IRMCx201
Application Developer's Guide

October 1, 2003
Version 4.0A

International
IOR Rectifier

Table of Contents

List of Figures

List of Tables

1 Introduction

This document is provided as a supplement to the datasheets for the IRMCx201 series products. It provides detailed information about the internal design and external interfaces of the products and describes how to configure the operation to conform to the requirements of your custom application. You should read this document if you are developing an application using the IRMCx201 products.

The document is divided into three main sections. In the Concepts section, system design concepts are presented and theory of operation is described in detail. This section will give you the background you need to begin your application development. The Techniques section provides practical "how-to" information, tips and examples to help you during your development process. The Reference section provides a complete definition of the host register map with a short description of each register and field. The registers are listed in sequential order for easy reference.

1.1 Control Structure Configuration

This section introduces some of the key configurable parameters used in the IRMCx201 system. Table 1 summarizes the system's configuration parameters and tunable gains. Determining proper settings for the parameters and gain values is the main focus of this document, with the necessary theory explained in the Concepts section and specific examples provided in the Techniques section.

Note: Changing some parameters may require the drive system to be in stand-by mode as opposed to PWM active mode.

Velocity Control Enable/Disable

With this switch, control mode can be configured as either Torque control or Velocity control. When disabling velocity control, reference input becomes torque command. When the velocity control is enabled, the reference input becomes the velocity command.

Slip Gain Enable/Disable (PMAC or induction machine)

The closed loop current control structure can be configured either for induction machine or permanent magnet AC machine. When slip gain is enabled, the feedforward slip gain path is added to a rotation angle to support indirect field orientation of induction machine. Torque producing current reference (Iq command) serves as the input to the slip gain.

Reference Select

The reference, whether it is for velocity reference or torque reference, can be connected to either ±10V analog input via a 12-bit A/D converter, or the host register interface. When analog input is selected, ±10V is converted bipolar digital value, which can be scaled by REF_scale in the associated multiply-divide block.

Current Feedback Interface Select (IR2175 or generic analog input interface)

The IRMCS201 has a built-in interface circuit for IR2175 motor current sensing high voltage ICs. With IR2175 with a shunt resistor, a 10-bit resolution of current feedback data can be obtained within 8.5 microseconds of conversion time. IR2175 IC is a simple yet high performance and low cost solution for servo drive application up to 3.7kW output power level. For a higher horsepower application, IR2175 becomes impractical due to power dissipation associated with a shunt resistor in series with a motor phase output. The IRMCS201 also supports an analog input type of interface such as a Hall-effect current sensor. This logic supports a low cost 12-bit A/D converter interface (ADS7818 family devices) with 2 channel simultaneous sample/hold and multiplexer. With this A/D interface, two phase current feedback data are sampled simultaneously and each converted to 12-bit data.

CEMF Feedforward Gain Enable/Disable

The output of q-axis current PI controller has an optional summing junction that adds a feedforward gain block. This gain block input is fed by the host interface register and can be connected to a velocity feedback to achieve CEMF feedforward control. This switch is also useful when the user needs V/Hz open loop control to disable current control PI regulators.

VD Enable

The output of d-axis current PI controller can be disconnected using this switch. When disabled, the d-axis voltage command is connected to the host register interface and the user can directly update/modify the value. This is particularly useful when the user needs an open loop V/Hz control

Parameter	Description
ID scale	d-axis current feedback scaler
IQ scale	q-axis current feedback scaler
V offset	Phase V current feedback offset adjustment
W offset	Phase W current feedback offset adjustment
Kp for IqId	Current regulator Proportional gain
Ki for IqId	Current regulator Integral gain
Vqlim	q-axis current PI amplifier output limit (for both positive/negative limit)
Vdlim	d-axis current PI amplifier output limit (for both positive/negative limit)
PWM Cval	PWM carrier frequency period
Deadtime	Deadtime
PWM mode	Asymmetrical or Symmetrical PWM
2/3 phase modulation select	2 phase or 3 phase modulation
FOC enable	Torque control enable
PWM enable	PWM enable
Slip gain	Slip gain for induction machine
Encoder Angle Scale Factor	Encoder angle scaler
Max Encoder count	Encoder line count
Z pulse initialization value	Z-pulse initialization count value
Speed scaler	Speed scaler
Encoder Type	Encoder type (Hall A/B/C multiplexed or independent)
Z pulse polarity	Logic sense polarity of z-pulse
Kp SPD	Velocity control PI amplifier Proportional gain
Ki SPD	Velocity control PI amplifier Integral gain
IqLim+	Speed regulator output positive limit
IqLim-	Speed regulator output negative limit

Table 1. Parameter Configuration

1.2 Constraints

The following are constraints for use of IRMCO201 with a custom hardware system.

Motor Current Sensing

As a default current sensing device, the system directly interfaces with two IR2175 high voltage ICs to facilitate current feedback from motor phase. If a different current sensing device is required, a 0-5V two-channel A/D converter can be used in lieu of IR2175. IRMCO201 interfaces to the ADS7818 (BurrBrown) A/D converter and has steering bit control for simultaneous sampling and two-channel multiplexing. IR2175 requires a series shunt resistor on the motor phase and is normally good at up to 1.5kW application with 320V DC bus operation.

XC2S300E-6 FPGA

The IRMCO201 only works with the Xilinx XC2S300E-6 FPGA. The object code is synthesized based on the target device, and is not compatible with other devices. The crystal oscillator is currently specified at 33.333 MHz.

DC Bus Voltage Feedback Interface

The IRMCO201 interfaces to the ADS7818 (BurrBrown) serial A/D converter (12 bit) for DC bus feedback.

Analog Reference Input Interface

A ±10V analog reference input can be interfaced with ADS7818 (BurrBrown) 12-bit serial A/D converter with analog multiplexer. The IRMCO201 provides the interface logic to these analog components.

1.3 Design Guidelines

Data range

Data range of Iq (Torque producing current in synchronously rotating frame) and Id (Field component current in synchronously rotating frame).

1 PN (Per Normal, 1PN equivalent to rated condition) = ±4095

Overload range = 3PN

Maximum data range = 4 PN = ±16383 including overload range plus additional room for overshoot

General nomenclature

1. Variables either ending or starting with "n" are negative logic indicating low true signals.
2. Unless specified with the associated data range, all variables are Boolean variables.

Motors

Motor can be either a permanent magnet motor or an induction motor. 2, 4, 6, or 8 pole motors are supported.

Hall A/B/C and Z-pulse Initialization

When permanent magnet motor is used, the Hall A/B/C data and z-pulse initialization data must be entered to the associated registers in the host register interface. These data are initial position data to make alignment to the initial rotor magnet position.

2 Concepts

This section provides background information and theory of operation for the major IRMCx201 design elements, including control structures, encoder and IR2175 interfaces and the Space Vector PWM module. The concepts are generally common to both permanent magnet motors and induction machines.

Figure 1 shows the detailed control structure of the IRMCx201. Functions are divided into five logic control units:

- Closed loop current control
- Closed loop velocity control
- Sequencing
- Fault handling and DC bus dynamic braking control
- Communication and external interfaces

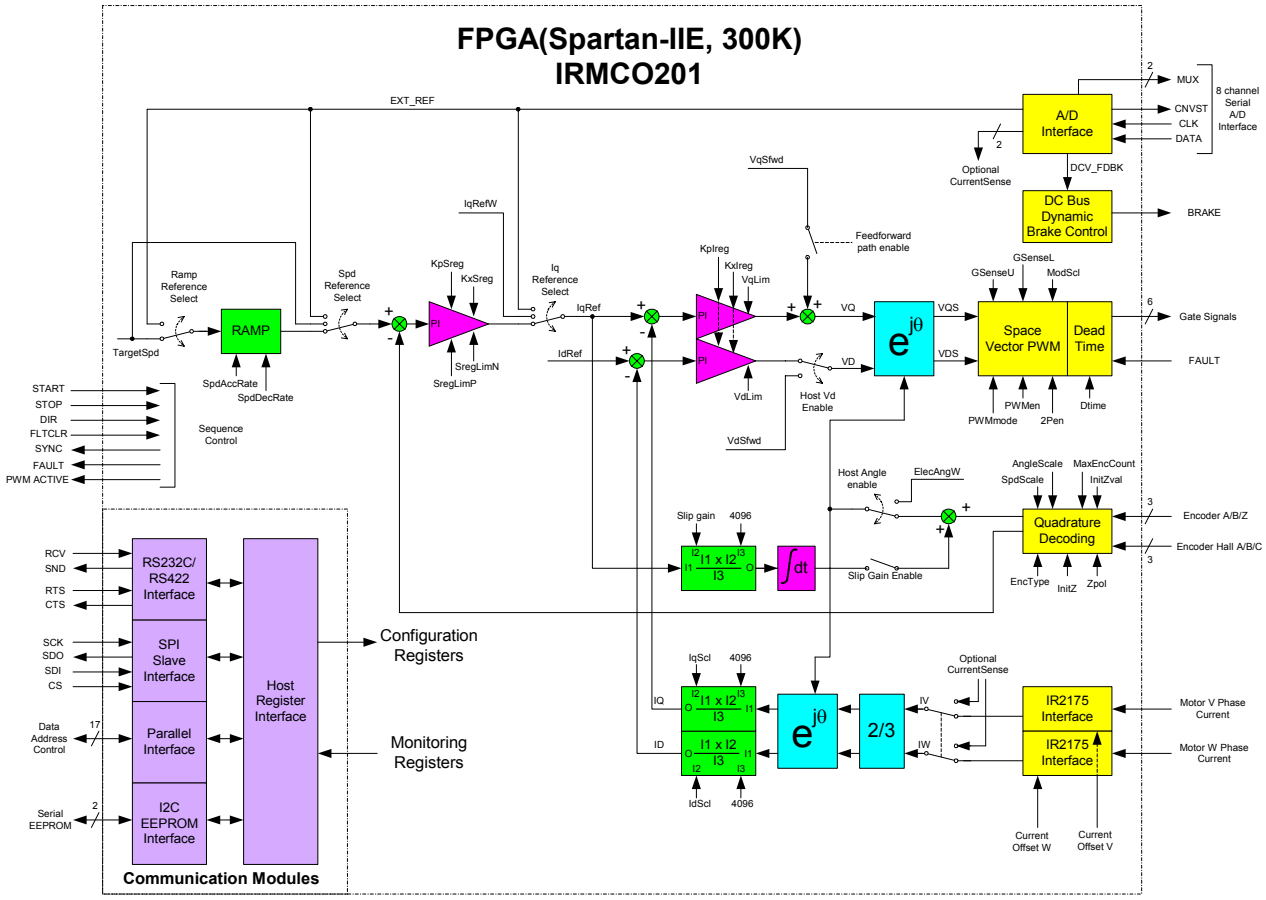


Figure 1. Detailed Control Structure

2.1 Closed Loop Current Control

The closed loop current control structure is based on the synchronously rotating frame field orientation. It employs two Proportional plus Integral (PI) amplifiers to independently regulate torque-producing current, namely Iq, and field flux component current, Id. These currents are oriented in synchronously rotating frame, and contains no AC component of excitation frequency.

The current feedback is obtained through the IR2175 high voltage current sensing IC, and directly transferred to the IRMCx20x FPGA. Then it is transformed to Id and Iq via a Vector rotator to decouple AC excitation component. Iq is compared against the command torque current while Id is compared to zero current reference. Two PI controllers produce the required voltage commands, namely Vd and Vq, which are transformed back to stationary frame via Vector rotator and the resulting voltages, Vds and Vqs, contain the AC excitation frequency component. These voltages are passed to the Space Vector PWM modulator and converted into six IGBT gate signals.

The user can configure the closed loop current control for induction machine based field orientation control in addition to permanent magnet motor. The configuration switch, "Slip Gain Enable", is provided to support induction machine vector control.

The user can also enable a feedforward control on the Iq regulator, which enhances dynamics of the PI controller by decoupling the CEMF (Counter Electro Mechanical Force) component.

The default current sensing device is IR2175. However, the user can choose an optional A/D interface (ADS7818) with other types of current sensing devices such as Hall effect current sensors. A 0 to 5V analog interface is provided for two channels with simultaneous sampling capability. A/D conversion takes five microseconds for each channel.

The entire closed loop current control is updated and computed once or twice every PWM carrier frequency period depending on the PWM mode. If symmetrical PWM is selected, then the update rate becomes once every PWM carrier frequency period. If asymmetrical PWM is selected, then the update becomes twice every PWM carrier frequency period.

2.2 IR2175 Current Sensing

Two channels of current feedback interface logic are provided in the IRMCx201 system. Each module measures the incoming varying duty period of the 130 kHz carrier frequency signal at the IR2175output. Measurement is performed for both carrier frequency period and on duty period at the same time using fast counters. Counting frequency is 133 MHz with a 33.3 MHz system clock.

The IR2175 are the unique high voltage ICs capable of measuring the motor phase current through an associated shunt resistor, which can generate ±260mV voltage range. The output of the IR2175 is an open drain with a 130 kHz fixed carrier frequency where the duty variance is linearly proportional to ±260 mV input voltage. The counting frequency is 133.3 MHz when the system clock crystal frequency is 33.3 MHz, which yields 10-bit resolution of the current measurement data from the IR2175.

The offset measurement is automatically added after the 10-bit current measurement has been calculated. The offset value must be calculated and supplied by external hardware or software.

The period measurement of both the carrier frequency period and the duty period of the IR2175 output signal are performed. For carrier frequency period measurement, there is a 16-stage averaging filter to smooth out the 130 kHz carrier period of the IR2175. The multiply/divide computation follows after completing both period measurements. Divide computation between the carrier frequency period and the duty period alleviates temperature drift of the incoming data off the IR2175, since variation of these periods uniformly moves in same direction as temperature changes.

The measured and adjusted data is coherently updated to the host digital system such as a microcontroller, DSP, or FPGA. A block diagram of the current measurement block is shown in Figure 2.

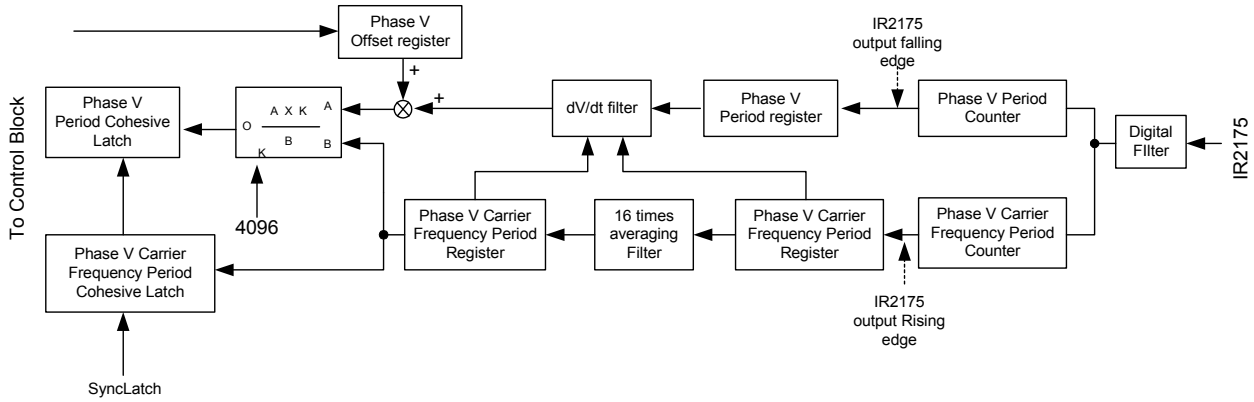


Figure 2. Current Feedback Measurement Block

The current feedback module requires a faster clock to count the duty period of the incoming pulse width modulated signal from the IR2175. This clock rate is designed to work with a frequency between 120 MHz and 133.3 MHz. Figure 3 depicts a simple time chart of counting.

The duty counter, shown as "TA" in Figure 3, captures/latches the value at the falling edge of IR2175 and reset. Then the counter waits for the next rising edge to start counting up. The carrier frequency counter ("TB") captures/latches the value at the rising edge of IR2175 and is immediately followed by re-counting at each IFB event.

At each IFB event, a multiply/divide operation is performed to cancel the temperature drift error of measurement. The following is the basic multiply/divide operation:

$$\frac{TA(n) \times 4096}{Filtered\_TB(n)}$$

Calculation starts immediately after the rising edge of the IR2175 signal as shown in Figure 3. This look-ahead calculation is required to minimize the latency of data availability of the calculation result.

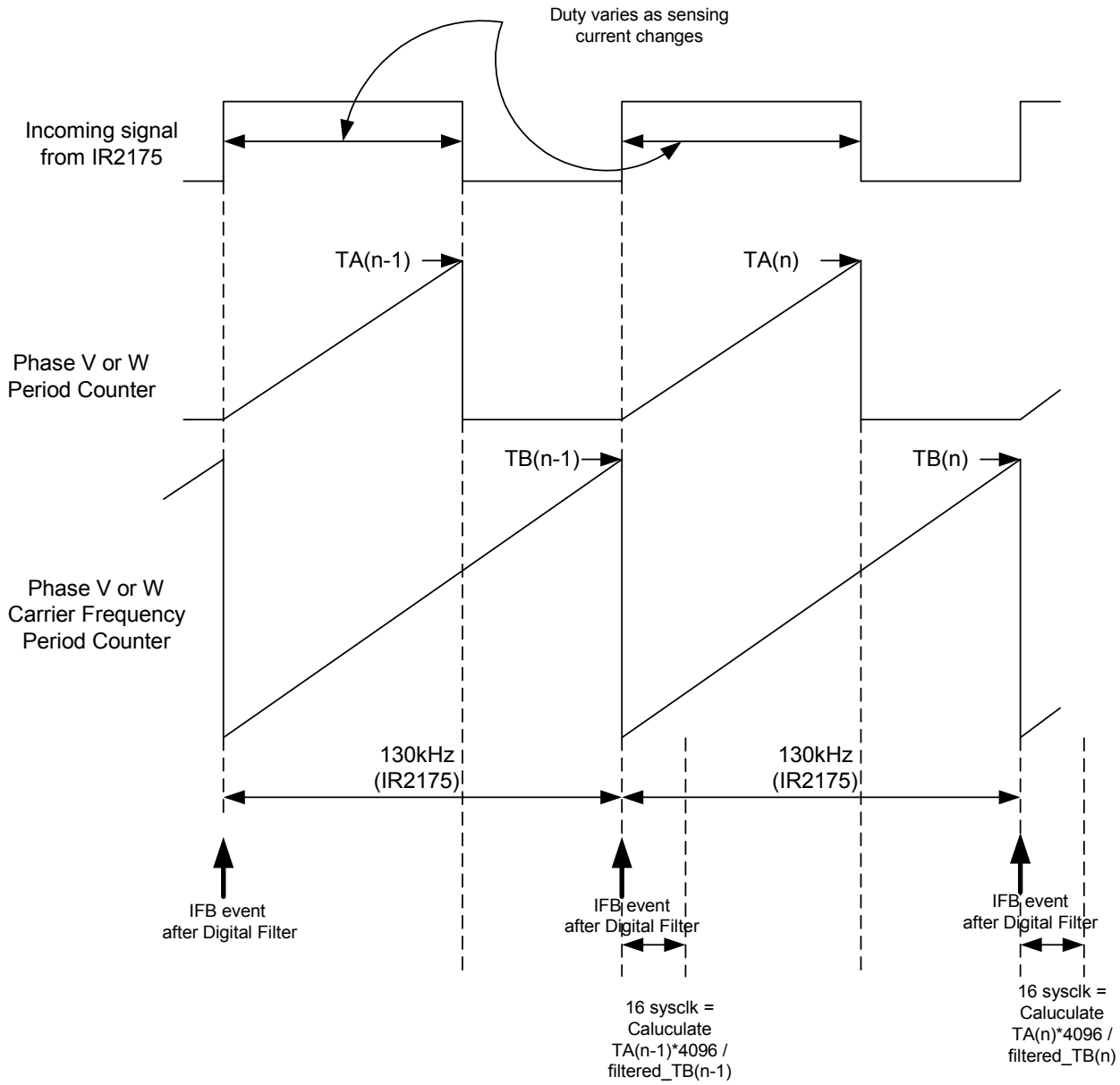


Figure 3. Current Feedback Calculation Timing

2.3 IR2175 Current Feedback Scaling

Maximum input voltage of IR2175	± 260 mV
Shunt resistor value for current sensing	10 mΩ (default)
Maximum Hardware Current (MHC)	± 26 Apeak
Guideline:	
Usable Control Range of current	± 20.8 Apeak to 23.4 Apeak
Overshoot current range	± 2.6 Apeak to ±5.2 Apeak

2.3.1 Selecting Current Sensing Shunt Resistor

Prior to adjusting the current feedback scaling gain, proper shunt resistors need to be selected. The basic guideline of choosing the right value is that the maximum hardware current(MHC) should be greater than the motor's peak overload current plus overshoot current amount. Overshoot current amount should be in the range of 10 to 20% of the maximum hardware current. For the IRMCS201 hardware system, 10mΩ is the minimum value to be selected due to IGBT current limitation. And as for default factory setting, 10mΩ shunt resistors are installed. If larger shunt resistor values are required, please contact the manufacturer, CADDOCK, with the CD2520FC series surface mount power resistor for the different value.

Trade-offs need to be considered, particularly when the MHC is not close to the motor overload rating; this mismatch may cause coarse resolution of current feedback, say, "Amps per Bit" or "Torque per Bit".

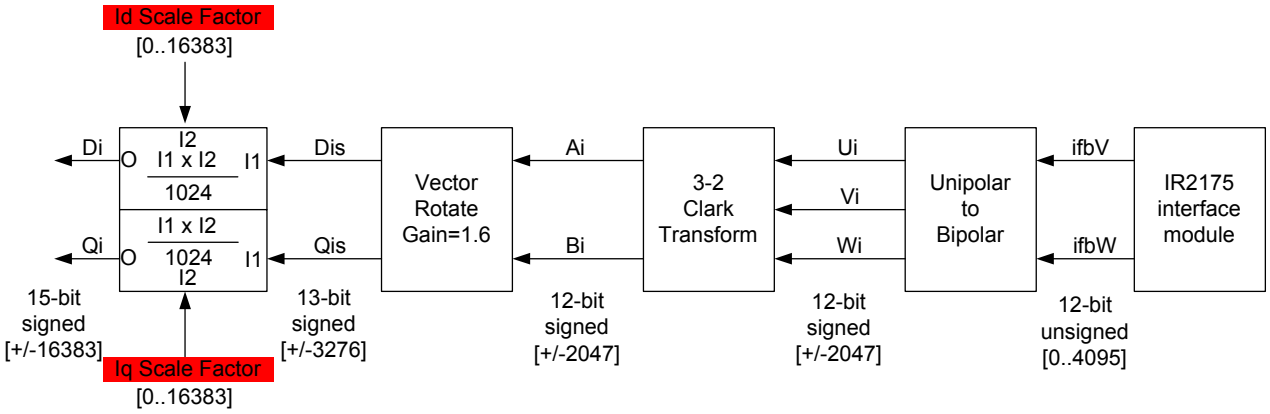


Figure 4. Current Feedback Scaling / Mapping

2.3.2 Adjusting Current Feedback Scaling for Permanent Magnet Motor

Current feedback scaling needs to be adjusted by two parameters, namely **Id scale factor** and **Iq scale factor**, through the host register interface. For a permanent magnet motor application, these two scaling factors are identical. The IRMCO201 design is based on the mapping of rated current (continuous current) to ±4095 digital values regardless of the motor current rating. The current control therefore regulates ±4095 as 100% rated torque for torque producing current, "Qi". If the motor has 300% overload capability, then the control regulates a maximum of ±12,287.

The user must know the detailed current feedback scaling path. Figure 4 illustrates the current feedback scaling/mapping implemented in IRMCO201. At the IR2175 interface module, 0 to 4095 is mapped to the maximum hardware current. Then it becomes ±3276 for the maximum hardware current mapping right after the vector rotator due to gain of 1.6. This number then needs to be scaled by user input parameters **Id scale factor** and **Iq scale factor**, which perform scale factor times input divided by 1024. For a permanent magnet motor application with Id = 0, the following equation can be used for determining these scaling gains.

$$Id\,/\,iq\ scale\ factor = \frac{MHC \times 4095 \times 1024}{RC \times 3909}$$

Where: MHC is Maximum hardware current (=260mV/Shunt resistor),
RC = Motor Rated Current in rms (continuous current)

Example:

Motor rated current (RC)	±2.7 Arms / ±3.8 Apeak
Motor overload current	±7.1 Arms / ±10 Apeak

Shunt resistor selection = 20mΩ, since a 10mΩ resistor will have too large unused current range relative to this motor current rating.

Maximum Hardware Current (MHC)	±13 Apeak
Usable Control Range of current	±10 Apeak
Overshoot current range	±3 A

Id/Iq scale factor = 13x 4095 x 1024 / (2.7 x 2047 x 1.414 * 1.647 * .82) = 5165

2.3.3 Adjusting Current Feedback Scaling for Induction Motor

Id scale factor and **Iq scale factor** need to be adjusted for an induction motor application. These two scaling factors are derived by different equations and therefore are usually different values. The IRMCO201 design is based on the mapping of rated current (continuous current) to be ±4095 digital values regardless of the motor current rating. The current control therefore regulates ±4095 as 100% rated torque for torque producing current, "Qi". If the motor has 300% overload capability, then the control regulates a maximum of ±12,287. The field component current, "Di" is also regulated 100% field flux with 4095 digital value regardless of the motor physical current or amount of magnetizing current.

In order to find Id scale factor and Iq scale factor, both torque producing current and magnetizing current need to be determined at a rated condition.

Motor nameplate data typically shows the rated current and either no load current or magnetizing current. Torque producing current can be derived from the following equation (all units are in rms):

$$Torque\ producing\ Current\,[TPC] = \sqrt{(rated\ current)^2 - (no\ load\ current[NLC])^2}$$

Then, iq scale factor can be derived from the equation:

$$Iq\ scale\ factor = \frac{MHC \times 4095 \times 1024}{TPC \times 2047 \times \sqrt{2} \times 1.647 \times .82}$$

Similarly, id scale factor can also be derived from the equation:

$$Id\ scale\ factor = \frac{MHC \times 4095 \times 1024}{NLC \times 2047 \times \sqrt{2} \times 1.647 \times .82}$$

where .82 is maximum modulation index of IR2175 and 1.647 is gain scale of the vector rotator in the current feedback path.

Example:

Motor rated current (RC)	±6.0 Arms / ±8.46 Apeak
Motor overload current:	±9.0 Arms / ±12.7 Apeak
No load current (NLC):	±3.8 Arms / ±5.37 Apeak
Maximum Hardware Current:	±13 Apeak

TPC = sqrt($6^2 - 3.8^2$) = sqrt(21.56) = 4.64 Arms
Iq scale factor = 13 x 4095 x 1024 / (4.64 x 2047 x 1.414 x 1.647 x .82) = 3004
Id scale factor = 13 x 4095 x 1024 / (3.8 x 2047 x 1.414 x 1.647 x .82) = 3669

2.3.4 Adjusting Slip Gain for Induction Motor

The slip gain needs to be adjusted for induction machine application. Figure 5 shows the slip gain block diagram. In order to determine the slip gain, the user must have the rated RPM information on the motor nameplate data, and the IRMCO201 closed loop current control update rate information.

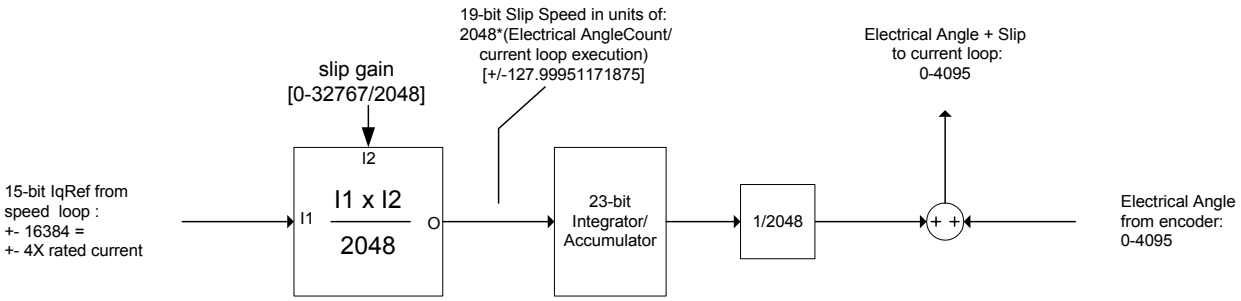


Figure 5. Slip Gain Block Diagram

The following equation is used to derive the slip gain:

$$Rated\ Hz = \frac{\frac{120 \times rated\ frequency}{polenumber} - rated\ RPM}{\frac{120 \times rated\ frequency}{pole\ number}} \times rated\ \ frequency$$

$$Internal\ slip\ speed = 2048 \times \frac{Rated\ Hz \times 4096}{current\ loop\ update\ rate}$$

$$slip\ gain = \frac{2048 \times Internal\ slip\ speed}{4096}$$

Example:

Rated RPM:	1725 rpm
Rated frequency:	60 Hz
Pole number:	4
Current loop update rate:	10 kHz

Rated Hz = ((120 * 60 / 4) – 1725) *60 / (120 * 60 / 4) = 2.5Hz
Internal slip speed = 2048 * 2.5 * 4096 / 10000 = 2097
Slip gain = 2097 * 2048 / 4096 = 1048

2.4 Closed Loop Velocity Control

The closed loop velocity control structure can be disabled to form torque mode control instead of velocity mode control. It contains one PI controller and a reference ramp block.

The closed loop velocity control is updated and executed once every two PWM carrier frequency periods. Therefore, if the user changes the PWM carrier frequency, then the velocity control update rate is automatically changed accordingly. The user cannot modify the velocity loop update independently from the PWM carrier frequency.

Input reference can be selected between analog input through A/D interface (ADS7818) or host register update command via communication modules.

2.5 Torque Mode Operation

Torque mode operation provides the user with the option of controlling motor speed from an external source. In this mode, the user supplies the speed ramp and PI controller, and provides IQREF (quadrature reference current) input through the host register interface in order to control torque (motor current). The user's algorithm should calculate and provide a new IQREF value at the start of each PWM period, which is indicated by the INT output signal. The EncCntR host read register provides encoder position data to be used in the calculation.

The following host register settings configure the system for torque mode operation:

- Set SpdLpRate to 0 in the Velocity Control write register group to disable the speed loop.
- Set IqRefSel in the SysConfig write register to 1 to enable use of the IqRefW write register.

Once the system is configured for torque mode operation, a quadrature reference current value should be written to the IqRefW host write register at the start of each PWM period.

2.6 Encoder Interface

The encoder interface module is included in IRMCx201. The interface assumes the encoder signal to be an incremental and rectangular pulse train waveform with differential output. The IRMCx201 can accept a maximum 2MHz encoder frequency. Quadrature signals are used for angle generation in the current control loop but not for speed measurement.

2.6.1 Initialization & Angle Generation

The Sanyo Denki P30B06040DXS00M motor, for example, has a wire saving type 2000 line count pulse encoder. Hall_A, Hall_B, Hall_C information is available only at power up and it gives the approximate 60-degree electrical angle position. This motor has 4 pole pairs (8-pole motor) and therefore it has 15 degrees resolution (360/4/6 = 15) mechanically. If the middle value of that range is taken, the maximum error of the initial position is 30 degrees electrically.

The motor can start running with Hall_A, Hall_B, Hall_C information and within one revolution, a z_pulse occurs, which gives precise angle information.

The counter EMF profiles can be measured by rotating the shaft manually. Zero angle is defined at negative-going zero crossing of the U phase counter EMF.

When you look at the motor from the front-end shaft side and the shaft is rotating CCW, the output waveforms are shown in Figure 6 in relation to the internal angle counters.

In this figure, a 4-pole motor (2-pole pairs) is assumed, and therefore two complete CEMF cycles are required for one revolution of the mechanical turn. The line-to-neutral voltage and the line-to-line voltage are CEMF waveforms. The line-to-line voltage can be measured at motor three phase voltage terminals. The top trace, Mechanical angle, is the internal up/down counter of the encoder Quadrature A/B pulses. The counter value needs to be initialized at power-up by sensing the Hall A/B/C signal position and reading the corresponding counter data. The counter value is initialized again upon receiving the first z-pulse with the predefined counter data. This two-stage initialization is required since the first initialization only provides a coarse angle resolution (maximum ±30 degree electrical angle) due to Hall A/B/C signals.

Hall A/B/C data need to be entered through the Motor Configuration dialog box in the ServoDesigner tool. The data are mechanical angle based counter value as shown in Figure 6's top trace. Initialization needs to be done so that the counter value becomes zero at the negative-going zero crossing of the phase U. It is also important that the mechanical angle counter rolls over to zero when it reaches MAX_COUNT, which is equal to the encoder line count times four (i.e. for 2000 line count encoder = 8000). It is also important to provide the data in the middle of Hall A/B/C state changes. A total of six counter values need to be entered for all three-bit combinations of Hall A/B/C. The z-pulse data also needs to be provided in a similar fashion.

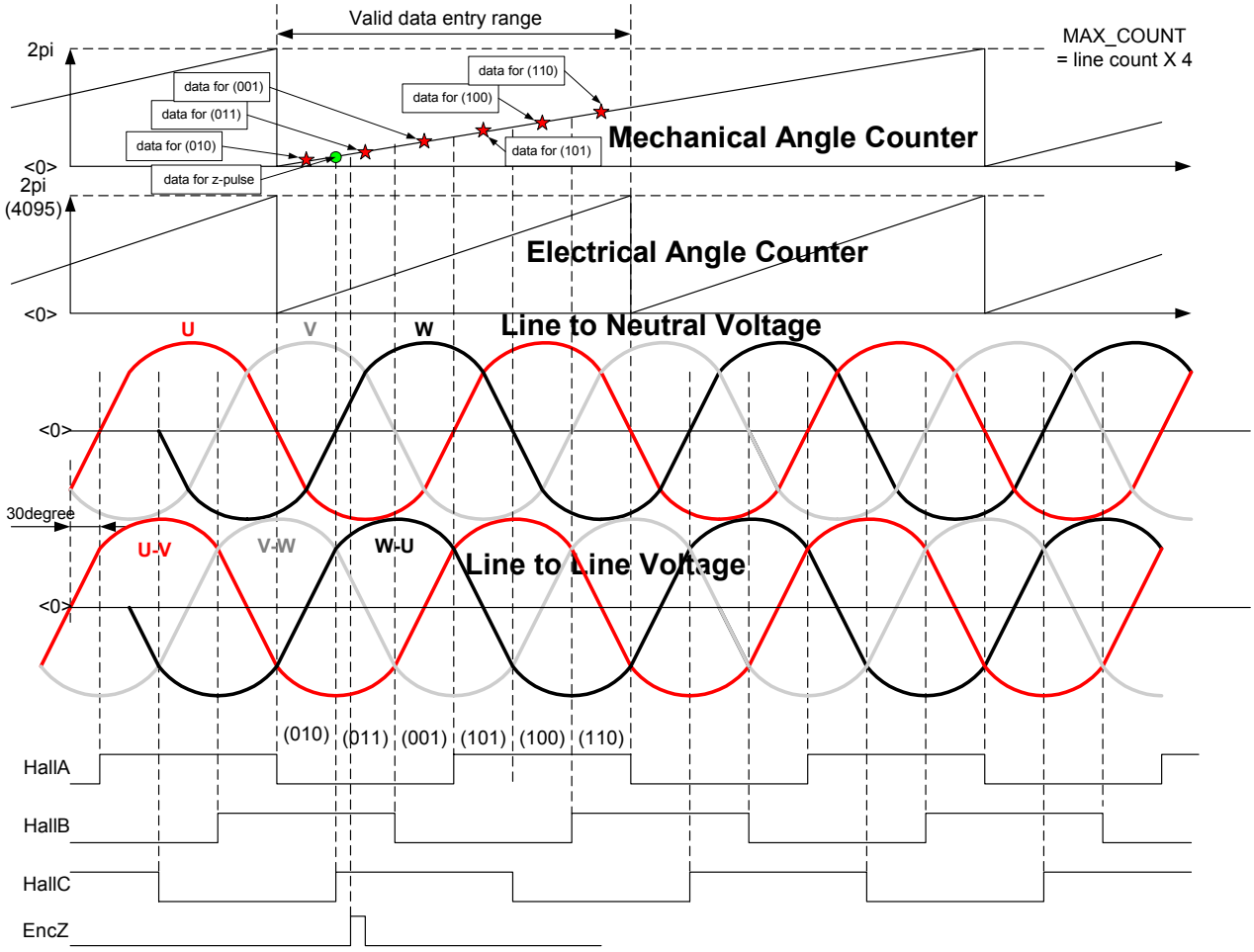


Figure 6. Counter-EMF Profile and Hall Signals

2.6.2 Initializing and Monitoring Encoder Counter Values

The encoder counter is initialized and monitored through the host register interface. Two counters are provided. The main mechanical angle counter is a 16-bit field that wraps to 0 when a configured maximum count (MAX_COUNT) is reached. This is the counter that must be initialized at power-up as described above. An auxiliary 32-bit counter that does not wrap is provided for optional use with robotic applications.

The maximum value for the 16-bit counter is configured using the MaxEncCnt field of the QuadratureDecode write register group. The maximum count is dependent on the encoder and should be set to represent a 360-degree physical angle, as described above. The 16-bit count value can be initialized by writing to the EncCntW field of the QuadratureDecode write register group. The current counter value can be read at any time from the EncCntR field of the QuadratureDecodeStatus read register group and represents the current encoder angle.

There is no maximum count setting for the 32-bit counter. The 32-bit count value can be initialized by writing to the EncCnt32bW field of the 32bitQuadDecode write register group. The current 32-bit counter value can be read at any time from the EncCnt32bR field of the 32bitQuadDecodeStatus read register group and can be used to determine the number of encoder revolutions, which in turn can be translated to the distance traveled by a robotic arm.

The two counters are independent. That is, initializing the 16-bit counter does not affect the value of the 32-bit counter and initializing the 32-bit counter does not affect the value of the 16-bit counter.

2.7 Space Vector PWM Module

The Space Vector PWM generation module accepts modulation index commands and generates the appropriate gate drive waveforms for each PWM cycle. This section describes the operation and configuration of the SVPWM module.

2.7.1 SVPWM Basic Theory and Transfer Characteristics

A three-phase 2-level inverter with dc link configuration can have eight possible switching states, which generates output voltage of the inverter. Each inverter switching state generates a voltage Space Vector (V1 to V6 active vectors, V7 and V8 zero voltage vectors) in the Space Vector plane (Figure 7). The magnitude of each active vector (V1 to V6) is 2/3 Vdc (dc bus voltage).

The Space Vector PWM (SVPWM) module inputs modulation index commands (U_Alpha and U_Beta) which are orthogonal signals (Alpha and Beta) as shown in Figure 7. The gain characteristic of the SVPWM module is given in Figure 8. The vertical axis of Figure 8 represents the normalized peak motor phase voltage (V/Vdc) and the horizontal axis represents the normalized modulation index (M).

Where : $M = Umag * Mod\_Scl * 10^{-4}$

$Umag = \sqrt{(U\_Alpha^2 + U\_Beta^2)}$ (-32768 <= U_Alpha, U_Beta <= 32767)

Mod_Scl : Input scaling factor (0 to 32767 range)

The inverter fundamental line-to-line Rms output voltage (Vline) can be approximated (linear range) by the following equation:

$Vline = Umag * Mod\_Scl * Vdc / \sqrt{6} / 2^{25}$ where dc bus voltage (Vdc) is in volts

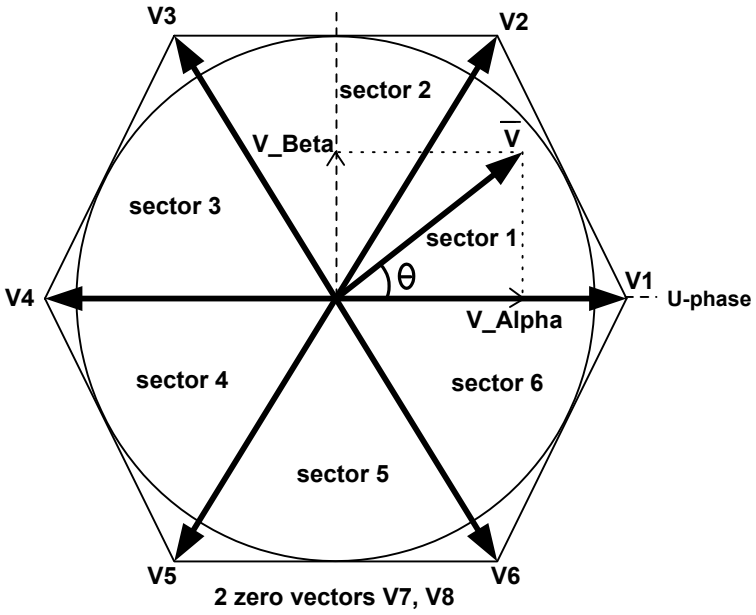


Figure 7. Space Vector Diagram

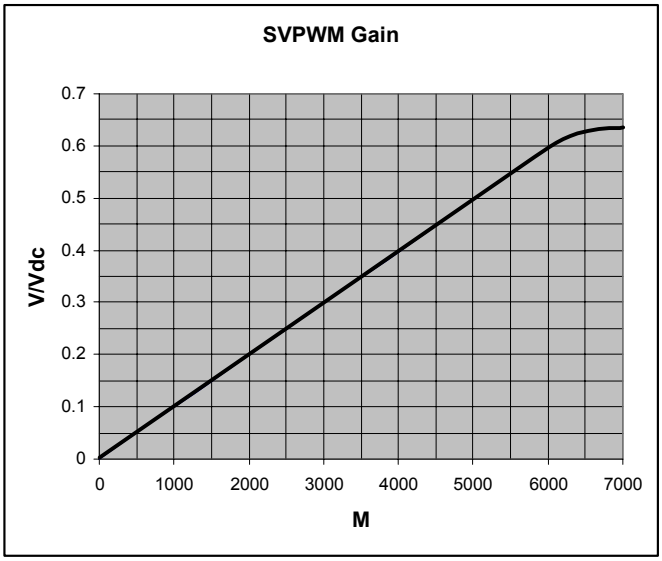


Figure 8. Transfer Characteristics

The maximum achievable modulation (Umag_L) in the linear operating range is given by:

$$Umag\_L = 2^{25} * \sqrt{3} / Mod\_Scl$$

Over modulation occurs when modulation Umag > Umag_L. This corresponds to the condition where the voltage vector in Figure 9 increases beyond the hexagon boundary. Under such circumstance, the Space Vector PWM algorithm will rescale the magnitude of the voltage vector to fit within the Hexagon limit. The magnitude of the voltage vector is restricted within the Hexagon; however, the phase angle (θ) is always preserved. The transfer gain (Figure 8) of the PWM modulator reduces and becomes non-linear in the over modulation region.

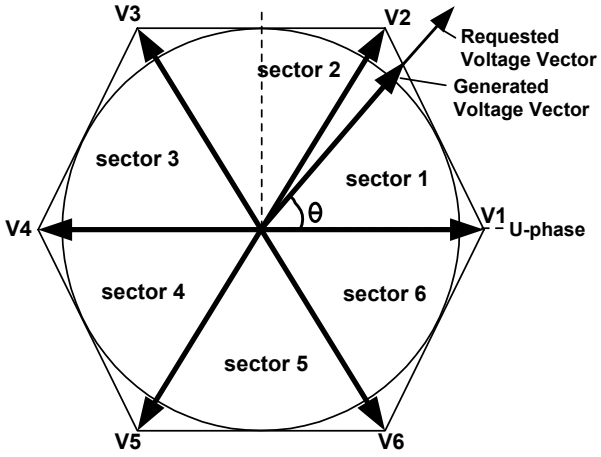


Figure 9. Voltage Vector Rescaling

2.7.2 PWM Operation

Referring to Figure 10, upon receiving the modulation index commands (U_Alpha and U_Beta) the sub-module SVPWM_Tm starts its calculations at the rising edge of the PwmLoad signal. The SVPWM_Tm module implements an algorithm that selects (based on sector determination) the active space vectors (V1 to V6) being used and calculates the appropriate time duration (w.r.t. one PWM cycle) for each active vector. The appropriated zero vectors are also being selected. The SVPWM_Tm module consumes 11 clock cycles typically and 35 clock cycles (worst case Tr) in over modulation cases. At the falling edge of nSYNC, a new set of Space Vector times and vectors are readily available for actual PWM generation (PhaseU, PhaseV, PhaseW) by sub module PwmGeneration. It is crucial to trigger PwmLoad at least 35 clock cycles prior to the falling edge of nSYNC signal; otherwise new modulation commands will not be implemented at the earliest PWM cycle.

Figure 10 (3-phase modulation) and Figure 11 (2-phase modulation) illustrates the PWM waveforms for a voltage vector locates in sector I of the Space Vector plane (Figure 7). The gating pattern outputs (PWMUH … PWMWL) include deadtime insertion (describe in later section).

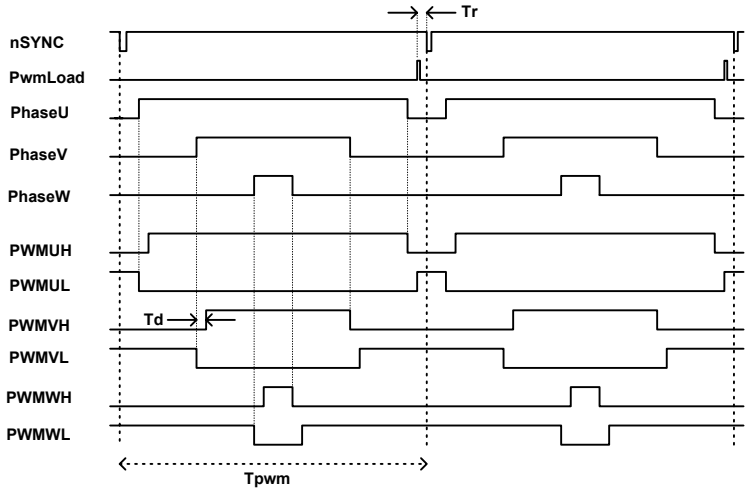


Figure 10. 3-phase Space Vector PWM

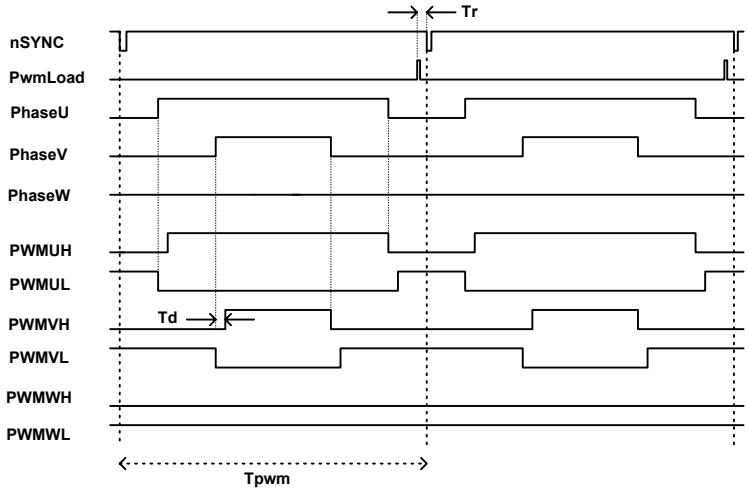


Figure 11. 2-phase (6-step PWM) Space Vector PWM

2.7.3 PWM Carrier Period

Input variable PwmCval controls the duration of a PWM cycle. It should be populated by the system clock frequency (Clk) and Pwm frequency (PwmFreq) selection. The variable should be calculated as:

$$PwmCval = Clk/(2 * PwmFreq) - 1$$

The input resolution of the Space Vector PWM modulator signals U_Alpha and U_Beta is 16-bit signed integer. However, the actual PWM resolution (PwmCval) is limited by the system clock frequency.

2.7.4 Deadtime Insertion Logic

Deadtime is inserted at the output of the PWM Generation Module. The resolution is 1 clock cycle, or 30 nsec at a 33.3 MHz clock and is the same as those of the voltage command registers and the PWM carrier frequency register.

The deadtime insertion logic chops off the high side commanded volt*seconds by the amount of deadtime and adds the same amount of volt*seconds to the low side signal. Thus, it eliminates the complete high side turn on pulse if the commanded volt*seconds is less than the programmed deadtime.

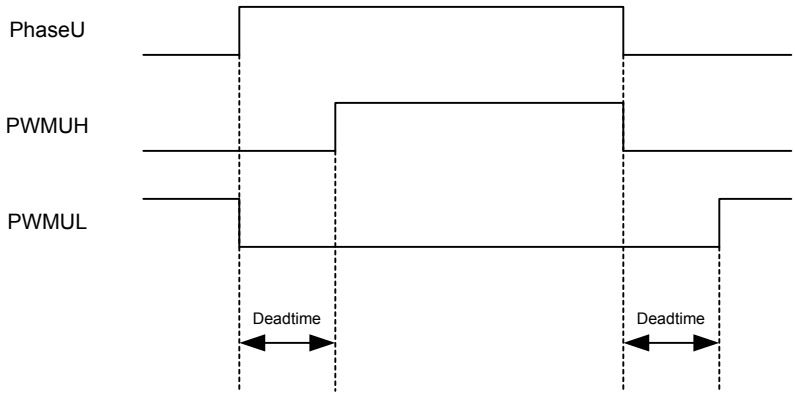


Figure 12. Deadtime Insertion

The deadtime insertion logic inserts the programmed deadtime between two high and low side of the gate signals within a phase. The deadtime register is also double buffered to allow "on the fly" deadtime change and control while PWM logic is inactive.

2.7.5 PWM Mode Select (PwmMode)

The nSYNC signal and the PWM internal Reload signal are completely simultaneous and synchronous events. The Reload signal resets the PWM counters inside the PwmGeneration module and it is an indicator reference of a new PWM cycle. The rate of signal generation of the SyncPulse and Reload signals is controlled via the configurable parameter PwmMode. Both signals can be generated at a 1:1 ratio or the SyncPulse can be generated once every four PWM ReLoad events so that the CPU loading and the PWM carrier frequency can be decoupled. *(*In other words, the PWM carrier frequency can be increased without increase of the CPU loading. This is beneficial for driving a low inductance AC machine such as a linear PM motor.)

Figure 13 shows the timing of nSYNC and Reload signals for the four different PwmMode settings. Asymmetrical mode operation (described in Section 2.7.6) can be obtained by setting PwmMode = 0.

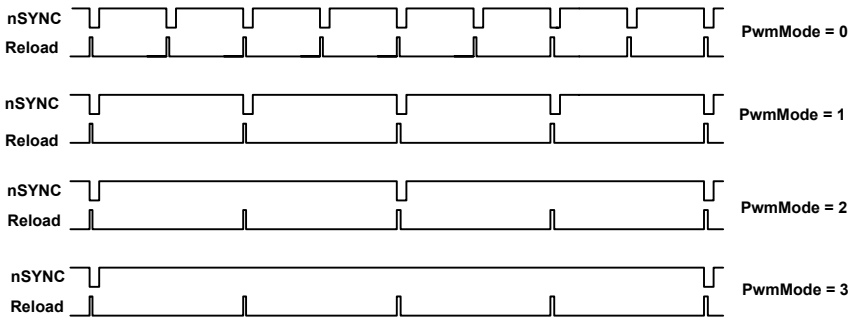


Figure 13. nSYNC and Reload Timing

2.7.6 Symmetrical and Asymmetrical Mode Operation

There are two modes (configured by PwmMode: 0 – Asymmetrical, 1 to 3 - Symmetrical) of operation available for PWM waveform generation, namely the Center Aligned Symmetrical PWM (Figure 10) and the Center Aligned Asymmetrical PWM (Figure 14). The volt-sec can be changed every half a PWM cycle (Tpwm) since PwmLoad occurs every half a PWM cycle (compare Figure 10 and Figure 14). With Symmetrical PWM mode, the inverter voltage can be changed at the rate of the inverter switching frequency. With Asymmetrical PWM mode operation (PwmMode = 0), the inverter voltage can be changed at 2 times the rate of the switching frequency. This will provide an increase in voltage control bandwidth, however, at the expense of increased current harmonics.

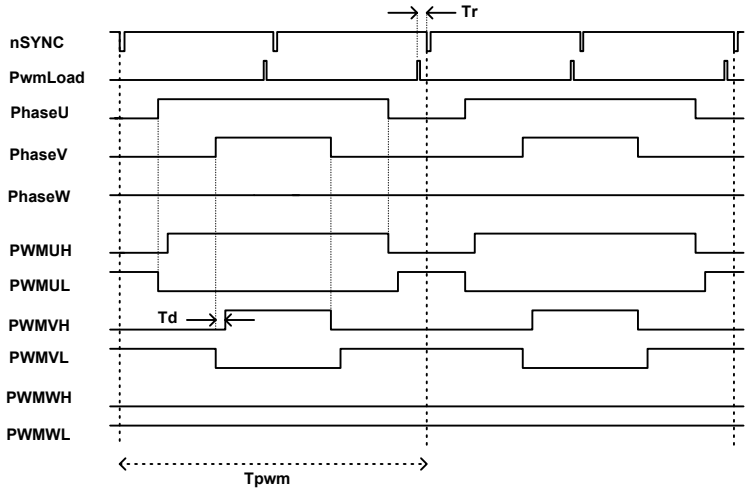


Figure 14. Asymmetrical PWM Mode

2.8 Communication and External Interfaces

This section describes the external interfaces supported by the IRMCx201. These include the host registers, which are accessed through an SPI or serial port; the discrete I/O interface used for standalone operation; and the analog I/O interface provided for diagnostic purposes.

2.8.1 Host Register Interface

A host computer controls the IRMCS201 FPGA using either its slave-mode Full-Duplex SPI port, or a standard serial port (RS-232 or RS-422). Both interfaces are always active and can be used interchangeably, although not simultaneously. Refer to the Reference section for more information.

2.8.2 Discrete I/O External Interface

The discrete I/O external interface signals provide a means of controlling basic motor operation without using the host register interface. In this mode of operation, the analog reference (described later in this section) is used to directly control the target speed.

Figure 15 shows a schematic diagram of the discrete I/O signals. The signals are described in Table 2.

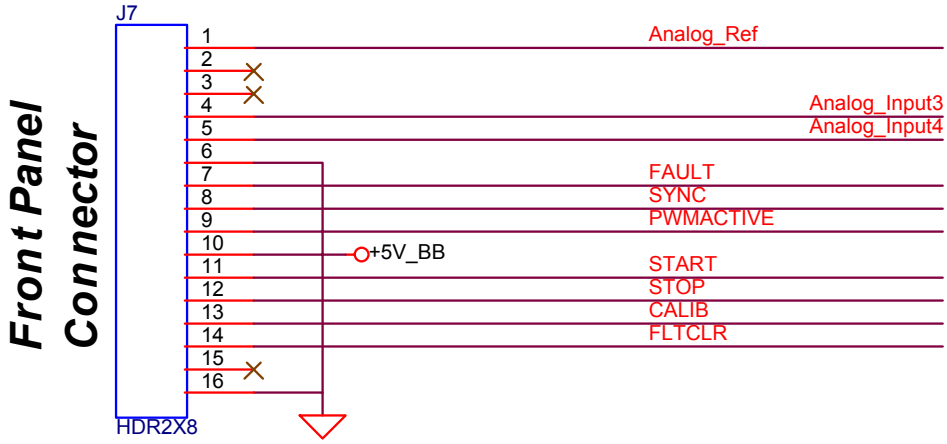


Figure 15. Discrete I/O Signals

Signal Name	Direction	Description
START	Input	A 100 nsec pulse on this signal enables PWM and FOC, equivalent to setting the PWMEN and FOCEN bits of the System Control register.
STOP	Input	A 100 nsec pulse on this signal disables PWM and FOC, equivalent to clearing the PWMEN and FOCEN bits of the System Control register.
CALIB	Input	Hold this signal high for 100 msec to initiate a calibration cycle (enables PWM and disables FOC).
FLTCLR	Input	A 1 µsec pulse on this signal clears a drive fault condition. Equivalent to setting the FAULT CLEAR bit of the Fault Control register.
FAULT	Output	This signal indicates the presence of a drive fault condition. The level is high when any of the bits in the Fault Status register are set.
SYNC	Output	This signal is held low for 3 µsec on each PWM period. (The falling edge indicates the start of the PWM period.)
PWMACTIVE	Output	High while PWM output is enabled, equivalent to the value of the PWMEN bit in the System Status register.

Table 2. External Interface Signal Description

NOTE: When the EXT_CTL bit in the System Configuration register is set to "0", the STOP signal is functional, but all other external interface signals are inactive.

Use the following procedure to configure the discrete I/O interface:

- Set the IfbOffsEnb bit in the System Control register to "1" if you want to us the CALIB signal to automatically calculate current feedback offset values.
- Write a "1" to the ExtCtrl bit in the System Configuration register to enable the external interface pins.
- Write a "1" to the RmpRefSel bit in the System Configuration register to select the reference A/D converter as the speed reference source.

2.8.3 Analog I/O Interface

IRMCx201 provides an analog input to control reference speed when using the discrete I/O interface and four channels of analog output for diagnostic use.

Analog Input

When using the external I/O interface, the user must supply ±10 volt analog reference input to control the reference speed. Figure 16 shows the typical hardware configuration. Using the analog reference input, a voltage level of +10V corresponds to maximum forward speed (16, 383) and –10V corresponds to maximum reverse speed (-16,384).

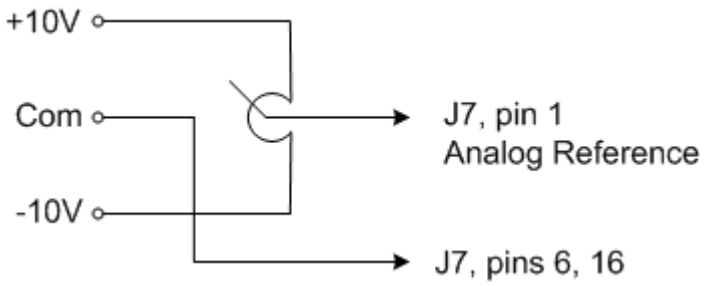


Figure 16. Analog External Reference Input

Analog Output

The diagnostic D/A interface provides four sources of diagnostic data and is intended for use with external RC filters for oscilloscope display. For each of the four signals, the user can select the data source from the items shown in Table 3. Each signal is encoded as a pulse-width modulated 8-bit value output at a frequency of 128 KHz. The data values are updated on each sync pulse. The values for each data source are scaled so that the valid range is represented as an 8-bit unsigned value. For example, the values of Qi and Di, which have an actual range of –16,384 to 16,383, are rescaled to the range 0 – 255 (so that 0 represents –16,384 and 255 represents 16, 383).

Value	Data Source
1	DC bus voltage
2	V phase current
3	W phase current
4	(not valid)
5	Speed PI reference
6	Speed PI feedback
7	Speed PI error
8	IQ ref
9	Q axis voltage (Qv)
10	D axis voltage (Dv)
11	Angle in quadrature counts
12	Q axis current (Qi)
13	D axis current (Di)
14	A axis (stationary frame) voltage Av
15	B axis (stationary frame) voltage Bv

Table 3. Data Sources for Analog Output

2.9 Sequencing Control

The sequencing control is contained in the IRMCx201 system to facilitate basic I/O sequencing. The signals shown in Table 4 can be directed either by local discrete I/O pins or host register interface. STOP is always activated by either the host interface register or the local STOP input pin.

Signal	Direction	Description
START	Input	Start motor. Edge triggered, low true signal. In order to start the motor, PWMEN signal needs to be asserted.
STOP	Input	Stop motor. Level sensitive, low true signal. This command initiates a coast-to-stop operation and immediately disables PWM output upon assertion.
FAULT	Output	Indicates a pending FAULT condition. It is cleared upon FLTCLR assertion. Low indicates FAULT.
FLTCLR	Input	Clear pending FAULT. Edge sensitive, low true signal.
CALIB	Input	Current feedback offset calibration. Level sensitive.
PWMEN	Input	PWM enable signal. Level sensitive, low true signal. This signal serves as a permissive signal for starting the motor.
SYNC	Output	Synchronized signal to PWM carrier frequency period. Stays high for the first half of the PWM cycle and stays low for the latter half of the PWM cycle. Thus this signal generates a 50% duty cycle signal.

Table 4. Sequencing Control Signals

Internally, IRMCx201 has three states: Stand-By or STOP state, RUN state, and FAULT state. Transitioning to each state can be caused either by initiation of the I/O pins described above or internal drive conditions such as overcurrent, overvoltage, etc. The state diagram is shown in Figure 17.

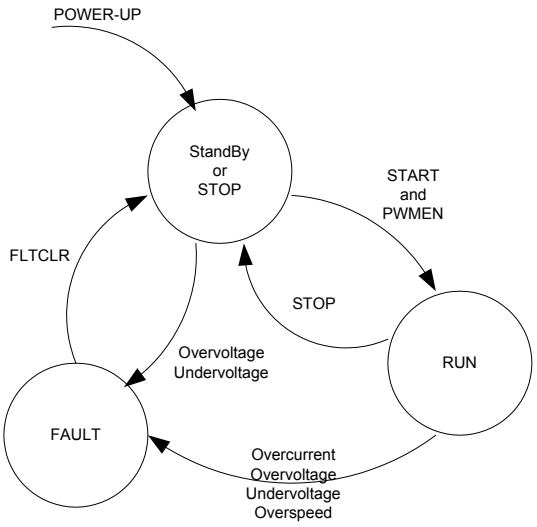


Figure 17. State Diagram and Sequencing

2.10 Fault Handling and DC Bus Dynamic Braking

The IRMCx201 system has built-in drive fault and protection features. Table 5 summarizes the types of drive fault conditions.

Fault	LED Indication	Status indication on Host Register Interface	Description
Overcurrent /Overtemperature	Static RED	Fault Status Read Register, field GATEKILL = 1	Overcurrent or overtemperature occurred. The IGBT gate driver (IR2136) disables gate drive outputs, momentarily latches a fault condition and asserts GATEKILL to the FPGA. This activates the fault latch inside the FPGA.
Overvoltage	Static RED	Fault Status Read Register, field OVER VOLT = 1	Overvoltage of the DC bus occurred. Only the fault latch inside the FPGA is activated.
Overspeed	Static RED	Fault Status Read Register, field OVER SPEED = 1	The speed of the motor exceeded the maximum speed. Only the fault latch inside the FPGA is activated.
Overrun	Static RED	Fault Status Read Register, field PTIME FAULT = 1	The computation of algorithm exceeded the selected PWM carrier frequency period. Only the fault latch inside the FPGA is activated.
Low voltage	Static RED	Fault Status Read Register, field LOW VOLT = 1	The bus voltage dropped below 120V. Only the fault latch inside the FPGA is activated.
DC bus power loss	LED off	N/A	DC bus power is not energized.
FPGA	Static RED	N/A	FPGA is not functioning indicating either configuration is not completed correctly and/or FPGA has a hardware problem itself.

Table 5. Drive Fault Conditions

When any drive fault occurs, the PWM output is disabled and the gate signals off the FPGA device are negated. This condition remains latched until Fault_Clear action is undertaken by the user. Fault_Clear, a level sensitive signal event, can be initiated either through the Fault Clear bit in the Fault Control host register or the FLTCLR discrete I/O external interface pin.

2.10.1 Gatekill Structure and Overcurrent/Overtemperature Fault

The IGBT module is an advanced intelligent power module (IRAMY20UP60A) and rated at a 600V/20A. This IGBT module contains an integrated high voltage gate drive IC (IR2136) with a low side shunt resistor. Complete short circuit/overcurrent protection is included along with a thermistor and its overtemperature protection circuit. Ground fault protection circuit is also equipped on board, and its output shares with overcurrent signal circuit output by a wired-OR connection. The signal is fed to an opto-coupler device to trigger the signal to FPGA pin, GATEKILL.

When overcurrent condition occurs, GATEKILL is asserted and momentarily latched within IR2136 for the programmed period which is approximately 9 milliseconds. After this period, the pending fault is automatically cleared by itself. Meanwhile triggered GATEKILL assertion latches and inhibits all PWM output gate signal off FPGA until user initiates FAULT CLEAR action.

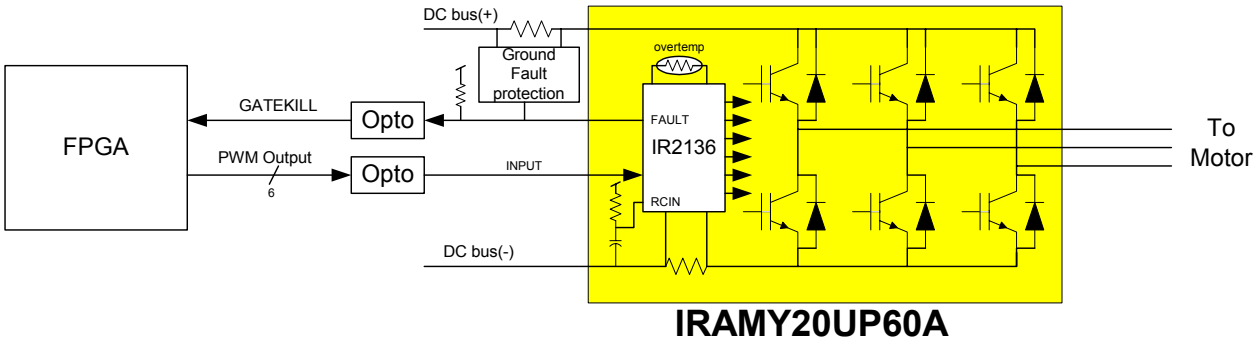


Figure 4. Protection Circuit Block Diagram

Figure 4 shows the protection circuit diagram. The IGBT module contains a low side shunt resistor with protection circuit. Inside of this module, there is an RC circuit connected to RCIN input of IR2136 which automatically initiates FAULT CLEAR in 9 milliseconds after assertion of FAULT. The IGBT module also contains overtemperature protection circuit which also shutdowns all IGBTs and performs automatic FAULT CLEAR as well. Overtemperature threshold level is set at approximately 110°C. IRMCS201 contains the ground fault protection circuit on the high side DC bus (+) node. The circuit senses positive ground fault current and send trigger signal to GATEKILL via a wired-OR FAULT signal.

Once any fault condition is detected, the IR2136 inside of the IGBT module momentarily latches the condition and initiates FAULT output and shutdown of all six IGBTs. Upon receiving the FAULT signal at its GATEKILL input, the FPGA disables all PWM gate signals and latches GATEKILL. It also disables PWM output, forcing the PwmEnbW and FocEnbW bits of the System Control register to "0".

To reset a fault condition, first write a "1" to the FltClr bit of the Fault Control register. This clears the fault in the FPGA. Then write a "0" to the FltClr bit to re-enable fault processing. Note that PWM output does not automatically restart after a fault condition is cleared.

2.10.2 DC Bus Faults and DC Bus Braking

Table 6 describes the actions taken at various DC bus voltage levels. When the Brake IGBT ON level is detected, the Brake IGBT is turned on and the BRAKE signal in the DC Bus Voltage read register is latched until the DC Bus voltage falls below the Brake IGBT OFF level, at which point Brake IGBT is turned off. Overvoltage and low voltage conditions produce corresponding drive faults, as described in Table 5.

DC Bus Action	DC Bus Level	Description
Overvoltage	410V	Overvoltage of the DC bus occurred. IGBT gate driver is disabled and the latched FAULT pending to the FPGA.
Brake IGBT ON	380V	Brake IGBT is turned ON above this voltage.
Brake IGBT OFF	360V	Brake IGBT is turned OFF below this voltage.
Nominal bus voltage range	360V-120V	Nominal DC bus voltage range
Low voltage	120V	Lowvoltage fault. PWM is disabled.

Table 6. DC Bus Voltage Level Management

2.11 LED Modes

The operating state of the FPGA is indicated by the LED module. There are three indication modes. Mode 1 indicates successful configuration of the FPGA. The LED is green in this mode. Thus, a green LED appears automatically right after successful completion of FPGA configuration following a power up.

A red LED indicates a drive fault condition. This is Mode 2.

The LED is not lit in Mode 3. This is a hardware fault condition. This means that either configuration data was not transferred to FPGA correctly or the FPGA itself has a hardware problem.

Mode	LED Indication	Description
Mode 1	Green	FPGA configuration has been done correctly and FPGA is functioning normally.
Mode 2	Red	A drive fault condition is pending.
Mode 3	Off	FPGA is not functioning indicating either configuration is not completed correctly and/or FPGA has a hardware problem itself.

Table 7. LED Modes

3 Techniques

3.1 Drive Parameter Setup

IRMCx201 includes an Excel workbook file that partially automates the procedure of calculating the appropriate values for configuration and tuning parameters. In the workbook, you enter motor nameplate data and parameters specific to your application, and Excel formulas calculate the appropriate values for certain motor configuration parameters and write registers. In Excel, you export these values to a text file, and in ServoDesigner, you can import the text file to fill in the motor configuration settings and register values. Then, when you execute the Starting Angle and Configure Motor function in ServoDesigner, the appropriate values are written to the FPGA.

The Excel Workbook File

The Excel workbook file is named IRMCS201-DriveParams.xls. Double click the file to open it in Excel.

At the bottom of the workbook window, you'll see a number of sheet tabs, which you can use to select the worksheet you want to display. You can use the arrows in the bottom left corner of the window to scroll through the sheet tabs.

The first tab, labeled "Calculation," selects a worksheet that shows the values used to calculate the exported parameter and register values. The second tab, "IRMCS201_Parameters," selects a worksheet that shows the results of the calculations. This is the worksheet that must be exported to a text file for use with ServoDesigner. The remaining tabs select motor setup worksheets, which have a format similar to that of the Calculation worksheet, but are pre-initialized with values appropriate for specific motors. Each tab is labeled with the name of the motor and an associated motor ID number in parentheses.

If you want to use one of the supported motors, first note its ID number (as shown on the sheet tab). Then click on the Calculation sheet tab and enter the motor ID number in the blue box at the top of the Calculation worksheet. When you press Enter or click to another field on the worksheet, the values from your selected motor setup worksheet are copied to the Calculation worksheet.

If you want to use an unsupported motor, you can modify one of the existing motor setup worksheets or use the last worksheet, labeled "New Motor." In addition to entering values specific to your motor (as described below), be sure to change the motor name and description at the top of the worksheet and the motor name on the sheet tab (double-click the label on the tab to change it). When your new motor setup worksheet is complete, click on the Calculation tab and enter the motor ID number in the blue box at the top of the Calculation worksheet to copy the values from your motor setup worksheet. You can modify values directly on the Calculation worksheet, but the values will be lost if you later copy another motor setup worksheet by entering a motor ID in the blue box.

Enter Drive Parameters in Excel

The first stage of configuring drive parameters involves entering the correct settings for the motor you're using and the specific requirements of your application. If you're using one of the supported motors, you can skip Steps 1 and 2.

Step 1. Initialize a motor setup sheet for your motor.

Click on the sheet tab that selects the motor setup worksheet for your motor. The first two lines of the motor setup worksheet describe the motor. An example is shown below. Double click on line 1, column B to change the motor name, and double click on line 2, column B to enter a description of the motor. The motor name and description are for your own use; they're not used in the calculations and not exported to ServoDesigner.

Double click on line 1, column E to specify the motor type: "1" for a permanent magnet motor or "2" for an induction motor.

	A	B	C	D	E	F	G	H	I	J
1	Motor:	P30B06040DXS00M			1	(1 : Permanent Magnet Motor, 2: Induction Motor)				
2		Sanyo Denki 400W								

Step 2. Enter Motor Information.
The motor information section of the motor setup worksheet contains parameter settings that you should be able to find in the datasheet for your motor or on its nameplate.

"========== Motor Information =========="			
Rated Speed	**3000**	rpm	
L_phase	**6.44E-03**	H	(line to line Inductance) / 2
R_phase	**1.4**	ohms/ph	(line to line Resistor) / 2
Rated Amps	**2.7**	Arms	
(NLC)No Load Current	**0**	Arms	
Inertia of Motor	**2.55E-05**	Kg-m2	
(Kt) Torque Constant	**0.533**	N-m/Arms	
(Ke) Voltage Constant	**18.6**	V ln-rms/krpm	
Poles	**8**		
Encoder PPR	**2000**	pulse/revolution	
Wire-Saving Encoder?	**TRUE**	(TRUE / FALSE)	

To enter a value for each parameter, double click in column B on the same line as the parameter name. Descriptions of each parameter are provided below.

Rated Speed	The rated speed of the motor (in RPM).
L_phase	Motor per phase inductance (in Henry).
R_phase	Per phase resistance of the motor **plus** cable (in ohms).
Rated Amps	The rated current of the motor (in Amps rms).
No Load Current	The no load current of the motor (in Amps rms). Needed only for induction motor.
Inertia of Motor	Motor inertia (in Kg-m^2)
Torque Constant (Kt)	Motor torque constant (in Newton-Meter per Amps rms).
Voltage Constant (Ke)	Motor voltage constant (in line-to-neutral rms volts per thousand rpm). Line to neutral voltage = line to line voltage / sqrt(3).
Poles	The number of motor poles.
Encoder PPR	Number of encoder pulses per revolution.
Wire-Saving Encoder	Encoder type. Enter TRUE for a wire-saving encoder; otherwise FALSE.

Step 3. Enter Application Information
The application information section of the motor setup worksheet contains parameter settings that describe the requirements of your specific application.

"========== Application Information =========="		
"---------- General ----------"		
Max RPM	4500	rpm
(Vdc_Nom) Nominal Vdc	310	Volts
(OvLoad) Max pu motor current at rated speed	3	pu
"---------- Speed Regulator Tuning ----------"		
Speed Regulator BW	200	rad/sec
Positive Speed Rate limit	1000	rpm/sec
Negative Speed Rate limit	1000	rpm/sec
Inertia of Load (measured)	0	Kg-m2
SpdLpRate	2	1 SpdLoop per this # of CurLoop
"---------- Current Limits ----------"		
Motoring Limit	200	%
Regen Limit	200	%
"---------- Inverter Switching Frequency ----------"		
(fc) Pwm carrier freq	10	KHz
Dead_Time	0.5	usec
" ---------- Current Regulator Tuning ---------- "		
(Ireg_BW) Current Reg BW	2500	rad/sec

The parameters are described below. To enter a value for each parameter, double click in column D on the same line as the parameter name.

Max RPM — This is the maximum speed (in rpm) required for your application. When motor speed exceeds this value, the system will generate an Overspeed trip fault. You may need to increase this value to make SpdScl less than 65535.

Nominal Vdc — Nominal DC bus voltage (in volts). For use with the IRMCS201 development platform, the nominal dc bus voltage should be set to 1.414 * ac input voltage (ac input voltage: USA 110V or 230, JAP 100V, UK 220V, etc.).

Max pu motor current at rated speed — This is the anticipated maximum current in per unit drawn by the motor at the motor's rated speed. Setting this parameter to 1 pu means that the system drives 100% rated current at the rated speed.

Speed Regulator BW — Speed regulator bandwidth (in rad/sec). The system may not tolerate high speed regulator bandwidth (due to mechanical coupling, gear box etc.), resulting in load mechanical resonance. If you're not sure how to set this parameter, start with a value of 10 rad/sec and raise it gradually as you tune the system.

Positive Speed Rate limit — This parameter defines the maximum changeable speed allowed for acceleration in second.

Negative Speed Rate limit — This parameter defines the maximum changeable speed allowed for decceleration in second.

Inertia of Load — Load inertia (in Kg-m^2). If load inertia is not specified in your design data, use your best estimate and adjust the value later when you fine-tune drive operation.

SpdLpRate — Number of current loops for each speed loop. For example, a value of 3 would process the speed loop on every third current loop. Range is 0 to 7.

Motoring Limit — Positive torque current limit (in percentage of rate current). Motoring power is energy transferred from the inverter to the motor while the motor is running. Maximum hardware current should be considered.

Regen Limit — Negative torque current limit (in percentage of rate current). Regenerative energy is transferred from the motor to the inverter when the motor decelerates. If the system does not contain a breaking resistor to absorb the regenerative energy, an increase in DC bus voltage (and potential trip fault) results. Maximum hardware current should be considered.

Pwm carrier freq — PWM carrier frequency. 10 KHz is the default setting for the IRMCS201 product. The setting of this parameter is a tradeoff between current ripple, inverter loss and EMI noise.

Dead Time — PWM dead time in usec to prevent IGBT shoot through.

Current Reg BW — Current regulator bandwidth (in rad/sec). Normally 1/10 of PWM carrier frequency is the maximum value.

Step 4. Enter Advanced Information (Platform Dependent)
The advanced information section of the motor setup worksheet contains parameter settings that are specific to the hardware platform. **If you are using the IRMCS201 development platform without modification, you do not need to modify these settings.**

"========== **Advance Information (Platform dependent)** =========="			
Note: Below values are fixed for IRMCS201 platform however can be changed for other platform			
(Clk) FPGA clock freq		**33.333**	MHz
Dc bus Scaling (Vdc_Scl)		**8.1875**	cts/volt
I_Torque	(I_Trq_Rated)	**4095**	cts for rated Amps
(Mod_Pk) - U_Alpha U_Beta max linear modulation		**2355**	cts
" ---------- **Desired Speed feedback Scaling** ---------"			
(Spd_Scale)		**16384**	cts/(Max RPM)
" ---------- **Current Feedback Scaling** ----------"			
Current Shunt Resistor		**10**	mOhm

The parameters are described below. To enter a value for each parameter, double click in column D on the same line as the parameter name.

FPGA clock freq — The frequency of the system clock, in MHz.

Dc bus Scaling — Scaling from internal count to voltage.

I_Torque — 4095 indicates 100% rated amps.

U_Alpha U_Beta max linear modulation
Maximum linear range of Space Vector PWM

Spd_Scale Maximum speed is mapped to this value internally.

Current Shunt Resistor Shunt resistor value for current feedback.

Step 4. Enter Commutation Information

This section of the motor setup worksheet contains parameter settings used in determining the starting position of the motor.

"========== Commutation Information =========="			
Angle of Z-pulse (based on UV line to line voltage)		272	degree
Mid Angle when Hall CBA is 001		120	degree
Mid Angle when Hall CBA is 010		240	degree
Mid Angle when Hall CBA is 011		180	degree
Mid Angle when Hall CBA is 100		0	degree
Mid Angle when Hall CBA is 101		60	degree
Mid Angle when Hall CBA is 110		300	degree

The parameters are described below. To enter a value for each parameter, double click in column D on the same line as the parameter name.

Angle of Z-pulse Enter the angle (in degrees) that z-pulse occurs in the reference of UV line to line voltage.

Mid Angle when Hall CBA is *cba*

For these six parameters, enter the angle (in degrees) that corresponds to each combination of Hall sensor values (*cba*).

Export Drive Parameters in Excel

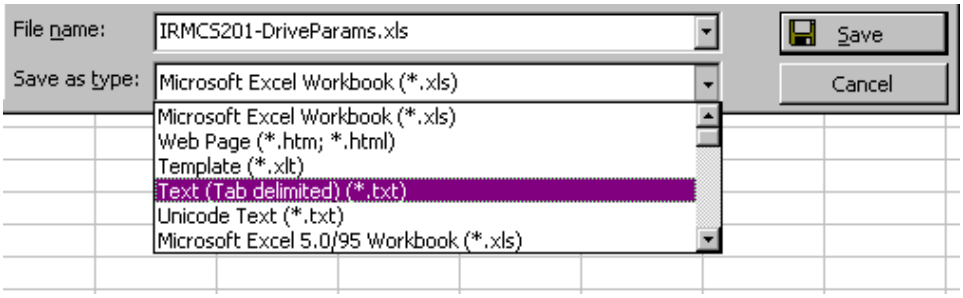
In the second stage of configuring drive parameters, the parameter settings you selected in the previous section are used to calculate values for a number of IRMCx201 FPGA write registers. The write register values and motor configuration parameters are written to a text file in a specific format defined for use with ServoDesigner.

Step 1. Save Your Settings

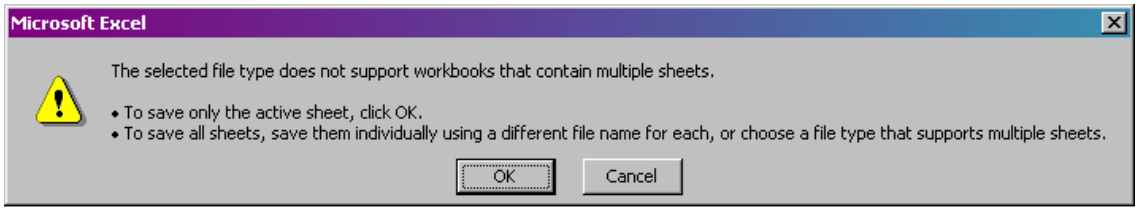
When you are satisfied with your settings, select Save from Excel's File menu to save your workbook file in ".xls" format.

Step 2. Export Drive Parameters

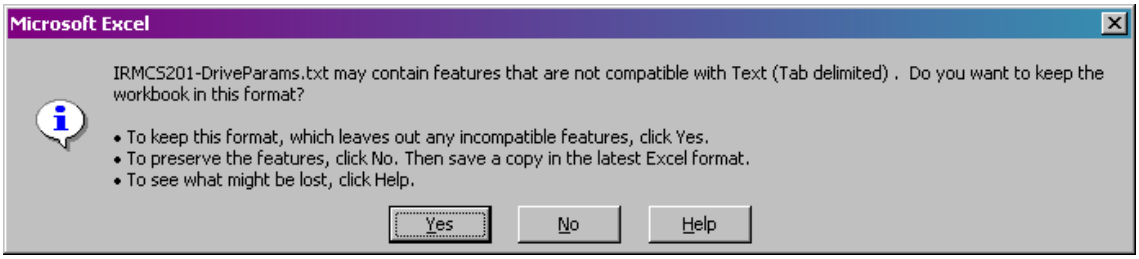
Click on the "IRMCS201_Parameters" tab at the bottom of the workbook window. This worksheet shows the register values that are calculated based on the settings in the Calculation worksheet. From Excel's File menu, select "Save As…". In the Save As dialog, select Save as type: "Text (Tab delimited) (*.txt)" as shown below. Then browse to a folder where you want to save the exported drive parameters file, specify a file name, and click Save.



Click OK when you see the following warning message:



Click Yes when you see a warning message like this:



Close Excel window.

Import Drive Parameters in ServoDesigner

The final stage of drive parameter configuration involves loading the drive parameter settings into your ServoDesigner database and writing the registers to the IRMCx201 FPGA. For information about how to use ServoDesigner, refer to the ServoDesigner User's Guide. In particular, Section 10.3 of that document describes the Import Drive Parameters feature.

The text file you exported from the Excel workbook contains two sections: Parameters and Registers.

The Parameters Section

The Parameters section specifies motor configuration parameters, which are saved in the ServoDesigner configuration file (.irc file). In ServoDesigner, you can view and modify these settings by selecting Motor Configuration from the Preferences menu. When you import the drive parameters text file into ServoDesigner, the motor configuration parameters in the import text file always replace the current settings in the ServoDesigner database.

The Registers Section

Each of the entries in the Register section of the file identifies a write register and a value to be stored in the register. In your ServoDesigner database, there are several locations where each register value can be used:

- In the register definition, the Value to Write is written to the corresponding FPGA register when you double click the register entry.
- Also in the register definition, the EEPROM Value to Write can be saved to EEPROM and used to initialize the FPGA register on power up.
- In the Function Definitions section, one or more functions may write the register value to the FPGA. (A function is set up to perform a sequence of operations automatically.)

When you import the drive parameters text file into ServoDesigner, you have several options for updating any or all of these register settings with the value specified in the file.

Step 1. Run ServoDesigner and Open a Database

Start ServoDesigner and select Open from the File menu. ServoDesigner configuration files have the file extension ".irc". Browse to locate a ServoDesigner configuration file and open it. (If you haven't already created a configuration file to use with your project, make a copy of the example file included with the release.)

Step 2. Import Drive Parameters
From the File menu, select Import, and from the Import sub-menu, select Drive Parameters. Browse to locate the text file you exported from Excel and click Open to open it. In the Import Drive Parameters dialog, select one of the three available modes and click OK. Depending on the mode you choose, ServoDesigner may prompt you for confirmation before modifying each register setting or group of settings. Refer to the ServoDesigner User's Guide for more information about the available modes of operation.

Step 3. Save the New Settings
The Import Drive Parameters function in ServoDesigner updates register values in the database you currently have open. If you want to save the new settings in your configuration file, you must select Save from the File menu before exiting ServoDesigner. If you don't do this, the updates will be lost, and you'll have to repeat the Import Drive Parameters function next time you open your configuration file.

Step 4. Write the Settings to the FPGA
The Import Drive Parameters function does not write any values to the IRMCx201 FPGA; it simply updates the register settings in your database. To transfer the register settings to the FPGA, you must either double click each write register individually (not recommended) or execute a function that writes the registers automatically. The Configure Motor function is pre-defined for this purpose. To execute the Configure Motor function, click the Configure Motor icon on the toolbar, or double click Configure Motor in the Function Definitions section of the tree view.

3.2 New Motor Adaptation for a Permanent Magnet Motor

This section provides a step-by-step procedure for determining the correct configuration parameters for a new permanent magnet motor, and describes how to store the parameters in the ServoDesigner tool. If you use the Excel workbook file as described in the "Drive Parameter Setup" section above, a portion of this procedure is automated and you can skip certain steps where noted.

A similar procedure for an induction motor is provided in Section 3.2.

3.2.1 Step 1 – System Setup

In this step, you need to prepare your motor, cables, computer and IRMCS201.

Step 1-1
Prepare IRMCS201 release 4.0A or later.

Step 1-2
Make an encoder cable and a power cable for your motor.

- Assemble 15-pin male D-Sub connector, referring to Figure1.
- Eleven pins are used: A+ (pin2), A- (pin3), B+ (pin4), B- (pin5), Z+ (pin6), Z- (pin7), HALL_A (pin10), HALL_B (pin11), HALL_C (pin12), 5V(pin1 or pin9) and GND (pin8 or pin15).
- Make sure that the encoder is a 5V type. If it is not a 5V type, proper modification is required.
- If hall sensors have differential output, connect only positive sides and leave negative sides open.

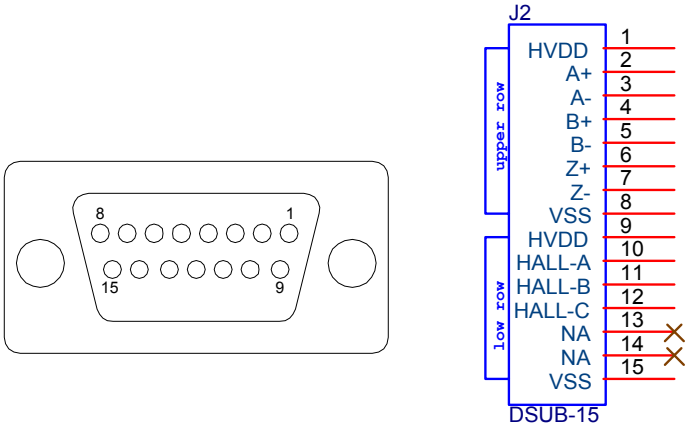


Figure 1. Encoder Interface Connector, J2

Step 1-3
Connect the encoder cable to J2 of IRMCS201 hardware.

Step 1-4
Connect 3-phase AC lines to R/S/T terminals of IRMCS201 hardware J1. Order doesn't matter. If single-phase AC lines are used, connect them to any two of R/S/T terminals

Step 1-5
Connect the power cable to U/V/W terminals of IRMCS201 hardware J1. Order doesn't matter, however use order as specified in motor datasheet if available.
Motor frame ground wire need to be connected to E terminal of J1 if available.

Step 1-6
Connect RS232 serial cable from the computer to IRMCS201 hardware J6.

3.2.2 Step 2 – Motor Sequencing and Encoder A/B Establishment

In this step, motor U/V/W phase sequence and encoder A/B will be determined.

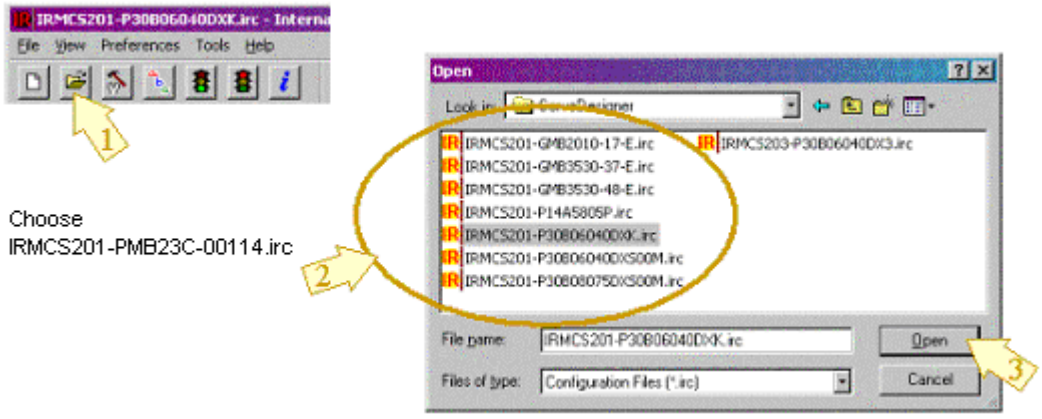
Step 2-1
Turn on the power and check the LED on IRMCS201 hardware is green.

- If the LED stays off, check your AC line and cables.
- If the LED is red, your hardware has a problem. Contact IR customer support.

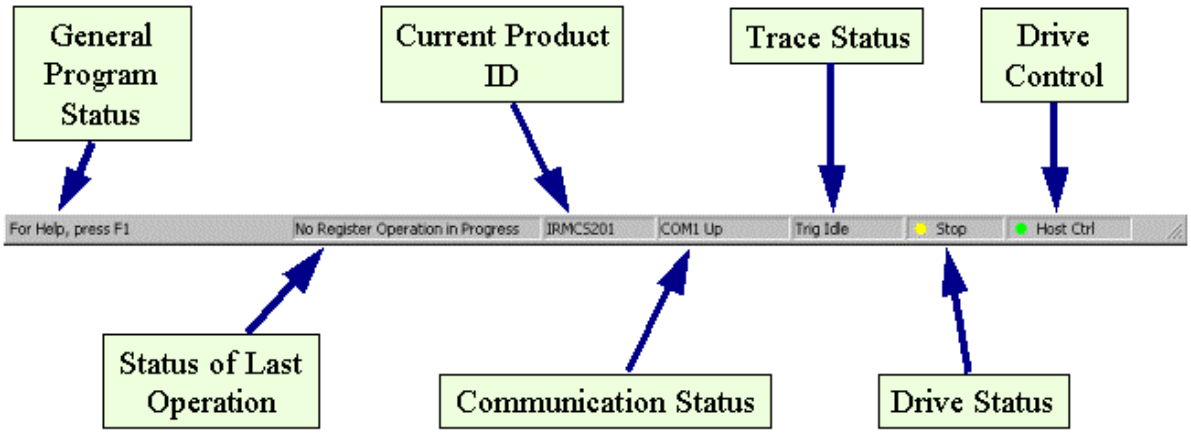
Step 2-2
Close all unnecessary program on your computer due to possible interaction with other applications.

Step 2-3
Execute 'ServoDesigner.exe'(Version 4.0.0.3 or later). Refer to the ServoDesigner User's Guide for questions about ServoDesigner if this is the first time.

Step 2-4
Open 'IRMCS201-PMB23C-00114.irc' as following display.



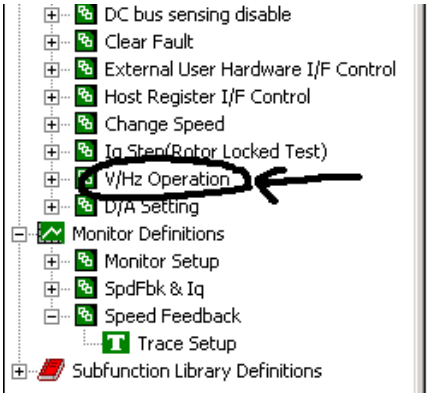
Step 2-5
Maximize the ServoDesigner's window and see right bottom of the window. It should display 'IRMCS201', 'COMx Up', 'Trig Idle', a yellow light with 'Stop' and a green light with 'Host Ctrl'.

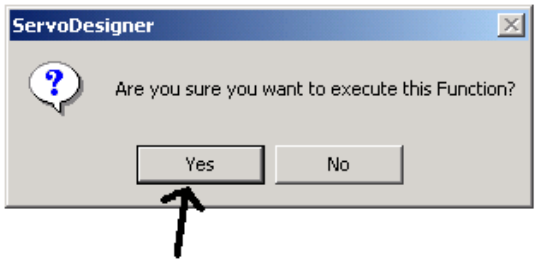


Step 2-6

Click 'Configure Motor' button.

Step 2-7
Expand 'Function Definition' and double click 'V/Hz Operation' and click on 'Yes'.





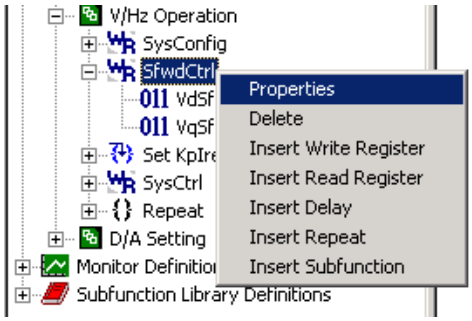
- If the motor shaft doesn't rotate and just vibrates, go to Step 2-8.
- If the motor shaft rotates in one direction, go to Step 2-11.
- If the motor shaft doesn't move at all, check the system setup.

Step 2-8

Click 'Stop motor' button.

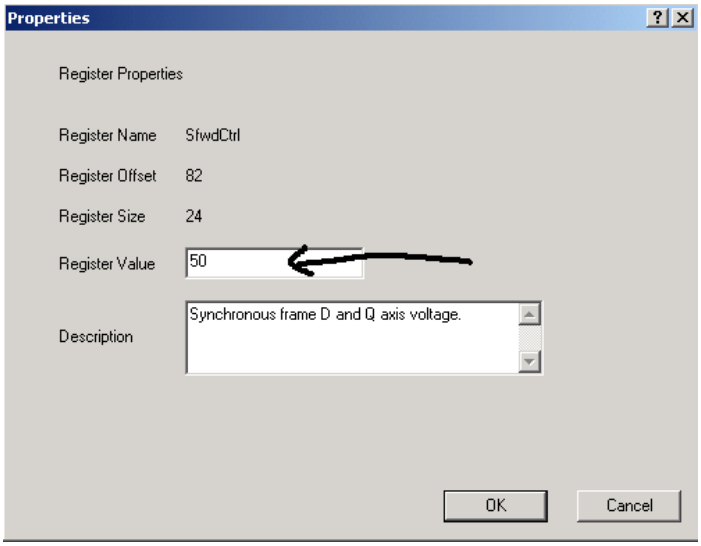
Step 2-9

Click '+' and, inside the 'V/Hz Operation' function, right click on 'SfwdCtrl'. Select Properties.



Step 2-10

Increase value in 'V/Hz Operation' by 10, click 'OK' and go to Step 2-7



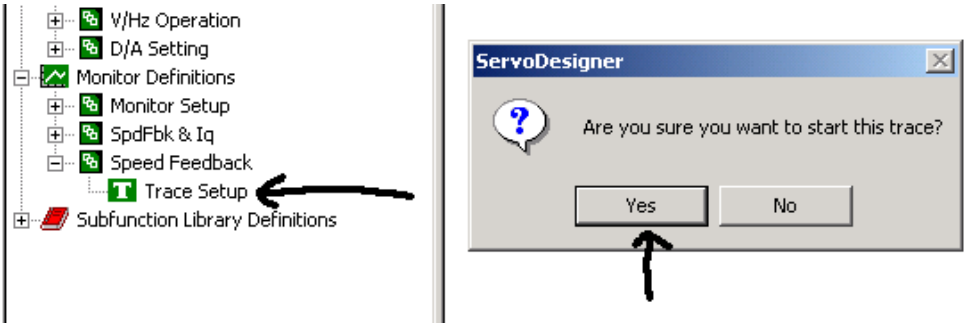
Step 2-11
Check the motor rotational direction at the motor front end (facing the motor front).

- If it's clockwise, go to Step 2-12.
- If it's counter clockwise, go to Step 2-14.

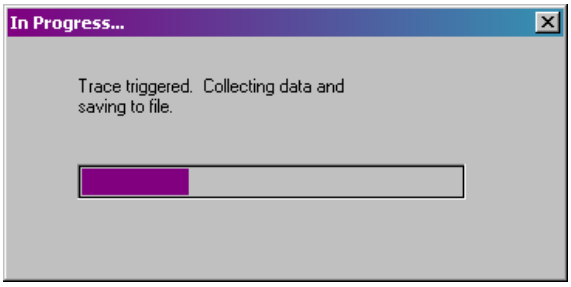
Step 2-12
Click 'Stop motor' button and turn off the power. Wait for a couple of minutes until the LED on IRMCS201 hardware goes off. Then proceed to Step 2-13.

Step 2-13
Swap 2 wires of power cable by moving 'V' phase wire to 'W' of J1 and 'W' phase wire to 'V'.
Turn on the power and go to Step 2-5.

Step 2-14
On the ServoDesigner window, go to 'Monitor Definitions' / 'Speed Feedback' / 'Trace Setup' and double click it. Then click 'Yes'.



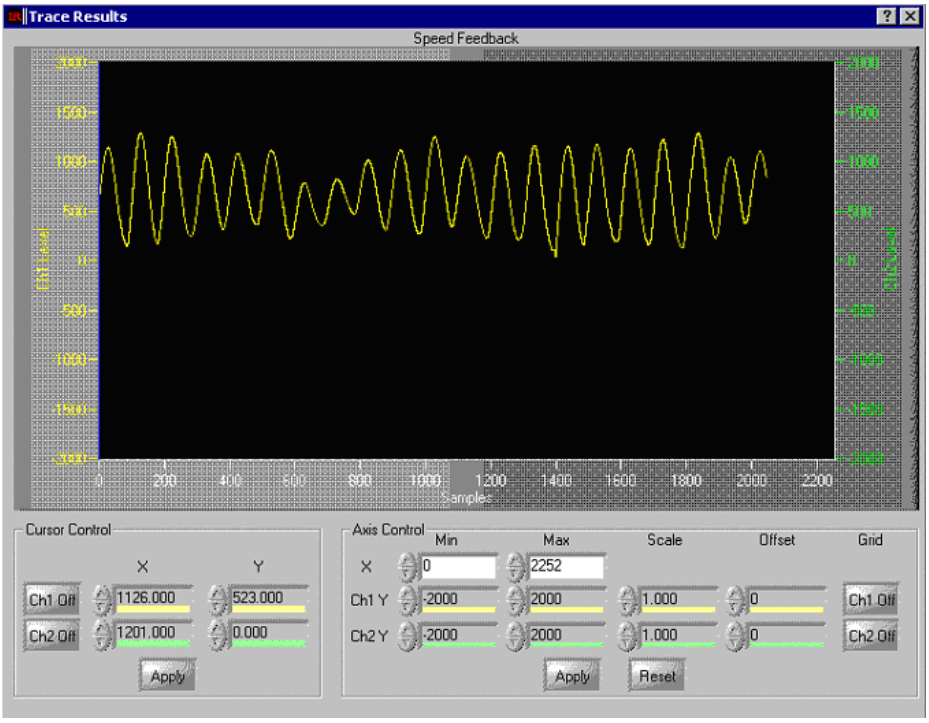
Step 2-15
Click 'Stop motor' button while progress bar is moving. Click 'OK' on ERROR message window. This error message happens because output file is not specified correctly.



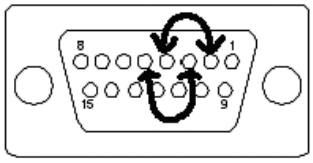
If progress bar window already disappeared and ERROR message window appeared before stopping the motor, click 'OK' on ERROR message window and then click 'Stop motor' button. It may be hidden behind Trace Result window.

Step 2-16
Check the Speed Feedback display (Yellow trace).

- If it's positive (greater than 0), go to Step 2-18.
- If it's negative (less than 0), go to Step 2-17.



Step 2-17
Turn off the power and wait for a couple of minutes until the LED on IRMCS201 hardware goes off.
Swap encoder A+ wire with encoder B+ wire and encoder A- wire with encoder B- wire in the connector going to J2.
In other words, change the connection between pin2 and pin4, and between pin3 and pin5.
Close the trace window and minimize the ServoDesigner.
Turn on the power and go to Step 2-5.



Step 2-18

Turn off the power and wait for a couple of minutes until the LED on IRMCS201 hardware goes off.
Put UVW labels on UVW wires respectively and remove UVW wires and E wire if connected from terminal block J1 on IRMCS201 hardware.
Now it's ready to go to Step3 Hall_ABC and Z_pulse Measurement.

3.2.3 Step 3 – Hall_ABC and Z_pulse Measurement

In this step, you need to prepare a 4-channel oscilloscope and measure UV line to line counter EMF voltage respect to Hall_ABC and Z_pulse.

Step 3-1
Set up your oscilloscope in order to measure Hall_ABC andU-V line to line counter EMF voltage.

- Set volt/div of all four channels to 10V/div.
- Set time/div to 10msec/div
- Hook up channel 1 probe to the motor 'U' wire and channel 1 ground to the motor 'V' wire.
- Hook up channel 2 probe to the test point 'HALL-C' and channel 2 ground to the test point 'VSS'.
- Hook up channel 3 probe to the test point 'HALL-B' and channel 3 ground to the test point 'VSS'.
- Hook up channel 4 probe to the test point 'HALL-A' and channel 4 ground to the test point 'VSS'.
- Set trigger source to channel 1, level to 10V, mode to normal, coupling to DC and slope to positive.

Step 3-2
Make sure the encoder cable is plugged in and the motor U/V/W are isolated one from another.
Turn on the power and rotate the motor shaft counter clockwise at the motor front end (facing the motor front) while monitoring channel 1. If it doesn't trigger, turn the shaft faster. Adjust trigger level, delay and time/div so that you can get one entire cycle of sinusoidal waveform like Figure 2.
Once you get a good waveform, save this waveform.

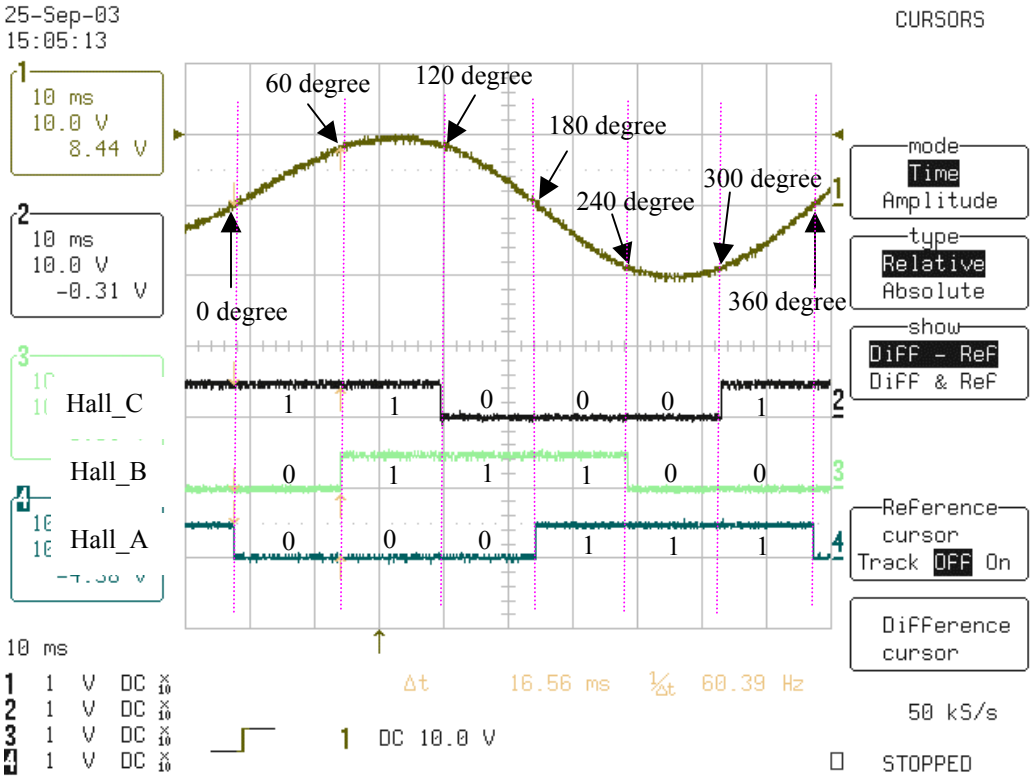


Figure 2. Sample Waveform of U-V counter EMF, Hall_C, Hall_B and Hall_A

Step 3-3

Turn off the power and set up your oscilloscope in order to measure the encoder Z_pulse and U-V line to line counter EMF voltage.

- Move channel 2 probe to the test point 'ENC-Z'.
- Set volt/div of channel 2 to 5V/div.
- Set trigger source to channel 2, trigger level to 2V and trigger mode to normal.
- Turn off channel 3 and channel 4.

Step 3-4
Turn on the power and rotate the motor shaft counter clockwise at the motor front end (facing the motor front) while monitoring channel 1. Adjust trigger level, delay and time/div so that you can get one entire cycle of sinusoidal waveform like Figure 3.
Once you get a good waveform, save this waveform, too.
Turn off the power and remove scope probes.

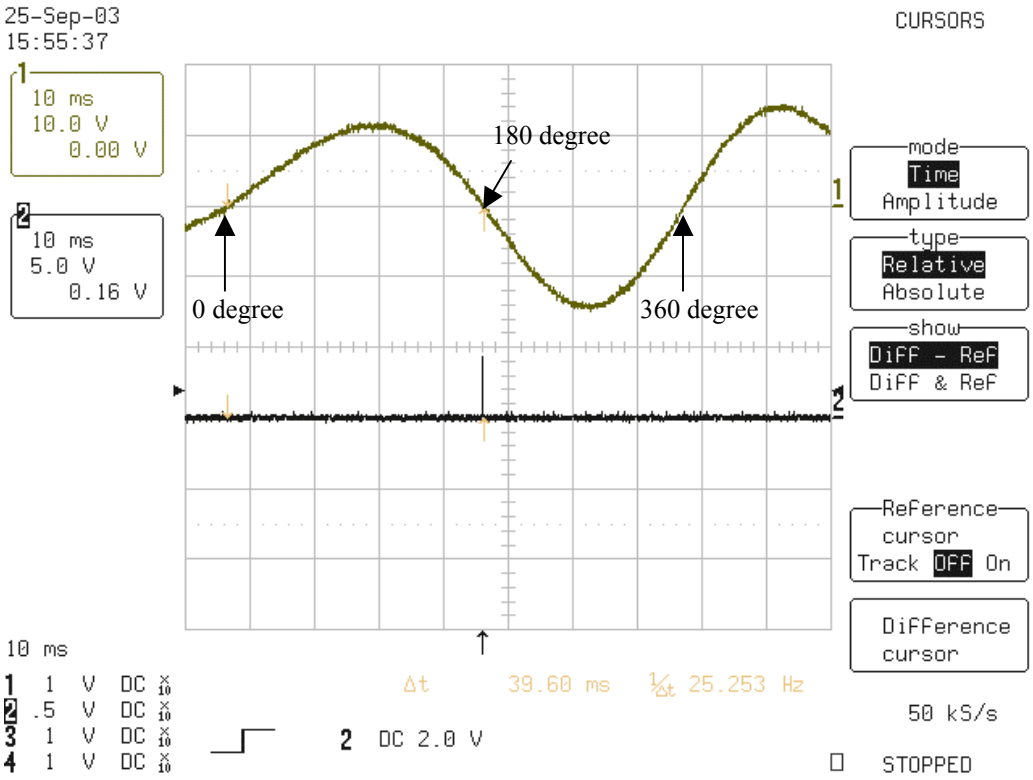
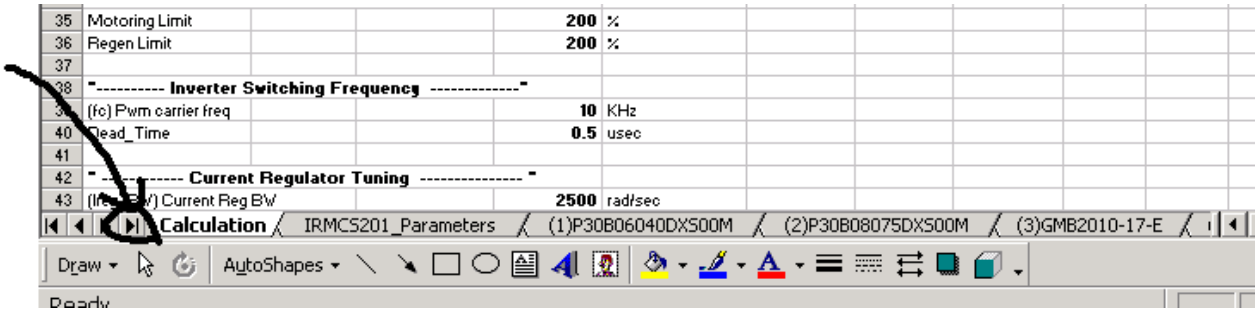


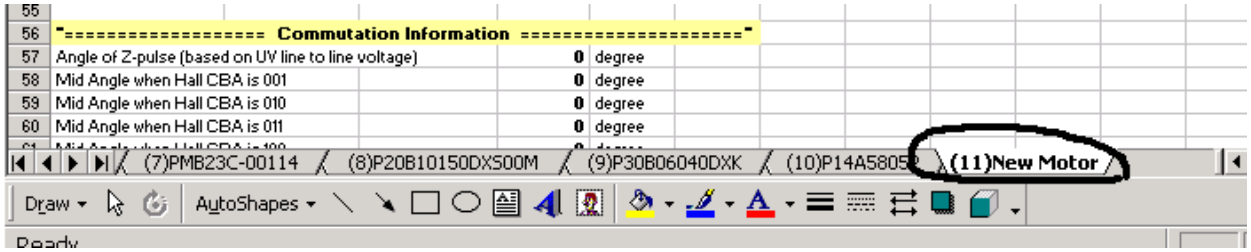
Figure 3. Sample Waveform of U-V counter EMF and Encoder Z_pulse

3.2.4 Step 4 – Entering Motor Parameters to Spread Sheet

In this step, you need to type all the information you have about your motor and make a text file for parameters.

Step 4-1
Open 'IRMCS201-DriveParams.xls' and select '(11) New Motor' Worksheet.





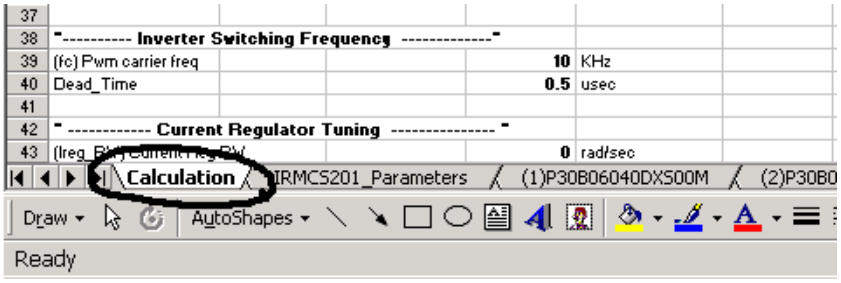
Step 4-2
Enter all the parameters. Refer to the '3.1 Drive Parameter Setup', if needed.
If any parameter is unknown, select one motor, which is most similar to yours, from provided sheets and copy from it.

Enter 'Commutation Information' based on your files saved. Zero angle is set as the point when U-V line to line counter EMF voltage cross zero from negative to positive. Hall CBA can change at each 30-degreesegment. Any transition from Hall sensors occurs only at the multiples of 30 degree. Choose from 0, 30,60,90, 120, 150, 180, 210, 240, 270, 300, and 330. Each HallCBA has 60-degree span. '000' and '111' don't exist.

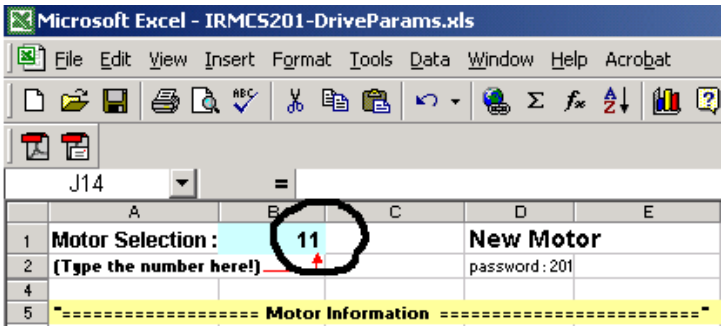
- In the example of Figure 3, it's obvious that Z_pulse occurs at 180 degree. Type 180 to 'Angle of Z-pulse'.
- In the example of Figure 2, when angle is 0-60 degree, Hall CBA is 100. Type the mid angle, which is 30 degree for this example, to 'Mid Angle when Hall CBA is 100' field. Fill out the rest fields.

56	"================== Commutation Information ===================="		
57	Angle of Z-pulse (based on UV line to line voltage)	180	degree
58	Mid Angle when Hall CBA is 001	270	degree
59	Mid Angle when Hall CBA is 010	150	degree
60	Mid Angle when Hall CBA is 011	210	degree
61	Mid Angle when Hall CBA is 100	30	degree
62	Mid Angle when Hall CBA is 101	330	degree
63	Mid Angle when Hall CBA is 110	90	degree

Step 4-3
- Go to 'Calculation' sheet.

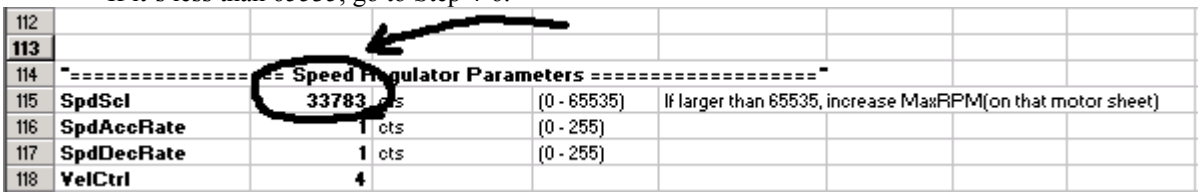


- Enter 11 to Motor Selection.



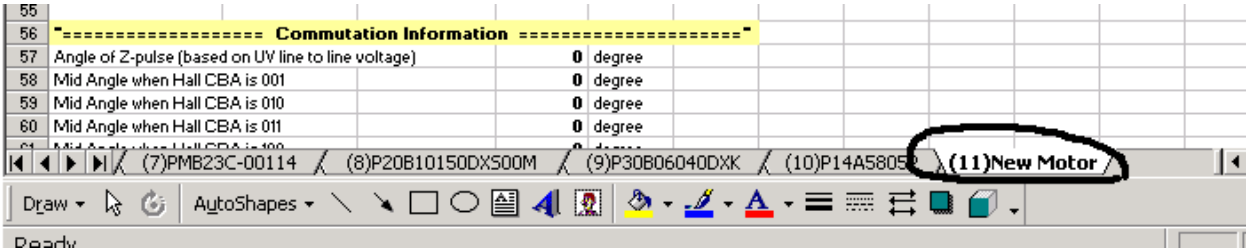
Step 4-4
Check 'SpdScl' value.

- If it's larger than 65535, go to Step 4-5.
- If it's less than 65535, go to Step 4-6.

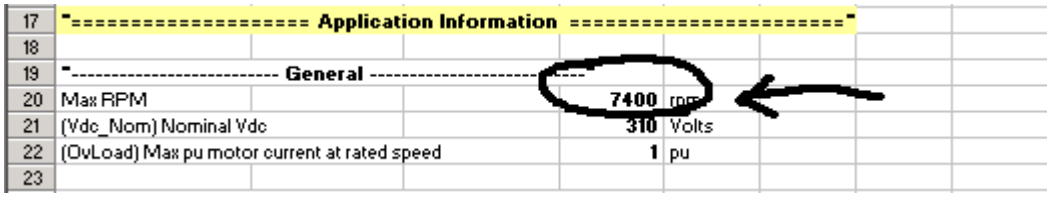


Step 4-5

- Go to '(11) New Motor' sheet.



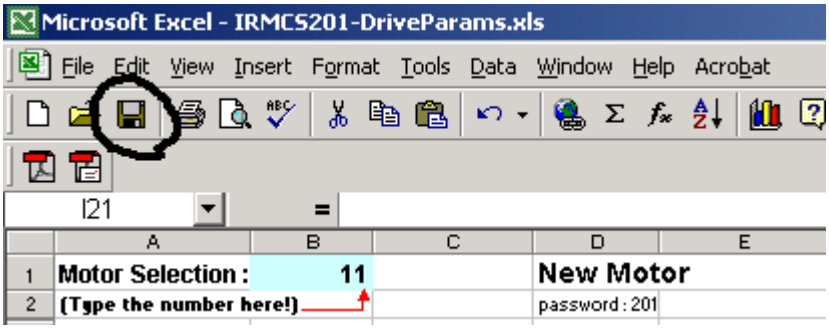
- Increase Max RPM by 1000.



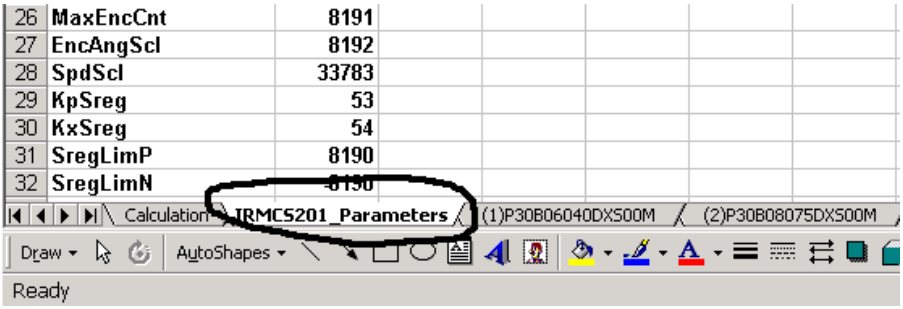
- Go to Step 4-3

Step 4-6

- Save Excel spread file.

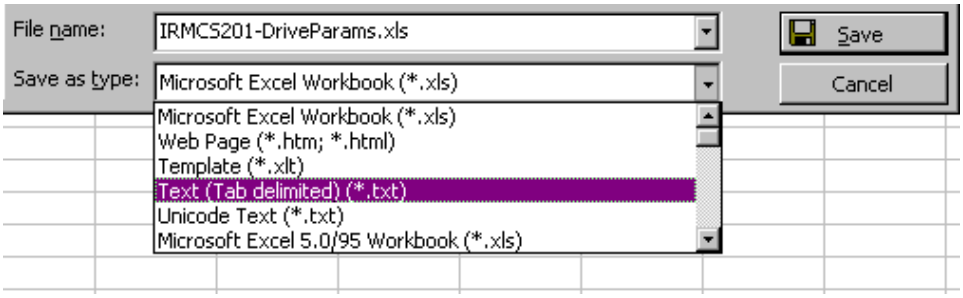


- Go to 'IRMCS201_Parameters' sheet.

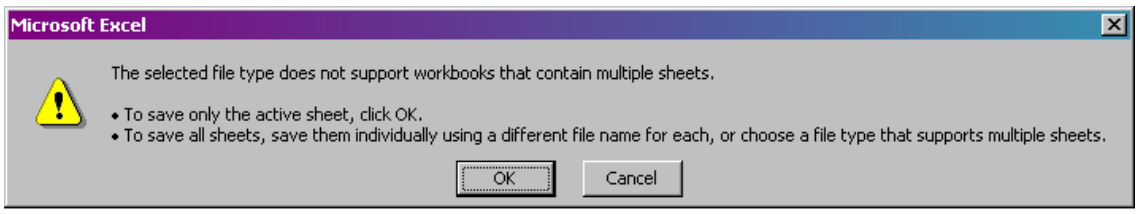


Step 4-7

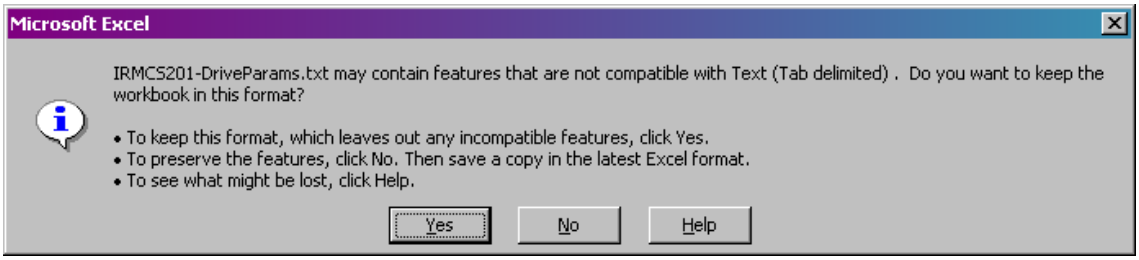
- Export parameters as text format. From Excel's File menu, select "Save As…". In the Save As dialog, select Save as type: "Text (Tab delimited) (*.txt)" as shown below. Then browse to a folder where you want to save the exported drive parameters file, specify a file name, and click Save.



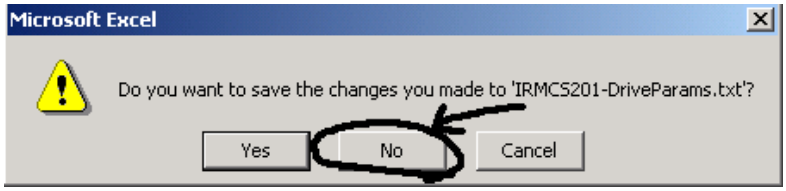
- Click 'OK' when you see the following warning message.



- Click 'Yes' when you see the following warning message.



- Close Excel Window and click 'No' when you see the following warning message.



3.2.5 Step 5 – Import Parameters to ServoDesigner and Run the Motor

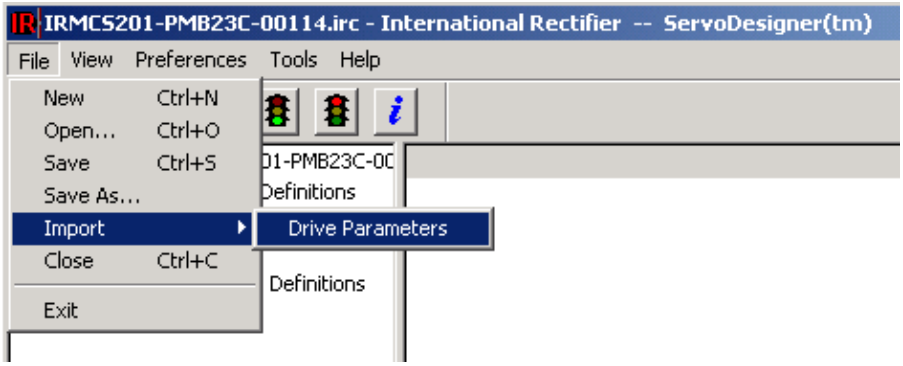
In this step, you import the text file to ServoDesigner and run the motor.

Step 5-1

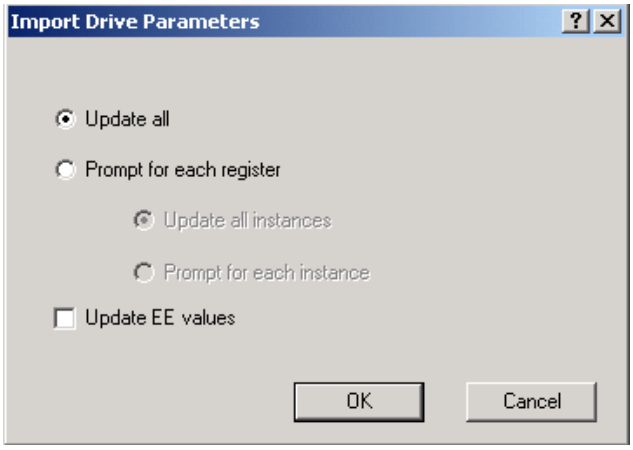
If you're following this procedure from Step 1, ServoDesigner was already executed in Step2 and 'IRMCS201-PMB23C-00114.irc' is opened. If not, please run ServoDesigner and open 'IRMCS201-PMB23C-00114.irc'. Maximize the ServoDesigner window.

Step 5-2

- From the File menu, select Import, and from the Import sub-menu, select Drive Parameters.



- Browse to locate the text file you exported from Excel and click Open to open it.
- In the Import Drive Parameters dialog, select 'Update all' unless you want something else. Depending on the mode you choose, ServoDesigner may prompt you for confirmation before modifying each register setting or group of settings. Refer to the ServoDesigner User's Guide for more information about the available modes of operation.



Step 5-3

- Connect your motor U/V/W/E wires back to U/V/W/E terminal of J1 on IRMCS201 hardware.
- Make sure system setup is OK and all probes are removed.
- Turn on the power and check LED and serial communication as we did in Step 2-1 and 2-5.

Step 5-4


Click 'Configure Motor' button.

Step 5-5


Click 'Starting Angle' button.

Step 5-6


Click 'Start Motor' button.

Step 5-7


Click 'Stop Motor' button.

Step 5-8

In Step 5-6, did motor start rotating without clunking?

- If yes, go to Step 5-11.
- If no, go to Step 5-9.
- If LED changes to red, go to Step 5-10.

Step 5-9

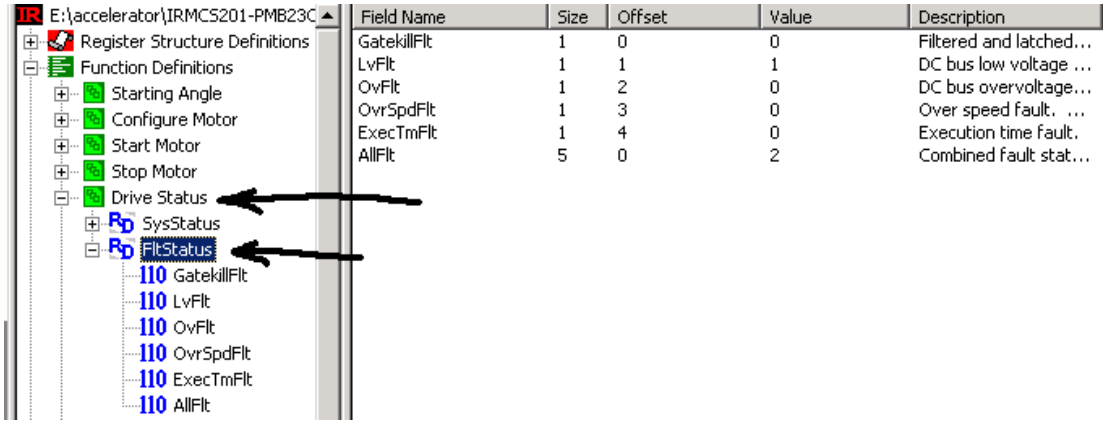
Check your spreadsheet values.

- Especially Hall CBA field and Z-pulse field are correct?
- Encoder PPR and Number of Pole are correct?

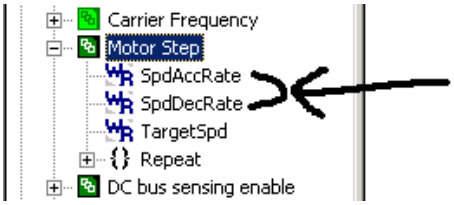
Step 5-10


Click 'Drive Status'.

Expand 'Drive Status' function by click '+' and select FltStatus. Look at the right hand side window to figure out which fault happened. Refer to IRMCS201 datasheet.



Step 5-11
Go to 'Motor Step' function and change 'SpdAccRate' and 'SpdDecRate' to 127 respectively by right click and selecting Properties.



Step 5-12
Run 'Motor Step' function by double click.
Does the motor changes speed from 0 rpm to certain high speed?
- If yes,

Step 5-13

Click 'Stop Motor' button.
Save this irc file as different name.

Now, you're ready to play with IRMCS201!

3.3 New Motor Adaptation for an Induction Motor

This section provides a step-by-step procedure for determining the correct configuration parameters for a new induction machine adaptation, and describes how to store the parameters in the ServoDesigner tool. If you use the Excel workbook file as described in the "Drive Parameter Setup" section above, a portion of this procedure is automated and you can skip certain steps where noted.

A similar procedure for a permanent magnet motor is provided in Section 3.2.

An induction motor can be used in place of a permanent magnet motor based on the following criteria.

1) Must be 230V 3 phase machine
2) Must be equal to or less than 1 kW or 6 Arms continuous current rating.
3) Must have an encoder at the back end of the motor, and be incremental A quadrature B type with 5V differential output. The maximum encoder line count is 5000 ppr.

3.3.1 Step 1 - Encoder Connector Assembly

- Assemble 15-pin male D-Sub connector (Digi-Key part # A23087-ND and A2072-ND), referring to Figure 18.
- Only six6 pins are used, because z-pulse is not necessary for an induction machine. The six pins are: A+ (pin2), A- (pin3), B+ (pin4), B- (pin5), 5V(HVDD, pin1 or pin9) and GND (VSS, pin8 or pin15).
- Make sure that the encoder is a 5V type. If it is not a 5V type, proper modification is required.
- Disable Z_pulse by connecting Z+ to GND and Z- to 5V.

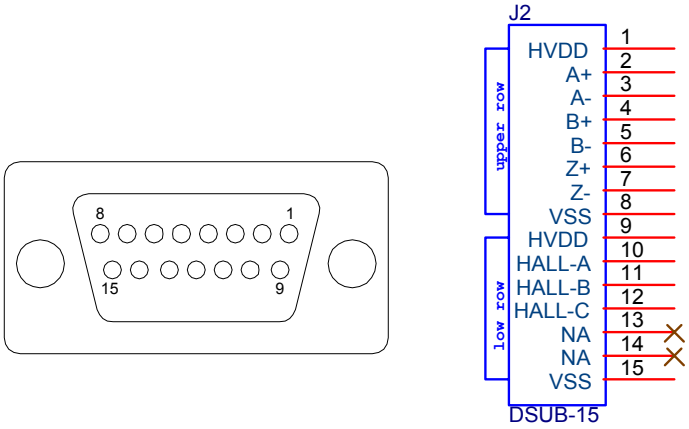


Figure 18. Encoder Interface Connector, J2

3.3.2 Step 2 - Encoder A/B Quadrature Polarity Establishment

- **Do not** connect motor U/V/W leads to Terminal Block J1 yet.
- Plug the encoder connector into J2.
- Rotate the motor shaft by hand in a counter-clockwise direction at the motor front end (facing the motor shaft).
- Using an oscilloscope, monitor the encoder A and B signals at test points 'ENC-A' and 'ENC-B' respectively.
- Verify that the B signal leads the A signal (see Figure 19)
- If A signal leads B signal, swap A+ with B+ and A- with B- by soldering wires on the connector pins.

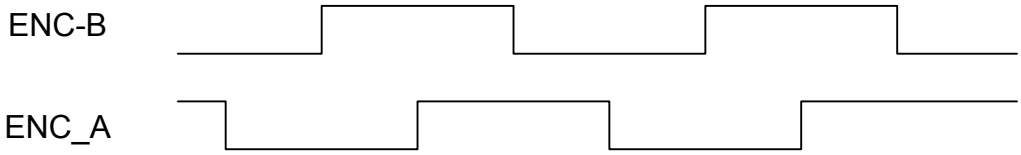


Figure 19. Sample Encoder A and B Signal Timing

3.3.3 Step 3 Motor Phase sequencing

- Unplug the encoder connector first.
- Connect the motor leads to U/V/W pins of terminal block J1.
- Run the ServoDesigner tool. (For detailed instructions, refer to the document "ServoDesigner User's Guide".)
- Open the configuration file "IRMCS201-P14A5805P.irc".
- Select Save As… from the File menu to create a new file using the name of the motor under test (your new motor).
- Execute the Configure Motor function by clicking the toolbar button (hammer and wrench icon) or double clicking the Configuration Motor function in the tree view (as shown in Figure 20).
- Execute the V/Hz Operation function by double clicking the name of the function in the tree view. The function will cause the motor shaft to turn. (The shaft may not turn if load is coupled. This step must be executed in a no load condition.)

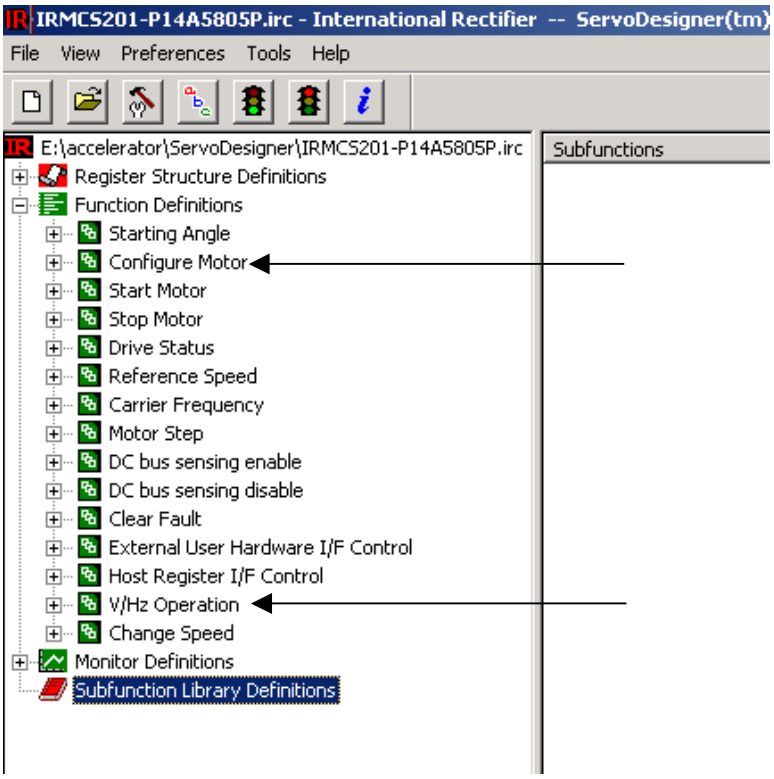


Figure 20. ServoDesigner Functions

- Verify the rotation direction. If it's counter-clockwise, go to Step 4.
- If the direction of rotation is clockwise, swap V with W and repeat Step 3.

3.3.4 Step 4 – Current Feedback Scaling

Id scale factor and **Iq scale factor** need to be separately adjusted for an induction motor application. These two scaling factors are derived by different equations and therefore are usually different values. The IRMCx201 design is

based on the mapping of rated current (continuous current) to be ±4095 digital values regardless of the motor current rating. The current control therefore regulates ±4095 as 100% rated torque for torque producing current, "Qi". If the motor has 300% overload capability, then the control regulates a maximum of ±12,287. The field component current, "Di" is also regulated 100% field flux with 4095 digital value regardless of the motor physical current or amount of magnetizing current.

If you use the Excel workbook, the Id and Iq scale factors are calculated for you. The following paragraphs describe how these values are determined.

In order to find Id scale factor and Iq scale factor, both torque producing current and magnetizing current need to be determined at a rated condition.

Motor nameplate data typically shows the rated current and either no load current or magnetizing current. Torque producing current can be derived from the following equation (all units are in rms):

$$Torque\ producing\ Current\ [TPC] = \sqrt{(rated\ current)^2 - (no\ load\ current[NLC])^2}$$

Then, iq scale factor can be derived from the equation:

$$Iq\ scale\ factor = \frac{MHC \times 4095 \times 1024}{TPC \times 2047 \times \sqrt{2} \times 1.647 \times .82}$$

Similarly, id scale factor can also be derived from the equation:

$$Id\ scale\ factor = \frac{MHC \times 4095 \times 1024}{NLC \times 2047 \times \sqrt{2} \times 1.647 \times .82}$$

where .82 is maximum modulation index of IR2175 and 1.647 is gain scale of the vector rotator in the current feedback path.

Example:

Motor rated current (RC)	±6.0 Arms / ±8.46 Apeak
Motor overload current:	±9.0 Arms / ±12.7 Apeak
No load current (NLC):	±3.8 Arms / ±5.37 Apeak
Maximum Hardware Current:	±13 Apeak

TPC = sqrt($6^2 - 3.8^2$) = sqrt(21.56) = 4.64 Arms
Iq scale factor = 13 x 4095 x 1024 / (4.64 x 2047 x 1.414 x 1.647 x .82) = 3004
Id scale factor = 13 x 4095 x 1024 / (3.8 x 2047 x 1.414 x 1.647 x .82) = 3669

3.3.5 Step 5 – Slip Gain adjustment, Current controller PI gain adjustment

The slip gain needs to be adjusted for induction machine application. Figure 5 shows the slip gain block diagram.

If you use the Excel workbook, the slip gain is calculated for you. The calculation requires the rated RPM information on the motor nameplate data and the IRMCx201 closed loop current control update rate information.

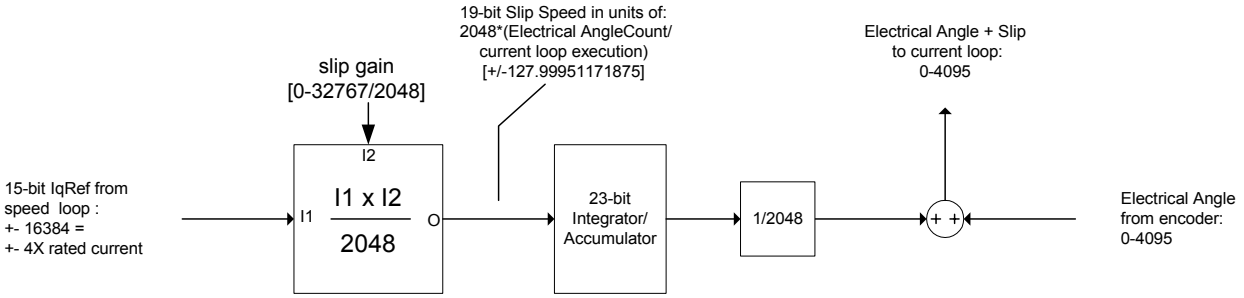


Figure 21. Slip Gain Block Diagram

The following equation is used to derive the slip gain:

$$Rated\ Hz = \frac{\frac{120 \times rated\ frequency}{polenumber} - rated\ RPM}{\frac{120 \times rated\ frequency}{pole\ number}} \times rated\ \ frequency$$

$$Internal\ slip\ speed = 2048 \times \frac{Rated\ Hz \times 4096}{current\ loop\ update\ rate}$$

$$slip\ gain = \frac{2048 \times Internal\ slip\ speed}{4096}$$

Example:

Rated RPM:	1725 rpm
Rated frequency:	60 Hz
Pole number:	4
Current loop update rate:	10 kHz

Rated Hz = ((120 * 60 / 4) – 1725) *60 / (120 * 60 / 4) = 2.5Hz
Internal slip speed = 2048 * 2.5 * 4096 / 10000 = 2097
Slip gain = 2097 * 2048 / 4096 = 1048

3.3.6 Step 6 – Encoder Configuration

If you are using the Excel workbook, you can skip the remaining steps (steps 6 – 8). Follow the instructions in the "Drive Parameter Setup" section instead.

In ServoDesigner, modify the maximum encoder counter value in the Configure Motor function. Expand the Configure Motor function in the tree view and locate the MaxEncCnt register entry. Right click on MaxEncCnt and select Properties. In the Register Value field of the Properties dialog, enter the maximum encoder counter value as follows:

MaxEncCnt = EncoderPPR * 4 – 1

Modify the encoder angle scale factor in the Configure Motor function. Right click on the EncAngScl register entry in the Configure Motor function and select Properties. In the Register Value field of the Properties dialog, enter the encoder angle scale factor value as follows:

EncAngScl = Number of Poles / 2 * 4096 * 4096 / (EncoderPPR * 4)

3.3.7 Step 7 – Speed Feedback Scaling, Speed controller PI gain adjustment

In ServoDesigner, modify the speed scale factor in the Configure Motor function. Expand the Configure Motor function in the tree view and locate the SpdScl register entry. Right click on SpdScl and select Properties. In the Register Value field of the Properties dialog, enter the speed scale factor value as follows:

SpdScl = 60 * 16383 * 33.333/32 *10^6 / (EncoderPPR * MaxRPM)

3.3.8 Step 8 – Save Values in ServoDesigner

In ServoDesigner, select Save from the File menu to save your changes to the Configure Motor function.

3.4 Speed-Controlled Servo Motor Initialization and Operation

This section describes the register-controlled configuration and operation for an example system that uses an IRMCx201 series FPGA with a host microprocessor, IRAM intelligent IGBT module, IR2175 current sense IC, and a Sanyo-Denki P30B06040DXS00M servomotor. The system is illustrated in Figure 22.

The IRMCS201 FPGA interfaces with a host microprocessor through a set of host registers, which are divided into write (control) and read (status) registers. The host microprocessor configures the IRMCx201 host write registers at power-on, and then uses the appropriate host write and read register fields for motor start, stop, and fault processing. For this example, we have assumed that the user is using the IRMCx201 internal torque loop and speed loop to control motor speed.

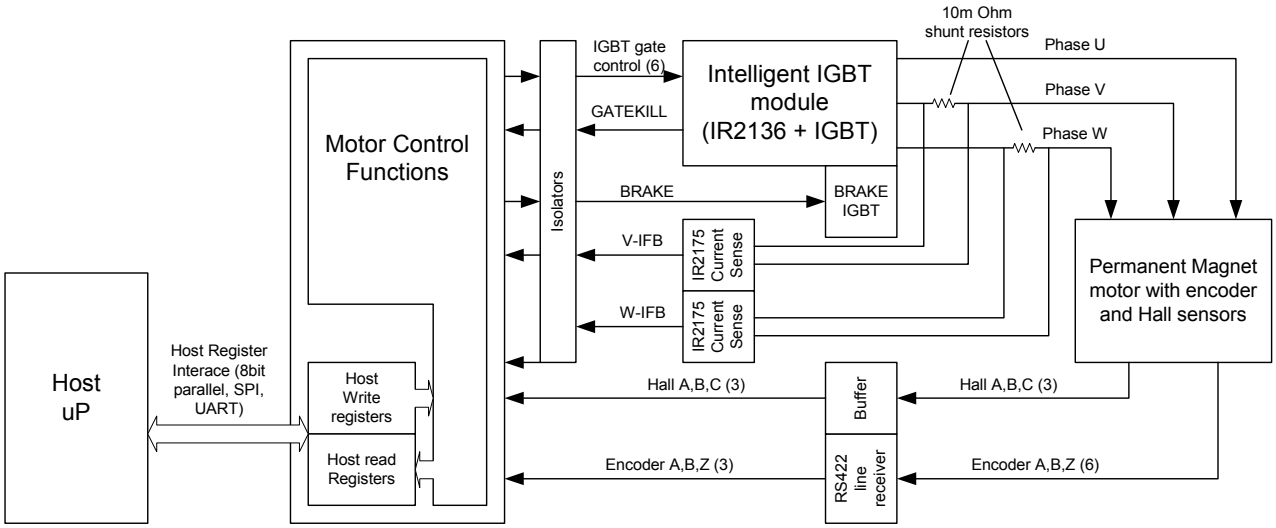


Figure 22. Host Microprocessor Controlled IRMCx201 System

3.4.1 Initialization

Table 8 gives IRMCX Motion Control Processor host write register initialization values for this example. These registers must be initialized at power-on before starting the motor. A brief description is provided for each field. Where noted, more detailed information is provided in the paragraphs following the table.

Field Name	Reset Value	Initial Value	Comment
Quadrature Decode Register Group			
EncCntW	0	-	Encoder Count Write Register. See description below for Hall Sensor initialization procedure.
MaxEncCnt	0	0x1F3F 7999	Maximum Encoder Count Value.[1]
ZEncCnt	0	0x158 (344)	Z-pulse Encoder Count Value.[1]
EncAngScl	0	8388 0x20C4	Encoder Angle Scale Factor.[1]
PwrOnRedSig	0	0	Power-on Reduced Signal Enable, only used for EEPROM standalone implementations.
ZpulsePol	0	1	Encoder Z-pulse polarity.
ZpulseEnb	0	1	Enable Z-pulse load from ZencCnt.

Field Name	Reset Value	Initial Value	Comment
CntEnb	0	1	Enable encoder counter.
PWM Configuration Register Group			
GatekillSns	0	0	IR2136 Gatekill signal sense, 0 = active low.
GateSnsL	0	0	IR2136 low side IGBT gate control sense, 0 = active low.
GateSnsU	0	0	IR2136 high side IGBT gate control sense, 0 = active low.
SD	0	1	Shutdown control (active low) to gate drive IC.
PwmPeriod	0	1666 0x682	Divider for 10Khz PWM.[1]
PwmDeadTm	0	17 0x11	Divider for 17 * .030 = .51 usec.
PwmConfig	0	1	Symmetrical, center-aligned PWM.
Current Feedback Configuration Register Group			
IdScl	0	4925 0x133D	Rotating frame "D" axis feedback current scale factor.[1]
IqScl	0	4925 0x133D	Rotating frame "Q" axis feedback current scale factor.[1]
System Control Register Group			
DcCompEnb	0	1	DC bus compensation enable. Setting this bit causes the current regulator output voltage to be automatically compensated for DC-bus voltage fluctuations.
IfbOffsEnb	0	1	Automatic current feedback offset application enable[1]
FocEnbW	0	0	FOC closed-loop control enable[2]
PwmEnbW	0	0	PWM output enable[2]
Torque Loop Configuration Register Group			
IdRef	0	0	Direct/Magnetizing axis reference current to current-loop PI controller. Used with external user velocity loop.
IqRef	0	0	Quadrature axis reference current to current-loop PI controller. Used with external user velocity loop.
KpIreg	0		Current regulator proportional gain
KxIreg	0		Current regulator integral gain
SlipGn	0	0	Slip-gain for induction motor implementation
Vdlim			Current regulator D-axis output limit
Vqlim			Current regulator Q-axis output limit
Velocity Control Register Group			
KpSreg	0		Speed regulator proportional gain
KxSreg	0		Speed regulator integral gain
SregLimP	0		Speed regulator output positive limit
SregLimN	0		Speed regulator output negative limit
SpdScl	0		Speed calculation scale factor
TargetSpd	0		Target motor speed
SpdAccRate	0		Motor acceleration rate
SpdDecRate	0		Motor deceleration rate
Fault Control Register			
FltClr	0	0	Fault clear register[2]
DcBusMEnb	0	1	Dc bus voltage monitor enable
SVPWM Scaler Register			
ModScl	0	5020 0x139C	SVPWM modulation scale factor. Set this value to PwmPer*sqrt(3)*4096/2355
Diagnostic PWM Control Register Group			

Field Name	Reset Value	Initial Value	Comment
PwmData0Sel	0	0	Diagnostic DAC Pwm #0 Data Select[2]
PwmData1Sel	0	0	Diagnostic DAC Pwm #1 Data Select[2]
PwmData2Sel	0	0	Diagnostic DAC Pwm #2 Data Select[2]
PwmData3Sel	0	0	Diagnostic DAC Pwm #3 Data Select[2]
System Configuration Register Group			
RmpRefSel	0	0	Speed ramp reference select.
HostVdEnb	0	0	Direct host D-axis voltage control enable
HostAngEnb	0	0	Direct host DQ voltage axis angle control enable
IqRefSel	0	0	Current regulator reference input select
SpdRefSel	0	0	Speed regulator reference input select
ExtCtrl	0	0	External user interface control enable (no-host standalone mode)
Direct Host Voltage Control Register Group			
VdSfwd	0	0	Host-controlled D-axis voltage.
VqSfwd	0	0	Host-controlled Q-axis voltage.
ElecAngW	0	0	Host-controlled Electrical Angle.
32-bit Quadrature Decoder Register Group			
EncCnt32bW	0	0	32-bit Encoder count register, for position control
EEPROM Control Register Group			
EeWrite	0	0	
EeRead	0	0	
EeRst	0	0	
EeAddr/RegMapCode	0	0	
EeDataW	0	0	
Hall Sensor Encoder Initialization Group (only for standalone mode)			
HallCBA001	0	0	
HallCBA010	0	0	
HallCBA011	0	0	
HallCBA100	0	0	
HallCBA101	0	0	
HallCBA110	0	0	

Table 8. IRMCx201 Write Register Initialization Values

Notes:

1. See description below for more information
2. These fields are also changed dynamically during motor operation. See description below for more details.

Encoder Maximum Count (MaxEncCnt)

When the encoder count reaches this value, the next count pulse resets the count register to 0. The encoder count tracks rotor angle, which is used for current loop d-q axis transformations. This register must be set to the count that corresponds to a 360-degree physical angle, which in this case is (2000 lines * 4) – 1 = 7999.

Encoder Z-pulse Count (ZEncCnt)

The encoder Z-pulse count specifies a count that is automatically loaded into the encoder count register each time the Z-pulse asserts. This action aligns the encoder count with the rotor angle so that the encoder can be used as an angle sensor. The Z-pulse value should be set such that the encoder count is 0 when the rotor angle is 0 relative to the stator U-phase MMF. For the 8-pole motor in this example, the Z-pulse occurs at an electrical angle of 62 degrees (mechanical angle of 15.5 degrees), so the ZEncCnt field should be set to 8000*15.5/360 = 344. Figure 23 shows the relationship between encoder count, Z-pulse count, physical angle, and electrical angle for this example. The Z-pulse count value can be determined either experimentally or from the motor manufacturer's data.

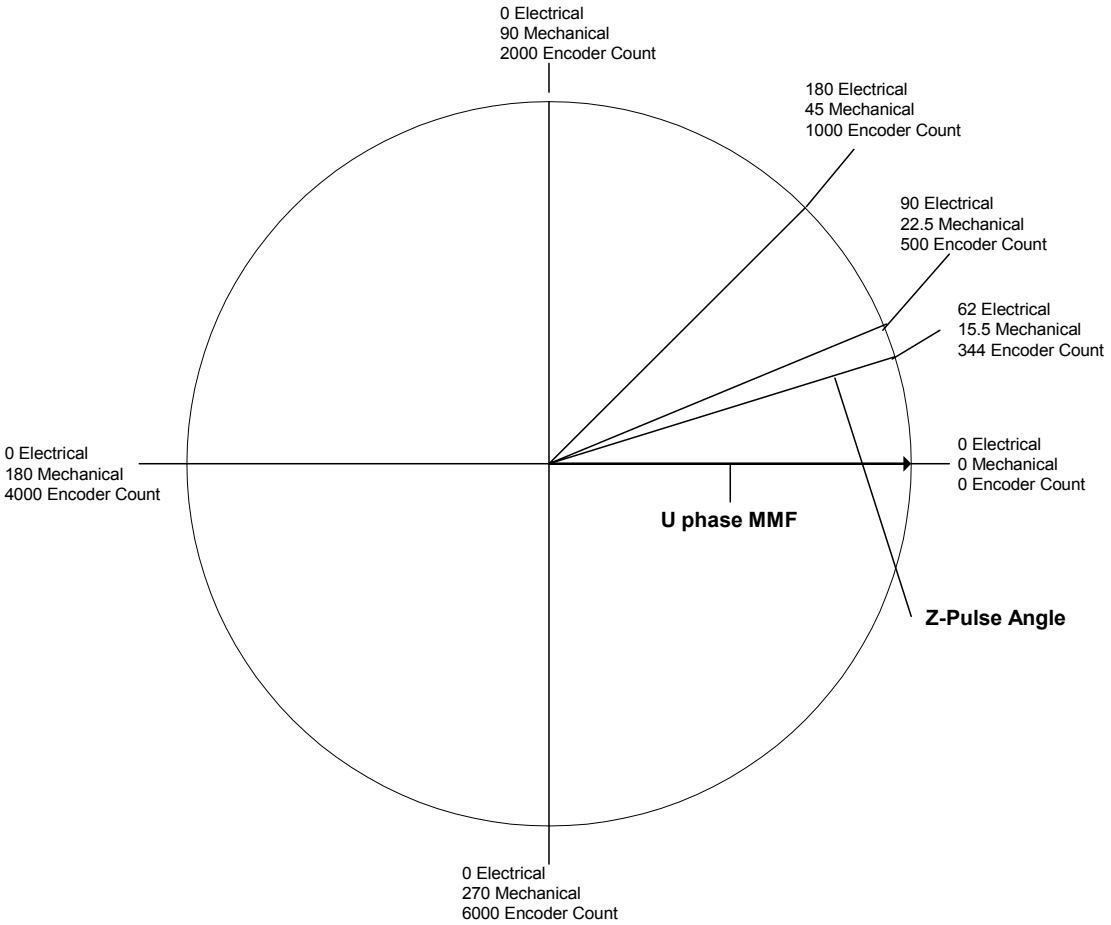


Figure 23. Motor Angle Relationships

Encoder Count Initialization via Hall Sensors (EncCntW)

At system power-up, the encoder count will not give the correct rotor angle representation until the first Z-pulse occurs. The host uses the Hall ABC sensors to approximate initial rotor position (to within 60 degrees) so that it can rotate the motor before the first Z-pulse. Figure 24 shows the Hall ABC values and the corresponding initial electrical angle and encoder count. During system initialization the host microprocessor must read the Hall ABC values from the Quadrature Decode Status Register Group Hall A,B,C register fields (PwrOnHall A,B,C fields for a wire-saving encoder) and use a lookup table to load the appropriate encoder count. The Hall ABC angle data can be determined either experimentally or from the motor manufacturer's data.

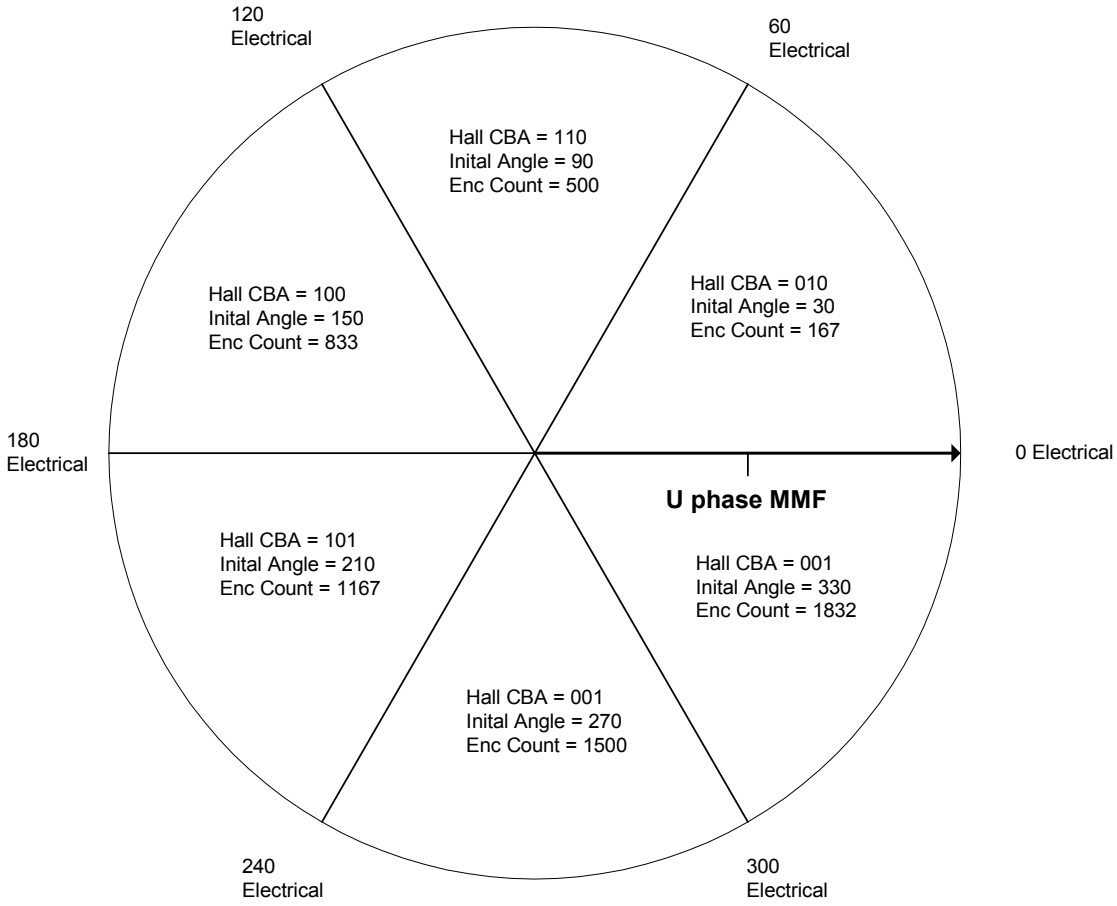


Figure 24. Hall Sensor and Initial Angle Values

Encoder Angle Scale Factor (EncAngScl)
The IRMCx201 FOC algorithm performs transformations to and from the rotating d-q frame using a 12-bit electrical angle with 0-4095 corresponding to angles from 0 – 359.9. The electrical angle is determined from the encoder count and the Encoder Angle Scale factor using the following equation:

ElecAng = ((MtrPoles / 2) * 4096 * (EncCnt) / (MaxEncCnt + 1)) MOD 4096

Where:
ElecAng is the motor electrical angle.
MtrPoles is the number of poles in the motor.
EncCnt in the encoder count.
MaxEncCnt is value in the the maximum encoder count field.

The EncAngScl parameter should be set to (MtrPoles / 2) * 4096 * 4096 / (MaxEncCnt + 1) so that the above operation can be performed in hardware with a single scaled multiplication as follows:

Electrical angle = EncAngScl * EncCnt / 4096

For this example EncAngScl = 4 * 4096 * 4096 / 2000 = 8388

Pwm Initialization (PwmConfig, PwmPeriod, PwmDeadTm)
Before starting the motor, the host must initialize the PWM parameters for the desired mode, frequency and dead time. For this example we use 10 kHz Symetrical center aligned PWM with a .5 us dead time, so that PwmConfig = 1, PwmPer = 33.333Mhz / (2 * 10 kHz) – 1 = 1666, PwmDeadTm = 17.

Current Feedback Scale Factors (IdScl, IqScl)

Figure 25 shows the calculation scaling for the IRMCx201 FOC current feedback path. The current feedback scale factors should be selected to scale the rotating frame d-q current so that it ranges from –4095 to +4095 at the input of the current loop PI controller summing junction. For this example, we are using a 10-milliohm current sense resistor with a rated motor current of 2.7 Amps or 3.82 Amps Peak. Referring to the figure, this means that the 15-bit programmable scale factor should be set to 4095 / (2900 * 3.82 / 13) = 4.81. Since the scale factors have 10 fractional bits, the final values for the IdScl and IqScl fields are 4.81 * 1024 = 4925. Please refer to the IR2175 datasheet for more information on IR2175-based current sensing.

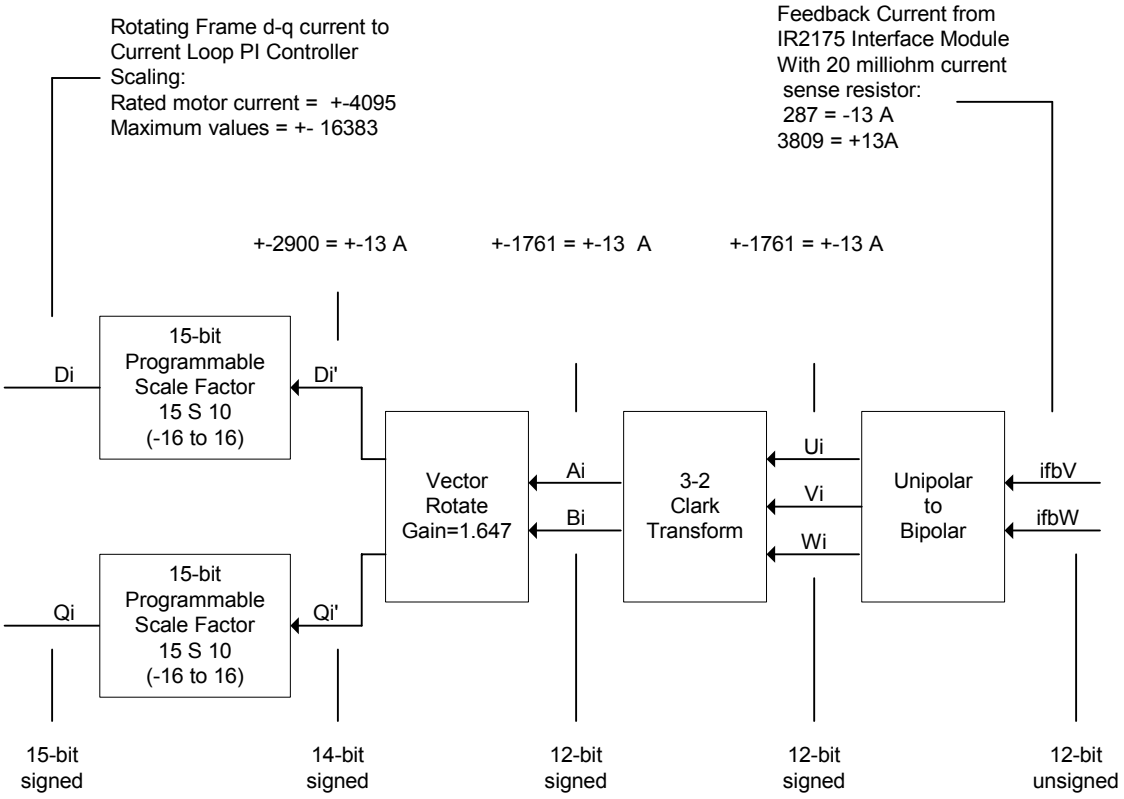


Figure 25. Current Feedback Scaling

3.4.2 Current Offset Calculation

Before running the motor, the IR2175 current measurement offset must be determined so that a correction can be applied to the current feedback before it is used in the FOC algorithm. To determine the IR2175 current measurement offset, we measure the V and W phase currents when the motor is stationary with no applied voltage. In this example, however, we cannot make a valid measurement when the phase voltage is completely off because the IR2175 receives its power from the motor line voltage via bootstrap capacitor. The IRMCx201 works around this by recognizing the condition where PwmEnbW = 1 and FocEnbW = 0, in which case it applies PWM with 0% duty cycle. With 0% PWM, the high side IGBT gate drivers are off, and the low side IGBT gate drivers are on most of the time, but switch off during the dead-time period on each PWM cycle. This short off period drives the IR2175 bootstrap power circuit, which allows a valid current measurement.

When the IRMCx201 detects the current offset calculation condition (PwmEnbW = 1, FocEnbW = 0), it automatically averages 4096 consecutive IR2175 current feedback measurements and stores the resulting current offsets in the IfbVOffs and IfbWOffs read registers. Since the IR2175 performs a current measurement every 1/120Khz = 8.33 usec, each offset calculation takes about 35 msec. In order to apply these offsets to the current feedback measurements, the host sets the IfbOffsEnb bit to "1". The host can also read the IfbVOffs and IfbWOffs registers for diagnostic purposes.

The host can also perform its own current offset calculations by reading the IvFbk and IwFbk registers and writing the desired offsets to the IfbOffsV and IfbOffsW write registers (note that these are different from the IfbVOffs and IfbWOffs read registers). When the IfbOffsEnb bot is set to "0", the IRMCx201 applies these values to the current feedback measurements instead of the automatically calculated values.

Regardless of the method used to determine current offset, the user's system must provide a mechanism for generating the calculation offset condition with the motor stationary and ensure the generation of valid offsets. The IR ServoDesigner tool, for example, accomplishes this each time the "Start Motor" command is executed as follows:

1. Set PwmEnbW = 1, FocEnbW = 0
2. Delay 100ms
3. Set PwmEnbW = 1, FocEnbW = 1 (starts the motor running)

3.4.3 Starting and Stopping the Motor

After initial configuration is complete, the host starts and stops the motor using the PwmEnbW and FocEnbW bits in the System Control Register. As an example, the following "Start Motor" sequence runs the motor at 2000 rpm using the acceleration rate that was set at initial configuration:

1. Set TargetSpeed = 7281 (16383 * 2000 / 4500)
2. Set PwmEnbW = 1, FocEnbW = 0 (start Ifb Offset calculation)
3. Delay 100ms (two IfbOffset calculations)
4. Set PwmEnbW = 1, FocEnbW = 1, IfbOffsEnb = 1, DcCompEnb = 1 (starts the motor running with auto-generated current offset and DC bus compensation)

To stop the motor, the host can use one of the following two methods:

1. Disable PWM and let the motor coast to a stop by setting PwmEnbW = 0 and FocEnbW = 0.
2. Bring the motor to a controlled stop using the deceleration ramp by setting TargetSpd = 0, monitoring the motor speed until it is near 0, and then setting PwmEnbW = 0 and FocEnbW = 0.

3.4.4 Emergency Stop

As a safety feature, the external user interface "Stop" pin can be asserted to disable PWM immediately at any time regardless of the IRMCx201 configuration. See Section 3.5 for more information on the external user interface.

3.4.5 Fault Handling

The IRMCx201 detects the several fault conditions that can occur during system operation. When a fault condition occurs, the IRMCx201 sets the fault status register and leaves PWM disabled until the host clears the fault condition and re-enables PWM and Field-Oriented-Control (FOC). An illustration of typical fault/fault-clear sequence is as follows:

1. When a fault condition occurs, the IRMCx201 clears the PwmEnbW, PwmEnbR, FocEnbW and FocEnbR bits, sets the appropriate bit in the fault status register, and turns on the red LED output pin. At this point the drive is disabled.
2. The host microprocessor, which has presumably been monitoring the FltStatus read register, detects the fault condition and initiates whatever operator alert is appropriate for the particular application. Note that the external LED will also light red if it is connected to the IRMCx201 red LED pin.
3. To clear the fault, the host microprocessor sets the FltClr write bit to "1" to clear the fault and then back to "0" to re-enable fault detection.

3.5 Standalone Operation and Register Initialization via Serial EEPROM

This section describes the register-controlled configuration and operation for an example system that uses the IRMCx201 system in standalone mode, which requires no initialization by a host microprocessor. In standalone mode, the IRMCx201 initializes the host write registers from an I^2C serial EEPROM at power up and receives control commands from the external hardware user interface signals during motor operation. The system described in this example is shown in Figure 26.

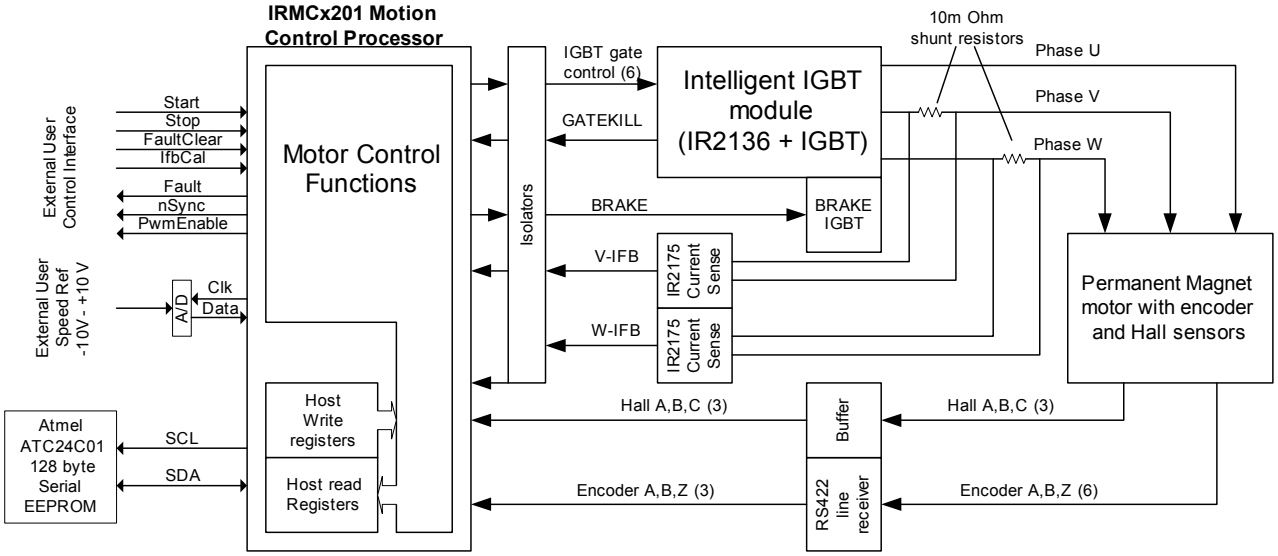


Figure 26. IRMCx201 Standalone System

3.5.1 Register Initialization via EEPROM

Each time the IRMCx201 powers up, it checks for valid EEPROM data by reading a single byte from EEPROM address 0x5D, which represents the IRMCx201 register map version code. If this value matches the IRMCx201 internal register map version, the IRMCx201 EEPROM initialization sequencer performs the following operations:

1. Read 128 sequential bytes from the EEPROM and store them in host write registers 0 - 0x7F.
2. Use the initial hall sensor values as an index to the hall initialization encoder values stored at EEPROM addresses 0x72-0x7D, and read the initial encoder value.
3. Write the initial encoder position to the EnCntW field to set the initial rotor angle.

If the user sets the location in the EEPROM that corresponds to the SystemConfig register group's ExtCtrl field to "1", motor operation can be controlled directly using the external user interface immediately after power-on host write register initialization.

3.5.2 Current Offset Calculation

When operating the IRMCx201 in standalone, the user initiates current offset calibration by asserting the IfbCal signal for a period of at leas 100ms. Before starting the motor, the user should assert IfbCal at least once while the motor is stationary. IfbCal assertion causes the condition PwmEnbW = 1, FocEnbW = 0 as described in the preceding section. The user can perform periodic offset calibration by asserting IfbCal as required by the particular application.

3.5.3 Starting and Stopping the Motor

To start or stop the motor in standalone mode, the user simply drives the "Start" or "Stop" signal to a logic "1". When both signals are asserted, the "Stop" signal has precedence. These signals are intended to be driven as pulsed signals with a minimum pulsewidth of 100 nsec. Note that driving "Stop" to a DC logic "1" disables the motor unconditionally, so that the "Stop" signal can also be used as an emergency kill switch.

3.5.4 Fault Processing

When the IRMCx201 detects a fault condition, it disables PWM and asserts the "Fault" signal. In standalone mode, the user clears the fault condition using the "FaultClear" signal. Fault processing is otherwise identical to that described Section 3.4.5.

4 Reference

This section provides reference material for the IRMCx201 products. It includes a description of the host register interface protocol with examples and a complete listing of the host registers in sequential order, with a short description of each field.

4.1 Register Access

A host computer controls the FPGA using either its slave-mode Full-Duplex SPI port, or a standard RS-232 port. Both interfaces are always active and can be used interchangeably, although not simultaneously. Control/status registers are mapped into a 128-byte address space.

4.1.1 SPI Register Access

When configured as an SPI device read only and read/write operations are performed using the following transfer format:

Command Byte	Data Byte 0	.....	Data Byte N

Data Transfer Format

Bit Position							
7	6	5	4	3	2	1	0
Read Only	Register Map Starting Address						

Command Byte Format

Data transfers begin at the address specified in the command byte and proceed sequentially until the SPI transfer completes. Note that accesses are read/write unless the "read only" bit is set.

4.1.2 RS-232 Register Access

The FPGA includes an RS-232 interface channel that provides a direct connection to the host PC. The software interface combines a basic "register map" control interface with a simple communication protocol to accommodate potential communication errors.

RS-232 Register Write Access

A Register write operation consists of a command/address byte, byte count, register data and checksum. When the FPGA receives the register data, it validates the checksum, writes the register data, and transmits and acknowledgement to the host.

Command / Address Byte	Byte Count	1-6 bytes of register data	Checksum

Register Write Operation

Command Acknowledgement Byte	Checksum

Register Write Acknowledgement

Bit Position							
7	6	5	4	3	2	1	0
1=Read/ 0=Write	Register Map Starting Address						

Command/Address Byte Format

Bit Position							
7	6	5	4	3	2	1	0
1=Error/ 0=OK	Register Map Starting Address						

Command Acknowledgement Byte Format

The following example shows a command sequence sent from the host to the FPGA requesting a two-byte register write operation:

0x2F	Write operation beginning at offset 0x2F
0x02	Byte count of register data is 2
0x00	Data byte 1
0x04	Data byte 2
0x35	Checksum (sum of preceding bytes, overflow discarded)

A good reply from the FPGA would appear as follows:

0x2F	Write completed OK at offset 0x2F
0x2F	Checksum

An error reply to the command would have the following format:

0xAF	Write at offset 0x2F completed in error
0xAF	Checksum

RS-232 Register Read Access

A register read operation consists of a command/address byte, byte count and checksum. When the FPGA receives the command, it validates the checksum and transmits the register data to the host.

Command / Address Byte	Byte Count	Checksum

Register Read Operation

Command Acknowledgement Byte	Register Data (Byte Count bytes)	Checksum

Register Read Acknowledgement (transfer OK)

Command Acknowledgement Byte	Checksum

Register Read Acknowledgement (error)

The following example shows a command sequence sent from the host to the FPGA requesting four bytes of read register data:

0xA0	Read operation beginning at offset 0x20 (high-order bit selects read operation)
0x04	Requested data byte count is 4
0xA4	Checksum

A good reply from the FPGA might appear as follows:

0x20	Read completed OK at offset 0x20
0x11	Data byte 1
0x22	Data byte 2
0x33	Data byte 3
0x44	Data byte 4
0xCA	Checksum

An error reply to the command would have the following format:

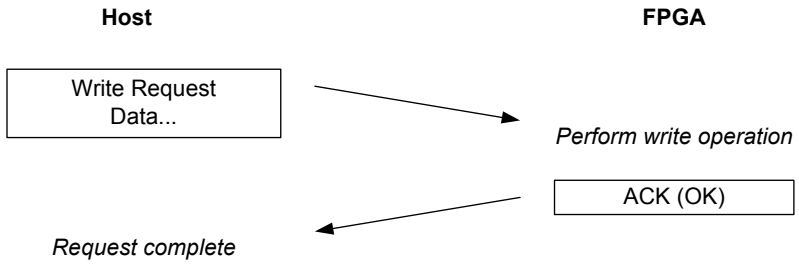
0xA0	Read at offset 0x20 completed in error
0xA0	Checksum

RS-232 Timeout

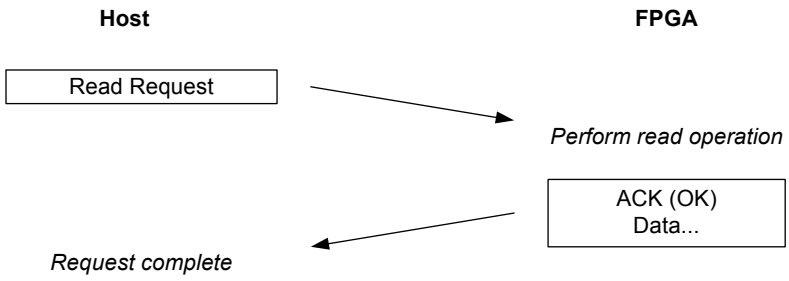
The FPGA receiver includes a timer that automatically terminates transfers from the host to the FPGA after a period of 32 msec.

RS-232 Transfer Examples

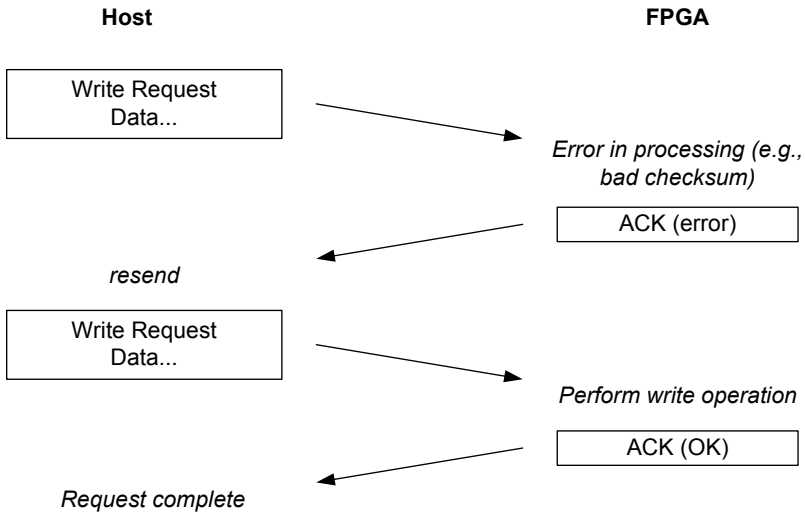
The following example shows a normal exchange executing a register write access.



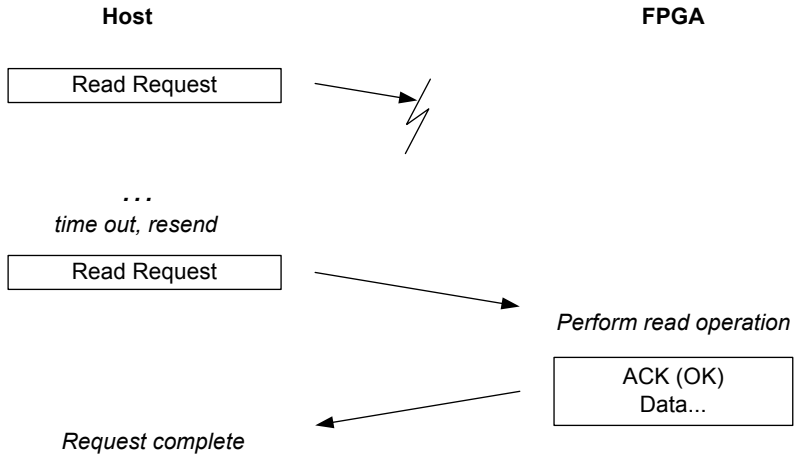
The example below shows a normal register read access exchange.



The following example shows a register write request that is repeated by the host due to a negative acknowledgement from the FPGA.



In the final example, the host repeats a register read access request when it receives no response to its first attempt.



4.2 Write Register Definitions

4.2.1 QuadratureDecode Register Group (Write Registers)

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x0	EncCntW (LSBs) (W)							
0x1	EncCntW (MSBs) (W)							
0x3	MaxEncCnt (LSBs) (W)							
0x4	MaxEncCnt (MSBs) (W)							
0x6	ZEncCnt (LSBs) (W)							
0x7	ZEncCnt (MSBs) (W)							
0x9	EncAngScl (LSBs) (W)							
0xA	EncAngScl (MSBs) (W)							
0xB	SPARE				PwrOn RedSig (W)	ZPulse Enb (W)	ZPulsePol (W)	CntEnb (W)

QuadratureDecode Write Register Map

Field Name	Access (R/W)	Field Description
EncCntW	W	New value for 16-bit Quadrature Decoder counter.
MaxEncCnt	W	Maximum value of 16-bit Quadrature Decoder counter. The encoder count is reset to 0 after this count has been reached. This maximum should be set to correspond to a 360-degree physical angle.
ZEncCnt	W	Encoder count value when the Z-pulse occurs. This value is loaded automatically in hardware when the Z-pulse occurs. (See ZPulseEnb and ZPulsePol fields below.)

Field Name	Access (R/W)	Field Description
EncAngScl	W	This value should be set to ((MtrPoles / 2) * (4096 * 4096) / (MaxEncCnt + 1), where MtrPoles is the number of motor poles. The value has the range 0 – 32767 and is used to convert the encoder count to an angle ranging from 0 - 4095 using the equation: Angle = ((MtrPoles / 2) * 4096 * (encoder count) / (MaxEncCnt + 1)) MOD 4096. (The current encoder count can be read from the EncCntR feld of the QuadratureDecodeStatus read register group.)
CntEnb	W	Encoder counter enable.
ZPulsePol	W	ZPULSE polarity. 1= load ZEncCnt on rising Z-pulse edge. 0= load ZEncCnt on falling Z-pulse edge.
ZPulseEnb	W	ZPULSE count initialization enable. When this bit is set, the encoder count is set to the ZEncCnt value at each Z-pulse edge as determined by the ZPulsePol field.
PwrOnRedSig	W	PowerOn Reduced signal enable. Set this bit in the EEPROM to enable EEPROM standalone initialization for a wire-saving encoder. When this bit is set, the EEPROM initialization uses the PwrOnHallA, PwrOnHallB, PwrOnHallC bits instead of the HallA, HallB, HallC bits to determine initial motor angle. (The Hall bits can be read from the QuadratureDecodeStatus read register group.)

QuadratureDecode Write Register Field Definitions

4.2.2 PwmConfig Register Group (Write Registers)

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0xC	Gatekill Sns (W)	SPARE	Gate SnsL (W)	Gate SnsU (W)	SyncSns (W)	BrakeSns (W)	SD (W)	SPARE
0xD	PwmPeriod (LSBs) (W)							
0xE	SPARE		PwmConfig (W)		PwmPeriod (MSBs) (W)			
0xF	PwmDeadTm (W)							

PwmConfig Write Register Map

Field Name	Access (R/W)	Field Description
SD	W	Shutdown control output to IR2137.
BrakeSns	W	Logic Sense for BRAKE signal output to gate driver IC. 0 = Active low, 1 = active high.
SyncSns	W	Logic Sense for PWM SYNC signal output to microprocessor. 0 = Active low, 1 = active high.
GateSnsU	W	Upper IGBT gate sense. 1 = active high gate control, 0 = active low gate control.
GateSnsL	W	Lower IGBT gate sense. 1 = active high gate control, 0 = active low gate control.
GatekillSns	W	GATEKILL signal sense. 1 = active high GATEKILL, 0 = active low GATEKILL.
PwmPeriod	W	This field is used to set the desired PWM frequency using the following equation: PwmPeriod = 33,333,333 / (2 * (PWM frequency)) – 1, where 33,333,333 is the system clock frequency (33.333Mhz). Note that while "PwmPeriod" is the name of this field, the actual PWM carrier period is 2 * (PwmPeriod + 1) * (System Clock Period = 30ns).
PwmConfig	W	PWM Configuration. 0 = Asymmetrical center aligned PWM, 1 = Symmetrical Center aligned PWM.
PwmDeadTm	W	Gate drive dead time in units of system clock cycles (e.g., 30 ns with 33 MHz clock).

PwmConfig Write Register Field Definitions

4.2.3 CurrentFeedbackConfig Register Group (Write Registers)

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x10	IfbOffsV (LSBs) (W)							
0x11	IfbOffsW (LSBs) (W)				IfbOffsV (MSBs) (W)			
0x12	IfbOffsW (MSBs) (W)							
0x13	IdScl (LSB) (W)							
0x14	IdScl (MSB) (W)							
0x15	IqScl (LSB) (W)							
0x16	IqScl (MSB) (W)							

CurrentFeedbackConfig Write Register Map

Field Name	Access (R/W)	Field Description
IfbOffsV	W	12-bit signed value for V phase current feedback offset. When the IfbOffsEnb bit in the SystemControl write register group is "0" this value is automatically added to each current measurement in hardware.
IfbOffsW	W	12-bit signed value for W phase current feedback offset. . When the IfbOffsEnb bit in the SystemControl write register group is "0" this value is automatically added to each current measurement in hardware.
IdScl	W	Rotating frame Id component current feedback scale factor. Constant used to scale current measurements before they are used in the field orientation calculation. This is a 15-bit fixed-point signed number with 10 fractional bits that ranges from –16 to + 16 + 1023 / 1024.
IqScl	W	Rotating frame Iq component current feedback scale factor. Constant used to scale current measurements before they are used in the field orientation calculation. This is a 15-bit fixed-point signed number with 10 fractional bits that ranges from –16 to + 16 + 1023 / 1024.

CurrentFeedbackConfig Write Register Field Definitions

4.2.4 SystemControl Register Group (Write Registers)

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x17	DcComp Enb	IfbOffs Enb	SPARE			AdcIfbEnb	Foc EnbW	Pwm EnbW

SystemControl Write Register Map

Field Name	Access (R/W)	Field Description
PwmEnbW	W	PWM Enable bit. Setting this bit to 1 or 0 sets the IGBT gate control signals to their active or inactive states. At power up the gate control output signals remain in a high-Z state. After PwmEnbW is set for the first time, the gate controls are driven to their active or inactive states according to the value of PwmEnbW. A fault condition clears this bit automatically in hardware.
FocEnbW	W	Field Orientated Control Enable bit. Setting this bit to 1 enables the FOC algorithm. Setting this bit to 0 resets the FOC algorithm and causes zero output voltage to be applied to the motor. A fault condition clears this bit automatically in hardware.
AdcIfbEnb	W	A/D Converter IFB Enable. Setting this bit to "1" caused current feedback measurement to be take from the ADS7818 A/D converter interface.

Field Name	Access (R/W)	Field Description
IfbOffsEnb	W	When IFB PwmEnbW = 1, and FocEnbW = 0, the Current feedback offset is calculated and saved in the CurrentFeedbackOffset read register group. When IfbOffsEnb = 1, the Current feedback offset values in the CurrentFeedbackOffset **Read** registers are applied to each current feedback measurement. When IfbOffsEnb = 0, the Current feedback offset values in the CurrentFeedbackConfig **Write** registers are applied to each current feedback measurement.
DcCompEnb	W	DC Bus Compensation enable. When this bit is set to "1", PWM output is compensated for using the following formula: PWM (comp) = PWM * 310 / DCBUSVOLTS where PWM (comp) is the compensated PWM output voltage; PWM is the uncompensated PWM output voltage; 310 is the nominal DC bus voltage; and DCBUSVOLTS is the actual DC bus voltage.

SystemControl Write Register Field Definitions

4.2.5 CurrentLoopConfig Register Group (Write Registers)

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x18	IqRefW – Quadrature Reference Current (LSBs) (W)							
0x19	IqRefW – Quadrature Reference Current (MSBs) (W)							
0x1A	KpIreg – Current Loop Proportional Gain (LSBs) (W)							
0x1B	KpIreg – Current Loop Proportional Gain (MSBs) (W)							
0x1C	KxIreg – Current Loop Integral Gain (LSBs) (W)							
0x1D	KxIreg – Current Loop Integral Gain (MSBs) (W)							
0x1E	IdRef – Direct/Magnetizing Reference Current (LSBs) (W)							
0x1F	IdRef – Direct/Magnetizing Reference Current (MSBs) (W)							
0x20	SlipGn (LSBs) (W)							
0x21	SlipGn (MSBs) (W)							
0x22	VqLim – Quadrature Current Output Limit (LSBs) (W)							
0x23	VqLim – Quadrature Current Output Limit (MSBs) (W)							

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x26	VdLim – Direct Current Output Limit (LSBs) (W)							
0x27	VdLim – Direct Current Output Limit (MSBs) (W)							

CurrentLoopConfig Write Register Map

Field Name	Access (R/W)	Field Description
IqRefW	W	15-bit signed quadrature current reference input from velocity loop.
KpIreg	W	15-bit signed current loop PI controller proportional gain. Scaled with 14 fractional bits for an effective range of 0 – 1.
KxIreg	W	15-bit signed current loop PI controller integral gain. Scaled with 19 fractional bits for an effective range of 0 - .03125.
IdRef	W	15-bit signed direct/magnetized current to D-axis current loop PI controller.
SlipGn	W	This parameter controls the slip speed for induction motor applications. SlipGn should be set to 2048 * 2048 * (Rated slip speed in Hz) / (Current loop update frequency). SlipGn MUST be set to 0 if slip is not desired.
VqLim	W	16-bit Quadrature current PI controller voltage output limit.
VdLim	W	16-bit Direct current PI controller voltage output limit.

CurrentLoopConfig Write Register Field Definitions

4.2.6 TraceBufferControl Register Group (Write Registers)

The Trace Buffer Control Register group manages the FPGA's diagnostic trace function. The FPGA contains an internal circular trace buffer consisting of 2048 four-byte entries. Each entry consists of two 16-bit items of sampled data: configurable data item "A" and configurable data item "B". Data is read from the trace buffer using the Trace Buffer Status Read Register group.

To use the trace function without the trigger:

1. Write a "1" to the TrcRst bit to reset the trace.
2. Select "A" and "B" data items by writing to the TrcDataASel and TrcDataBSel registers.
3. Specify the number of samples to be collected (1 – 2048) by writing to the PstTrigCnt register.
4. Set TrigEdge to zero.
5. Set the Arm and ForceTrig bits to start collecting data and generate an immediate trigger so collection will stop after PstTrigCnt samples.
6. Read TrcSt (in the TraceBufferStatus read Register group) until its value is 3 (done collecting data).
7. Read TrcDataA and TrcDataB (in the TraceBufferStatus read Register group) 2048 times to retrieve the entire trace buffer. The valid samples are in the last PstTrigCnt buffer entries retrieved. NOTE: To read the data samples properly, you must either read each TrcDataA and TrcDataB together in a single 32-bit operation, or read each TrcDataB first, before reading TrcDataA. This is because the FPGA's internal trace buffer pointer increments to the next sample on a read of TrcDataA (LSBs).

To use the trace with trigger:

1. Write a "1" to the TrcRst bit to reset the trace.

2. Select "A" and "B" data items by writing to the TrcDataASel and TrcDataBSel registers. (The trigger is always based on the data "A" item.)
3. Specify the number of samples to be collected after the trigger occurs (1 – 2048) by writing to the PstTrigCnt register.
4. Set TrigLvl to the value of data item "A" at which you want the trigger to occur.
5. Set TrigEdge to 1 or 2 depending on whether you want the trigger to occur on the falling or rising edge of the data item "A" value.
6. Set the Arm bit to start collecting data. (Do not set the ForceTrig bit.)
7. When the trigger occurs, the value of TrcSt (in the TraceBufferStatus Read Register group) changes to 2. Read TrcSt until its value changes to 3 (done collecting data).
8. Read TrcDataA and TrcDataB (in the TraceBufferStatus read Register group) 2048 times to retrieve the entire trace buffer. The oldest samples are retrieved first, so the samples collected after the trigger occurred are in the last PstTrigCnt buffer entries. NOTE: To read the data samples properly, you must either read each TrcDataA and TrcDataB together in a single 32-bit operation, or read each TrcDataB first, before reading TrcDataA. This is because the FPGA's internal trace buffer pointer increments to the next sample on a read of TrcDataA (LSBs).

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x2B	SPARE	TrigEdge		SPARE				
0x2C	PstTrigCnt (LSBs)							
0x2D	TrcRst	Arm	ForceTrig	PstTrigCnt (MSBs)				
0x2E	TrcDataBSel				TrcDataASel			
0x2F	TrigLvl (LSBs)							
0x30	TrigLvl (MSBs)							

TraceBufferControl Write Register Map

Field Name	Access (R/W)	Field Description
TrigEdge	W	Trigger edge for threshold level triggering. 0 = Disabled, 1 = Rising edge, 2 = Falling edge.
PstTrigCnt	W	Post Trigger count. Specifies number of samples to collect immediately following a trace trigger occurrence.
ForceTrig	W	Force trace trigger. Setting this self-clearing bit to 1 initiates a trace trigger, which causes data collection to cease immediately after another PstTrigCnt samples.
Arm	W	Trace arm. Setting this self-clearing bit to 1 starts data collection to the circular trace buffer.
TrcRst	W	Trace reset. Setting this self-clearing bit to 1 resets the trace state to Idle.

Field Name	Access (R/W)	Field Description
TrcDataASel, TrcDataBSel	W	Selects data items for display on channels "A" and "B": 1 = DC Bus Voltage 2 = V phase current 3 = W phase current 5 = Speed PI Reference 6 = Speed PI Feedback 7 = Speed PI Error 8 = IQ Ref 9 = Q axis voltage Qv 10 = D axis voltage Dv 11 = 12-bit electrical angle 12 = Q axis current Qi 13 = D axis current Di 14 = A axis (stationary frame) voltage Av 15 = B axis (stationary frame) voltage Bv
TrigLvl	W	Trigger threshold level. When this field is non-zero, a trace trigger is initiated each time the channel "A" data passes through this value as indicated by the TrigEdge field.

TraceBufferControl Write Register Field Definitions

4.2.7 VelocityControl Register Group (Write Registers)

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x31	SPARE				SpdLpRate			SPARE
0x32	KpSreg – Velocity loop proportional gain (LSBs) (W)							
0x33	KpSreg – Velocity loop proportional gain (MSBs) (W)							
0x34	KxSreg – Velocity loop integral gain (LSBs) (W)							
0x35	KxSreg – Velocity loop integral gain (MSBs) (W)							
0x36	SregLimP – Velocity loop positive Limit (LSBs) (W)							
0x37	SregLimP – Velocity loop positive Limit (MSBs) (W)							
0x38	SregLimN – Velocity loop negative Limit (LSBs) (W)							
0x39	SregLimN – Velocity loop negative Limit (MSBs) (W)							
0x3A	SpdScl – Speed Scale Factor (LSBs)							

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x3B	SpdScl – Speed Scale Factor (MSBs)							
0x3C	TargetSpd – Setpoint/target speed (LSBs)							
0x3D	TargetSpd – Setpoint/target speed (MSBs)							
0x3E	SpdAccRate – Acceleration							
0x3F	SpdDecRate – Deceleration							

VelocityControl Write Register Map

Field Name	Access (R/W)	Field Description
SpdLpRate	W	Speed loop update rate: 0 = disabled, N = update speed loop immediately before every Nth current loop update.
KpSreg	W	15-bit velocity loop proportional gain, in fixed point with 5 fractional bits. Range = 0 - 512.
KxSreg	W	15-bit velocity loop integral gain, in fixed point with 13 fractional bits. Range = 0 - 2.
SregLimP	W	16-bit speed PI controller output positive limit.
SregLimN	W	16-bit speed PI controller output negative limit (2's complement).
SpdScl	W	Motor Speed Scale factor. The user should set SpdScl = 60 * 16383 * (33.333MHz/32) / (Max RPM * Encoder PPR) / 2, which will result in a Spd value ranging ±16384 corresponding to ± Max RPM.
TargetSpd	W	Velocity loop speed setpoint in SPEED units, which are determined by the user via the SpdScl register setting.
SpdAccRate	W	Velocity loop acceleration in units of SPEED / Velocity loop execution or SPEED / (SpdLpRate / PWM period).
SpdDecRate	W	Velocity loop deceleration in units of SPEED / Velocity loop execution or SPEED / (SpdLpRate / PWM period).

VelocityControl Write Register Field Definitions

4.2.8 FaultControl Register Group (Write Registers)

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x42	SPARE						FltClr	DcBusM Enb

FaultControl Write Register Map

Field Name	Access (R/W)	Field Description
DcBusMEnb	W	DC Bus monitor enable. 1 = Monitor DC bus voltage and generate appropriate brake signal control and disable PWM output when voltage fault conditions occur. GatekillFlt and OvrSpdFlt faults cannot be disabled. DC bus voltage thresholds are as follows: Overvoltage – 410V Brake On – 380V Brake Off – 360V Nominal – 310V Undervoltage off – 140V Undervoltage – 120V
FltClr	W	This bit clears all active fault conditions. The user should monitor the FaultStatus read register group to determine fault status and set this bit to "1" to clear any faults that have occurred. A fault condition automatically clears the PwmEnbW and FocEnbW bits in the SystemControl write register group. Note that this bit also directly controls the output 2137 FLTCLR pin. After clearing a fault, the user must explicitly set this bit to "0" to re-enable fault processing.

FaultControl Write Register Field Definitions

4.2.9 SVPWMScaler Register Group (Write Registers)

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x44	ModScl (LSBs) (W)							
0x45	ModScl (MSBs) (W)							

SVPWMScaler Write Register Map

Field Name	Access (R/W)	Field Description
ModScl	W	Space vector modulator scale factor. This register, which depends on the PWM carrier frequency, should be set as follows: ModScl = PwmPeriod * sqrt(3) * 4096 / 2355 where PwmPeriod is the value in the PwmConfig write register group's PwmPeriod register.

SVPWMScaler Write Register Field Definitions

4.2.10 DiagnosticPwmControl Register Group (Write Registers)

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x4E	PwmData1Sel				PwmData0Sel			
0x4F	PwmData3Sel				PwmData2Sel			

DiagnosticPwmControl Write Register Map

Field Name	Access (R/W)	Field Description
PwmData0Sel, PwmData1Sel, PwmData2Sel, PwmData3Sel	W	Selects diagnostic data items for output on DAC PWM pins 0-3. These pins are intended for use with external RC filters for oscilloscope diagnostic display: 1 = DC Bus Voltage 2 = V phase current 3 = W phase current 5 = Speed PI Reference 6 = Speed PI Feedback 7 = Speed PI Error 8 = IQ Ref 9 = Q axis voltage Qv 10 = D axis voltage Dv 11 = 12-bit electrical angle 12 = Q axis current Qi 13 = D axis current Di 14 = A axis (stationary frame) voltage Av 15 = B axis (stationary frame) voltage Bv

DiagnosticPwmControl Write Register Field Definitions

4.2.11 SystemConfig Register Group (Write Registers)

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x50	ExtCtrlW	SpdRefSel		IqRefSel		HostAng Enb	HostVd Enb	RmpRef Sel

SystemConfig Write Register Map

Field Name	Access (R/W)	Field Description
RmpRefSel	W	Speed Ramp reference select. 0= TargetSpd field of the VelocityControl write register group, 1 = External analog reference.
HostVdEnb	W	Host D-Axis current control enable. When this bit is set, the D-Axis PI Controller is disconnected from the forward path vector rotator, which then takes its input from the VdSfwd field of the DirectHostVoltageControl write register group.
HostAngEnb	W	Host electrical angle control enable. When this bit is set, the vector rotator takes its angle input from the ElecAngW field of the DirectHostVoltageControl write register group.
IqRefSel	W	Selects the source for the Q-Axis PI controller IQREF input: 0 = Speed PI controller output 1 = IqRefW field of the CurrentLoopConfig write register group 2 = Reference A/D converter input.
SpdRefSel	W	Selects the source for the Speed PI controller reference input: 0 = Internal Accel/Deccel ramp generator 1 = TargetSpd field of the VelocityControl write register group 2 = Reference A/D converter input.
ExtCtrlW	W	Setting this bit to "1" enables direct control of basic motor operation via the external User Interface pins. When this bit is "1", the FocEnbW and PwmEnbW bits in the SystemControl write register group are ignored.

SystemConfig Write Register Field Definitions

4.2.12 DirectHostVoltageControl Register Group (Write Registers)

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x52	VdSfwd (LSBs) (W)							
0x53	VqSfwd (LSBs) (W)				VdSfwd (MSBs) (W)			
0x54	VqSfwd (MSBs) (W)							

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x55	ElecAngW (LSBs) (W)							
0x56	SPARE				ElecAngW (MSBs) (W)			

DirectHostVoltage Control Write Register Map

Field Name	Access (R/W)	Field Description
VdSfwd	W	12-bit signed value for synchronous frame direct current when host direct current control is enabled. This field is typically used for V/Hz control.
VqSfwd	W	12-bit signed value for synchronous frame quadrature voltage that is added to the Q-Axis PI-controller output. This field is typically used for feedforward or V/Hz control.
ElecAngW	W	12-bit electrical angle used when host electrical angle control is enabled. This field is typically used for V/Hz control.

DirectHostVoltageControl Write Register Field Definitions

4.2.13 32bitQuadDecode Register Group (Write Registers)

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x58	EncCnt32bW (bits 0-7) (W)							
0x59	EncCnt32bW (bits 8-15) (W)							
0x5A	EncCnt32bW (bits 16-23) (W)							
0x5B	EncCnt32bW (bits 24-31) (W)							

32bitQuadDecode Write Register Map

Field Name	Access (R/W)	Field Description
EncCnt32bW	W	New value for 32-bit Quadrature Decoder counter.

32bitQuadDecode Write Register Field Definitions

4.2.14 EepromControl Registers (Write Registers)

At power up, the write registers can be optionally initialized with values stored in EEPROM. The EepromControl write register group and EepromStatus read register group are used to read and write these EEPROM values. Since the EeAddrW write register (which selects the EEPROM offset to read or write) does not require initialization at power up, the location corresponding to that register in EEPROM (at offset 0x5D) is used to store a register map version code. At power on, the FPGA initializes the write registers from EEPROM only if the version code stored at this offset in EEPROM matches its internal register map version code (which can be read from the RegMapVerID field of the ProductIdentification read register group).

To enable write register initialization at power up, write the appropriate register map version code to EEPROM at offset 0x5D. To disable write register initialization at power up, write a zero (or any non-matching version code) to offset 0x5D of the EEPROM.

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x5C	SPARE					EeWrite	EeRead	EeRst
0x5D	EeAddrW / RegMapVersCode (W)							
0x5E	EeDataW (W)							

EepromControl Write Register Map

Field Name	Access (R/W)	Field Description
EeRst	W	Self-clearing EEPROM reset. Writing a "1" to this bit resets the I2C EEPROM interface.
EeRead	W	Self-clearing I2c EEPROM Read. Writing a "1" to this bit initiates an EEPROM read from the byte located at EEPROM address EeAddrW. After setting this bit the user should poll the EeBusy bit in the EepromStatus read register group to determine when the read completes and then read the data from EeDataR in the EepromStatus read register group.
EeWrite	W	Self-clearing EEPROM Write. Writing a "1" to this bit initiates an EEPROM write from the data byte in EeDataW to the EEPROM address EeAddrW.
EeAddrW	W	EEPROM Address Register. Contains the address for the next EEPROM read or write operation.
EeDataW	W	EEPROM Data Register. Contains the data for the next EEPROM write operation.

EepromControl Write Register Field Definitions

4.2.15 HallSensorEncoderInit (Write Registers – EEPROM only)

These values must be set in the EEPROM for initial encoder count/angle initialization in the EEPROM standalone (i.e. operation without a host program). EEPROM initialization logic automatically loads the appropriate value into the encoder counter at power-on based on the HALL A/B/C sensor values. These values are present only in the EEPROM since they serve no purpose after power on.

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x72	HallCBA001(LSBs)							
0x73	HallCBA001(MSBs)							
0x74	HallCBA010 (LSBs)							
0x75	HallCBA010 (MSBs)							
0x76	HallCBA011(LSBs)							
0x77	HallCBA011(MSBs)							
0x78	HallCBA100 (LSBs)							
0x79	HallCBA100 (MSBs)							
0x7A	HallCBA101(LSBs)							
0x7B	HallCBA101(MSBs)							
0x7C	HallCBA110 (LSBs)							
0x7D	HallCBA 110 (MSBs)							

HallSensorEncoderInit Register Map

Field Name	Access (R/W)	Field Description
HallCBA*nnn*	W (EEPROM ONLY)	Initial encoder count for Hall Sensor [C, B, A] value [*n*, *n*, *n*].

HallSensorEncoderInit Field Definitions

4.3 Read Register Definitions

4.3.1 QuadratureDecodeStatus Register Group (Read Registers)

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x0	EncCntR (LSBs) (R)							
0x1	EncCntR (MSBs) (R)							
0x3	SPARE	PwrOn HallC	PwrOn HallB	PwrOn HallA	SPARE	HallC	HallB	HallA

QuadratureDecodeStatus Read Register Map

Field Name	Access (R/W)	Field Description
EncCntR	R	Current value of 16-bit Quadrature Decoder counter.
HallA, HallB, HallC	R	Hall Sensor A/B/C values.
PwrOnHallA, PwrOnHallB, PwrOnHallC	R	Hall Sensor A/B/C values at power-on for reduced-wire encoder interface.

QuadratureDecodeStatus Read Register Field Definitions

4.3.2 SystemStatus Register Group (Read Registers)

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x7	Start	Stop	SPARE	PwrID		ExtCtrlR	Foc EnbR	Pwm EnbR

SystemStatus Read Register Map

Field Name	Access (R/W)	Field Description
PwmEnbR	R	PWM Enable bit status.
FocEnbR	R	FOC Enable bit status.
ExtCtrlR	R	Reflects the status of the ExtCtrlW bit in the System Configuration write register (address 0x50).
PwrID	R	Power ID. 0 = 3 kW, 1 = 2 kW, 2 = 500 W.
Stop	R	User Interface "STOP" digital input status.
Start	R	User Interface "START" digital input status.

SystemStatus Read Register Field Definitions

4.3.3 DcBusVoltage Register Group (Read Registers)

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0xA	DcBusVolts (LSBs)							
0xB	SPARE			Brake	DcBusVolts (MSBs)			

DcBusVoltage Read Register Map

Field Name	Access (R/W)	Field Description
DcBusVolts	R	DC Bus Voltage. Data range is 0 - 4095, which corresponds to a DC bus voltage between 0 and 500 volts.
Brake	R	Brake signal status. 0 = Brake signal active.

DcBusVoltage Read Register Field Definitions

4.3.4 FocDiagnosticData Register Group (Read Registers)

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0xC	IvFbk - V Phase IFB Raw Current (LSBs) (R)							
0xD	IwFbk - W Phase IFB Raw Current (LSBs) (R)				IvFbk - V Phase IFB Raw Current (MSBs) (R)			
0xE	IwFbk - W Phase IFB Raw Current (MSBs) (R)							
0xF	Id – Synchronous Frame Direct Current (LSBs) (R)							

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x10	Id – Synchronous Frame Direct Current (MSBs) (R)							
0x11	Iq – Synchronous Frame Quadrature Current (LSBs) (R)							
0x12	Iq – Synchronous Frame Quadrature Current (MSBs) (R)							
0x13	Ud – Synchronous Frame Direct Voltage (LSBs) (R)							
0x14	Ud – Synchronous Frame Direct Voltage (MSBs) (R)							
0x15	Uq – Synchronous Frame Quadrature Voltage (LSBs) (R)							
0x16	Uq – Synchronous Frame Quadrature Voltage (MSBs) (R)							
0x17	UAlpha – Stationary Frame Alpha Voltage (LSBs) (R)							
0x18	UBeta – Stationary Frame Beta Voltage (LSBs) (R)				UAlpha – Stationary Frame Alpha Voltage (MSBs) (R)			
0x19	UBeta – Stationary Frame Beta Voltage (MSBs) (R)							

FocDiagnosticData Read Register Map

Field Name	Access (R/W)	Field Description
IvFbk, IwFbk	R	Offset-corrected V and W phase raw current from the IR2175 current sensor. Values range from 0 - 4096, where 2048 corresponds to 0 current. The current feedback scale factors IdScl and IqScl in the CurrentFeedbackConfig write register group and the current sense resistor value determine the full scale current value.
Id, Iq	R	Synchronous or rotating frame direct and quadrature current values in 2's complement representation. The full scale current values range from –16384 to 16383.
Ud, Uq	R	Synchronous or rotating frame direct and quadrature voltage values in 2's complement representation. Data ranges are ± VdLim for Ud and ± VqLim for Uq as specified in the CurrentLoopConfig write register group.
UAlpha, UBeta	R	Stationary frame Alpha and Beta voltage output component values. Data range is ± VdLim or ± VqLim (as specified in the CurrentLoopConfig write register group), whichever is larger.

FocDiagnosticData Read Register Field Definitions

4.3.5 FaultStatus Register Group (Read Registers)

The Fault Status register records fault conditions that occur during drive operation. When any of these fault conditions occur, the PWM output is automatically disabled. The user should monitor this register continuously for fault conditions. A fault condition can be cleared by writing a "1" to the FaultClr bit in the FaultControl write register group. (This does not automatically re-enable PWM output.)

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x1E	SPARE			ExecTm Flt	OvrSpdFlt	OvFlt	LvFlt	GatekillFlt

FaultStatus Read Register Map

Field Name	Access (R/W)	Field Description
GatekillFlt	R	Filtered and latched version of IR2137 FAULT output.
LvFlt	R	DC bus low voltage fault. This fault occurs if the DC bus drops below 120V.
OvFlt	R	DC bus overvoltage fault. This fault occurs if the DC bus voltage exceeds 410V.
OvrSpdFlt	R	Over speed fault. This fault occurs whenever the motor reaches the positive or negative limits. The user should use the scale factor in the SpdScl field of the VelocityControl write register group to scale the motor speed so that it falls between -16384 and +16383 with these limits as the over speed condition.
ExecTmFlt	R	Execution time fault.

FaultStatus Read Register Field Definitions

4.3.6 TraceBufferStatus Register Group (Read Registers)

The data registers in the TraceBufferStatus Register group access a single entry in the trace buffer. A read of the TrcDataA (LSBs) register causes the internal trace buffer pointer to increment to the next entry. To retrieve all data items for a single trace buffer entry, read the TrcDataA and TrcDataB registers as a single 32-bit access, or read the TrcDataA (LSBs) register last.

See the description of the TraceBufferControl write register group for more information about using the trace function.

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x1F	SPARE	TrcSt		SPARE				
0x20	TrcDataA (LSBs) (R)							
0x21	TrcDataA (MSBs) (R)							

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x22	TrcDataB (LSBs) (R)							
0x23	TrcDataB (MSBs) (R)							

TraceBufferStatus Read Register Map

Field Name	Access (R/W)	Field Description
TrcSt	R	Trace data collection state. 0 = Idle, 1 = Armed/Collecting, 2 = Triggered, 3 = Done Collecting Data.
TrcDataA	R	Data "A" item from current trace buffer location.
TrcDataB	R	Data "B" item from current trace buffer location.

TraceBufferStatus Read Register Field Definitions

4.3.7 VelocityStatus Register Group (Read Registers)

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x26	Spd (LSBs)							
0x27	Spd (MSBs)							

VelocityStatus Read Register Map

Field Name	Access (R/W)	Field Description
Spd	R	Current motor speed in SPEED units. (See the description of SpdScl in the VelocityControl write register group.)

VelocityStatus Read Register Field Definitions

4.3.8 CurrentFeedbackOffset Register Group (Read Registers)

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x30	IfbVOffs (LSBs) (R)							
0x31	IfbWOffs (LSBs) (R)				IfbVOffs (MSBs) (R)			
0x32	IfbWOffs (MSBs) (R)							

CurrentFeedbackOffset Read Register Map

Field Name	Access (R/W)	Field Description
IfbVOffs, IfbWOffs	R	Current feedback offset values from the last IFB Offset calculation. These values are automatically applied to each current feedback measurement value whenever the IfbOffsEnb bit in the SystemControl write register group is set.

CurrentFeedbackOffset Read Register Field Definitions

4.3.9 32bitQuadDecodeStatus Register Group (Read Registers)

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x34	EncCnt32bR (bits 0-7) (R)							
0x35	EncCnt32bR (bits 8-15) (R)							
0x36	EncCnt32bR (bits 16-23) (R)							
0x37	EncCnt32bR (bits 24-31) (R)							

32bitQuadDecodeStatus Read Register Map

Field Name	Access (R/W)	Field Description
EncCnt32bR	R	Current value of 32-bit Quadrature Decoder counter.

32bitQuadDecodeStatus Read Register Field Definitions

4.3.10 EepromStatus Registers (Read Registers)

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x38	SPARE							EeBusy
0x39	EeDataR (R)							
0x3A	EeAddrR (R)							

EepromStatus Read Register Map

Field Name	Access (R/W)	Field Description
EeBusy	R	I2C EEPROM Interface busy bit. The user should wait for this bit to clear before initiating EEPROM read or write operations.
EeDataR	R	EEPROM Data Register. Contains the data from the last EEPROM read operation. Note that writing to the EeRst field in the EepromControl write register group invalidates this register.
EeAddrR	R	EEPROM Address read register shows the value stored in EEPROM at the offset of the EeAddrW write register (0x5D). Since this address in the EEPROM contains the BPFPGA register map version, the user can read this field to determine whether or not the write registers were initialized at power on.

EepromStatus Read Register Field Definitions

4.3.11 FOCDiagnosticDataSupplement Register Group (Read Registers)

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x3C	ElecAngR (LSBs) (R)							
0x3D	SPARE				ElecAngR (MSBs) (R)			
0x3E	SpdRef (LSBs) (R)							
0x3F	SpdRef (MSBs) (R)							

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x40	SpdErr (LSBs) (R)							
0x41	SpdErr (MSBs) (R)							
0x42	IqRefR (LSBs) (R)							
0x43	IqRefR (MSBs) (R)							

FOCDiagnosticDataSupplement Read Register Map

Field Name	Access (R/W)	Field Description
ElecAngR	R	Electrical angle.
SpdRef	R	Speed PI controller reference input.
SpdErr	R	Speed PI controller error.
IqRefR	R	Speed PI controller output.

FOCDiagnosticDataSupplement Read Register Field Definitions

4.3.12 ProductIdentification Registers (Read Registers)

Byte Offset	Bit Position							
	7	6	5	4	3	2	1	0
0x7C	ProductID (R)							
0x7D	RegMapVerID (R)							
0x7E	RevCodeID (LSBs) (R)							
0x7F	RevCodeID (MSBs) (R)							

ProductIdentification Read Register Map

Field Name	Access (R/W)	Field Description
ProductID	R	Product identification code.
RegMapVerID	R	Current register map version code.
RevCodeID	R	FPGA Revision Code. Revision code format is "XX.XX", where each "X" is a 4-bit hexadecimal number.

ProductIdentification Read Register Field Definitions

International
IOR Rectifier

WORLD HEADQUARTERS: 233 Kansas St., El Segundo, California 90245, Tel: (310) 322 3331
EUROPEAN HEADQUARTERS: Hurst Green, Oxted, Surrey RH8 9BB, UK Tel: ++ 44 1883 732020
IR CANADA: 7321 Victoria Park Ave., Suite 201, Markham, Ontario L3R 2Z8, Tel: (905) 475 1897
IR GERMANY: Saalburgstrasse 157, 61350 Bad Homburg Tel: ++ 49 6172 96590
IR ITALY: Via Liguria 49, 10071 Borgaro, Torino Tel: ++ 39 11 451 0111
IR FAR EAST: 171 (K&H Bldg.), 30-4 Nishi-ikebukuro 3-Chome, Toshima-ku, Tokyo Japan Tel: 81 3 3983 0086
IR SOUTHEAST ASIA: 315 Outram Road, #10-02 Tan Boon Liat Building, Singapore 0316 Tel: 65 221 8371
http://www.irf.com

Sales Offices, Agents and Distributors in Major Cities Throughout the World. ***Data and specifications subject to change without notice.***

