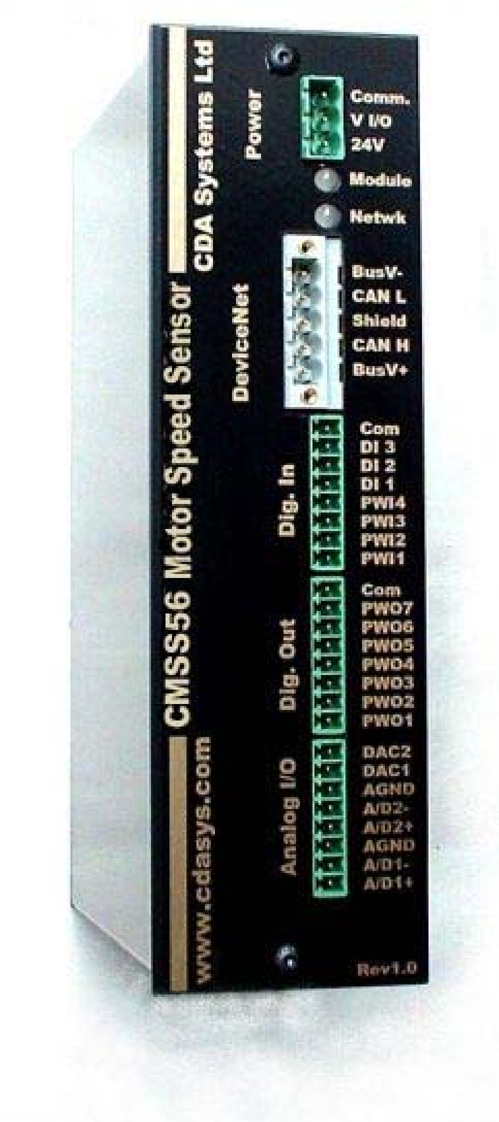
PRELIMINARY

CMSS56

Commutated Motor Speed Sensor



481 Baker St.
London, Ontario
N6C 1Y1 Canada
Tel: (519) 438-2280
Fax: (519) 438-0811
email: admin@cdasyscom

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

REVISION HISTORY

REVISION	DESCRIPTION	ENG	DATE
0.1	Initial Document	ER	12 Jan, 04

APPLICATION OF CMSS56

The CMSS56 module can accurately determine the rotary speed of a DC commutated/brushless motor based solely on the motor’s supply current. The module monitors the ripple in the motor current due to commutation and back-EMF to track the motor speed. The module goes one step further and monitors how confident it is in its ability to perform its intended function.

Connection of Wiring

The input to the module can be a typical current shunt, Hall effect sensor, or other current measurement device. The analog inputs to the module are rugged, electrically isolated from the main power supply, and have programmable gain.

The module has a secondary analog input channel that can be used to measure some other physical property at the same time. Both analog input channels are electrically isolated from the module’s supply current, but they do share a common ‘analog ground’ signal. If using the secondary analog input channel to perform measurements, either the current should be measured with a low-side shunt (Figure 2) or with a Hall effect sensor (Figure 3).

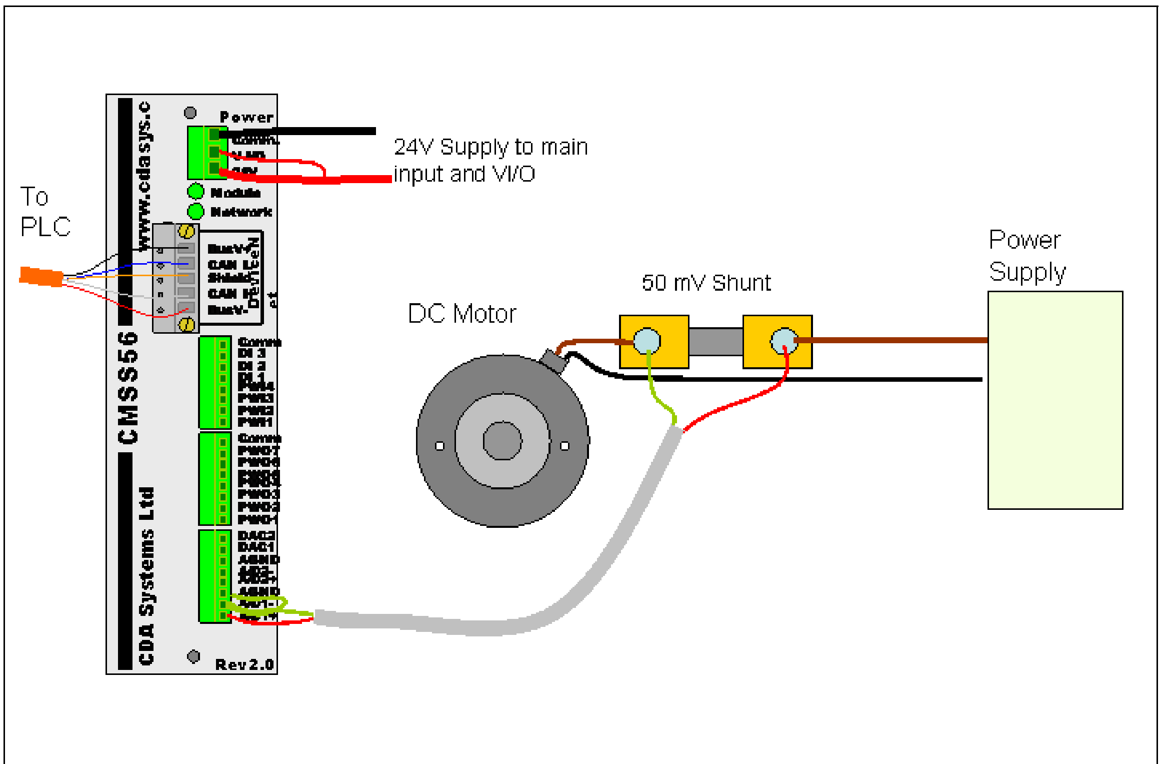


Figure 1: Basic Connection of Speed Sensor – High-Side Shunt

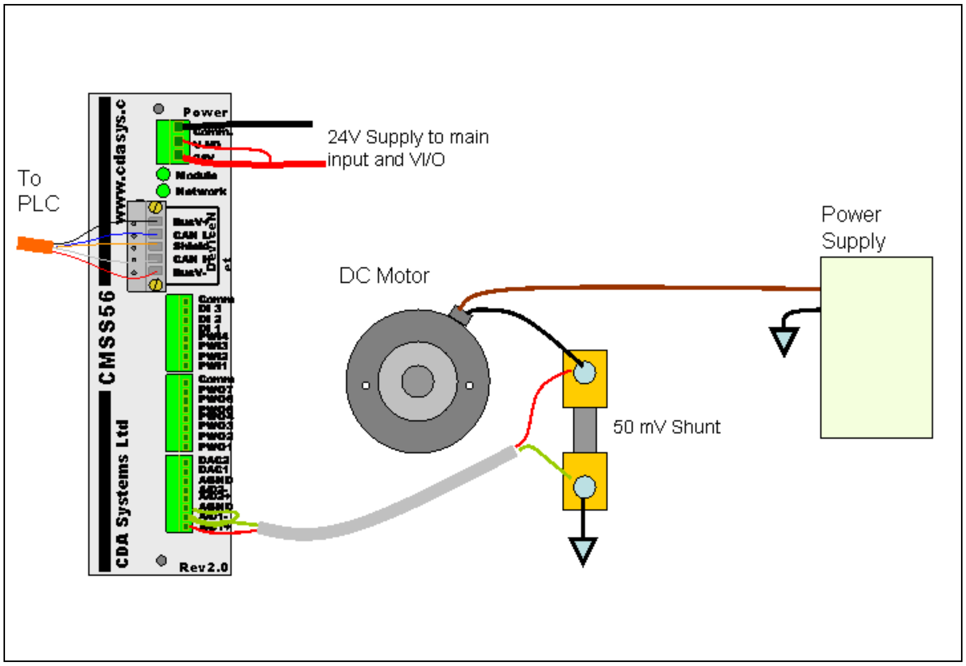


Figure 2: Alternate Configuration - Low Side Shunt

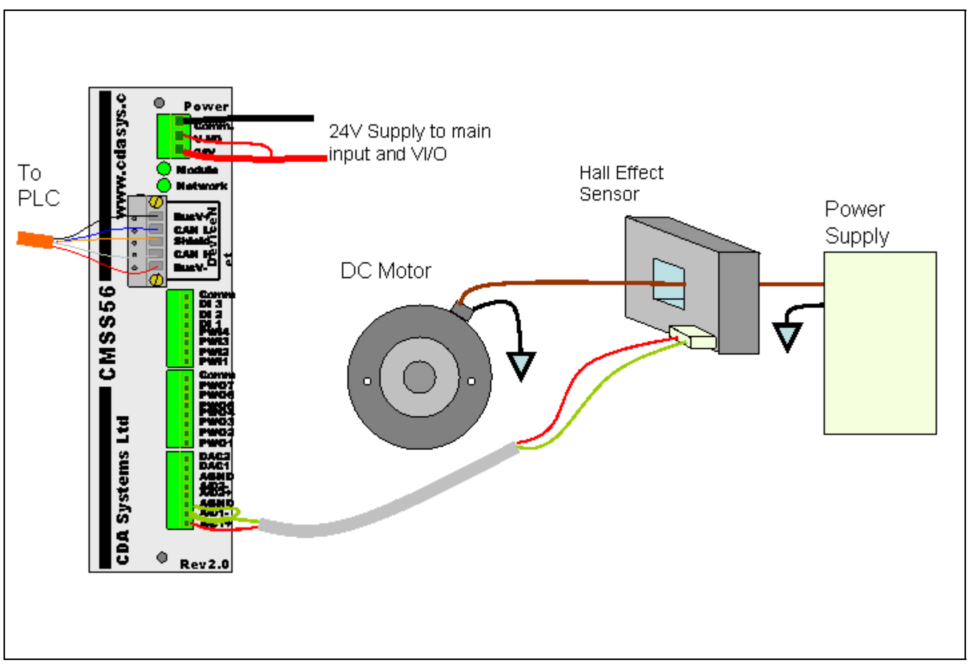


Figure 3: Alternate Configuration - Hall Effect Sensor

BACKGROUND OF MEASUREMENT

The CMSS56 module uses the motor's commutation current ripple to determine motor speed. The frequency of the ripple is equal to the number of commutation segments times the frequency of rotation. The module requires the number of commutator segments, plus a few characteristics of the motor's current noise and ripple to perform its intended function.

The module re-calculates the motor speed on each motor commutation and presents its results in real-time. The calculated speed is based on the motor's last single full rotation.

The module also provides an 'average' speed output which calculates the motor speed based on several rotations of the motor.

Error Detection

Many modern DC motors, especially some of the newer brushless motors, have a very low ripple current which will turn up as micro-volts to a few milli-volts on a typical industrial current shunt used to measure motor current. Often the 'noise' level across one of these shunts is actually larger in signal level than the ripple due to commutation current.

The module's signal processing algorithms were designed to allow the module to accurately make its intended measurement in the presence of noise. The module's software analyzes its own results to determine its confidence in the accuracy of the measurement. It presents a 'Confidence' bit which is set only when the processed input waveform appears consistently accurate. The module also presents an output 'error count' which is incremented each and every time that the module suspects that it could have been 'tricked' by input noise.

Analog Channel Gain & Scaling

The CMSS56 modules have programmable gain controls for each of the analog inputs. The programming bit patterns and their corresponding gain setting are outlined in Table 1 on page 4 of this document. The 'full scale' input level is a function of the programmed gain. The full-scale input at the module's A/D converter is set to 1.0 volts. Thus if the input gain is specified to be 20, the module's full-scale input level is 50mV.

One of the module's input variables is 'full scale input' **value.** The module's 'full scale input' value is written to the module such that when a full-scale input **level** is presented to the module's analog input, the specified 'full scale input' **value** will be presented to the PLC. The 'value' can be any meaningful 16-bit number.

One recommended scaling for current is set to 256 times the full-scale input current (in amperes). The most significant byte of the two-byte output is the integer units (ie. amperes for channel 1). The least significant byte is in binary fractional form, where hexadecimal '80' is 0.5 Amperes, and the least significant bit represents 1/256 amperes (3.9 milli-amperes). *NOTE: The user can invent any scale that they wish. For example, the result could be scaled in milli-amperes, where the full-scale value is 50,000.)* Examples are given on page 5 of this document.

Each time that an input gain is changed, the module re-calibrates the DC input offset error. This calibration will take a few hundred milli-seconds, after which normal operation will continue.

Table 1: Module Input Gain Settings

Module Input (GainA, GainB)	Module Gain (V/V)	Full-scale Voltage
0	0	Not Applicable
1	1	1V
2	2	500mV
3	5	200mV
4	10	100mV
5	20	50mV
6	50	20mV
7	100	10mV

Example of Gain & Scaling

Supposing you had a motor connected up through a 50 Ampere (50 mV) shunt, with expected peak current of 35 amperes. On the secondary channel you need to measure RMS vibration of the motorized device with an accelerometer with a sensitivity of 100 mV/G, and a maximum acceleration of 0.65 G.

Recommended Steps to Setup of Module

1. Determine input gain settings to make optimal use of the module's A/D converter resolution.

   For motor current channel, we can check Table 1 and find that the gain setting of 20 works perfectly, having 50 mV as the full-scale input.

   For the accelerometer channel, we calculate the maximum input voltage to the module to be 0.65G * 100 mV/G = 65 mV. Referring to Table 1, we find a gain setting of 10 will give us a full-scale input voltage of 100 mV.

2. Determine the 'Full Scale Value' parameters for each channel.

   For the motor current channel, we have a full-scale input of 50 mV which happens to represent 50 Amperes. The scaling we choose is actually arbitrary, but there are considerations. We could choose our scaling to be in milli-amperes, or other arbitrary scaling.

   Our 'full-scale value' is represented by a positive 16-bit integer, giving us possible numbers between 1 and 65535. If we were to choose a value of '50' as our full scale value, our results would be accurate to within 0.5 Amperes… not very accurate. If we were to choose a value of 50,000, the result will be in milli-amperes.

   A third alternative might be 256 times the current (ie. 256 * 50 = 12,800). With this as a selected scaling, the most significant byte of the 16-bit number is the integer portion of the number defining current.

   For the accelerometer channel, our full-scale input is 1G. 1 would certainly not be a good choice for the 'full-scale value'. A value of 1000 would have the output represented in milli-G's.

Setup Speed Tracking Parameters

The speed tracking algorithm needs user input about the motor before it can operate. The algorithm tracks the motor speed via its commutation characteristics. The primary two things that the module needs to know about the motor are the number of commutator segments (or number of commutations per rotation for brushless motors), and the noise floor in the current when the motor is running.

The module has a series of digital filters that can be used to reduce error due to high frequency noise. If the maximum speed of the motor is known, then an appropriate filter can be selected to remove noise above that frequency.

Noise Current

The noise current input parameter is used to help the module distinquish between changes in current due to commutation and changes due to electrical noise such as arcing.

The Noise Current parameter is a 16-bit parameter, scaled in the same units as the FullScale Current input parameter.

For example, if the FullScale Current variable is set to 50,000 to represent 50 Amperes, and a noise current of 0.75 amperes is determined to be optimal, the Noise Current parameter should be set to

$$NoiseCurrent\_Value = \frac{NoiseCurrent(Amps)}{FullScale\_Current(Amps)} * FullScale\_Current\_Value$$

$$NoiseCurrent\_Value = \frac{0.75}{50} * 50000 = 750$$

The module has an output measurement that can be used to assist in determining a value for this variable. The module measures peak-to-peak current. The module can directly measure the noise current with the motor off. The maximum number seen on this measurement is the minimum value that should be used in the Noise Current parameter. By running the motor under test, you can see what the peak to peak current of the motor is. The value for Noise Current should be no more than 1/3 of this value.

Hysteresis

This input parameter is a percentage. It is the percentage of the 'clean' peak-to-peak signal current to be used for speed tracking. Values from 0 to 100 are valid. Values greater than 100 will be treated as 100.

Input Filt

This input parameter is used to select a filter for eliminating high-frequency noise that could cause jitter and/or error in the motor speed tracking algorithm. If the maximum motor speed is known, the filter frequency should be selected to remove frequencies above the motor's maximum speed times the number of commutations per rotation.

Table 2: Input Filter Frequencies

Byte Value	Frequency
0	No Filter
1	250 Hz
2	500 Hz
3	750 Hz
4	1000 Hz
5	1500 Hz
6	2K Hz
7	3 KHz
8	5 KHz
9	7.5 KHz
10	10 KHz
11	15 KHz

Example:
For example, if a 12-bar motor has a maximum frequency of 3500 RPM, we can select our frequency target by:

$$Frequency = \frac{RPM}{60} * Bars$$

$$Frequency = \frac{3500}{60} * 12 = 700 Hz$$

Looking at Table 2, we see that 750 Hz is the next filter frequency higher than our calculated 700 Hz, so we should set our Input Filt parameter to a '3'.

HARDWARE DESCRIPTION

The CMSS56 module is a high-powered digital signal processor (DSP) module. It has an electrically isolated analog I/O section with two channels of 16-bit accurate A/D converters, programmable gain controls, and optional D/A converters.

The module has a digital I/O section that is controlled/monitored from the PLC interface.

The module's main power input expects a 24V DC input, fused at 1.6 Amperes. The digital outputs of the module are separately powered with 5 to 24V DC. The Digital output section must be powered whether in use or not.

The module is capable of accepting daughter-boards that can intercept all analog I/O, allowing for easy expansion capability, such as the addition of accelerometer or other signal conditioning.

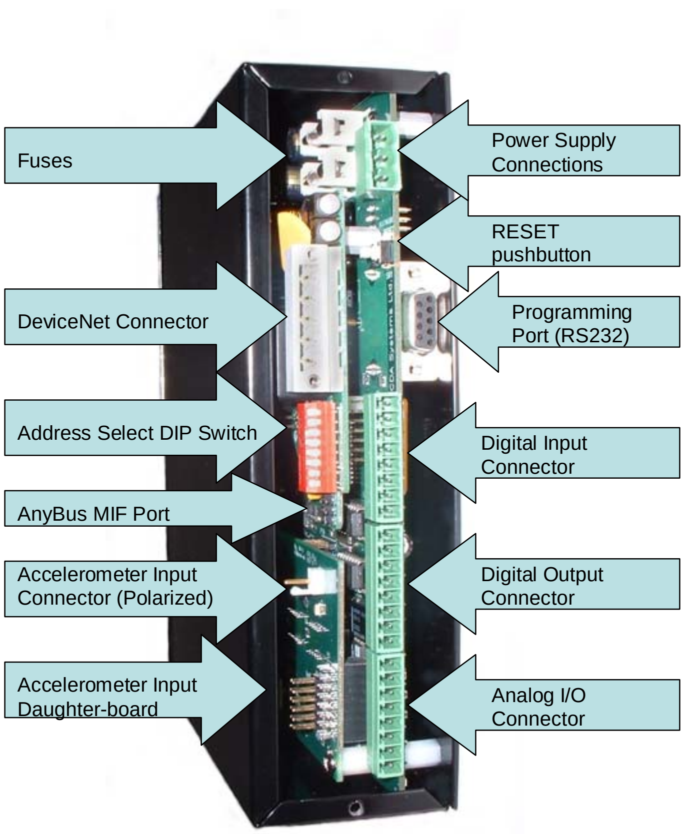


Figure 4: Components Behind Front Panel

Configuring DeviceNet Address/Baud Rate

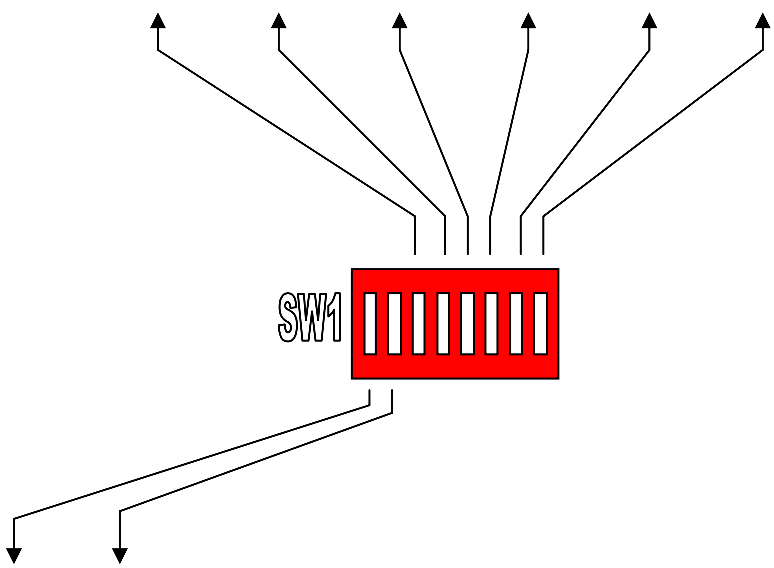
The DeviceNet address and baud rate are selected via a DIP switch on the module’s printed circuit board. In order to access the DIP switch, the front panel must be removed from the chassis by removing the two front panel screws. The DIP switch is directly behind the front panel. (See Figure 4) The front panel is connected to the accelerometer daughter-board via a polarized connector which can be disconnected and re-connected if required.

The ‘Closed’ position for the dip switch is the position closer to the printed circuit board that the switch is mounted to.

NOTE: The module must be powered down and up again before new settings will take effect.

Figure 5: DeviceNet Baud/Address Select Dip Switch Settings

BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	MAC ID
Closed	Closed	Closed	Closed	Closed	Closed	0
Closed	Closed	Closed	Closed	Closed	Open	1
Closed	Closed	Closed	Closed	Open	Closed	2
…….	…….	…….	…….	…….	…….	…….
Open	Open	Open	Open	Open	Open	63



Bit 8	Bit 7	Baud Rate
Closed	Closed	125 kbps
Closed	Open	250 kbps
Open	Closed	500 kbps
Open	Open	Reserved

APPLICATION DATA

The algorithm several results to the DeviceNet interface. The data returned includes the motor speed, and current measurements. Also included are RMS measurements for the secondary input channel.

Table 3: DeviceNet Input Data (To PLC)

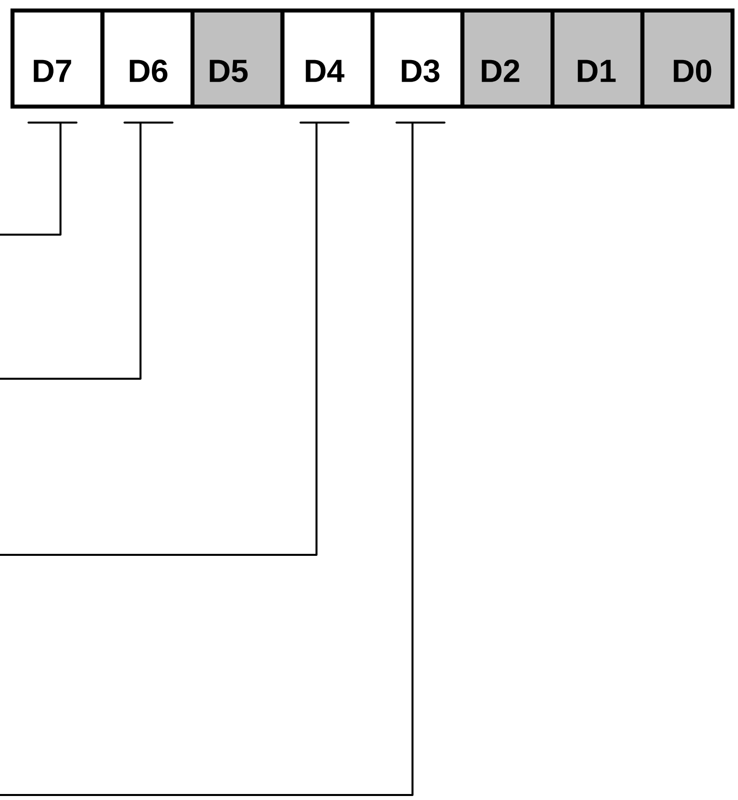
Byte Addr	Data	Description
0	Module_Stat	Byte indicating module status. See page 12.
1	Speed_Stat	Byte indicating speed measurement status. Indicates errors and speed stability. See page 13.
2-3	Speed_inst	Motor speed for last rotation. (RPM). Note LSB first in byte map. Positive integer.
4-5	Speed_Ave	Motor speed averaged over previous 125mS. (RPM) Positive integer.
6-7	Current_RMS	**RMS** current into motor. Value displayed is scaled as discussion in section Analog Channel Gain & S, on page 4. Positive integer. Note that data is LSB first.
8-9	Current_AVE	**Average** current into motor. Value displayed is scaled as per discussion in section Analog Channel Gain & S, on page 4. Positive integer. Note that data is LSB first.
10-11	Current_PK2PK	**Peak to peak** motor current due to noise and commutation. Value displayed is scaled as per discussion in section Analog Channel Gain & S, on page 4. Positive integer. Note that data is LSB first.
12-13	Ch2_RMS.	RMS measure from secondary channel for the previous rotation of the motor. Value determined by input variable **FullScale Input2 Value**. (See Table 4 for description). Note that output is LSB first. Positive integer.
14-15	Ch2_RMS_Ave	RMS average measure of secondary channel the previous N 100 ms measurements, where N is input parameter Num_RMS_Avg, byte 10 of module input parameters. Output scaling same as for **Ch2_RMS_Inst**, above. Note that output is LSB first. Positive integer.
16	Error_Count	Count of commutation errors detected since the motor started turning. 'Error' is defined as having one or more commutation times varying by more than 30 percent from the average commutation time over the previous N commutations, where N is the number of commutator segments in the motor. This number is expected to be incremented during fast acceleration or deceleration. Once the motor reaches a stable speed, this number should remain constant. Note that the value is reset after motor speed dropped to 0. LSB first. Positive integer.
17	WD_Count	Watchdog Count. 8-bit counter, incremented each time output measurement is updated. Positive integer.
18	Gain1	Input gain for A/D channel 1. Actual gain value set by bits specified in input variable InputGain1 (see Table 1, pg 4)
19	Gain2	Input gain for A/D channel 2. Actual gain value set by bits specified in input variable InputGain2 (see Table 1, pg.4)
20	InByte	Bits B0-3 correspond with digital inputs PWI1-4. Bits B4-B6 correspond with inputs DI1-DI3

Table 4: DeviceNet Output Data (From PLC)

Byte Addr	Data	Description
0	Mode	Mode of operation. Reserved for specialty applications. Write 0 for normal operation.
1	Bars	Number of bars in DC commutated motor
2	Input_filt	Selects from a series of low-pass filters for noise rejection.
3	Hysteresis %	Variable selects hysteresis for motor speed sensing algorithm. This variable specifies the percentage of the peak to peak ripple to use as a threshold for detection algorithm.
4-5	FullScale Current	Full scale current value in amperes on analog input. Dependant on variable InputGain. See discussion in section Analog Channel Gain & S, on page 4.
6-7	Noise_Current	Value controls algorithm for speed detection. If non-zero, value specifies the amplitude of noise spikes that are to be ignored for proper hysteresis action of the algorithm.
8-9	FullScale Input2 Value	Value represented by a full-scale signal on channel 2 input. Dependant on variable InputGain2. See discussion in section Analog Channel Gain & S, on page 4.
10	Num_RMS_Avg	Number of samples to use in average of RMS measure. (# 100ms measurements to average)
11	InputGain1	Bit pattern specifying input gain for motor current channel. For a 50 mV shunt, gain should be set to 20. Variable Ifs (byte 2 of this table) should be set to the full scale current rating of the shunt. See Table 1 on page 4 of this document for gain scaling. Undefined bits are reserved and should be set to 0.
12	InputGain2	Bit pattern specifying input gain for secondary input channel. See Table 1 on page 4 of this document for gain scaling. Undefined bits are reserved and should be set to 0
13	OutByte	For future compatibility, write 0 to this location.

Module Status Byte

The Module Status byte provides information about the operation of the module. The bits are depicted below:



Module Running – Set to 1 after successful initialization.

Module Calibrating – Set to 1 when calibrating DC offset after gain change. 0 otherwise

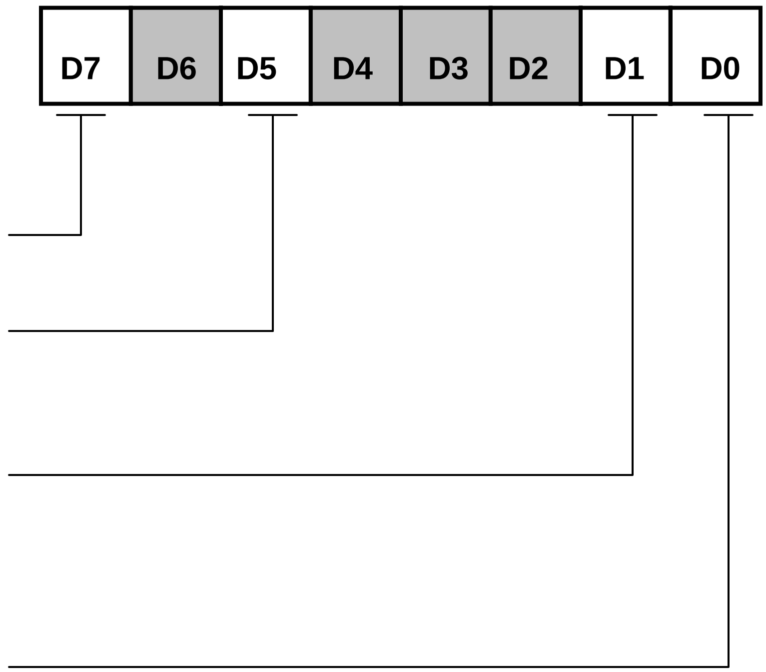
Current Input Clipping – Set to 1 when input found clipping at A/D converter. Remains high for 250 milliseconds after clipped sample detected.

Secondary Input Clipping – Set to 1 when input found clipping at A/D converter. Remains high for 250 milliseconds after clipped

NOTE – Other bits currently undefined

Speed Status Byte

The Speed Status byte provides information about the operation motor speed tracking algorithm. The bits are depicted below:



Software Fault – Should always be 0. If found to be a 1, report to factory.

Current Polarity – 0 if motor current polarity correct, 1 if reversed

Confidence – Set to 1 when no commutation fault for 125 milliseconds while motor running.

Speed Stable – Set to 1 if motor speed remained constant (less than 1% shift) over sample period used in average calculation. See DeviceNet parameter Num_RMS_Avg

NOTE – Other bits currently undefined

APPENDIX I : EDS File for CMSS1

NOTE: The latest EDS file can be found on the CDA Systems web site at www.cdasys.com.