NCS32100 Reference Design User's Guide

TND6377/D

Introduction

The NCS32100 reference design includes all the necessary electronics and firmware to evaluate and demonstrate a full rotational inductive encoder. It is a design example highlighting the electrical connections and the firmware used on the NCS32100 to provide a select number of features and performance using a PCB sensor. The NCS32100 reference design is a base starting point upon which specific application designs can be built. The NCS32100 datasheet is available as separate supporting documentation. The STR−NCS32100−GEVK evaluation board uses the reference design directly and is available for further understanding and evaluation of the NCS32100. The datasheet details all hardware capabilities including some that are not used in this reference design.

Reference Design Features

- Full Absolute Encoder Output (20−bit Resolution + 24 bits multi−turn)
- Sensor capable of ±50 arcsec Accuracy (using a 38 mm diameter PCB rotor and stator)
- 5 V Half−Duplex RS−485 Interface
- Backup Battery Mode Capable
- Over Temperature Readout
- Battery Readout for detecting low battery voltage

ANATOMY OF A FULL ROTARY ENCODER SYSTEM USING THE NCS32100

A complete rotary encoder system requires a PCB rotor, PCB stator, a center shaft, and housing, in addition to the NCS32100. A cross section of this system is shown below in Figure 1. The Stator PCB houses the NCS32100 and any connectors that are needed to power the device and connect the data pins outside the module housing. This reference manual focuses on the stator circuitry and connections to the NCS32100. Although this reference design uses a 38 mm PCB sensor, the size of the rotor and stator can be designed for larger or smaller diameters.

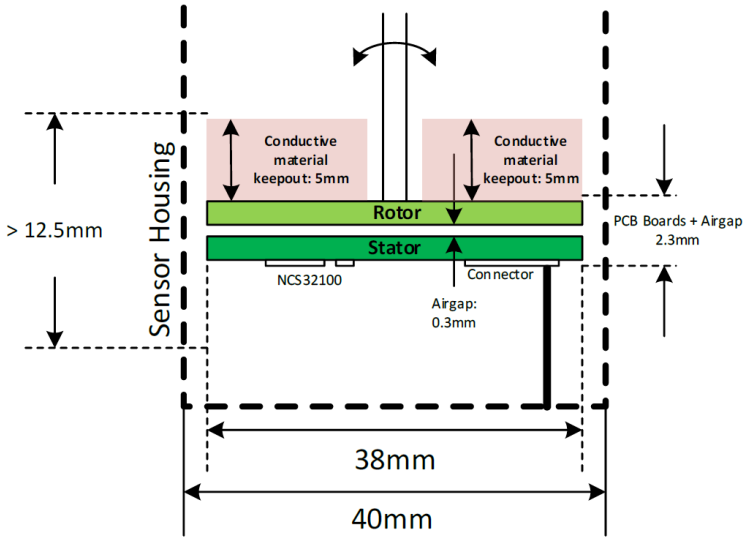


Figure 1. Full Rotary Encoder Anatomy

OPERATION DESCRIPTION

A simplified block diagram of the NCS32100 reference design (Based on the STR−NCS32100−EVK) is shown in Fig 2.

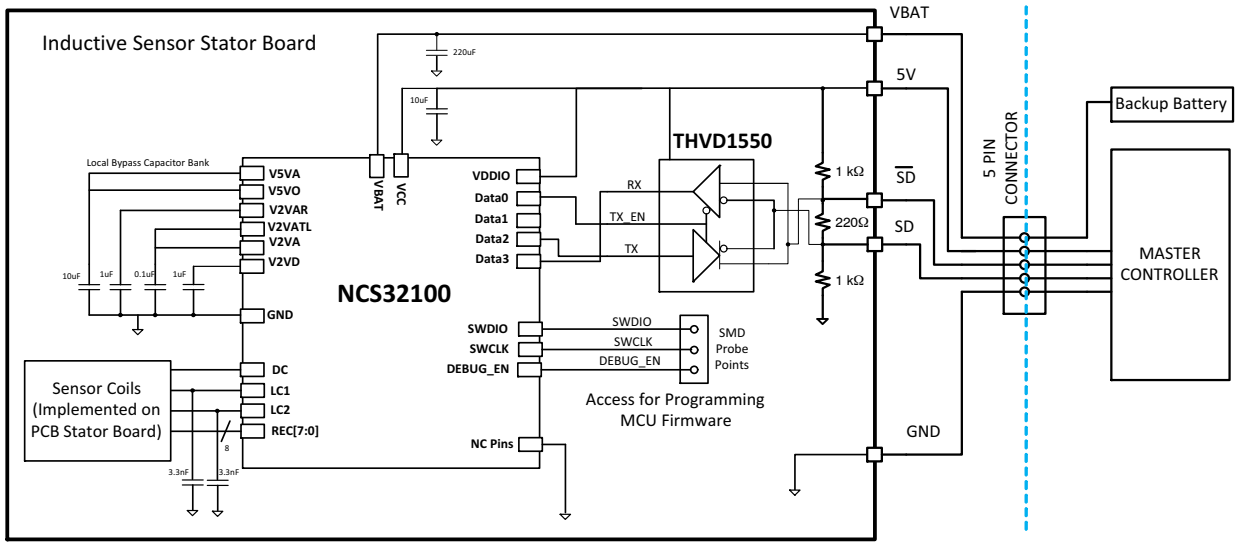


Figure 2. NCS32100 Reference Design Block Diagram Left of the dashed line is contained on the Stator PCB Board)

The NCS32100 drives the sensor excitation coils with LC1 and LC2, and reads inductively coupled responses on the REC[7:0] pins. The receiver coils are located on the stator PCB with the NCS32100, and their electrical signals vary with the position of the nearby rotor PCB. The received signals are translated by the NCS32100 to an absolute angle. Speed data is also calculated by the NCS32100. An RS−485 driver (THVD1550, ADM485, or equivalent) is used to translate the UART receive commands from the master controller and response from the NCS32100 to 5V RS485 twisted pair for the data transmission over long cabling. All circuitry needed for a complete rotational sensor is contained on the stator PCB to demonstrate the small form factor of the complete sensor. This is represented in Figure 2 by everything to the left of the dashed line. In application, the rotor and stator boards would be housed as a contained module with only the 5 pin RS485 connector accessible.

Table 1. TYPICAL PARAMETERS SPECIFIC TO THE NCS32100 REFERENCE DESIGN

(for full specifications of the NCS32100 refer to the NCS32100 Datasheet)

Rating	Symbol	Value	Unit
Supply Voltage Range	V_{CC}	3.3V – 5.5V	V
Operating Ambient Temperature Range	T_{AMB}	−40 to 85	°C
Multi−turn Count Range		24	Bits
Position Resolution		20	Bits
Velocity Resolution		20	Bits
Supply Standby Current	$I_{STANDBY}$	200	μA
Supply Dynamic Current (Including RS−485 Driver)	$I_{DYNAMIC}$	80	mA
RS485 Transmission Rate (UART based)		2.5	MHz
Data Communication Response Time	T_{RESP}	3.0 ±0.5	μs
RS485 Differential Load Resistance	R_{LOAD}	54	Ω
RS485 Driver Differential output rise/fall time		400	ns
Ready after power up time	T_{READY}	50	ms
Eccentricity Tolerance	$D_{MISALIGN}$	±0.25	mm
Rotor to Stator Airgap Tolerance	D_{AIRGAP}	0.2−0.5	mm

Table 1. TYPICAL PARAMETERS SPECIFIC TO THE NCS32100 REFERENCE DESIGN
(for full specifications of the NCS32100 refer to the NCS32100 Datasheet) (continued)

Back to Back Command Timing	T_{SPER}	62.5	μs
Memory and Register Write Time (Note 1)	T_{SMEM}	15	ms

NOTE: Additional relevant mechanical specifications for the housing and mounting of the rotor and stator are not detailed in this reference manual. Mechanical tolerances are outside of the reference design scope, such as mass, moment of inertial, friction torque, shaft misalignment, max angular acceleration, and mechanical longevity.

1. Minimum delay between last received byte and new command byte. Does not apply to memory read and write which has a minimum of 37.5 μs.

External Connector

The NCS32100 reference design requires a 5−pin connection (2 RS485 data signals, PWR, GND, and VBAT battery backup) from the sensor to the external master. The pinout in Table 2 below is compatible with common 5 pin connectors used in industrial applications. VBAT should be tied to ground through a 20 kΩ resistor if it is not being used.

Table 2. INDUCTIVE ENCODER CONNECTOR DEFINITION

Function	Notes
VCC	5V main power supply
GND	Connect to system return
VBAT	External Backup Battery Supply
SD	Serial Data Line (RS485+)
SD_bar	Inverse Serial Data Line (RS485−)

External Interface Data Packet Format Description

The NCS32100 reference design uses a half−duplex 2 wire RS485 interface for communication with an external controller (SD and SD_bar). The NCS32100 acts as the slave, only sending data when requested by the master. The encoder communication is broadly classified into 4 types of commands (**data** readout, configuration **register read/write** commands, **NVM read/write** commands, and **special** diagnostic commands) and 2 modes of operations (Run Mode and Validation Mode). **NVM read/write** commands, and **special** diagnostic commands. Each of these formats are detailed below. The interface protocol is orchestrated by the NCS32100 internal MCU, and is firmware defined.

Data Readout Formatting

Each supported data command has a unique data ID, or op−code. These op−codes are defined in Table 3. Each byte is 10−bits long, beginning with a start bit, 8 bits of information, and ending with a stop bit. The master initially sends the command byte to start the communication. The encoder returns information based on the data ID. The NCS32100 always returns the command byte back to the master as the first byte of its response. In the case of data readout, the encoder returns the command field followed by the status byte, followed by the data bytes, then followed by the CRC field as the last byte. Figure 3 below shows the data readout format. Refer to Table 1 for rate at which back−to−back commands can be processed (T_{SPER}).

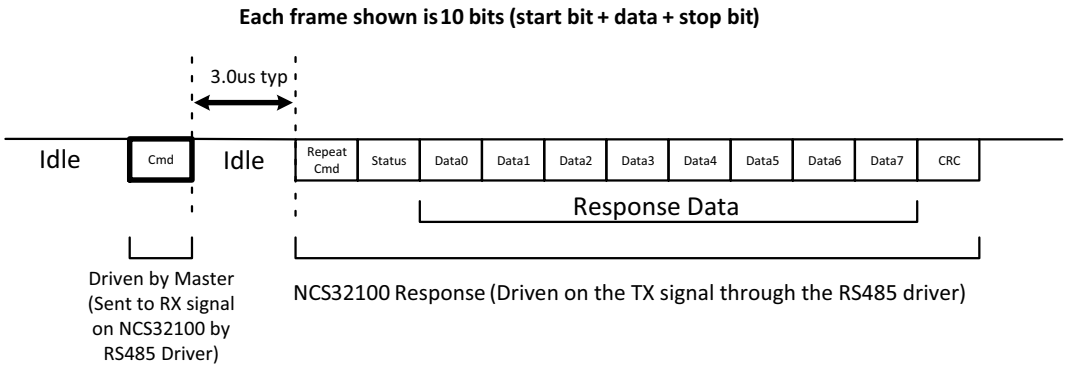


Figure 3. Data Readout Communication Format Definition. The SD signal is shown. SD_bar is omitted for clarity.

NOTE: Number of bytes in response data frame will depend on allocated bytes as defined in Table 6. Not all the data readout commands will send back 11 bytes. Some will respond with fewer bytes.

NVM 8−bit Memory Read / Write Formatting

For reading and writing single byte entries to the NCS32100 8−bit registers, the master sends the register address field (and register data field for write) followed by the CRC. These commands are for storing and accessing bytes that are 8 bits in length. The formats shown below in Figures 4 and 5 show a write communication and a read communication using the ‘writeMem’ and ‘readMem’ commands. Details on supported memory addresses are detailed in Table 7. Refer to Table 1 for rate at which readMem and writeMem can be sent at.

After issuing a writeMem command, allow T_{SMEM} of processing time before sending another command (see T_{SMEM} in Table 1). The NCS32100 will not respond to commands during this time. readMem does not require T_{SMEM} processing time.

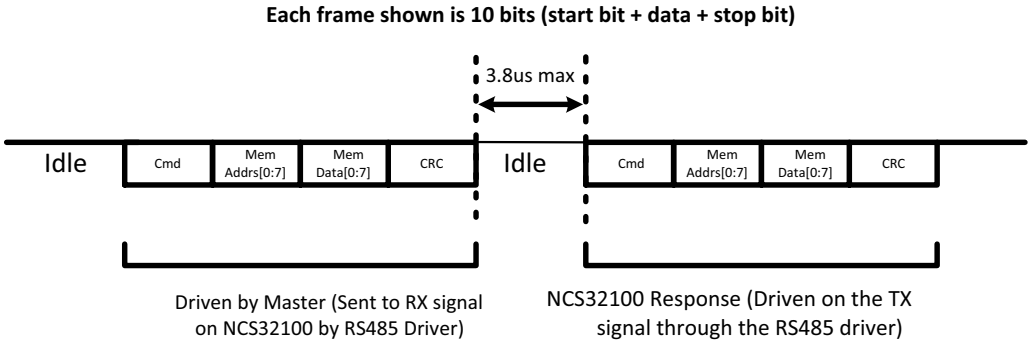


Figure 4. Memory Write Communication Format Definition. (“writeMem” in Table 3.)

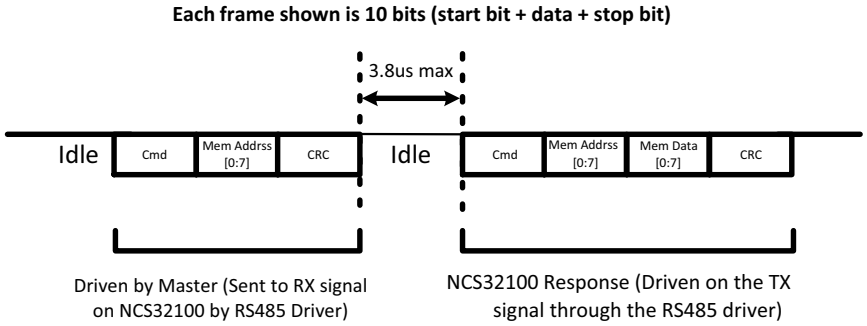


Figure 5. Memory Read Communication Format Definition. (“readMem” in Table 3.)

NVM 16−bit Register Read / Write Formatting

Various configuration words used in the NCS32100 are 16 bits (2 bytes) in length. For writing and reading these words, a format is used that supports 2 bytes. This format is shown in Figure 6 and 7 below for writing and reading.

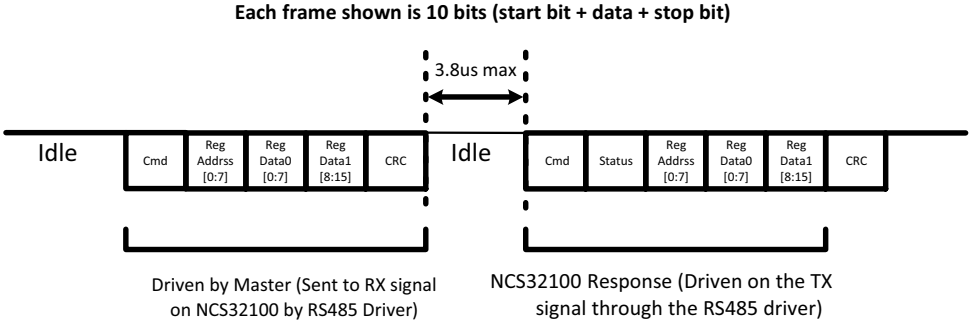


Figure 6. Register Write Communication for 16−bit Words. (“writeReg” in Table 3.)

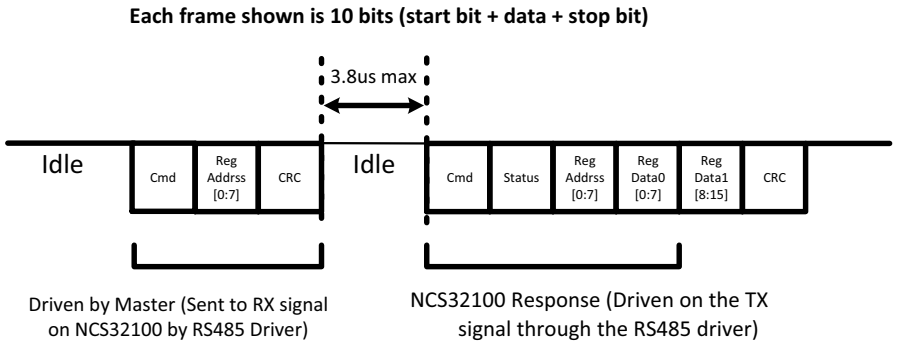


Figure 7. Register Read Communication for 16−bit Words. (“readReg” in Table 3.)

Special Command Formatting

Some of the commands offered by the NCS32100 Reference Design do not follow the standard formatting. These commands are used for system debug and diagnostics and are not intended to be used during normal operation. These commands will be explained individually in a later section. These commands still appear in Table 3 below but are labeled as “Special” indicating that their responses do not follow the standard formatting. The formatting for these commands will be defined individually in later sections of this manual.

Command Field Definition

As defined in the sections above, each communication between the external master and the NCS32100 (slave) uses a command byte. The command byte format is shown below in Figure 8. The frame header allows the NCS32100 to recognize that a command is coming. The first 3 bits contain a “010” sync code, and the remaining bits are purposed with the command code. A parity bit is included at the end. Special commands may not follow the the standard format, and may have a different Sync Code.

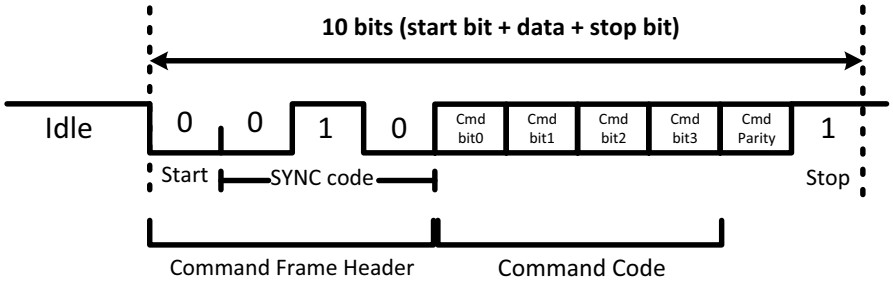


Figure 8. Command Field (Send by the Master, repeated in the response from the Slave as shown in Figure 3.)

The available command codes are listed below in Table 3.

Table 3. COMMAND CODE DEFINITIONS

Command	OP Code				Parity	Command Type	Description
	Cmd0	Cmd1	Cmd2	Cmd3			
getPosition (0x02)	0	0	0	0	0	Data	Returns 20−bit absolute position data
resetErrors (0xC2)	0	0	0	1	1	Data	Resets all errors back to 0
resetMultiTurn (0x62)	0	0	1	1	0	Data	Sets Multi−Turn Count to 0
getEncoderID (0x92)	0	1	0	0	1	Data	Returns the encoder ID
getMultiTurn (0x8A)	1	0	0	0	1	Data	Returns 24−bit Multi−turn Data
getBattV (0x4A)	1	0	0	1	0	Data	Returns Digitized Battery Voltage Measurement. The measurement is a 16−bit value that should be divided by 100 to get the battery measurement in volts.
getBundle (0x1A)	1	1	0	0	0	Data	Returns Position, multi−turn count, encoder ID, and encoder error.
getSpeed (0xDA)	1	1	0	1	1	Data	Returns 20−bit speed data
resetPosition (0xBA) (Note 2)	1	1	1	0	1	Data	Sets current sensor position as absolute 0 (Rotor must be static for at least 15ms while resetPostion command is running)
Memory Access Commands							
readReg (0x52)	0	1	0	1	0	Register Read	For reading 16−bit values from the NCS32100 register memory
writeReg (0x2A) (Note 2)	1	0	1	0	0	Register Write	For writing 16−bit values to the NCS32100 register memory
readMem (0xEA)	1	0	1	1	1	NVM Read	Reads a single byte from NCS32100 memory (See Figure 4 for details)
writeMem (0x32) (Note 2)	0	1	1	0	0	NVM Write	Writes a single byte to NCS32100 memory (See Figure 3 for details)
Special Commands							
runAcquisition (0xF2) (Note 3)	0	1	1	1	1	Special	Collect a set of 800 ADC sample sets and stores them in the NCS32100 MCU flash so that they can be used for external calibration
getOneADCSet (0x07)	0	0	0	0	0	Special	Does not use standard sync pulse. Gets one sample set from NCS32100.
getAcquiredData (0x7A) (Note 3)	1	1	1	1	0	Special	Gets one specified sample set from NCS32100 (only after ‘runAcquisition’ has been run).
selfCalibrate (0xA2)	0	0	1	0	1	Special	Runs NCS32100 self−calibration. Command must be sent 10 consecutive times.
getVersionInfo (0x8F)	1	0	0	0	1	Special	Returns the current firmware version number

2. Allow 15 ms of processing time before issuing another command to the NCS32100.
3. These commands are available when Diagnostic mode is enabled. See memory page 7, address 0x0A.

Status Field Definition

The status field is used for communicating the status of the encoder to the master and is sent with every data communication response prior to the data fields. The status field follows the repeat of the command field in the NCS32100 response. Figure 9 below defines the bits in the status field. The information bits are only used to inform the master that a write command was used to write to a read only register. If such an event occurs, the information bit 0 is set to 1 and the target register of the write command is not written but remains read only. Status byte information resets to 0 at power on/reset (POR). The 'Encoder Error' bits and 'Communication Error' bits are defined below in Tables 4 and 5. If errors are present in the Status byte, the data present must be considered to be invalid until the errors are cleared.

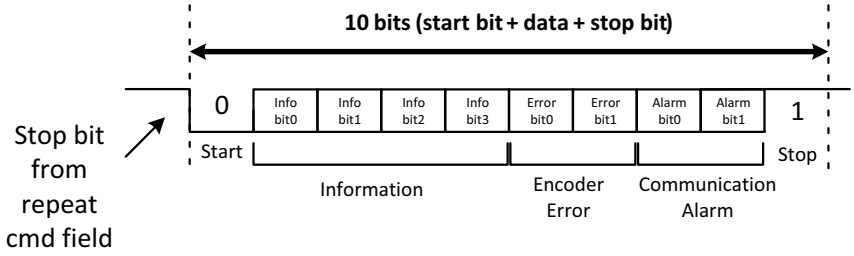


Figure 9. Status Field Definition

Table 4. STATUS ERROR BITS (as reported in the status frame)

Error Bit 0	Error Bit 1	Description
x	1	Over Temperature Error, or Multi−turn Error, or Battery Error
1	x	Received CRC error
0	0	No Status errors to report

Table 5. FRAME ERROR BIT DEFINITION (for status frame)

Communication Error	Alarm Bit 0	Alarm Bit 1	Description
Parity Error	1	x	Parity is incorrect
Stop Bit Error	x	1	Stop bit is 0 instead of the expected 1
No Error	0	0	No Frame errors to report

Response Data Field Definition

Response data is returned by the NCS32100 when commands requesting data are sent by the master (see Table 3). Eight data bytes are used to communicate the response data. Each data byte is organized with the LSB first as shown in Figure 10 below.

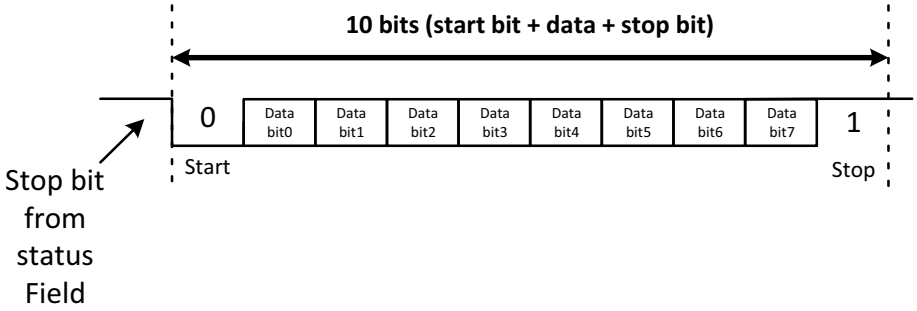


Figure 10. Data Field Definition

The tables below define how the response data bytes are used for each data command. The blank data bytes indicate that the field is not used and will be omitted in the response. Responses with omitted bytes will be shorter in length.

Table 6. DATA FIELD BIT LOCATIONS (blank boxes indicate data field is omitted)

Command From Master	Data Byte 0	Data Byte 1	Data Byte 2	Data Byte 3	Data Byte 4	Data Byte 5	Data Byte 6	Data Byte 7
getPosition	Position bits 0–7	Position bits 8–15	Position bits 16–19					
getSpeed	Speed bits 0–7	Speed bits 8–15	Speed bits 16–19					
getEncoderID	ID bit 0–7							
GetMultiTurn (Note 5)	Count bits 0–7	Count bits 8–15	Count bits 16–23	0x00	0x00	0x00	0x00	0x00
getBatt	Batt Data 0–7	Batt Data 8–15						
getBundle	Position bits 0–7	Position bits 8–15	Position bits 16–19	ID bits 0–7	Count bits 0–7	Count bits 8–15	Count bits 16–23	Error bits 0–7
resetMultiTurn (Note 6)	Count bits 0–7	Count bits 8–15	Count bits 16–23	0x00	0x00	0x00	0x00	0x00
resetPosition (Note 4)	Position bits 0–7	Position bits 8–15	Position bits 16–19	0x00	0x00	0x00	0x00	0x00
resetErrors	Position bits 0–7	Position bits 8–15	Position bits 16–19	0x00	0x00	0x00	0x00	0x00

4. Allow T_{SMEM} of processing time before issuing another command to the NCS32100.
5. Data Bytes 3 through 7 are output as 0x00. A total of 11 bytes are used for this message.
6. If tracking multi−turn count, it is recommended to issue resetMultiTurn at startup configuration since value may initialize to non−zero number.

Data byte 7 is used when the “getBundle” command is sent to communicate error codes. Table 7 below defines the meaning of each bit in data byte 7 for the “getBundle” command. If data byte 7 contains all 0’s, then there are no errors, and the error bit in the status frame will also be 0. If the error bit in the status frame is 1, then the getBundle command can be sent to get more information on the type of error as defined below.

Table 7. ERROR BIT DEFINITION

Error	Bit	Description
Over Speed	0	Speed has exceeded 6000 rpm (Accuracy may not be in spec)
Reserved	1	Reserved
Sensor Error	2	REC Gains are Railed (Sensor connections are shorted or not correct)
Multi−Turn Overflow	3	24 bit multi−turn count has rolled over back to 0
Over Temperature Error	4	Temperature has exceeded TMAX (TMAX is set in the TMAX register)
CRC Error	5	Calculated CRC did not match received CRC.
Low Battery	6	Battery Voltage is less than BATT_MIN (BATT_MIN is set in the BATT_MIN register)
Battery Alarm	7	Asserted when battery is less than 3.1 V

The Multi−turn overflow will assert every time the count reaches 0 as shown below. This assumes that the multi−turn count is interpreted as a signed number.

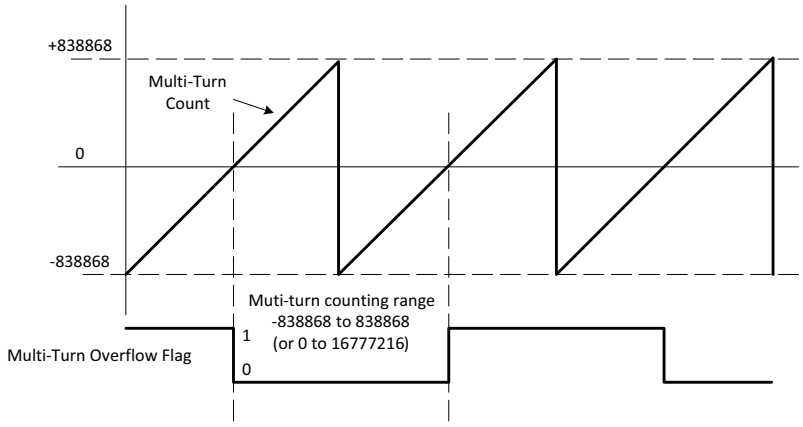


Figure 11. Multi−turn Overflow Flag Explanation

Reset Commands

The resetPosition, resetMultiTurn, and errorReset commands must be transmitted 10 times within 10 ms of each other while the rotary sensor is stationary before they will be reset. This is required to avoid the unintentional reset of these values, and to allow the user the ability to ensure that the shaft is intentionally stationary. Running these reset commands on a moving sensor will create ambiguous and inaccurate results. After issuing a reset command, 15 ms of processing time is required before another command can be issued to the NCS32100.

CRC Definition

The Cyclic Redundancy Check (CRC) is calculated for every transmission from the NCS32100. This includes both data responses, register read/write responses, and read/write memory responses. The CRC is calculated by the internal MCU using the following equation.

$G(x) = X^8 + 1(X = CRCbit0 \sim CRCbit7)$.

If the CRC from a packet sent by the external master does not calculate correctly, CRC error bit should be set in the status byte, and bit 5 should be set in the Error byte that is returned with the ‘getBundle’ command. The data is arranged LSB first. All 8 bits of each field are used to calculate the CRC.

The format of the CRC is shown below in Figure 12.

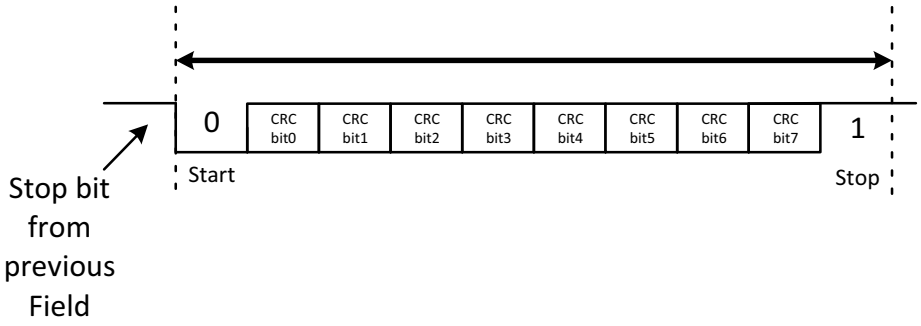


Figure 12. CRC Field Definition

Below is an example of a full transaction with the NCS32100. In this example the ‘getBundle’ command is sent and the NCS32100 responds with the 11 bytes as defined in Figure 3.

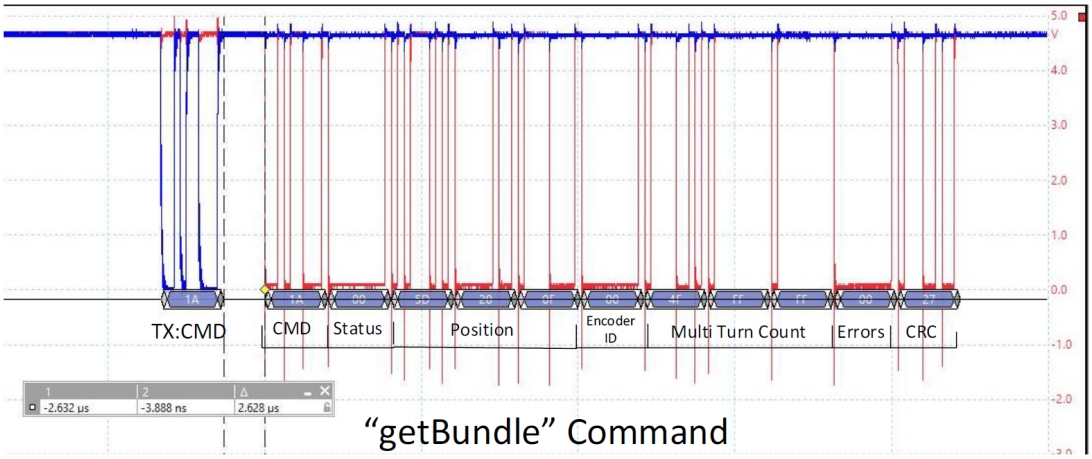


Figure 13. Example Position Acquisition Transaction (RS485 differential pair signal omitted for clarity)

SPECIAL Commands

The NCS32100 reference design supports 4 special commands. These include the “getOneADCSet” command, the “getAquiredData” command, the “getVersionInfo” command and the “selfCalibration” command. The formatting and purpose of each of these commands will be detailed below.

Getting Raw Sensor ADC Values

The inductively coupled signals from the PCB stator are connected to the NCS32100 REC pins. The REC0, REC1, and REC2 pins are connected to the stator coarse coils. The REC5, REC6, and REC7 pins are connected to the stator fine coils. The REC3 and REC4 pins are used to calculate the DC offset of the sensor. All of these signals are rectified and digitized by the NCS32100’s internal 12−bit differential ADCs. These digitized signals are representative of the raw sensor signals, and therefore can be very useful to the user to verify correct sensor design and correct configuration and connection to the NCS32100. The ‘getOneADCSet’ command allows the user to collect a full raw ADC sample set from the NCS32100. These ADC values are signed values. This includes all the samples that are used to calculate a single rotor position. This command does not use the standard sync pulse that the other commands use. It is strictly for diagnostic purposes. Figure 14 below defines the formatting for this command. Each ADC sample is represented with 2 bytes as a 16−bit SIGNED number. The samples come out LSB first in the following order: Course0, Coarse1, Coarse2, CoarseDC, Fine0, Fine1, Fine2, and FineDC. There is a CRC included at the end.

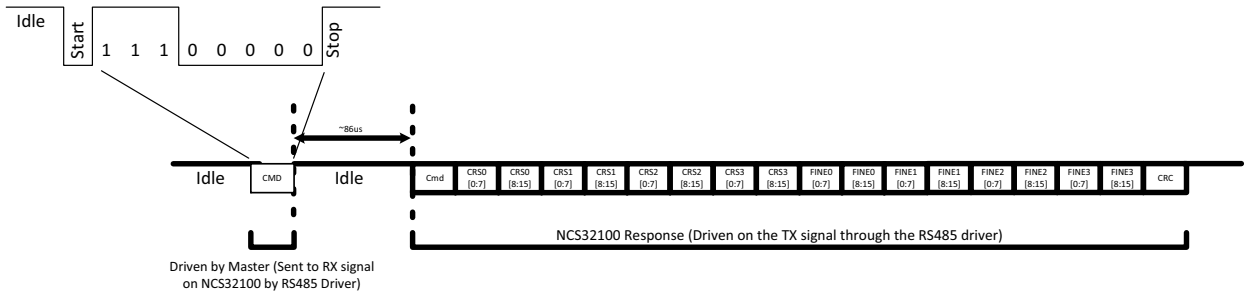


Figure 14. Command and Response Format for the ‘getOneADCSet’ Command

The response gives 3 coarse coil measurements, 1 coarse coil offset measurement, 3 fine coil measurements, and 1 fine coil offset measurement.

NCS32100 Self−Calibration

The NCS32100 has the capability to self−calibrate without a reference encoder (using quadratic interpolation). If the mechanical alignment and tilt are tight and repeatable, then this self−calibration may be all that is needed to achieve accuracies better than 50 arcsec. The NCS32100 will collect ADC samples during this time and store them in the MCU flash. The samples sets are then used to feed the quadratic interpolation algorithm which adjusts the 16 Clarke Coefficients (stored in registers 0x12 – 0x21). The Clarke coefficients are applied during the Clarke transform, which translates the 3−phase sensor signals into a sine and cosine waveform pair for both the fine coil system and the coarse coil system. The 16 Clarke coefficients are adjusted during the calibration routine to result in a sine / cosine pair that is as close to ideal as possible. The self−calibration routine will minimize accuracy error caused by all harmonics beyond the fundamental. A 360−degree single period fundamental or low spatial frequency error is caused by mis−alignment plus tilt of the rotor relative to the shaft. The NCS32100 reference sensor can tolerate some misalignment if there is no tilt of the rotor relative to the shaft. It can also tolerate some rotor tilt relative to the shaft if there is not mis−alignment, however, both tilt and mis−alignment together will cause a single period 360 degree error.

The self−calibration is initiated by a “0xA2” command from the master (see Table 3). The command ”0xA2” must be sent 10 consecutive times, with less than 10 milliseconds between each command. After each command is issued from the master, the NCS32100 will respond, acknowledging the calibration command, and how many times it has successfully received the calibration command. After the 10th successful command is received, the NCS32100 will begin the self−calibration process. At this time, the rotor must be rotated at a rate below 100 RPM. The rotor can be turned in either direction for calibration, but, once the rotor begins turning in one direction (for example, clockwise), the direction of rotation must not be changed during the calibration process. The NCS32100 collects 800 ADC samples at specific points of the three−phase signals (zero−crossings and phase−crossings), and stores them in the internal MCU. Since the self−calibration routine looks for signal crossings, an exact rotation speed is not required. The calibration sample acquisition time is proportional to the rotation speed. For example, ff the rotation speed is below 10 RPM, then sample collection may take longer than 10 seconds, causing the self calibration process to take longer. If no rotor movement is detected, the calibration routine will timeout after 3 seconds. These samples are utilized for the self−calibration, and the calibration coefficients (see Table 9) will be updated to achieve position calibration. Upon completion, the NCS32100 responds with a summed squared error value and mismatch value that can be used to determine the accuracy of the resulting calibration. The summed squared error value is a fixed point 64−bit number that represents the summation of all the error terms during the calibration. The minimum and maximum mismatch values (32−bit signed) are another indicator of calibration quality. The absolute values of mismatch may not exceed 255 for correct performance. Keeping the mismatch below 64 is recommended. Figure 15 and 16 below shows the format for the command and response for self−calibration. Once the response has been received, the calibration is complete, and the user can stop rotating the rotor.

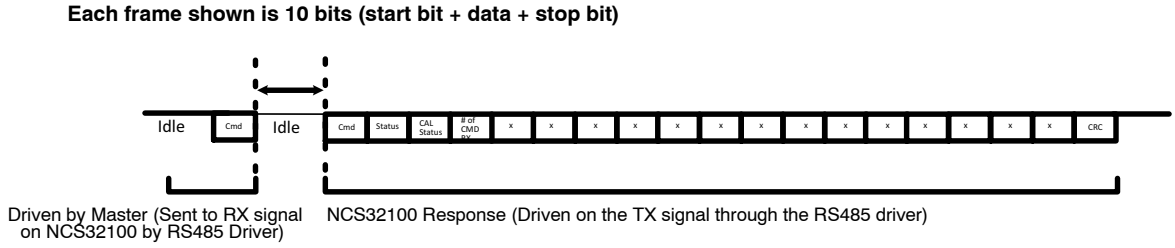


Figure 15. Command and Response Format for the ‘selfCalibrate’ Command (Response 1 through 10)

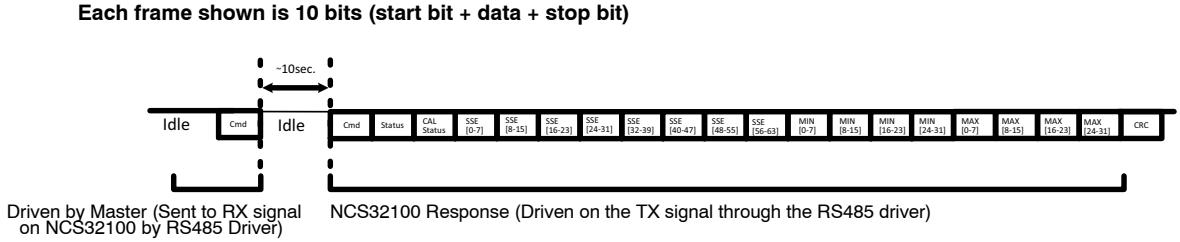


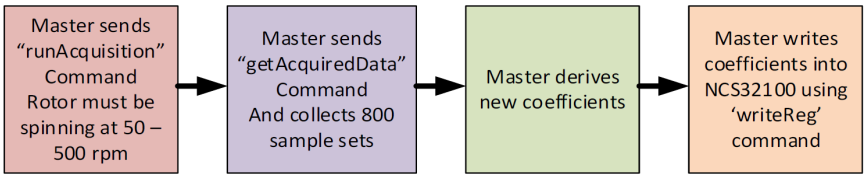
Figure 16. Command and Response Format for the ‘selfCalibrate’ Command (Final Response)

Table 8. CALIBRATION STATUS BYTE

Calibration Status	
Value	**Description**
0	Failed
1	Time out (No rotor movement detected)
2	Calibration not started (1 to 9 command inputs received)
3	Starting Calibration (10th command received)
4	Invalid
5	Completed

External Calibration

External calibration is possible if the user chooses to do so. The user may do this to reduce the calibration time by using an external processor with more processing power. To complete an external calibration, the user must follow 4 steps.



The "runAcquisition" command tells the NCS32100 to collect 800 sample sets and store them in the MCU flash. The "getAcquiredData" command allows the master to get the 800 sample sets from the NCS32100. This command can get one sample set at a time. Once the master has the raw ADC samples, a variety of methods can be used to derive coefficients for calibration. Once the new calibration coefficients have been calculated, the user must store the updated values in the NCS32100 configuration registers (addresses 0x12 through 0x21) for the calibration to take effect. These calibration coefficients will be held in NVM to maintain proper calibration through subsequent power cycles. The format for the 'runAcquisition' command is shown below in Figure 17 runAcquisition can only be run in validation mode (see memory register Page 7 0x0A).

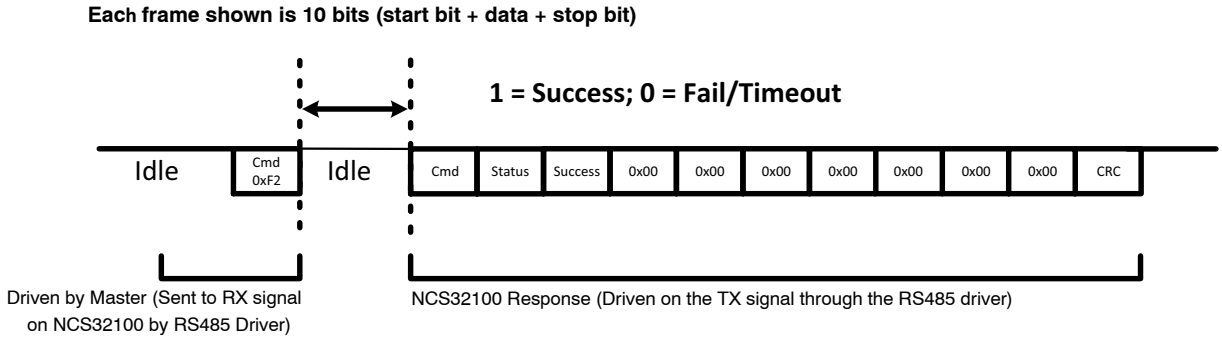


Figure 17. Command and Response Format for the 'runAcquisition' Command

The format for the 'getAcquiredData' command is show below in Figure 18. The sample number is indicated in the command and verified in the NCS32100 response. Using this command, the master can get all 800 sample sets from the NCS32100, or only a portion of them if needed. runAcquisition can only be run in validation mode (see memory register Page 7 0x0A).

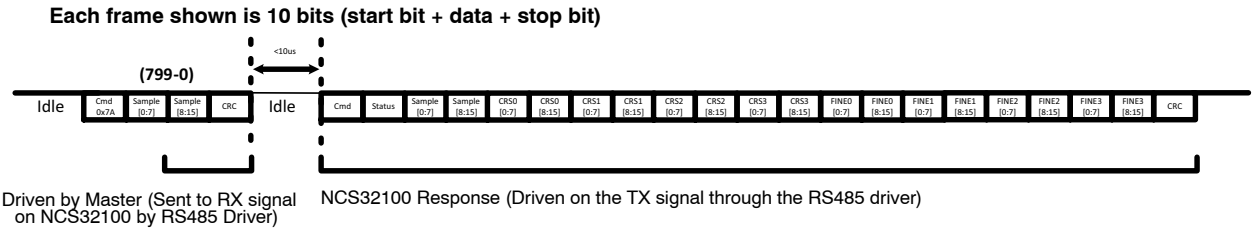


Figure 18. Command and Response Format for the getAcquiredData Command

"getVersionInfo" Command

The "getVersionInfo" command is used to verify the current version of the firmware that is flashed to the NCS32100. The response to this command provides the type of encoder the firmware was written for, as well as the firmware version number along with the analog trim code that was set during production test. This information can be used to identify the current firmware version. The response format is shown below.

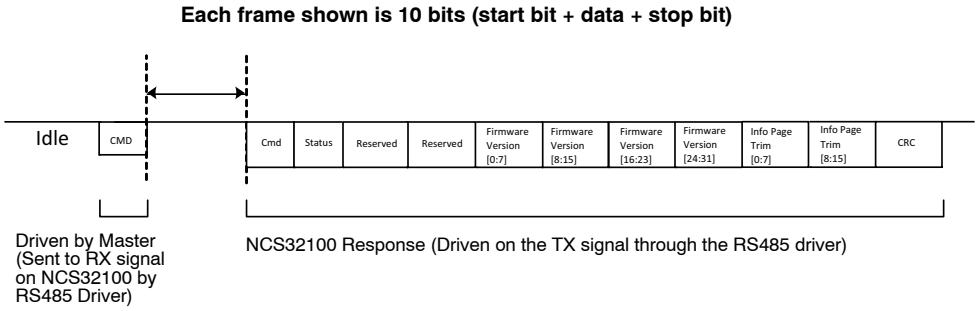


Figure 19. Command and Response Format for the 'getVersionInfo' Command

User Accessible Memory

The NCS32100 non−volatile memory accessible to the user has been divided up into 2 sections (firmware defined). The first section is for reading and storing 8−bit addressable values and is accessed using the 'writeMem' and 'readMem' commands. The second section is reserved for 16−bit configuration registers. These are accessed using the 'readReg' and 'writeReg' commands.

8−bit Addressable Memory

This memory space is organized to occupy 7 pages with 127 addressable bytes each. To change pages, the user must write the new page to address 127 of the current page. The default page upon power up is 0. The address is sent out LSB first as shown below in Figure 20.

After issuing a 'writeReg' or 'writeMem' command, allow 15 ms of processing time before issuing another command to the NCS32100.

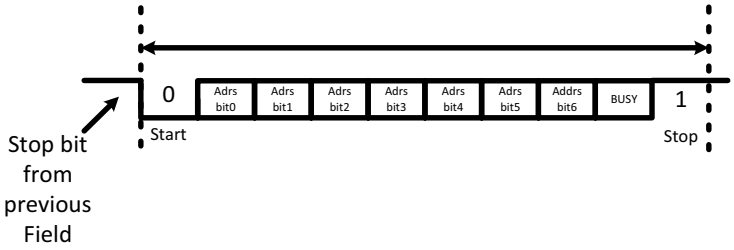


Figure 20. Address Byte in the 'readMem' and 'writeMem' Frames

Addresses 0 through 126 on pages 0 through 5 are open to the user to store anything that they want to maintain through a power cycle. Address location 127 on each page is reserved

to hold the current page number, and the page location is reset at every power cycle (POR).

Table 9. 8−BIT ADDRESSABLE MEMORY LOCATIONS ACCESSIBLE BY THE USER

Register Name	R/W	Access Command	Address	Description
Open NVM Page0	R/W	readMem / writeMem	0x00 to 0x7E page0	Page 0 Address available for customer use
Page Listing	R/W	readMem / writeMem	0x7F Page0	Reserved for Current Page Listing
Open NVM Page1	R/W	readMem / writeMem	0x00 to 0x7E page1	Page 1 Addresses available for customer use
Page Listing	R/W	readMem / writeMem	0x7F Page1	Reserved for Current Page Listing
Open NVM Page2	R/W	readMem / writeMem	0x00 to 0x7E page2	Page 2 Addresses available for customer use
Page Listing	R/W	readMem / writeMem	0X7F Page2	Reserved for Current Page Listing
Open NVM Page3	R/W	readMem / writeMem	0x00 to 0x7E page3	Page 3 Addresses available for customer use
Page Listing	R/W	readMem / writeMem	0x7F Page3	Reserved for Current Page Listing
Open NVM Page4	R/W	readMem / writeMem	0x00 to 0x7E page4	Page 4 Addresses available for customer use
Page Listing	R/W	readMem / writeMem	0x7F Page4	Reserved for Current Page Listing
Open NVM Page5	R/W	readMem / writeMem	0x00 to 0x7E page5	Page 5 Addresses available for customer use
Page Listing	R/W	readMem / writeMem	0x7F Page5	Reserved for Current Page Listing
Velocity Resolution	R/W	readMem / writeMem	0x00 Page 7	Sets the number of resolution bits for the velocity readout. Can be set from 1 to 20. Default value is 16 bit resolution.
BATT_ALARM	R	readMem	0x03 Page 7	1 byte: Reads back the battery alarm. This can range from 0 V to 4 V. The value is represented in 10 mV increments. There is an offset of 1.00 V to allow values from 1 V to 3.0 V to be represented. Ex. 2.75 V → h'AF (175 decimal + offset of 100 = 275 → 2.75 V)
T_MAX (over−heat temp threshold)	R/W	readMem / writeMem	0x04 Page 7	Setting for the temperature alarm. T_MAX temperature is stored as a 7−bit value that represents 1C to 125C. The most significant bit is always 1 (bit7)
TEMP	R	readMem	0x05 Page 7	The current substrate temperature. Temperature is stored as an 8−bit signed value representing −128C to +127C.
SFE_STATUS	R	readMem	0x06 Page 7	Allows the user to read additional status information. See the SFE_STATUS register contents section for more details.
BAT_MIN	R/W	readMem / writeMem	0x07 Page 7	Battery voltage (in Volts) at which the low battery alarm flag will assert. This can range from 0 V to 4 V. 2.75 V is the default value. The value is represented in 10 mV increments. There is an offset of 1.00 V to allow values from 1 V to 3.0 V to be represented. Ex. 2.75 V → h'AF (175 decimal + offset of 100 = 275 → 2.75 V)
CALIBRATION_ITERATION	R/W	readMem / writeMem	0x08 Page 7	1 byte: Number of times that the last 16 steps of calibration will run. Repeating these steps more than 4 times may not result in better performance for most sensors. Maximum value of 255. Default of 4.

Table 9. 8−BIT ADDRESSABLE MEMORY LOCATIONS ACCESSIBLE BY THE USER (continued)

Register Name	R/W	Access Command	Address	Description
BATT_TEMP_PIREAD	R/W	readMem / writeMem	0x09 Page 7	These settings allow the user to enable or disable features that may not be required. Disabling PI read will disable Speed error, Multiturn overflow and Sensor error detection. Disabling the Battery, Temperature or Speed monitoring allows for lower and more stable latency in response time. 0b111 − All enabled (Default) 0b001 − Only Battery voltage read 0b010 − Only temperature read enabled 0b100 − PI read on every 10 ms.
VALIDATION_EN	R/W	readMem / writeMem	0x0A Page 7	1 bit: Enable validation mode commands (runAcquisition, getAcquired, getCalibrationInfo). Enabled by default.
UART_NOISE_REDUCTION	R/W	readMem / writeMem	0x0B Page 7	1 bit: UART deglitch option. Glitches on UART RX line, less than 50nsec in width, will be ignored 0b1: enable UART deglitch filter (default) 0b0: disable UART deglitch filter.

16−bit Addressable Configuration Registers

The ‘readReg’/‘writeReg’ commands allow the user to write and read specific configuration data to the NCS32100. ‘readReg’/‘writeReg’ are for 16−bit register locations used strictly for NCS32100 configuration. These register only need to be written by the user if there is a design change to the sensor or if the filter settings need to be updated. The registers for secondary calibration are also accessible by the readReg and writeReg commands. The default firmware sets these registers during the startup routine to the correct configuration for the reference design sensor. The following table lists all available registers supported by the NCS32100 reference design and their memory locations. All register content passed over the external interface is written / presented in an LSB first format. These registers are all defined in the NCS32100 datasheet.

Backup−Battery Mode

The NCS32100 reference design firmware is programmed to go into a low power battery mode anytime the VCC voltage falls below the VBAT voltage. During this time, all external communication between the NCS32100 is not possible because the internal level shifters and drivers are powered down. The internal MCU wakes up every 3 ms to update the turns count value, and then goes back to sleep. This will continue until the VCC power is restored. This mode allows the turns count to be tracked up to 6000 rpm when the power is lost if the device is connected to a backup battery.

The wake up period can be configured by the user by writing to register 0x90. This register is 21 by default, and it can be changed to one of 9 values found in the table in the register 0x90 definition. The battery mode algorithm works by moving between a 3 ms wakeup to the wakeup defined by the user in register 0x90. It changes the wakeup period based on the movement of the rotor. If the rotor is not moving, then the algorithm will change the wakeup period to the user defined value in 0x90. If the rotor is detected to move, then the wakeup period will be changed to the 3 ms wakeup until it is detected that the rotor is not moving anymore. This is done to conserve battery power. It is assumed that the rotor will not be moving most of the time battery mode is engaged. The user is allowed to change to slower wakeup time so that they can guarantee turns counts will not be lost during the worst case acceleration from stand still. The max acceleration from stand still is application dependent.

NCS32100 Secondary Calibration

The purpose of the secondary calibration method is to remove any single period 360 degree error that may exist after a self−calibration has been executed. The NCS32100 has 16 registers (register 0x80 through 0x8F) for storing 16 secondary coefficients. These coefficients are applied to the NCS32100 output position as it is passed through the MCU on its way to the external master. The final output position is calculated as the following:

$$\text{Output Position} = \text{Internal Position} + \text{BinN coefficient}$$

Where the “BinN coefficient” is one of the 16 coefficients stored in the secondary calibration registers. The 16 bins correspond to 22.5 degree sections as follows (Table 10):

Table 10. Secondary Calibration Registers

Register	Name	Description
0x80	Bin0[15:0]	Applied to positions from 348.75 – 11.25 degrees
0x81	Bin1[15:0]	Applied to positions from 11.25 – 33.75 degrees
0x82	Bin2[15:0]	Applied to positions from 33.75 – 56.25 degrees
0x83	Bin3[15:0]	Applied to positions from 56.25 – 78.75 degrees
0x84	Bin4[15:0]	Applied to positions from 78.75 – 101.25 degrees
0x85	Bin5[15:0]	Applied to positions from 101.25 – 123.75 degrees
0x86	Bin6[15:0]	Applied to positions from 123.75 – 146.25 degrees
0x87	Bin7[15:0]	Applied to positions from 146.25 – 168.75 degrees
0x88	Bin8[15:0]	Applied to positions from 168.75 – 191.25 degrees
0x89	Bin9[15:0]	Applied to positions from 191.25 – 213.75 degrees
0x8A	Bin10[15:0]	Applied to positions from 213.75 – 236.25 degrees
0x8B	Bin11[15:0]	Applied to positions from 236.25 – 258.75 degrees
0x8C	Bin12[15:0]	Applied to positions from 258.75 – 281.25 degrees
0x8D	Bin13[15:0]	Applied to positions from 281.25 – 303.75 degrees
0x8E	Bin14[15:0]	Applied to positions from 303.75 – 326.25 degrees
0x8F	Bin15[15:0]	Applied to positions from 326.25 – 11.25 degrees

Here is an example of what a single period 360-degree error might look like as a result of an accuracy test performed against a reference encoder.

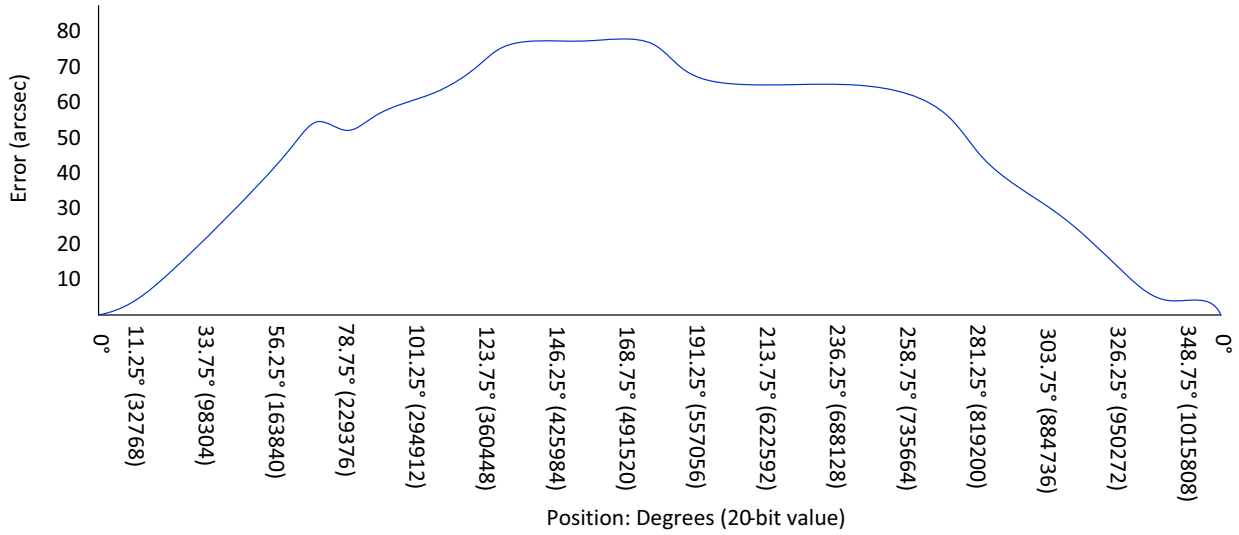


Figure 21.

If this error is sectioned into 16 bins, the average error for each section can be calculated. The negation of these values are what should be stored in registers 0x80 through 0x8F after they have been converted from arcsec to 20−bit position offset values. Error could be calculated in terms of 20−bit position values to avoid the conversion step. The following figure illustrates this process.

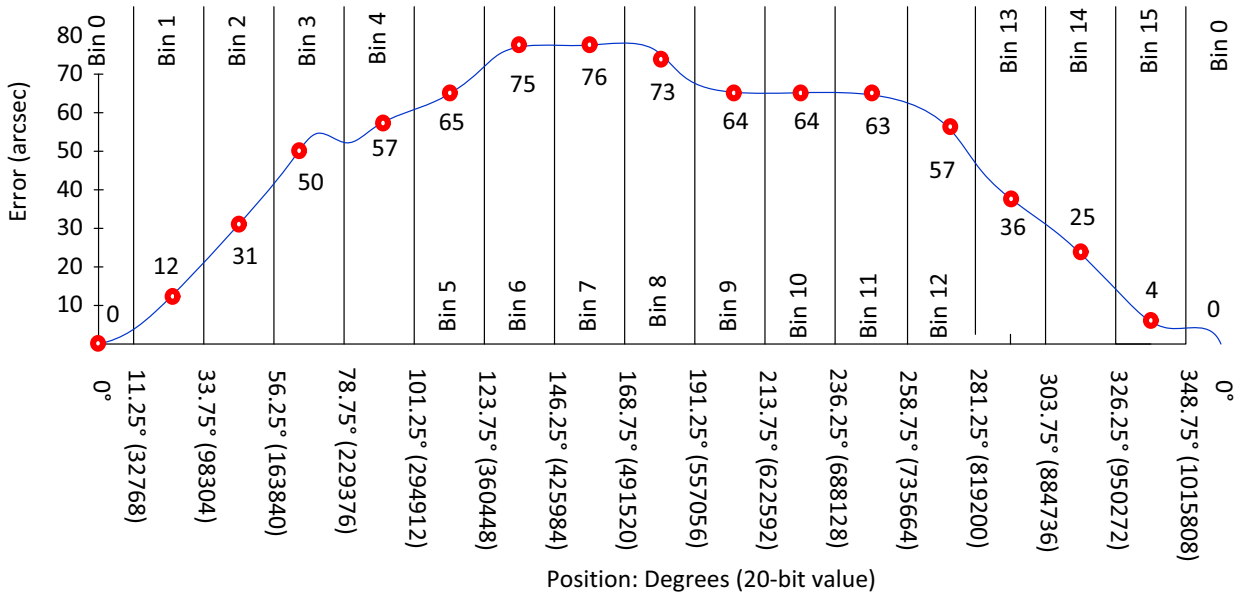


Figure 22.

Error in arcsec can be converted to a secondary calibration coefficient position offset by dividing by 3600(360) and multiplying by 1,048,576.

Table 11. REGISTER VALUES DERIVED FOR EXAMPLE SECONDARY CALIBRATION

Bin	Error (arcsec)	Converted to 20−bit Position	Coefficient to load into NCS32100
0x80	0	0	0
0x81	12	10	−10
0x82	31	25	−25
0x83	50	40	−40
0x84	57	46	−46
0x85	65	53	−53
0x86	75	61	−61
0x87	76	61	−61
0x88	73	59	−59
0x89	64	52	−52
0x8A	64	52	−52
0x8B	63	51	−51
0x8C	57	46	−46
0x8D	36	29	−29
0x8E	25	20	−20
0x8F	4	3	−3

With the secondary calibration values in place, an accuracy test with a reference encoder will perform as shown below where the offsets have been applied and the fundamental error has been removed to a first order.

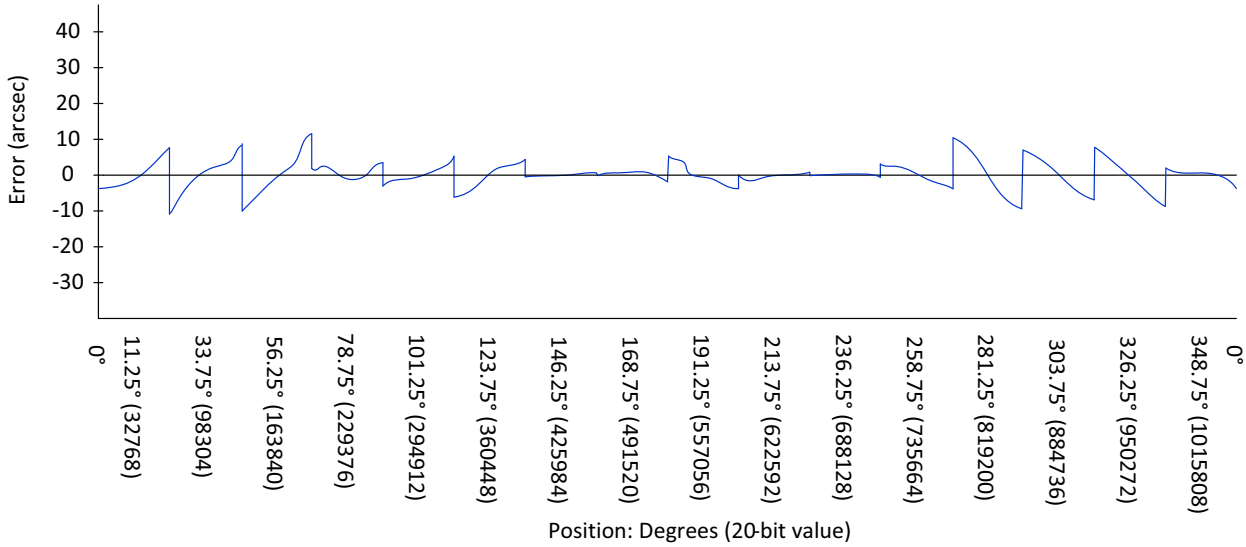


Figure 23.

If no secondary calibration is needed or wanted, then registers 0x80 through 0x8F can be left as their default 0x00 values and no offsets will be applied to the output position.

To calculate and set the secondary calibration coefficient, follow the steps below.

1. Connect the NCS32100 Encoder inline with a reference encoder or a constant speed method.
2. Set the 0-position index of the NCS32100 to the 0 position of the reference encoder. This is achieved by moving the shaft to the reference encoder 0 index location, and then setting the NCS32100 index to that location by sending the "resetPosition" command 10 times.
3. Complete a full 360-degree sweep stopping at a minimum of 64 positions along the way. At each point, take a position measurement from the reference encoder, and one measurement from the NCS32100. Ensure that the rotor is not moving or jittering during these measurements. If static measurements are difficult to acquire in a production setting, then a constant velocity method can be used where the coefficients are set to reduce the output position variance from an ideal linear fit.
4. Use the acquired data to compute the accuracy error between the NCS32100 DUT and the reference encoder.
5. Separate the error numbers for the full 360-degree sweep into 16 equal bins. For each bin, calculate the average error. The average error for each bin is the correction value that needs to be applied to the output position as it crosses through each respective bin.
6. Load the correction values into the NCS32100, where the bin 0 correction goes in register 0x80 and so on.

CONFIGURATION

The analog front end internal to the NCS32100 communicates with the embedded MCU via a 16−bit 40 MHz parallel bus. The NCS32100 firmware includes subroutine functions that enables communication with the front end on this parallel bus. Data can be acquired such as position, and velocity. The firmware allows access to the internal configuration registers that control the functionality of the part. These configuration registers are loaded from the MCU non−volatile memory as part of the start−up routine. Configuration can be changed after initial startup. The reference firmware allows for reading and writing these configuration registers, and any changes will be saved to the MCU non−volatile memory. Once the part is configured and data is received across the internal parallel bus, the embedded MCU can format the data and transmit it to an external master per request according to the protocol definition implemented in the firmware. Please refer to the NCS32100 reference design manual for a detailed description of the firmware functionality and the interface implementation. The NCS32100 is loaded with the reference design firmware by default, which properly handles the configuration during start−up.

NCS32100 Configuration Table

The table below details the configuration registers accessed by the embedded MCU. The purpose of each register is defined in Appendix A. These registers are all properly handled in the reference design firmware. Modification can be made by the user as needed. The NCS32100 is highly configurable. Application notes are provided upon request detailing advanced configuration options.

Table 12. NCS32100 SHADOW REGISTERS TO BE LOADED BY THE EMBEDDED MCU

Address	Access Type	Description	Default Value
0x00	R	LC Oscillator Gain DAC (Status only)	0x0000 (Note 6)
0x01	R	PGA Gain DAC (Status only)	0x0000 (Note 6)
0x03	R/W	Digital Control (Status only)	0x0000
0x04	R	Status	0x0000 (Note 6)
0x10	R/W	Selection Matrix for Coarse Block	0xF249
0x11	R/W	Selection Matrix for Fine Block	0x0B6D
The next 16 registers (0x12 through 0x21) hold the calibration coefficients used to calibrate out any asymmetries in the PCB sensor and the analog front−end. See the calibration section for more details.			
0x12	R/W	Clarke Transform Imaginary Coefficient 0 for Coarse Block	0x0000 (Note 8)
0x13	R/W	Clarke Transform Imaginary Coefficient 1 for Coarse Block	0x0000 (Note 8)
0x14	R/W	Clarke Transform Imaginary Coefficient 2 for Coarse Block	0x0000 (Note 8)
0x15	R/W	Clarke Transform Imaginary Coefficient 3 for Coarse Block	0x0000 (Note 8)
0x16	R/W	Clarke Transform Real Coefficient 0 for Coarse Block	0x0000 (Note 8)
0x17	R/W	Clarke Transform Real Coefficient 1 for Coarse Block	0x0000 (Note 8)
0x18	R/W	Clarke Transform Real Coefficient 2 for Coarse Block	0x0000 (Note 8)
0x19	R/W	Clarke Transform Real Coefficient 3 for Coarse Block	0x0000 (Note 8)
0x1A	R/W	Clarke Transform Imaginary Coefficient 0 for Fine Block	0x0000 (Note 8)
0x1B	R/W	Clarke Transform Imaginary Coefficient 1 for Fine Block	0x0000 (Note 8)
0x1C	R/W	Clarke Transform Imaginary Coefficient 2 for Fine Block	0x0000 (Note 8)
0x1D	R/W	Clarke Transform Imaginary Coefficient 3 for Fine Block	0x0000 (Note 8)
0x1E	R/W	Clarke Transform Real Coefficient 0 for Fine Block	0x0000 (Note 8)
0x1F	R/W	Clarke Transform Real Coefficient 1 for Fine Block	0x0000 (Note 8)
0x20	R/W	Clarke Transform Real Coefficient 2 for Fine Block	0x0000 (Note 8)
0x21	R/W	Clarke Transform Real Coefficient 3 for Fine Block	0x0000 (Note 8)
The remaining register control the extrapolation algorithms, filtering, and gain settings			
0x22	R/W	Velocity Coefficient MSB for Absolute Algorithm Extrapolation	0x0000
0x23	R/W	Velocity Coefficient LSB for Absolute Algorithm Extrapolation	0x7E75
0x26	R/W	Velocity Coefficient MSB for Digital Filter Extrapolation	0x0084
0x27	R/W	Velocity Coefficient LSB for Digital Filter Extrapolation	0x0561
0x2A	R/W	Low Pass Filter for Digital Filter	0x003F
0x2B	R/W	Digital Control	0x0020
0x2F	R/W	Channel Select	0x0000
0x40	R/W	LC Oscillator Gain DAC Control	0x0000
0x41	R/W	LC Oscillator Gain Time Control	0x01F4
0x42	R/W	PGA Coarse Gain DAC Control	0x0000
0x43	R/W	PGA Fine Gain DAC Control	0x0000
0x44	R/W	PGA Gain Time Control	0x01F4
0x45	R/W	PGA Coarse Offset DAC Control	0x0000
0x46	R/W	PGA Fine Offset DAC Control	0x0000
0x47	R/W	PGA Offset Time Control	0x01F4
0x48	R/W	Angle Extrapolator Time Control	0x0000
0x4C	R/W	Normal Wakeup Delay	0x1000

Table 12. NCS32100 SHADOW REGISTERS TO BE LOADED BY THE EMBEDDED MCU (continued)

Address	Access Type	Description	Default Value
0x4D	R/W	Startup Delay Time MSB	0x0006
0x4E	R/W	Startup Delay Time LSB	0x1A80
0x4F	R/W	DSP Wakeup Delay	0x1000
0x50	R/W	Autozero Angle MSB	0x0000
0x51	R/W	Autozero Angle LSB	0x0000
0x57	R/W	Input/Output Control	0x0008
0x5F	R/W	Sensor Selection	0x0013
0x80	R/W	Secondary Calibration Bin 0	0x0000
0x81	R/W	Secondary Calibration Bin 1	0x0000
0x82	R/W	Secondary Calibration Bin 2	0x0000
0x83	R/W	Secondary Calibration Bin 3	0x0000
0x84	R/W	Secondary Calibration Bin 4	0x0000
0x85	R/W	Secondary Calibration Bin 5	0x0000
0x86	R/W	Secondary Calibration Bin 6	0x0000
0x87	R/W	Secondary Calibration Bin 7	0x0000
0x88	R/W	Secondary Calibration Bin 8	0x0000
0x89	R/W	Secondary Calibration Bin 9	0x0000
0x8A	R/W	Secondary Calibration Bin 10	0x0000
0x8B	R/W	Secondary Calibration Bin 11	0x0000
0x8C	R/W	Secondary Calibration Bin 12	0x0000
0x8D	R/W	Secondary Calibration Bin 13	0x0000
0x8E	R/W	Secondary Calibration Bin 14	0x0000
0x8F	R/W	Secondary Calibration Bin 15	0x0000
0x90	R/W	Battery mode wakeup period setting	0x0100

7. NCS32100 will determine values based on sensor characteristics.
8. Registers 0x12 through 0x21 will be populated with results of self calibration, or user input based on off−line calibration.

APPENDIX A: INTERNAL CONFIGURATION REGISTER DEFINITIONS

LC OSCILLATOR GAIN DAC

Address 0x00 − Default Value 0x0000

This register shows the excitation coil oscillator gain. It is a **read only** register. It is periodically updated by the NCS32100 to get the desired receiver coil magnitudes.

Bit	Name	Description
15:8		Reserved.
7:0	USER_LC_GAIN_DAC_FSM[7:0]	LC Oscillator Gain drive strengths. The current LC Oscillator Gain DAC value. **Code** **Gain Calculation** 0b'00xxxxxx 0.4 + (0.05 * N) mA 0b'01xxxxxx 3.6 + (0.1 * N) mA 0b'10xxxxxx 10 + (0.2 * N) mA 0b'11xxxxxx 22.8 + (0.4 * N) mA where 'N' is the 'xxxxxx' portion of the code

RECEIVER PIN AMPLIFIER GAIN (DAC SETTINGS)

Address 0x01 − Default Value 0x0000

This register shows the programmable amplifier gain setting. It is a **read only** register. The programmable gain amplifier is used to amplify the receiver coil signals before they are digitized. The programmable gain amplifier settings are periodically updated by the NCS32100 to get the best possible dynamic range on the sensor receiver signals.

Bit	Name	Description
15:14		Reserved.
13:8	USER_PGA_GAIN_DAC_FSM_COARSE[5:0]	Programmable Amplifier Gain settings for Coarse Coils Block. The current PGA Gain DAC value used by the Coarse block. NOTE: The PGA Gain G_{PGA} is related to USER_PGA_GAIN_DAC_COARSE[5:0] by the equation: G_{PGA} = (1.189^USER_PGA_GAIN_DAC_COARSE[5:0]) * 10.
7:6		Reserved.
5:0	USER_PGA_GAIN_DAC_FSM_FINE[5:0]	Programmable Amplifier Gain settings for Fine Coils Block. The current PGA Gain DAC value used by the Fine block. NOTE: The PGA Gain G_{PGA} is related to USER_PGA_GAIN_DAC_FINE[5:0] by the equation: G_{PGA} = (1.189^USER_PGA_GAIN_DAC_FINE[5:0]) * 10.

DIGITAL CONTROL

Address 0x03 − Default 0x0000

The Digital Control Register allows the user to turn the angle extrapolation feature on or off. It also allows the user to run the open coil detect check.

Bit	Name	Description
15	USER_ABSALGO_EXTRP_DIS	User Absolute Algorithm Extrapolation Disable. Disable the angle extrapolation in the absolute algorithm when the LC oscillator gain, PGA gain or PGA offset are updated
14		Reserved.
13	USER_OPENCOILDET_RUN	USER Open Coils Detection Run. Run the open coils detection FSM. The bit is reset to 0 when the FSM is finished.
12		Reserved.
11	USER_TURN_RST	Turn Counter Reset. Reset the Turn counter to 0. The register bit is automatically reset to 0
10:0		Reserved.

STATUS

Address 0x04 − Default 0x0000

The status register holds the results of the open coil detect test.

Bit	Name	Description
15:14		Reserved.
13	USER_OPENCOILDET_FLAG	User Open Coil Detection Flag. A 1 signifies that there is an unconnected coil.
12:0		Reserved.

SELECTION MATRIX FOR COARSE BLOCK

Address 0x10 − Default 0x0000

This register holds the setting for the connection between the REC pins and the sensor coarse loops.

Bit	Name	Description
15:14	SEL_MATRIX_RECT3P_COARSE[1:0]	Selection Matrix for Rectifier3 Positive Input on Coarse block. See table below for decode.
13:12	SEL_MATRIX_RECT3N_COARSE[1:0]	Selection Matrix for Rectifier3 Negative Input on Coarse block. If SEL_MATRIX_RECT3N_COARSE[1:0] = SEL_MATRIX_RECT3P_COARSE[1:0] then the rectifier3 negative input of the Coarse block is connected to DC. Otherwise see table below for decode.
11:10	SEL_MATRIX_RECT2P_COARSE[1:0]	Selection Matrix for Rectifier2 Positive Input on Coarse Block. See table below for decode.
9:8	SEL_MATRIX_RECT2N_COARSE[1:0]	Selection Matrix for Rectifier2 Negative Input on Coarse Block. If SEL_MATRIX_RECT2N_COARSE[1:0] = SEL_MATRIX_RECT2P_COARSE[1:0] then the rectifier2 negative input of the Coarse block is connected to DC. Otherwise see table below for decode.
7:6	SEL_MATRIX_RECT1P_COARSE[1:0]	Selection Matrix for Rectifier1 Positive Input on Coarse Block. See table below for decode.
5:4	SEL_MATRIX_RECT1N_COARSE[1:0]	Selection Matrix for Rectifier1 Negative Input on Coarse Block. If SEL_MATRIX_RECT1N_COARSE[1:0] = SEL_MATRIX_RECT1P_COARSE[1:0] then the rectifier1 negative input of the Coarse block is connected to DC. Otherwise see table below for decode.
3:2	SEL_MATRIX_RECT0P_COARSE[1:0]	Selection Matrix for Rectifier0 Positive Input on Coarse Block. See table below for decode.
1:0	SEL_MATRIX_RECT0N_COARSE[1:0]	Selection Matrix for Rectifier0 Negative Input on Coarse Block. If SEL_MATRIX_RECT0N_COARSE[1:0] = SEL_MATRIX_RECT0P_COARSE[1:0] then the rectifier0 negative input of the Coarse block is connected to DC. Otherwise see table below for decode.

Table 13. COARSE BLOCK RECTIFIER CONTROL SELECTION

Coarse Register Input	Channel Select	Pin
00	0	REC0
00	1	REC4
01	0	REC1
01	1	REC5
10	0	REC2
10	1	REC6
11	0	REC3
11	1	REC7

SELECTION MATRIX FOR FINE BLOCK

Address 0x11 − Default 0x0000

This register holds the setting for the connection between the REC pins and the sensor fine loops.

Bit	Name	Description
15:14	SEL_MATRIX_RECT3P_FINE[1:0]	Selection Matrix for Rectifier3 Positive Input on Fine Block. See table below for decode.
13:12	SEL_MATRIX_RECT3N_FINE[1:0]	Selection Matrix for Rectifier3 Negative Input on Fine Block. If SEL_MATRIX_RECT3N_FINE[1:0] = SEL_MATRIX_RECT3P_FINE[1:0] then the rectifier3 negative input of the fine block is connected to DC. Otherwise see table below for decode.
11:10	SEL_MATRIX_RECT2P_FINE[1:0]	Selection Matrix for Rectifier2 Positive Input on Fine Block. See table below for decode.
9:8	SEL_MATRIX_RECT2N_FINE[1:0]	Selection Matrix for Rectifier2 Negative Input on Fine Block. If SEL_MATRIX_RECT2N_FINE[1:0] = SEL_MATRIX_RECT2P_FINE[1:0] then the rectifier2 negative input of the fine block is connected to DC. Otherwise see table below for decode.
7:6	SEL_MATRIX_RECT1P_FINE[1:0]	Selection Matrix for Rectifier1 Positive Input on Fine Block. See table below for decode.
5:4	SEL_MATRIX_RECT1N_FINE[1:0]	Selection Matrix for Rectifier1 Negative Input on Fine Block. If SEL_MATRIX_RECT1N_FINE[1:0] = SEL_MATRIX_RECT1P_FINE[1:0] then the rectifier1 negative input of the fine block is connected to DC. Otherwise see table below for decode.
3:2	SEL_MATRIX_RECT0P_FINE[1:0]	Selection Matrix for Rectifier0 Positive Input on Fine Block. See table below for decode.
1:0	SEL_MATRIX_RECT0N_FINE[1:0]	Selection Matrix for Rectifier0 Negative Input on Fine Block. If SEL_MATRIX_RECT0N_FINE[1:0] = SEL_MATRIX_RECT0P_FINE[1:0] then the rectifier0 negative input of the fine block is connected to DC. Otherwise see table below for decode.

Table 14. FINE BLOCK RECTIFIER CONTROL SELECTION

Coarse Register Input	Channel Select	Pin
00	0	REC4
00	1	REC0
01	0	REC5
01	1	REC1
10	0	REC6
10	1	REC2
11	0	REC7
11	1	REC3

The next 16 registers hold the calibration coefficients used to calibrate out any sensor non−linearity. See the calibration section for more details. These registers are updated by the calibration routine every time that it is run.

CLARKE TRANSFORM IMAGINARY COEFFICIENT 0 FOR COARSE BLOCK

Address 0x12 − Default 0x0000

Bit	Name	Description
15:0	CLARKE_COARSE_IM_A[15:0]	Filter 0 Clarke Transform Imaginary Coefficient for Coarse Block. An Imaginary Coefficient of the Clarke Transform. This coefficient is applied to the Filter/Rectifier 0 output of the Coarse Block. This register is a fixed−point number with 14 sign bits, in Q format as Q2.14

CLARKE TRANSFORM IMAGINARY COEFFICIENT 1 FOR COARSE BLOCK

Address 0x13 − Default 0x0000

Bit	Name	Description
15:0	CLARKE_COARSE_IM_B[15:0]	Filter 1 Clarke Transform Imaginary Coefficient for Coarse Block. An Imaginary Coefficient of the Clarke Transform. This coefficient is applied to the Filter/Rectifier 1 output of the Coarse Block. This register is a fixed−point number with 14 sign bits, in Q format as Q2.14

CLARKE TRANSFORM IMAGINARY COEFFICIENT 2 FOR COARSE BLOCK

Address 0x14 − Clarke Transform Imaginary Coefficient C for Coarse Block

Bit	Name	Description
15:0	CLARKE_COARSE_IM_C[15:0]	Filter 2 Clarke Transform Imaginary Coefficient for Coarse Block. An Imaginary Coefficient of the Clarke Transform. This coefficient is applied to the Filter/Rectifier 2 output of the Coarse Block. This register is a fixed−point number with 14 sign bits, in Q format as Q2.14

CLARKE TRANSFORM IMAGINARY COEFFICIENT 3 FOR COARSE BLOCK

Address 0x15 − Default 0x0000

Bit	Name	Description
15:0	CLARKE_COARSE_IM_D[15:0]	Filter 3 Clarke Transform Imaginary Coefficient for Coarse Block. An Imaginary Coefficient of the Clarke Transform. This coefficient is applied to the Filter/Rectifier 3 output of the Coarse Block. This register is a fixed−point number with 14 sign bits, in Q format as Q2.14

CLARKE TRANSFORM REAL COEFFICIENT 0 FOR COARSE BLOCK

Address 0x16 − Default 0x0000

Bit	Name	Description
15:0	CLARKE_COARSE_RE_A[15:0]	Filter 0 Clarke Transform Real Coefficient for Coarse Block. A Real Coefficient of the Clarke Transform. This coefficient is applied to the Filter/Rectifier 0 output of the Coarse Block. This register is a fixed−point number with 14 sign bits, in Q format as Q2.14

CLARKE TRANSFORM REAL COEFFICIENT 1 FOR COARSE BLOCK

Address 0x17 − Default 0x0000

Bit	Name	Description
15:0	CLARKE_COARSE_RE_B[15:0]	Filter 1 Clarke Transform Real Coefficient for Coarse Block. A Real Coefficient of the Clarke Transform. This coefficient is applied to the Filter/Rectifier 1 output of the Coarse Block. This register is a fixed−point number with 14 sign bits, in Q format as Q2.14

CLARKE TRANSFORM REAL COEFFICIENT 2 FOR COARSE BLOCK

Address 0x18 − Default 0x0000

Bit	Name	Description
15:0	CLARKE_COARSE_RE_C[15:0]	Filter 2 Clarke Transform Real Coefficient for Coarse Block. A Real Coefficient of the Clarke Transform. This coefficient is applied to the Filter/Rectifier 2 output of the Coarse Block. This register is a fixed−point number with 14 sign bits, in Q format as Q2.14

CLARKE TRANSFORM REAL COEFFICIENT 3 FOR COARSE BLOCK

Address 0x19 − Default 0x0000

Bit	Name	Description
15:0	CLARKE_COARSE_RE_D[15:0]	Filter 3 Clarke Transform Real Coefficient for Coarse Block. A Real Coefficient of the Clarke Transform. This coefficient is applied to the Filter/Rectifier 3 output of the Coarse Block. This register is a fixed−point number with 14 sign bits, in Q format as Q2.14

CLARKE TRANSFORM IMAGINARY COEFFICIENT 0 FOR FINE BLOCK

Address 0x1A − Default 0x0000

Bit	Name	Description
15:0	CLARKE_FINE_IM_A[15:0]	Filter 0 Clarke Transform Imaginary Coefficient for Fine Block. An Imaginary Coefficient of the Clarke Transform. This coefficient is applied to the Filter/Rectifier 0 output of the Fine Block. This register is a fixed−point number with 14 sign bits, in Q format as Q2.14

CLARKE TRANSFORM IMAGINARY COEFFICIENT 1 FOR FINE BLOCK

Address 0x1B − Default 0x0000

Bit	Name	Description
15:0	CLARKE_FINE_IM_B[15:0]	Filter 1 Clarke Transform Imaginary Coefficient for Fine Block. An Imaginary Coefficient of the Clarke Transform. This coefficient is applied to the Filter/Rectifier 1 output of the Fine Block. This register is a fixed−point number with 14 sign bits, in Q format as Q2.14

CLARKE TRANSFORM IMAGINARY COEFFICIENT 2 FOR FINE BLOCK

Address 0x1C − Default 0x0000

Bit	Name	Description
15:0	CLARKE_FINE_IM_C[15:0]	Filter 2 Clarke Transform Imaginary Coefficient for Fine Block. An Imaginary Coefficient of the Clarke Transform. This coefficient is applied to the Filter/Rectifier 2 output of the Fine Block. This register is a fixed−point number with 14 sign bits, in Q format as Q2.14

CLARKE TRANSFORM IMAGINARY COEFFICIENT 3 FOR FINE BLOCK

Address 0x1D − Default 0x0000

Bit	Name	Description
15:0	CLARKE_FINE_IM_D[15:0]	Filter 3 Clarke Transform Imaginary Coefficient for Fine Block. An Imaginary Coefficient of the Clarke Transform. This coefficient is applied to the Filter/Rectifier 3 output of the Fine Block. This register is a fixed−point number with 14 sign bits, in Q format as Q2.14

CLARKE TRANSFORM REAL COEFFICIENT 0 FOR FINE BLOCK

Address 0x1E − Default 0x0000

Bit	Name	Description
15:0	CLARKE_FINE_RE_A[15:0]	Filter 0 Clarke Transform Real Coefficient for Fine Block. A Real Coefficient of the Clarke Transform. This coefficient is applied to the Filter/Rectifier 0 output of the Fine Block. This register is a fixed−point number with 14 sign bits, in Q format as Q2.14

CLARKE TRANSFORM REAL COEFFICIENT 1 FOR FINE BLOCK

Address 0x1F − Default 0x0000

Bit	Name	Description
15:0	CLARKE_FINE_RE_B[15:0]	Filter 1 Clarke Transform Real Coefficient for Fine Block. A Real Coefficient of the Clarke Transform. This coefficient is applied to the Filter/Rectifier 1 output of the Fine Block. This register is a fixed−point number with 14 sign bits, in Q format as Q2.14

CLARKE TRANSFORM REAL COEFFICIENT 2 FOR FINE BLOCK

Address 0x20 − Default 0x0000

Bit	Name	Description
15:0	CLARKE_FINE_RE_C[15:0]	Filter 2 Clarke Transform Real Coefficient for Fine Block. A Real Coefficient of the Clarke Transform. This coefficient is applied to the Filter/Rectifier 2 output of the Fine Block. This register is a fixed−point number with 14 sign bits, in Q format as Q2.14

CLARKE TRANSFORM REAL COEFFICIENT 3 FOR FINE BLOCK

Address 0x21 − Default 0x0000

Bit	Name	Description
15:0	CLARKE_FINE_RE_D[15:0]	Filter 3 Clarke Transform Real Coefficient for Fine Block. A Real Coefficient of the Clarke Transform. This coefficient is applied to the Filter/Rectifier 3 output of the Fine Block. This register is a fixed−point number with 14 sign bits, in Q format as Q2.14

VELOCITY COEFFICIENT MSB FOR ABSOLUTE ALGORITHM EXTRAPOLATION

Address 0x22 − Default 0x0000

This register allows for the tuning of the extrapolation algorithm. Please refer to the extrapolation tuning section for more details.

Bit	Name	Description
15:8		
7:0	ABSALGO_EXTRP_KV[23:16]	Velocity Coefficient MSBs for Absolute Algorithm Extrapolation. The coefficient used in the velocity portion in the angle extrapolation algorithm.

VELOCITY COEFFICIENT LSB FOR ABSOLUTE ALGORITHM EXTRAPOLATION

Address 0x23 − Default 0x0000

This register allows for the tuning of the extrapolation algorithm. Please refer to the extrapolation tuning section for more details.

Bit	Name	Description
15:0	ABSALGO_EXTRP_KV[15:0]	Velocity Coefficient LSBs for Absolute Algorithm Extrapolation. The coefficient used in the velocity portion in the angle extrapolation algorithm.

VELOCITY COEFFICIENT MSB FOR DIGITAL FILTER EXTRAPOLATION

Address 0x26 − Default 0x0000

This register allows for the tuning of the extrapolation algorithm. Please refer to the extrapolation tuning section for more details.

Bit	Name	Description
15:8		
7:0	DIG_FILTERS_EXTRP_KV[23:16]	Velocity Coefficient MSBs for the Delay Extrapolator. The coefficient used in the velocity portion of the Delay extrapolator.

VELOCITY COEFFICIENT LSB FOR DIGITAL FILTER EXTRAPOLATION

Address 0x27 – Default 0x0000

This register allows for the tuning of the extrapolation algorithm. Please refer to the extrapolation tuning section for more details.

Bit	Name	Description
15:0	DIGFILTERS_EXTRP_KV[15:0]	Velocity Coefficient LSBs for Digital Filter Extrapolation. The coefficient used in the velocity portion of the Delay extrapolator.

LOW PASS FILTER FOR DIGITAL FILTER

Address 0x2A – Default 0x0000

This register allows for the tuning of the digital low pass filter that is applied to the digitized receiver coil samples.

Bit	Name	Description
15:6		Reserved.
5:4	LPF_K_ANG[1:0]	Low Pass Filter Coefficient for Angle = 00 then the angle LPF coefficient is 16(default); = 01 then the angle LPF coefficient is 32; = 10 then the angle LPF coefficient is 64; = 11 then the angle LPF coefficient is 128.
3:2	LPF_K_VEL[1:0]	Low Pass Filter Coefficient for Velocity = 00 then the velocity LPF coefficient is 16 (default); = 01 then the velocity LPF coefficient is 32; = 10 then the velocity LPF coefficient is 64; = 11 then the velocity LPF coefficient is 128.
1:0	Reserved	Reserved.

DIGITAL CONTROL

Address 0x2B –Default 0x0000

This register allows the user to control the source of the output data. Below is a graphical representation of the MUX options

Bit	Name	Description
15:6		Reserved.
5:4	ANGLE_OUTPUT_MUX	Angle Output Mux The angle coming out of the DSP block is the 00 = Delay Extrapolated angle 01 = Delay Extrapolated angle 10 = Digital LPF angle 11 = Absolute Algorithm angle
3		Reserved.
2	CAL_TURN_DIRECTION	0 = Position set to increasing for clockwise rotation in calibration routine. 1 = Position set to increasing for counter clockwise rotation in calibration routine.
1	ADCCAL_AT_POR_DIS	0 = ADC Calibration at POR Disable (default) 1 = ADC Calibration is not run during startup routine
0	OPENCOILDET_AT_POR_DIS.	0 = Open Coil Detection at POR Disable. 1 = Open Coil Detection FSM is not run during startup routine.

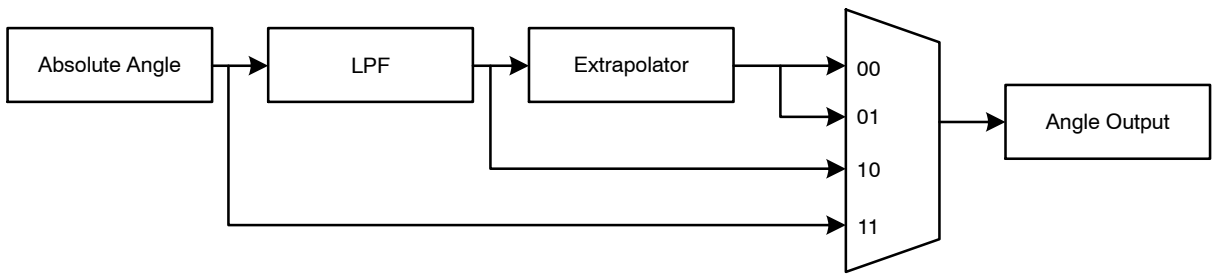


Figure 24.

CHANNEL SELECT

Address 0x2F − Default 0x0000

This register controls where the outputs of each ADC go. There are two ADCs, and their mapping to the fine and coarse coils is as follows:

Bit	Name	Description
15:1		Reserved.
0	FINE_CHANB_SEL	When 1, Receiver pins 0−3 are mapped to the Fine block. When 0, Receiver pins 0−3 are mapped to the Coarse block. Receiver pins 4−7 are mapped to the opposite block.

LC OSCILLATOR GAIN DAC CONTROL

Address 0x40 − Default 0x0000

This register allows the user to bypass the automatically calculated value for the LC oscillator gain setting. In order to force a gain setting, the user must set bit 15 and put the desired gain value in bit 7:0. This will turn off the automatic periodic gain adjustment and instead use the value set in bits 7:0.

Bit	Name	Description
15	SHDW_LC_GAIN_DAC_EN	LC Oscillator Gain DAC Enable. The LC Oscillator is forced to use the SHDW_LC_GAIN_DAC register value.
14:8		Reserved.
7:0	SHDW_LC_GAIN_DAC[7:0]	LC Oscillator Gain DAC. The LC Oscillator current $I_{LC,OSC}$ is related to the DAC value from the equation: $I_{LC,OSC}$ = 2^(SHDW_LC_GAIN_DAC[7:6]) * (3 + SHDW_LC_GAIN_DAC[5:0]/64) μA.

LC OSCILLATOR GAIN TIME CONTROL

Address 0x41 − Default 0x0000

This register allows the user to change the periodic update timing for the LC oscillator gain.

Bit	Name	Description
15:0	SHDW_LC_GAIN_TIME[15:0]	LC Oscillator Gain Time Control. The LC Oscillator Gain controller updates the LC Oscillator Gain DAC value every SHDW_LC_GAIN_TIME[15:0] * 2 μs (2 μs = 80 clock periods at 40 MHz). If SHDW_LC_GAIN_TIME[15:0]=0 then the FSM does NOT update the LC Oscillator Gain DAC value.

PGA COARSE GAIN DAC CONTROL

Address 0x42 − Default 0x0000

This register allows the user to bypass the automatically calculated value for the coarse PGA gain setting. In order to force a gain setting, the user must set bit 15 and put the desired gain value in bit 5:0. This will turn off the automatic periodic gain adjustment and instead use the value set in bits 5:0.

Bit	Name	Description
15	SHDW_PGA_COARSE_GAIN_DAC_EN	PGA Gain DAC Enable Coarse Block. The Coarse block PGA Gain DAC is forced to use the SHDW_PGA_COARSE_GAIN_DAC register value.
14:6		Reserved.
5:0	SHDW_PGA_COARSE_GAIN_DAC[5:0]	PGA Gain DAC Coarse Block. The PGA Gain G_{PGA} is related to SHDW_PGA_GAIN_DAC[5:0] by the equation: G_{PGA} = (1.189^SHDW_PGA_GAIN_DAC[5:0]) * 10, for codes 0 to 22.

PGA FINE GAIN DAC CONTROL

Address 0x43 – Default 0x0000

This register allows the user to bypass the automatically calculated value for the fine PGA gain setting. In order to force a gain setting, the user must set bit 15 and put the desired gain value in bit 5:0. This will turn off the automatic periodic gain adjustment and instead use the value set in bits 5:0.

Bit	Name	Description
15	SHDW_PGA_FINE_GAIN_DAC_EN	PGA Gain DAC Enable Fine Block. The Fine block PGA Gain DAC is forced to use the SHDW_PGA_FINE_GAIN_DAC register value.
14:6		Reserved.
5:0	SHDW_PGA_FINE_GAIN_DAC[5:0]	PGA Gain DAC Fine Block. The PGA Gain G_{PGA} is related to SHDW_PGA_FINE_GAIN_DAC[5:0] by the equation: G_{PGA} = (1.189^SHDW_PGA_GAIN_DAC[4:0]) * 10, for codes 0 to 22.

PGA GAIN TIME CONTROL

Address 0x44 – Default 0x0000

This register allows the user to change the periodic update timing for the PGA gain.

Bit	Name	Description
15:0	SHDW_PGA_GAIN_TIME[15:0]	PGA Gain Time Control Both blocks update the PGA Gain DAC value every SHDW_PGA_FINE_GAIN_TIME[15:0] * 2 μs (2 μs = 80 clock periods at 40 MHz). If SHDW_FINE_PGA_GAIN_TIME[15:0]=0 then the controllers do NOT update the PGA Gain DAC values.

PGA COARSE OFFSET DAC CONTROL

Address 0x45 – Default 0x0000

This register allows the user to bypass the automatically calculated value for the coarse PGA offset setting. In order to force a gain setting, the user must set bit 15 and put the desired gain value in bit 7:0. This will turn off the automatic periodic gain adjustment and instead use the value set in bits 7:0.

Bit	Name	Description
15	SHDW_PGA_COARSE_OFFSET_DAC_EN	PGA Offset DAC Enable Coarse Block. The Coarse block PGA Offset DAC is forced to use the SHDW_PGA_COARSE_OFFSET_DAC register value.
14:8		Reserved.
7:0	SHDW_PGA_COARSE_OFFSET_DAC[7:0]	PGA Offset DAC. SHDW_PGA_COARSE_OFFSET_DAC[7:0] is a 2 μs complement value covering an offset range of [−32mV, 32mV]. The DAC LSB is equivalent to 250 μV.

PGA FINE OFFSET DAC CONTROL

Address 0x46 – Default 0x0000

This register allows the user to bypass the automatically calculated value for the fine PGA offset setting. To force a gain setting, the user must set bit 15 and put the desired gain value in bit 7:0. This will turn off the automatic periodic gain adjustment and instead use the value set in bits 7:0.

Bit	Name	Description
15	SHDW_PGA_FINE_OFFSET_DAC_EN	PGA Offset DAC Enable Fine Block. The Fine block PGA Offset DAC is forced to use the SHDW_PGA_FINE_OFFSET_DAC register value.
14:8		Reserved.
7:0	SHDW_PGA_FINE_OFFSET_DAC[7:0]	PGA Offset DAC. SHDW_PGA_FINE_OFFSET_DAC[7:0] is a 2 complement value covering an offset range of [−32 mV, 32 mV]. The DAC LSB is equivalent to 250 μV.

PGA OFFSET TIME CONTROL

Address 0x47 − Default 0x0000

This register allows the user to change the periodic update timing for the PGA offset.

Bit	Name	Description
15:0	SHDW_PGA_COARSE_OFFSET_TIME[15:0]	PGA Offset Time Control x Fine/Coarse Block. Both blocks update the PGA Offset DAC value every SHDW_PGA_COARSE_OFFSET_TIME[15:0] * 2 μs (2 μs = 80 clock periods at 40 MHz). If SHDW_PGA_COARSE_OFFSET_TIME[15:0]=0 then the controllers do NOT update the PGA Offset DAC values.

ANGLE EXTRAPOLATOR TIME CONTROL

Address 0x48 − Default 0x0000

This register controls the update rate of the angle extrapolation feature.

Bit	Name	Description
15:0	SHDW_LC_GAIN_EXTRP_TIME[15:0]	Angle Extrapolation Time Control. The Angle Extrapolation runs for SHDW_LC_GAIN_EXTRP_TIME [15:0] * 2 μs (2 μs = 80 clock periods at 40 MHz) after a qualifying event. If SHDW_LC_GAIN_EXTRP_TIME [15:0]=0 then the Angle Extrapolator does NOT run.

NORMAL WAKEUP DELAY

Address 0x4C − Default 0x1000

This register controls the number of clock cycles used to wait for the analog to settle after transitioning to the Functional state from the sleep state.

Bit	Name	Description
15:0	LP_Normal_Delay_Time[15:0]	Number of clock cycles used to wait for the analog to settle after coming out of sleep mode.

STARTUP DELAY TIME MSB

Address 0x4D − Default 0x0006

This register controls the startup timing of analog blocks in the NCS32100. This is the amount to time that the digital sequencing is held in reset to allow settling time for the analog front end circuitry to settle during the initial startup routine.

Bit	Name	Description
15:3		Reserved
2:0	STARTUP_SETTLE[18:16]	Startup Settle MSB Number of clock cycles used to wait for the analog to settle during startup.

STARTUP DELAY TIME LSB

Address 0x4E − Default 0x1A80

This register controls the startup timing of analog blocks in the NCS32100. This is the amount to time that the digital sequencing is held in reset to allow settling time for the analog front end circuitry to settle during the initial startup routine.

Bit	Name	Description
15:0	STARTUP_SETTLE[15:0]	Startup Settle LSB Number of clock cycles used to wait for the analog to settle during startup.

DSP WAKEUP DELAY

Address 0x4F Default: 0x1000

This register controls the startup timing of the DSP block in the NCS32100. This is the amount to time that the digital sequencing is held in reset to allow settling time for the analog front−end circuitry to settle during battery mode when the device transitions between a sleep state and a low power.

Bit	Name	Description
15:0	DSP_WAKEUP_DLY_TIME[15:0]	Number of clock cycles used to wait for the analog to settle after transitioning to the Low power state from the Sleep state.

AUTOZERO ANGLE MSB

Address 0x50 − Default 0x0000

This register holds the most significant bits that define the encoder 0 index position. It is an offset from the true index that is applied to the position output so that the user can set a custom index. This value is held in non−volatile memory.

Bit	Name	Description
15:4		Reserved.
3:0	SHDW_AUTOZERO_ANGLE[19:16]	Autozero Angle MSB. Value used to offset the Mechanical angle from the Absolute Angle algorithm

AUTOZERO ANGLE LSB

Address 0x51 − Default 0x0000

This register holds the least significant bits that define the encoder 0 index position. It is an offset from the true index that is applied to the position output so that the user can set a custom index. This value is held in non−volatile memory.

Bit	Name	Description
15:0	SHDW_AUTOZERO_ANGLE[15:0]	Autozero Angle LSB. Value used to offset the Mechanical angle from the Absolute Angle algorithm

INPUT/OUTPUT CONTROL

Address 0x57 − Default 0x0000

This register allows the user to select the direction of the external Data pins.

Bit	Name	Description
15:4		Reserved.
3	DATA3_OUT_EN	DATA3 Output Enable. = 0 then the package pin DATA3 is configured to be an output; = 1 then the package pin DATA3 is configured to be an input.
2	DATA2_OUT_EN	DATA2 Output Enable. = 0 then the package pin DATA2 is configured to be an output; = 1 then the package pin DATA2 is configured to be an input.
1	DATA1_OUT_EN	DATA1 Output Enable. = 0 then the package pin DATA1 is configured to be an output; = 1 then the package pin DATA1 is configured to be an input.
0	DATA0_OUT_EN	DATA0 Output Enable. = 0 then the package pin DATA0 is configured to be an output; = 1 then the package pin DATA0 is configured to be an input.

SENSOR SELECTION

Address 0x5F − Default 0x0000

This register should be set based on the number of coarse and fine loops that exist in the sensor.

Bit	Name	Description
15:5		Reserved.
4	SENSOR_COARSE_SEL	Sensor Coarse Selection = 0 then the Coarse sensor period is 3 times the mechanical angle period; = 1 then the Coarse sensor period is 5 times the mechanical angle period;
3		Reserved.
2:0	SENSOR_FINE_SEL[2:0]	Sensor Fine Selection = 000 then the Fine sensor period is 8 times the mechanical angle period; = 001 then the Fine sensor period is 16 times the mechanical angle period; = 010 then the Fine sensor period is 32 times the mechanical angle period; = 011 then the Fine sensor period is 64 times the mechanical angle period; = 100 then the Fine sensor period is 128 times the mechanical angle period; = 101 then the Fine sensor period is 256 times the mechanical angle period;

SECONDARY CALIBRATION COEFFICIENTS

Address 0x80 through 0x8F – Default 0x0000

These registers hold the 16 secondary calibration coefficients that are used to offset the output position in each of 16 bins respectively. These values are position offsets.

Bit	Name	Description
0x80	Bin0[15:0]	Bin0 position offset (covers position from 348.75 – 11.25 degrees)
0x81	Bin1[15:0]	Bin1 position offset (covers position from 11.25 – 33.75 degrees)
0x82	Bin2[15:0]	Bin2 position offset (covers position from 33.75 –56.25 degrees)
0x83	Bin3[15:0]	Bin3 position offset (covers position from 56.25 – 78.75 degrees)
0x84	Bin4[15:0]	Bin4 position offset (covers position from 78.75 – 101.25 degrees)
0x85	Bin5[15:0]	Bin5 position offset (covers position from 101.25 – 123.75 degrees)
0x86	Bin6[15:0]	Bin6 position offset (covers position from 123.75 – 146.25 degrees)
0x87	Bin7[15:0]	Bin7 position offset (covers position from 146.25 – 168.75 degrees)
0x88	Bin8[15:0]	Bin8 position offset (covers position from 168.75 – 191.25 degrees)
0x89	Bin9[15:0]	Bin9 position offset (covers position from 191.25 – 213.75 degrees)
0x8A	Bin10[15:0]	Bin10 position offset (covers position from 213.75 – 236.25 degrees)
0x8B	Bin11[15:0]	Bin11 position offset (covers position from 236.25 – 258.75 degrees)
0x8C	Bin12[15:0]	Bin12 position offset (covers position from 258.75 – 281.25 degrees)
0x8D	Bin13[15:0]	Bin13 position offset (covers position from 281.25 – 303,75 degrees)
0x8E	Bin14[15:0]	Bin14 position offset (covers position from 303.75 – 326.25 degrees)
0x8F	Bin15[15:0]	Bin15 position offset (covers position from 326.25 – 11.25 degrees)

BATTERY MODE WAKEUP PERIOD

Address 0x90 – Default 0x0100

This register allows the user to set the wakeup period during battery mode. The NCS32100 goes into battery mode anytime VCC < VBAT. In battery mode the NCS32100 wakeup periodically to check the position and update the multi−turn count if a full revolution has passed. The shorter the wakeup period, the more current will be pulled from VBAT.

Bit	Name	Description			
15:0	WAKEUP_PER	WAKEUP_PER	Period Timing (ms)	Average current draw on VBAT (mA)	Max speed at which turns count will be tracked (rpm)
		3	3	10	6600
		4	5	5	4000
		21	16	1.45	1250
		31	24	1	830
		42	31	0.8	650
		55	41	0.6	487
		72	53	0.5	370
		100	73	0.4	270
		200	140	0.3	140

NCS32100 Raw Outputs

The table below shows the format for each of the raw data outputs from the NCS32100. These can be converted mathematically to degrees or radians as desired.

Table 15. ANGLE, TURN, AND VELOCITY DATA FORMAT

Data	Bits	Sign	Range	Description
Angle	20	Unsigned	[0, 2^20] Normalized rotation	Angle is an unsigned 20 bits measurement of the absolute angle in 1 revolution. The angle is normalized to 2*π therefore the user has to multiply angle by 2*π to get the angle in radians.
Turn	24	Signed	[−8,388,608:8,388,607]	Turn count is an unsigned 24 bits measurement of the complete revolution number. An overflow/underflow flag is associated with the Turn. The overflow flag will assert / de−assert when count reaches 0. See diagram
Velocity	20	Signed	[−7.5e−4, 7.5e−4] (This covers guaranteed accuracy range of 6000 rpm) 20 bits will represent higher speeds at reduced accuracy.	Velocity is a signed 2's complement 20 bits measurement of the angular speed. Velocity is normalized to (2*π)/1us therefore the customer has to multiply the Velocity by (2*π)/1us to get the physical value of the angular speed. These calculations are left to the master. Velocity data from the SFE is passed through.

Below are examples that show how these formats can be converted to radians and degrees if desired.

Angle Conversion

Example: Received Angle hex value of 0x62626 (Decimal value is 402,982)

Table 16. POSITION CONVERSION EXAMPLE

To convert to radians	
1. Divide received value by scale (2^20).	402,982/1,048,576 = 0.384314 rotations
2. Multiply by 2π to get radians	0.384314 * 2π = 2.41471 radians
Convert to Degrees	
1. Divide received value by scale (2^20).	402,982/1,048,576 = 0.384314 rotations
2. Multiply by 360 to get degrees	0.384314 * 360 = 138.353 Degrees

Velocity Conversion

Example: Received velocity is hex 0xBAADF (signed decimal −289,937)

Table 17. VELOCITY CONVERSION EXAMPLE

To convert to radians	
1. Divide received value by scale (2^29).	−289,937 / 536,870,912 = −0.00054 rotations/μs
2. Multiply by 2π to get radians	−0.00054 * 2π = −0.003393 radians/μs
Convert to Degrees	
1. Divide received value by scale (2^29).	−289,937 / 536,870,912 = −0.00054 rotations/μs
2. Multiply by 360 to get degrees	−0.00054 * 360 = −0.1944 Degrees/ μs

CONCLUSION

The presented NCS32100 reference design is intended to be used as a starting point for users in designing their own end applications. This design does not utilize all possible features of the NCS32100, nor does it discuss PCB sensor design in detail. The full configuration details are available in the NCS32100 datasheet. A firmware appendix containing a detailed explanation of the firmware used in the NCS32100 reference design is available upon appropriate request.

PUBLICATION ORDERING INFORMATION

LITERATURE FULFILLMENT:
Email Requests to: orderlit@onsemi.com

onsemi Website: www.onsemi.com

TECHNICAL SUPPORT
North American Technical Support:
Voice Mail: 1 800−282−9855 Toll Free USA/Canada
Phone: 011 421 33 790 2910

Europe, Middle East and Africa Technical Support:
Phone: 00421 33 790 2910
For additional information, please contact your local Sales Representative

◊