] when initiating a calibration command, followed by an 8 μ s reference pulse. The frequency is centered at 1 MHz if bit D4 is logic [1].

The 33991 can be deceived into calibrating faster or slower than the optimal frequency by sending a calibration pulse longer or shorter than the intended 8 μ s. As long as the count remains between four and 15 there is no clock calibration flag. For applications requiring a slower calibrated clock, i.e., a motor designed with a gear ratio of 120:1 (8 microsteps/degrees), a longer calibration pulse is required. The device allows a SPI selectable slowing of the internal oscillator, using the PECR command so the calibration divisor safely falls within the four to 15 range when calibrating with a longer time reference. For example, for the 120:1 motor the pulse would be 12 μ s instead

of 8 μ s. The result of this slower calibration results in the longer step times necessary to generate pointer movements meeting the acceleration and velocity requirements. The resolution of the pointer positioning decreases from 0.083°/microstep (180:1) to 0.125°/microstep (120:1). The pointer sweep range increases from approximately 340° to over 500°.

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

Pointer Deceleration Waveshaping

Constant acceleration and deceleration of the pointer results in choppy movements when compared to air core movements. Air core behavior can be simulated with appropriate wave shaping during deceleration only. This shaping is achieved by adding repetitive steps at several of the last step values. An example is illustrated in Figure 8.

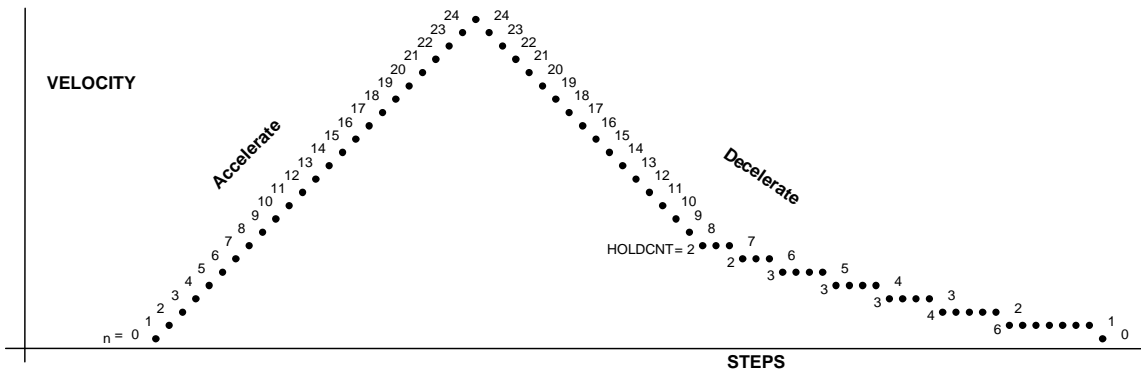


Figure 8. Deceleration Waveshaping

Return to Zero Calibration

Many stepper motor applications require the integrated circuit (IC) detect when the motor is stalled after commanded to return to the zero position for calibration purposes. Stalling occurs when the pointer hits the end stop on the gauge bezel, usually at the zero position. It is important when the pointer reaches the end stop it immediately stops without *bouncing* away from the stop.

The 33991 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. During an RTZ event, all commands related to the gauge being returned are ignored, except when the RTZR bit D1 is used to disable the event, or when the RTZR bits D3 and D2 are changed in order to look at different RTZ accumulator bits. Once an RTZ event is initiated, the device will report back via the SO pin, indicating 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. Reaching the end stop, the device reports the RTZ success to the microcontroller via the SO pin. The RTZ automatically disables allowing other commands to be valid. In the event the master determines an RTZ sequence is not working properly, for example the RTZ taking too long, it can disable the command via the RTZR bit D1.

RTZCR bits D12:D5 are written to preload the accumulator with a predetermined value assuring 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 D4. This operating mode allows the master to monitor the RTZ event using the accumulator information available in the SO status bits D15:D8. Once the optimal value is determined, the RTZ event can be turned off using the RTZR bit D1.

During an RTZ event, the pointer is returned counter-clockwise (CCW) using full steps at a constant speed determined by the RTZCR D3:D0 bits during RTZ configuration. See Figure 9. Full steps are used because only one coil of the motor is being driven at any time. The coil not being driven is used to determine whether the pointer is moving. If the pointer is moving, a back EMF signal can be processed and detected in the non-driven coil. This is achieved by integrating the signal present on an opened end of the non-driven coil while grounding the opposite end.

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 back EMF 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. When the threshold value is not exceeded, the drive sequence is stopped when RTZR bit D4 is logic [0]. If bit D4 is logic [1], the RTZ movement will continue indefinitely until the RTZR bit D1 is used to stop the RTZ event.

A pointer not on a full step location, or in magnetic alignment prior to the RTZ event, may result in a false RTZ detection. More specifically, an RTZ event beginning from a non-full step position may result in an abbreviated integration, interpreted as a stalled pointer. Similarly, if the magnetic fields of the energized coils and the rotor are not aligned prior to initiating the RTZ, the integration results may mistakenly indicate the pointer has stopped moving.

Advancing the pointer by at least 24 microsteps clock-wise (CW) to the nearest full step position, e.g., 24,30, 36, 42, 48 ...prior to initiating an RTZ, ensures the magnetic fields are aligned. Doing that increases the chances of a successful pointer stall detection. It is important the pointer be in a static, or commanded position before starting the RTZ event. Since the time duration and the number of steps the pointer moves prior to reaching the commanded position can vary depending on its status at the time a position change is communicated, the master should assure sufficient time has elapsed 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 24 microstep steps clock-wise (CW) 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 took by monitoring and counting the ST4 (ST5) status bit transitions, confirming the pointer is again in a static position.

Only one gauge at a time can be returned to the zero position. 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, will be 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 pointer back to the previous zero position and then re-calibrating the pointers.

After completion of a RTZ, the 33991 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 $\pm 0.5^\circ$. 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 should be advanced by at least 24 microsteps to the nearest full step prior to initiating the RTZ.

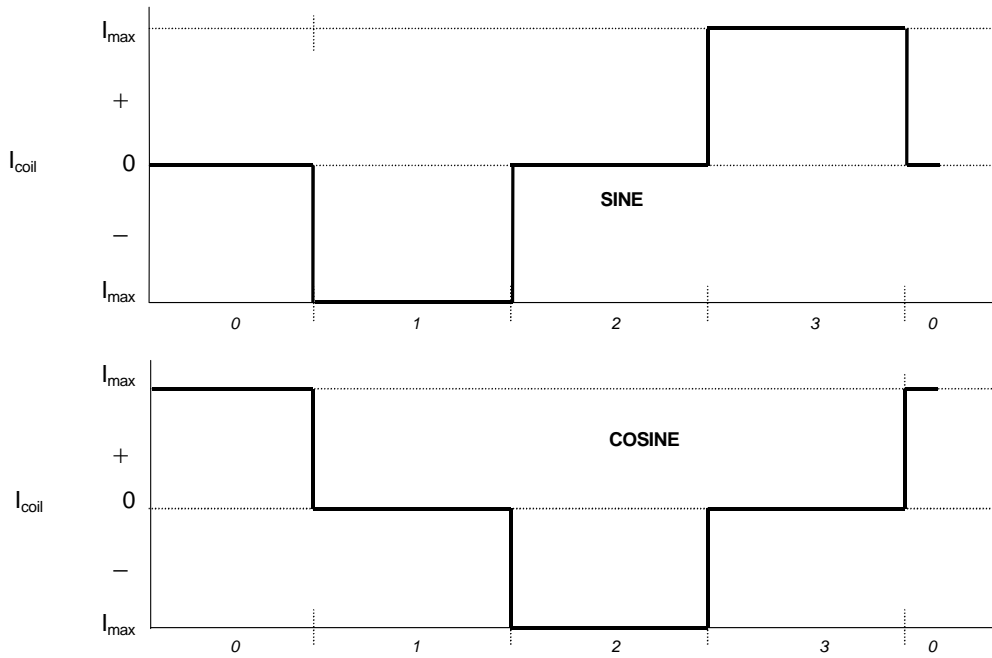


Figure 9. Full Steps (CCW)

RTZ Output

The non-driven coil is analyzed determining the state of the motor during an RTZ event. The 33991 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 33991 after an internal or external reset and prior to SPI communication. An internal reset occurs during V_{DD} power-up. An external reset is initiated by the \overline{RST} pin driven to a logic [0]. With the exception of the RTZ_{CR} full step time, 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, SO flag ST6 is set, indicating an under voltage 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 33991 device indicates each of the following faults as they occur:

- Over temperature fault
- Out-of-range voltage faults
- Clock out of spec

Over current faults are not reported directly. It is likely, however, an over current condition will become a thermal issue and be reported.

Over Temperature Fault Requirements

The 33991 incorporates over temperature protection circuitry, shutting off the affected Gauge Driver when excessive temperatures are measured. In the event of a thermal overload, the affected gauge driver is automatically disabled. The over temperature fault is flagged via ST0 and/or ST1. Its respective flag continues to be set until the affected gauge is successfully re-enabled, provided the junction temperature has fallen below the hysteresis level.

Over Voltage Fault Requirements

The device is capable of surviving V_{PWR} voltages within the maximum specified in the *Maximum Ratings* table. V_{PWR} levels resulting in an Over Voltage Shut Down condition can result in uncertain pointer positions. Therefore, the pointer position should be re-calibrated. The master is then notified of an over voltage event via the ST6 flag on the SO pin. Over voltage detection and notification occurs regardless of whether the gauge(s) are enabled or disabled.

Note: There is no way to distinguish between an over voltage fault and an under voltage fault from the status bits. If there is no external means for the microcontroller to determine the fault type, the gauges should be routinely enabled following the transition to logic [0] of ST6.

Over Current Fault Requirements:

Output currents are limited to safe levels, then the device will rely on Thermal Shutdown to protect itself.

Under Voltage Fault Requirements

Severe under voltage V_{PWR} conditions may result in uncertain pointer positions; therefore, recalibration of the pointer position may be advisable. During an under voltage event, the state machine and outputs continues to operate, although the outputs may be unable to reach the higher voltage levels. The master is notified of an under voltage event via the SO pin. Under voltage detection occurs regardless whether the gauge(s) are enabled, or disabled.

Note: There is no way to distinguish between over voltage and under voltage faults from the status bits. If there is no external means for the microcontroller to determine the fault type, the gauges should be routinely enabled following the transition to logic [0] of ST6.

Electrical Requirements

All voltages specified are measured relative to the device ground pins, unless otherwise noted. Current flowing into 33991 is positive, while current flowing out of the device is negative.

Resets (Sleep Mode)

The device can reset internally or externally. If the V_{DD} level falls below the V_{DDUV} level, specified in the *Static Electrical Characteristics*, the device resets and powers up in the Default mode. Similarly, If the RST pin is driven to a logic [0], the device will reset to its default state. The device will consume 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.

APPLICATION INFORMATION

The 33991 is an extremely versatile device, used in a variety of applications. Table 15, and the following sample code, provides a step-by-step example of configuring using many of the features designed into the IC. This example is intended to give a generic overview of how the device could be used. Further, it is intended to familiarize users with some of its capabilities. In Steps 1-9, the gauges are enabled, the clock is calibrated, the device is configured for RTZ, and the pointers are calibrated with the RTZ command. Steps 1-9 are

representative of the first steps after power-up. Maximum velocity is set in Step 10, if necessary. In Steps 11 and 12, pointers are commanded to the desired positions by the master. These steps are the most frequently used during normal operation. Steps 13 -15 place the pointer close to the zero position prior to the initiation of the RTZ commands in Steps 16-19. Step 20 disables the gauges, placing them into a Low Quiescent Current mode.

Table 15. GDIC SETUP, CONFIGURATION, & USAGE EXAMPLE

Step Number	Command	Description	Table/Figure Number
1	PECR	a. Enable Gauges. - Bit PE0: Gauge 0. - Bit PE1: Gauge 1.	Table 3 Figure 7
		b. Clock Calibration - Bit PE3: enables calibration procedure. - Bit PE4: set clock f =1 MHz maximum or nominal.	
		Send 8 μ s pulse on \overline{CS} to calibrate 1 MHz clock.	
2	RTZCR	Set RTZ Full Step Time - Bits RC3:RC0.	Tables 9-10
		Set RTZ Blanking Time - Bit RC4.	Tables 9-10
		Preload RTZ Accumulator. - Bits RC12:RC5.	Table 11
		Check SO for an Out of Range Clock Calibration. - Is bit ST7 logic level 1? If so, repeat Steps 1 and 2.	Table 12
3	POS0R	a. Move pointer to position 24 prior to RTZ.	Table 5

Table 15. GDIC SETUP, CONFIGURATION, & USAGE EXAMPLE (continued)

Step Number	Command	Description	Table/Figure Number
4	POS1R	Move pointer to position 24 prior to RTZ.	Table 6
		Check SO to see if Gauge 0 has moved. - Is bit ST4 logic level 1? If so, gauge 0 has moved to the first microstep.	Table 12
5	PECR	Send null command to see if gauges have moved. - Bit PE12.	Table 3
		Check SO to see if Gauge 0 (Gauge 1) has moved - Is bit ST4 (ST5) logic level 1? If so, Gauge 0 (Gauge 1) has moved another microstep. Keep track of the movement. If 24 steps are finished, and both gauges are at a static position, then RTZ. Otherwise repeat steps a) and b).	Table 12
6	RTZ	a. 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 RZ3:RZ2 are used to select the RTZ accumulator bits clocking out on the SO pin.	Table 7
		b. Select the RTZ accumulator bits clocking out on the SO bits ST15:ST8. These will be used if characterizing the RTZ. - Bits RZ3: RZ2 are used to select the bits.	Table 8
7	PECR	a. Check the Status of the RTZ by sending the null command to monitor SO bit ST2 - Bit PE12 is the null command.	Table 3
		Is ST2 logic level 0? If not, Gauge 0 is still returning and null command should be resent.	Table 12
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.	Tables 7-8
9	PECR	a. Check the Status of the RTZ by sending the null command to monitor SO bit ST3 - Bit PE12 is the null command.	Table 3
		Is ST3 logic level 0? If not, Gauge 1 is still returning. Null command should be resent.	Table 12
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 the acceleration table.	Table 4
11	POS0R	Position Gauge 0 pointer - Bits P0 11: P0 0: Desired Pointer Position.	Table 5
		Check SO for Out of Range V_{PWR} - Bit ST6 logic level 1. If so, RTZ after valid V_{PWR} .	Table 12
		Check SO for Over Temperature bit ST0 logic level 1. If so, enable driver again. If ST0 continues to indicate Over Temperature, shut down Gauge 0. If ST2 returns to normal, reestablish the zero reference by RTZ command.	
12	POS1R	Position Gauge 1 pointer - Bits P1 11:P1 0: Desired Pointer Position	Table 6
		Check SO for Out of Range V_{PWR} bit ST6 logic level 1. If so, RTZ after valid V_{PWR} .	Table 12
		Check SO for Over Temperature bit ST1 logic level 1. If so, enable driver again. If ST1 continues to indicate over temperature, shut down Gauge 1. If ST1 returns to normal, reestablish the zero reference by RTZ command.	
13	POS0R	a. Return the pointers close to zero position using POS0R	Table 5
		b. Move pointer position at least 24 microsteps CW to the nearest full step prior to RTZ.	

Freescale Semiconductor, Inc.

Table 15. GDIC SETUP, CONFIGURATION, & USAGE EXAMPLE (continued)

Step Number	Command	Description	Table/Figure Number
14	POS1R	Return the pointers close to zero position using POS1R	Table 6
		Move pointer position at least 24 microsteps CW to the nearest full step position prior to RTZ.	
		Check SO to see if Gauge 0 has moved. - Is bit ST4 logic level 1? If so, Gauge 0 has moved to the first microstep.	Table 12
15	PECR	Send null command to see if gauges have moved. - Bits PE12	Table 3
		Check SO to see if Gauge 0 (Gauge 1) has moved. - Is bit ST4 (ST5) logic level 1? If so, Gauge 0 (Gauge 1) has moved another microstep. Keep track of movement. If 24 steps are finished and both gauges are at a static position, then RTZ. Otherwise repeat steps a) and b).	Table 12
16	RTZ	a. Return one gauge at a time to the zero stop using RTZ. Command bit RZ0 selects the gauge bit RZ1 used to enable or disable an RTZ. - Bits RZ3:RZ2 are used to select the RTZ accumulator bits clocking out on the SO pin.	Tables 7-8
		b. Select the RTZ accumulator bits clocking out on the SO bits ST15:ST8. These will be used if characterizing the RTZ. - Bits RZ3:RZ2 are used to select the bits.	
17	PECR	a. Check the status of the RTZ by sending the null command to monitor SO bit ST2 - Bit PE12 is the null command.	Table 3
		Is ST2 logic level 0? If not, Gauge 0 is still returning. Null command should be resent.	Table 12
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.	Tables 7-8
19	PECR	a. Check the Status of the RTZ by sending the null command to monitor SO bit ST3 - Bit PE12 is the null command.	Table 3
		Is ST3 logic level 0? If not, Gauge 1 is still returning. Null command should be resent.	Table 12
20	PECR	Disable both gauges and go to standby bit PE0:PE1 are used to disable the gauges Put the device to sleep. - $\overline{\text{RST}}$ pin is pulled to logic 0.	Table 3

SAMPLE CODE

/ The following example code demonstrates a typical set up configuration for a M68HC912B32. */*
/ This code is intended for instructional use only. Motorola assumes no liability for use or */*
/ modification of this code. It is the responsibility of the user to verify all parameters, variables, */*
/ timings, etc. */*

```

void InitGauges(void)
{
    /* Step 1 */
    Command_Gauge(0x00,0x03);           /* Enable Gauges */
    Command_Gauge(0x00,0x08);         /* Clock Cal bit set */
        /* 8 uSec calibration */
        PORTS = 0x00;                   /* Enable GDIC CS pin - PORTS2 */
        For (cnt = 0; cnt < 5; cnt++)
        {
            NOP;                       /* Wait for 8 uSec calibration */
        }
        PORTS = 0x04;                   /* Disable GDIC CS pin - PORTS2 */

    /* Step 2 */
    Command_Gauge(0xA0,0x21);         /* Send RTZCR values */
    Command_Gauge(0x10,0x00);         /* Null Read to get SO status */
        /* Check SO bits for Out of Range Clock Calibration */
        If ((status & 0x80) != 0)
        /* If Clock is out of range then recalibrate 8 uSec pulse */

    /* Step 3 */
    Command_Gauge(0x40,0x18);         /* Send position to gauge0 */

    /* Step 4 */
    Command_Gauge(0x60,0x18);         /* Send position to gauge1 */
        /* Check SO bit ST4 to see if Gauge 0 has moved */
        If((status & 0x10) != 0)
        /* If ST4 is logic 1 then Gauge 0 has moved to the first microstep */

    /* Step 5 */
    Command_Gauge(0x10,0x00);         /* Null Read to get SO status */
        /* Check SO bit ST4 to see if Gauge 0 has moved */
        If((status & 0x10) != 0)
        /* If it has moved, then keep track of position */
        /* Wait until 24 steps are finished then send a RTZ command (Step 7) */

    /* Step 6 */
    Command_Gauge(0x80,0x02);         /* Send RTZ to Gauge 0 */

    /* Step 7 */
    Command_Gauge(0x10,0x00);         /* Null Read to get status */
        /* Read Status until RTZ is done */
        While ((status & 0x04) != 0)
            { Command_Gauge(0x10,0x00); }

```



```

/* Step 8 */
Command_Gauge(0x80,0x03);          /* Send RTZ to Gauge 1 */

/* Step 9 */
Command_Gauge(0x10,0x00);          /* Null Read to get status */
/* Read Status until RTZ is done */
While ((status & 0x08) != 0)
    { Command_Gauge(0x10,0x00); }

/* Step 10 */
Command_Gauge(0x23,0xFF);          /* Send velocity */

/* Step 11 */
Command_Gauge(0x4F,0xFF);          /* Send position to gauge0 */
/* Check SO bits for Out of Range Vpwr and Overtemperature */
If((status & 0x40) != 0)
/* If bit ST6 is logic 1 then RTZ after valid Vpwr */

If((status & 0x01) != 0)
/* If bit ST0 is logic 1 then enable driver again.
/* If ST0 continues to indicate over temperature, then shut down Gauge 0. */
/* If ST2 returns to normal, then reestablish the zero reference by RTZ command. */

/* Step 12 */
Command_Gauge(0x6F,0xFF);          /* Send position to gauge1 */
/* Check SO bits for Out of Range Vpwr and Over-Temperature */
If((status & 0x40) != 0)
/* If bit ST6 is logic 1 then RTZ after valid Vpwr */

If((status & 0x01) != 0)
/* If bit ST0 is logic 1 then enable driver again.
/* If ST0 continues to indicate Over-Temperature, then shut down Gauge 1. */
/* If ST2 returns to normal, then reestablish the zero reference by RTZ command. */

/* Step 13 */
Command_Gauge(0x40,0x00);          /* Send position to Gauge 0 */
/* Return the pointers close to zero position */
Command_Gauge(0x40,0x18);          /* Send position to Gauge 0 */
/* Move the pointer at least 24 microsteps CW to the nearest full step */

/* Step 14 */
Command_Gauge(0x60,0x00);          /* Send position to Gauge 1 */
/* Return the pointers close to zero position */
Command_Gauge(0x60,0x18);          /* Send position to Gauge 1 */
/* Move the pointer at least 24 microsteps CW to the nearest full step */

/* Check SO bit ST4 to see if Gauge 0 has moved */
If((status & 0x10) != 0)
/* If ST4 is logic 1 then Gauge 0 has moved to the first microstep */

```

```

/* Step 15 */
    Command_Gauge(0x10,0x00);          /* Null Read to get status */
    /* Check SO bit ST4 to see if Gauge 0 has moved */
    If((status & 0x10) != 0)
    /* If it has moved, then keep track of position */
    /* Wait until 24 steps are finished then send a RTZ command (Step 17) */

/* Step 16 */
    Command_Gauge(0x80,0x02);          /* Send RTZ to Gauge 0 */

/* Step 17 */
    Command_Gauge(0x10,0x00);          /* Null Read to get status */
    /* Read Status until RTZ is done */
    While ((status & 0x04) != 0)
        {Command_Gauge(0x10,0x00);}

/* Step 18 */
    Command_Gauge(0x80,0x03);          /* Send RTZ to Gauge 1 */

/* Step 19 */
    Command_Gauge(0x10,0x00);          /* Null Read to get status */
    /* Read Status until RTZ is done */
    While ((status & 0x08) != 0)
        {Command_Gauge(0x01,0x00);}

/* Step 20 */
    Command_Gauge(0x00,0x00);          /* Disable Gauges and go into Standby */
    /* Put device to sleep by setting RSTB to logic 0 */
}

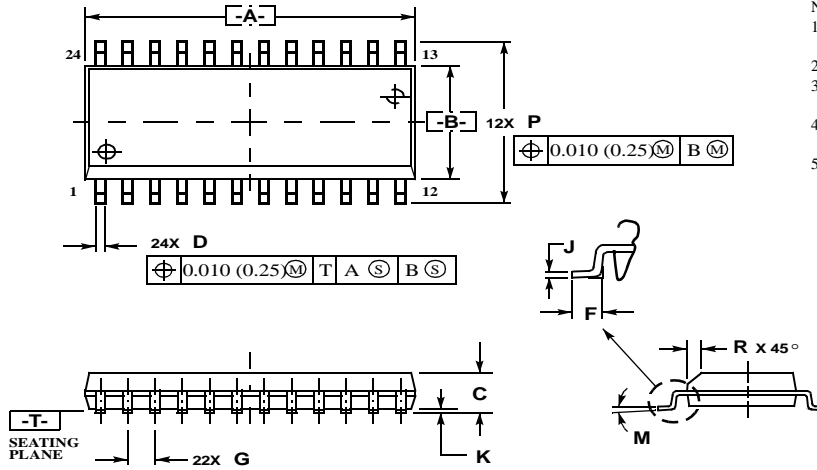
void Command_Gauge(char MSB, char LSB)
/*This subroutine sends the GDIC commands on the SPI port */
{
    PORTS = 0x00;                       /* Chip select low (active) */
    SP0DR = MSB;                         /* send first byte of gauge command */
    While ((SP0SR & 0x80) == 0);         /* wait for Rxflag (first byte) */
    RTZdata = SP0DR;                     /* Read status MSB */
    SP0DR = LSB;                         /* send second byte of command */
    While ((SP0SR & 0x80) == 0);         /* wait for Rxflag (second byte) */
    status = SP0DR;                      /* read status LSB */
    PORTS = 0x04;                       /* Chip select high (deactivated) */
}

/* Motorola Semiconductor Products Sector */
/* October 4, 2002 */

```

PACKAGE DRAWING

DW SUFFIX
 PLASTIC PACKAGE
 CASE 751E-04
 SOICW
 Issue E



NOTES:

1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION.
4. MAXIMUM MOLD PROTRUSION 0.15 (0.006) PER SIDE.
5. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.13 (0.005) TOTAL IN EXCESS OF D DIMENSION

DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	15.25	15.54	0.601	0.612
B	7.40	7.60	0.292	0.299
C	2.35	2.65	0.093	0.104
D	0.35	0.49	0.014	0.019
F	0.41	0.90	0.016	0.035
G	1.27 BSC		0.050 BSC	
J	0.23	0.32	0.009	0.013
K	0.13	0.29	0.005	0.011
M	0°	8°	0°	8°
P	10.05	10.55	0.395	0.415
R	0.25	0.75	0.010	0.029

Freescale Semiconductor, Inc.

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. Motorola and are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

MOTOROLA and the Stylized M Logo are registered in the US Patent and Trademark Office. All other product or service names are the property of their respective owners.

© Motorola, Inc. 2002

HOW TO REACH US:

USA/EUROPE/LOCATIONS NOT LISTED: Motorola Literature Distribution: P.O. Box 5405, Denver, Colorado 80217.

1-303-675-2140 or 1-800-441-2447

JAPAN: Motorola Japan Ltd.; SPS, Technical Information Center, 3-20-1 Minami-Azabu. Minato-ku, Tokyo 106-8573 Japan.

81-3-3440-3569

ASIA/PACIFIC: Motorola Semiconductors H.K. Ltd.; Silicon Harbour Centre, 2 Dai King Street, Tai Po Industrial Estate, Tao Po, N.T.,

Hong Kong. 852-26668334

TECHNICAL INFORMATION CENTER: 1-800-521-6274



MOTOROLA

**For More Information On This Product,
Go to: www.freescale.com**

MC33991/D