AN11517

Field Oriented Control (FOC) of PMSM motor using LPC15xx

Rev. 1.4 — 2 June 2014 Application note

Document information

Info	Content
Keywords	LPC15xx, FOC, SVPWM, PMSM, QEI
Abstract	This application note discusses the implementation of Field-Oriented Control (FOC) for a Permanent Magnet Synchronous Motor (PMSM) using space vector pulse width modulation with two shunts each connected to each phase of the motor for current measurement. Sensored FOC operation is demonstrated using QEI sensor.

Revision history

Rev	Date	Description
1.4	20140602	Added terminal command and source code release info.
1.3	20140328	Updated section 3.
1.2	20140304	Updated Fig 12 and Fig 13.
1.1	20140225	Updates throughout.
1	20140206	Initial version.

Contact information

For additional information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

1. Introduction

Control of Electronically Commutated (EC) motors can be achieved in various ways. The most straightforward control strategy is 'six-step' or 'trapezoidal control' which sequentially commutates each phase of the Brushless Direct Current (BLDC) motor. Position feedback of the rotor is given by a Hall sensor or encoder interface. In six-step control only two out of three phases are powered. Sensorless control is possible by measuring the induced voltages called back-EMF of the non-powered 'floating' phase.

Another type of control is sinusoidal control using Space Vector Pulse Width Modulation (SVPWM). This type of controller varies the phase currents depending on the rotor angle eliminating the torque ripple of a six-step control. The downside of sinusoidal commutation is that it attempts to control time-varying motor currents with a time intolerant Proportional-Integral (PI) controller algorithm, and does not account for interactions between the phases. This causes a performance loss at high speeds.

A more advanced approach is Field-Oriented Control (FOC). This technique is discussed in this application note. With this control strategy each phase is driven sinusoidally and permanently powered in a way that the magnetic field inside the motor is regulated to the most optimal value. Phase currents can be measured individually at the same time with two shunts by which the current flowing through the third phase can also be calculated.

A sensored version of FOC for position feedback of the rotor features an encoder interface. Hall sensors are also applicable by interpolating the rotor angle using a timer. Sensorless operation can be demonstrated by implementation of a current estimator. Below table shows different methods to control electronically commutated motors.

Table 1. Different control and feedback methods

	Hall	Encoder	Sensorless: Back-EMF measurement	Sensorless: Current measurement
Six-step	✓	✓	✓	⊘
Sinusoidal	✓ [1]	✓	⊘	✓
FOC	✓ [1]	✓	⊘	✓

[1] Angle must be interpolated with a timer.

This application note specifically talks about control of PMSM in QEI sensored mode with NXPs highly configurable LPC15xx family of microcontrollers.

The LPC15xx devices are equipped with useful features for motor control applications, which make software implementation of motor control algorithms easier and off-load the CPU.

- The SCTimer/PWM and ADC subsystem ensures easy and reliable control of time critical peripherals. It provides a mechanism implemented in hardware for synchronization of the ADC's with the PWM for exact timing of the sample moments.
- In order to use SVPWM in combination with current vector reconstruction the flexibility of the PWM block in the LPC15xx family is used to a full extent. The PWM match registers are equipped with a shadow register (match reload registers) to allow an atomic change of the PWM configuration without disturbing the currently generated PWM outputs. Automatic update of the PWM registers reduces CPU

overhead. SCTimer/PWM block is used to generate double edge PWM which reduces harmonics in the motor.

- Two channels of ADC0 are used to sample the two phase currents. The signals are sampled in one clock cycle at 36 MHz and converted to 12-bit values at a rate of 1.5 MHz. This fast sampling method helps to get samples when required by the FOC algorithm. If needed LPC15xx devices can be operated up to 2M Samples/sec sampling rate.
- The Quadrature Encoder Interface (QEI) monitors the position and velocity, using hardware registers without generating interrupts.
- The LPC15xx family features a Cortex M3 core with dedicated instruction and data busses to get maximum performance out of the CPU and still being deterministic.
- The 32-bit processing power and powerful instructions of the ARM Cortex-M3 core of the LPC15xx series compensates for the math-intensive FOC algorithms.

Fig 1 shows the LPC15xx PWM analog subsystem block diagram. This block diagram shows the interconnections between SCTimer/PWM, ADC and the analog comparator.

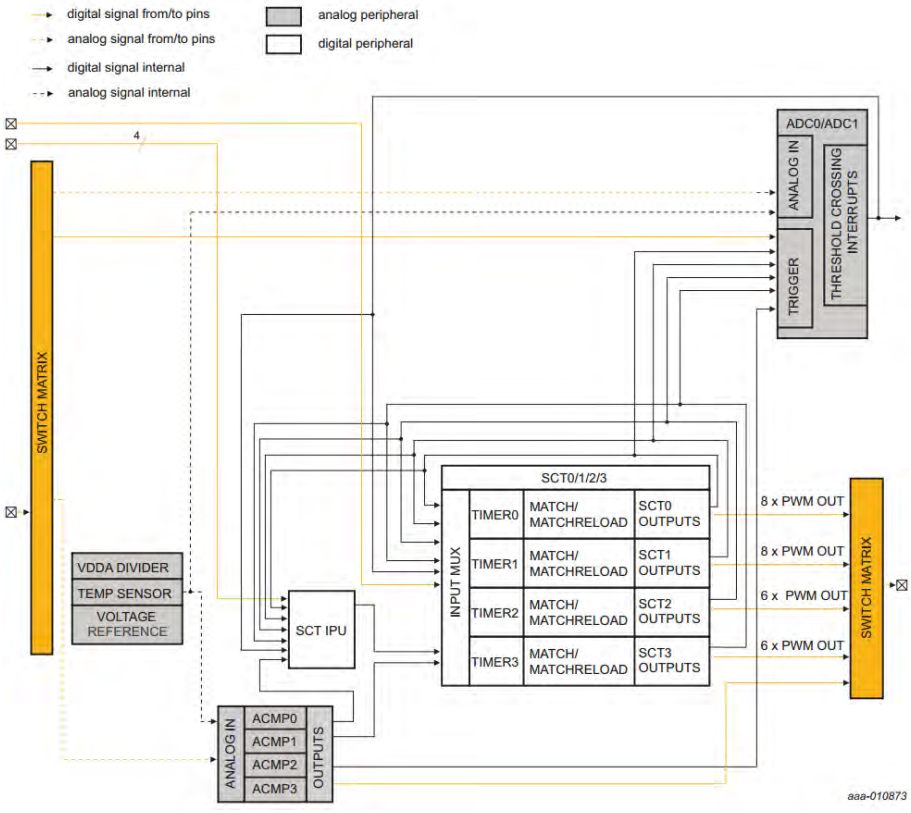

Fig 1. LPC15xx SCTimer/PWM Analog subsystem

The SCTimer/PWM combines the features of a timer and a state machine allowing development of sophisticated solutions in the digital control field.

In its full-featured implementation, the block provides up to 16 independent programmable hardware resources (named “events”), which can be configured to be active (to perform actions) when programmable conditions based on timed matches, or I/O signals, or a combination of them, are verified. This is defined in the control register for the each hardware event resource.

Each of the hardware events can be configured to drive an output signal, trigger an IRQ, a DMA transfer, or influence the timer behavior itself (by stopping, starting, resetting or halting the timer).

For defining a time-based event, or the time-based part of a combined time-based and I/O based event, the SCTimer/PWM provides dedicated registers which will hold the counter match value. These registers are associated to “shadow registers” which can be changed by the application at runtime, to modify the match point; the new value will be loaded from the shadow register into the match register when the associated timer gets limited (when its counter is reset to zero if in up-counting mode, or reverses direction if in up-down counting mode).

The SCTimer/PWM is also able to provide more complex sequencing, by introducing the concept of “states”. Each event is capable of letting a state machine (associated to each timer via a state register) jump from one state to another during operation. Each of these user-defined states can be configured to “filter” a specific subset of all possible events which have been defined for the SCTimer/PWM. In this way it is possible to configure events to be active only in specific states.

As a consequence, it is possible to associate the behavior of the SCTimer/PWM with a state machine diagram, and the system will be able to react differently to certain events depending on the specific state the SCTimer/PWM is currently into.

A demonstration kit was developed to demonstrate FOC for a PMSM at 17.5 kHz PWM frequency. This kit uses an LPC15xx-based LPC1549 LPCXpresso board, an LPC1549 LPCXpresso motor control board with a three-phase power inverter, and supports a Quadrature Encoder Interface (QEI) and a hall sensor. It also supports sensorless FOC control. For FOC control, current measurement is done by two shunts placed at phase A and phase B. The board also supports sensored or sensorless six-step control with back-EMF measurement. This application note discusses the usage of the demonstration board for FOC motor control in QEI sensor mode.

The PMSM FOC software is used to control the PMSM motor. The PMSM FOC software performs pulse width modulated field oriented control for use with PMSM. This software is specifically intended for use with PMSM motors. It is specifically targeted for operation on the NXP LPC15xx ARM-Cortex-M3 class microcontroller.

This software makes use of specific LPC15xx features to perform FOC motor control, and is intended to be tuned for operation of a specific motor and/or load. This software will also operate standard BLDC motors, however, improved performance will be noted with actual PMSM motors.

This application note assumes familiarity with the following NXP Semiconductors documents:

- LPC1549 Xpresso v2_schem_Rev_B2
- LPCXpresso Motor Control Board Revision B
- NXP LPC15XX Data Sheet
- NXP LPC15XX User Manual

- Teknic PMSM Motor(M2310P-LN-04K) Specification

The various topics covered in this application note are as follows:

1. SVPWM
2. Hardware
3. Software

2. SVPWM

2.1 PMSM and BLDC difference

PMSM and BLDC motors are both permanent magnet-based motors with the same basic structure, consisting of permanent magnets on the rotor and windings on the stator. The main difference between BLDC and PMSM is the drive signal for which it is designed. A PMSM is designed for a sinusoidal drive, while a BLDC is designed for a trapezoidal drive. The advantage of a sinusoidal driven motor is the minimized torque ripple that results in a much quieter motor, both electrically and mechanically. The current harmonics are in the switching frequency range resulting in a lower audible noise, lower motor core losses and a reduced current peak. The disadvantages are higher switching losses due to an extra phase that has to be powered constantly.

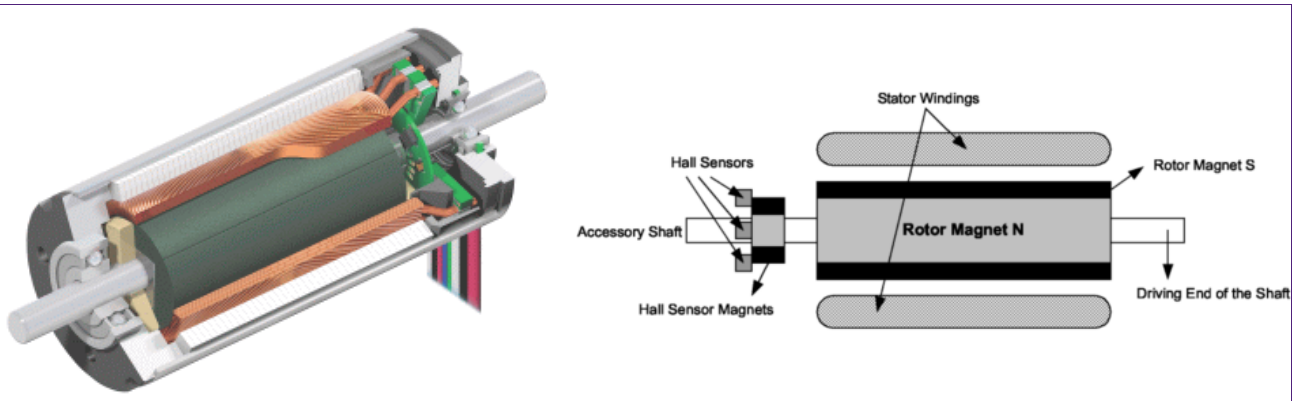

Fig 2. Brushless DC/PMSM motor

For a sensorless operating BLDC motor, the back EMF has to be trapezoidal. Therefore the rotor is designed to have a uniform flux profile in contrast to the sinusoidal flux profile of a PMSM. A BLDC requires a low winding inductance so it typically uses surface mounted rotor magnets. The high inductance of a PMSM motor makes it less suitable for trapezoidal control.

2.2 FOC basics

The PMSM motor in Fig 2 has two main components: stator windings and a permanent magnet. The term synchronous means the magnetic fields generated by the stator and the rotor rotate at the same frequency. In other words, PMSM motors do not have the slip that is normally seen in induction motors.

The basic operating theory of FOC is to measure and regulate the magnetic field in the motor. Optimum leverage for the magnetic forces is achieved when the direction of the permanent magnetic field of the rotor is perpendicular to the induced magnetic field of the stator windings. This creates torque on the rotor axis. This principle is illustrated in Fig 3.

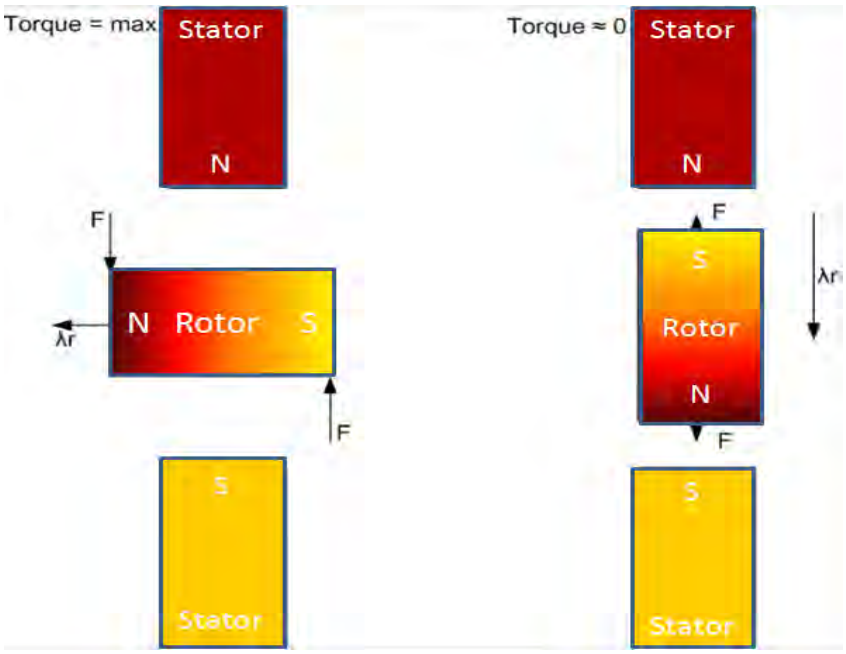

Fig 3. Torque on different field directions

The direction and magnitude of the induced magnetic field is equal to the sum of the current vectors of the different phases. Thus by measuring the currents through each phase, the total current vector can be calculated. Because the PMSM is a three-phase motor, each phase is separated by 120°.

A 'Clarke transformation' is used to convert a 3-phase system into a 2-phase coordinate system. This frame is called the static reference frame. The quadrature-phase components can be calculated using only two phases of the 3-phase system. As shown in Fig 4 I_α and I_β are the current vectors in the two phase system. I_a, I_b and I_c are the three phase current flowing through three windings of a three phase PMSM motor.

$$I_\alpha = I_a - \tfrac{1}{2}(I_b - I_c) \quad (1)$$

$$I_\beta = \tfrac{1}{2}\sqrt{3}(I_b - I_c) \quad (2)$$

With $I_a + I_b + I_c = 0$,substitution of I_b with $-I_a - I_c$ and scaling with a factor 2/3 this leads to a non power-invariant transformation:

$$I_\alpha = I_a \quad (3)$$

$$I_\beta = \left(I_a + 2 \cdot I_b\right) / \sqrt{3} \quad (4)$$

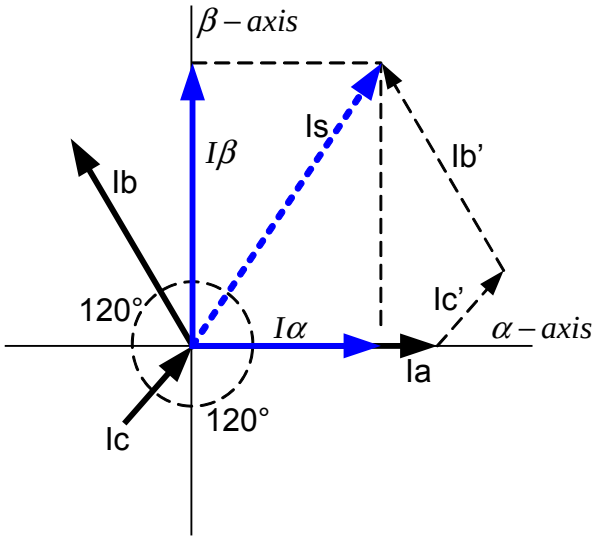

Fig 4. Clarke transformation

The key of FOC is to regulate the magnetic field such that it is perpendicular to the magnetic field of the rotor. By applying a voltage vector to the windings of the motor, the field is controlled. This voltage vector is translated to three duty cycles for each phase by the SVPWM system described in the next paragraph. Both axes of the voltage vector are regulated with a Proportional Integral (PI) controller.

The input of the controller is a single axis of the current vector where the output matches the equivalent axis of the voltage vector. Because the motor spins at high rpm, the field must also spin at the same high frequency. A PI regulator can only be optimally calibrated for a fixed frequency.

With the wide speed-range of the motor, an additional step is needed for the PI controller to work optimally. The current vector in the static reference frame (α, β) of the windings can be transformed into the dynamic frame (d, q) of the rotating permanent magnet to create a speed invariant system that can be regulated by a PI controller.

A Park transformation is used to transform the static reference frame to a dynamic reference frame. In a zero speed situation, the quadrature component (q) produces torque while the direct component (d) only produces unwanted forces in the motor bearings. For a higher speed range a negative d-component can be used to perform field weakening, which is not covered in this application note.

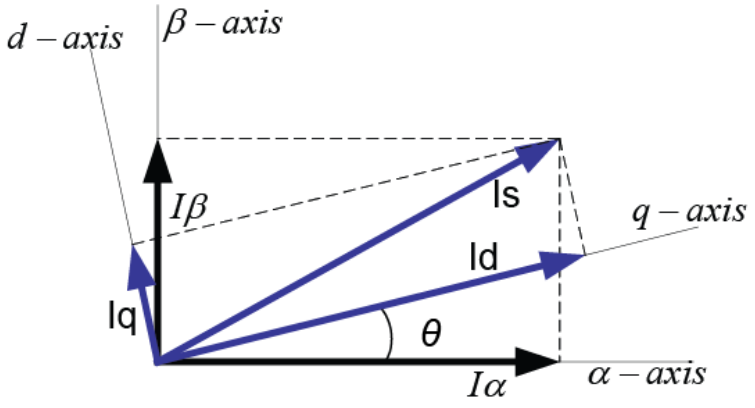

Fig 5. Park transformation

$$I_d = I_\beta \cdot \sin\theta + I_\alpha \cdot \cos\theta \quad (5)$$

$$I_q = I_\beta \cdot \cos\theta - I_\alpha \cdot \sin\theta \quad (6)$$

The outputs of the PI controllers are represented as voltage vectors V_q and V_d. An inverse Park transformation is used to go back from the rotating reference frame to the stationary reference frame producing V_α and V_β. Basically it needs to go back to the three-phase currents to control the motor. This technique is called SVPWM described in paragraph 2.3.

Inverse Park:

$$V_\alpha = V_d \cdot \cos\theta - V_q \cdot \sin\theta \quad (7)$$

$$V_\beta = V_d \cdot \sin\theta + V_q \cdot \cos\theta \quad (8)$$

2.3 SVPWM

SVPWM is a technique to generate three-phase PWM signals. The diagram of a three-phase inverter is shown in Fig 6.

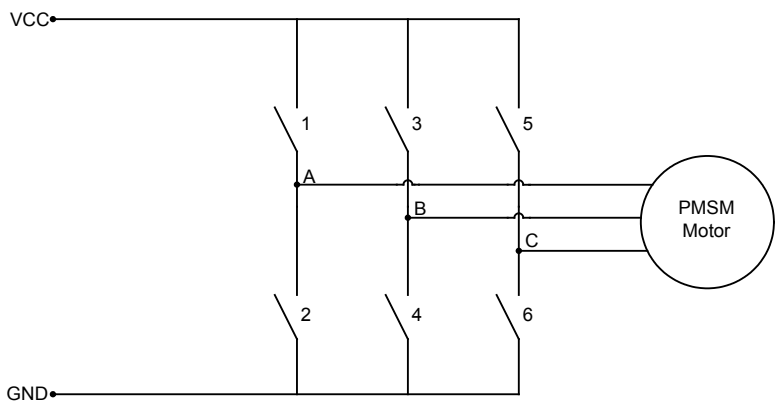

Fig 6. Three-phase inverter

Each phase of the three-inverter outputs can be driven high or low. In every branch there is always one switch open and one switch closed.

If both switches are closed, a short-circuit situation is created. This means that there are eight possibilities depending on the switches that are turned on. The two states where all three outputs are connected to the positive or the negative supply are called zero states or zero vectors. In that case there is no potential difference between any of the phases.

Table 2. Switching states

	Space vector	Switching state	ON switches
Zero vector	V7	[111]	1.3.5
	V0	[000]	2.4.6
Active vector	V1	[001]	2.4.5
	V2	[010]	2.3.6
	V3	[011]	2.3.5
	V4	[100]	1.4.6
	V5	[101]	1.4.5
	V6	[110]	1.3.6

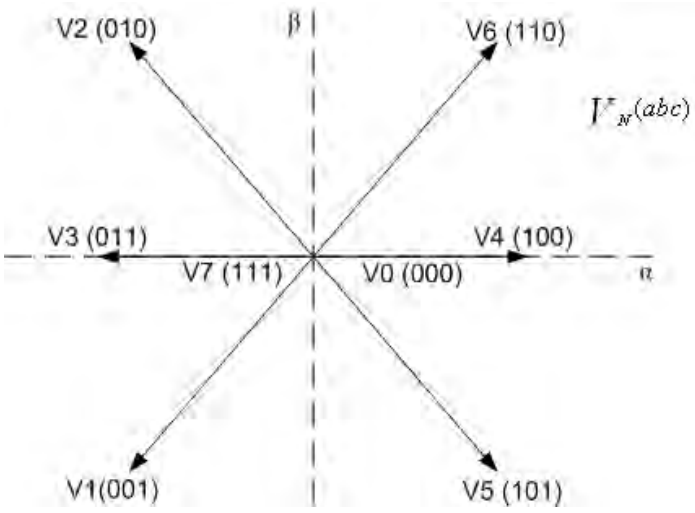

Fig 7. SVPWM star

Fig 7 shows how the active vectors divide the plane into six sectors. Depending on the sector that the reference voltage is in, two adjacent vectors are chosen. The binary representations of two adjacent basic vectors differ in only one bit. The vectors are time weighted in a sample period T to produce the desired output voltage.

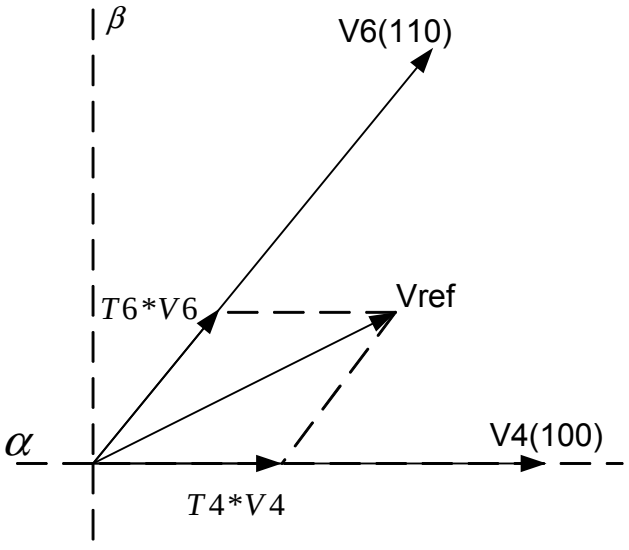

Fig 8. Reference vector

The length of the vectors T4 and T6 represent the amount of the time V4 and V6 are active and vary depending on the angle and amplitude of the reference voltage vector. In other words, T4 represents the on-time of V4, which activates switches 1, 4 and 6 as can be seen from Table 2. Dividing the actual time T4 by T produces a normalized time vector T4. The on-time of the vectors T4 and T6 can be calculated with the sine rule:

$$T4 = V_{ref} \cdot \left(\cos(\alpha) - \tfrac{1}{3}\sqrt{3} \cdot \sin(\alpha)\right) \quad (9)$$

$$T6 = 2 \cdot V_{ref} \cdot \frac{1}{\sqrt{3}} \cdot \sin(\alpha) \quad (10)$$

When V_{ref} is smaller than V4, which is typically maximum $\frac{2}{3}$ V_{dc}, one of the zero vectors (V0 or V7) have to be used to produce a no-torque condition for a period equal to:

$$T0 = 1 - T4 - T6 \quad (11)$$

Different strategies are possible in sequencing V4, V6 and one of the zero vectors V0 or V7. The strategy that induces the least harmonic distortions on the output line voltage uses a center aligned seven-segment technique. The sequence V0-V4-V6-V7 is used in the first half period, and V7-V6-V4-V0 in the second half period. The sequences are symmetrical. An example period with seven segments is shown in Fig 10. In the middle of the pattern, zero-vector V7(111) is selected instead of zero-vector V0(000). To go from V6 to V7 requires only one of three branches of the three-phase inverter to be switched. This minimizes the number of switching actions.

2.4 Duty cycle calculation

The next step in SVPWM control is to calculate the actual duty cycle value for each phase. First it is important to determine which sector is currently active by evaluating the rotor angle. For calculation, use only the adjacent vectors in the given sector.

Let's assume for example that Vref = 10 V, Vdc = 12 V and the rotor angle equals 20. This angle refers to the first sector of the SVPWM hexagon, therefore V4(100), V6(110), V0(000) and V7(111) are used for calculation.

$$T4 = \frac{10V}{12V} \cdot \left(\cos(20) - \tfrac{1}{3}\sqrt{3} \cdot \sin(20)\right) \cdot 100\% = 62\% \quad (12)$$

$$T6 = 2 \cdot \frac{10V}{12V} \cdot \frac{1}{\sqrt{3}} \cdot \sin(20) \cdot 100\% = 33\% \quad (13)$$

$$T0{,}7 = 100\% - T4 - T6 = 5\% \quad (14)$$

The duty cycle equals the sum of the corresponding components of the adjacent vectors multiplied by their 'on-time'. Remember that a seven-segment SVPWM pattern is chosen where half the time of the zero-vectors V0 is used and V7 for the other half. When V0 is used, none of the switches are active, so the contributing value to the PWM will always be zero and therefore can be neglected.

PWM = V4 component * 'on-time'	+ V6 component * 'on-time'	+ V7 component * ½ 'on-time'
PWM1 = 1 * 62%	+ 1 * 33%	+ 1 * ½ * 5% = 97,5%
PWM2 = 0 * 62%	+ 1 * 33%	+ 1 * ½ * 5% = 35,5%
PWM3 = 0 * 62%	+ 0 * 33%	+ 1 * ½ * 5% = 2,5%

This calculation requires the angle and the length of the voltage reference vector. The output of the PI regulation however consists of the α and β components of the voltage vector. A smarter approach of calculating the PWM values is to transform the two-phase voltage vector into a three phase vector V1, V2 and V3 using a modified inverse Clarke transformation.

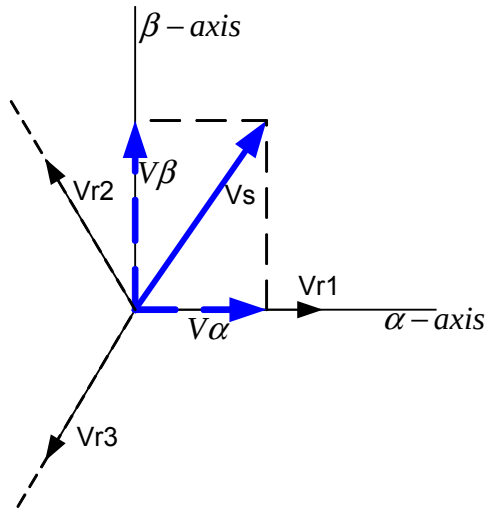

Fig 9. Inverse Clarke transformation

$$V_{r1} = V_{\beta} \tag{15}$$

$$V_{r2} = -\tfrac{1}{2}V_{\beta} + \tfrac{1}{2}\sqrt{3} \cdot V_{\alpha} \tag{16}$$

$$V_{r2} = -\tfrac{1}{2}V_{\beta} - \tfrac{1}{2}\sqrt{3} \cdot V_{\alpha} \tag{17}$$

By reversing V_{α} and V_{β} a reference axis is created that is shifted thirty degrees from the SVPWM star. As a result two of the three phases Vr1, Vr2 and Vr3 are symmetrically bound to the SVPWM star, and one axis is exactly opposite. The values on the two bound axes are equal to the time vectors T4 and T6.

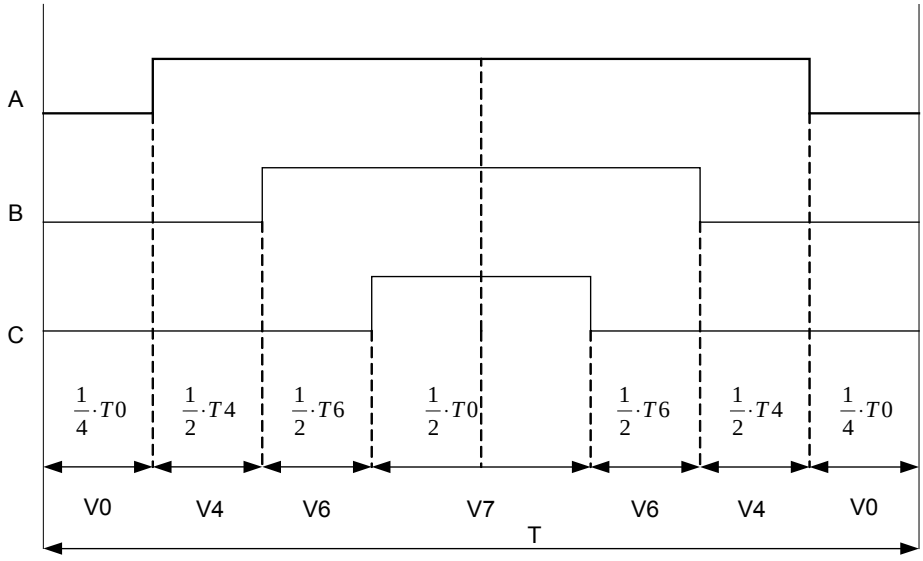

Fig 10. Seven-segment SVPWM period

After normalizing the voltage vectors the duty cycle can then be calculated. A full period T is equal to T = T0 + T4 + T6. From Fig 10 the duty cycle of phase C is equal to:

$$C = \tfrac{1}{2}T0 \quad (18)$$

Substitution for T0 = 1 – T4 – T6 leads to:

$$C = \tfrac{1}{2}(1 - T4 - T6) \quad (19)$$

$$B = C + T6 \quad (20)$$

$$A = B + T4 \quad (21)$$

For calculation of the real duty cycle, substitute values T4 and T6 for the two voltage reference vectors that correspond with the bound axes of that sector. Therefore, in each sector, the duty cycle calculation is done by a different combination of vectors for the different phases. The easiest way to implement this in software is to use a state machine depending on the sign of the voltage vectors.

2.5 FOC implementation

Fig 11 shows the high level block diagram implemented with LPC15xx to control PMSM motor. This diagram shows the FOC algorithm embedded inside two PI control loops.

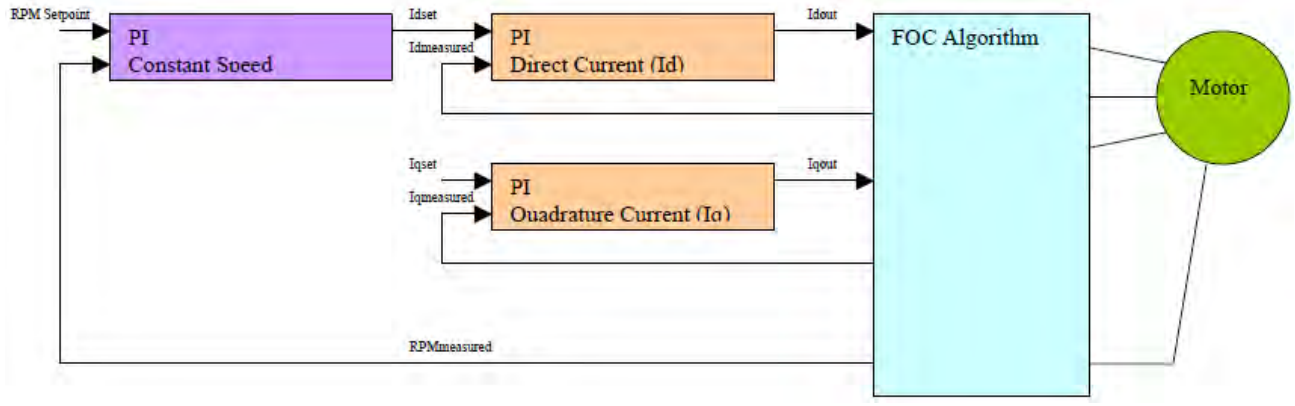

Fig 11. FOC algorithm embedded inside two PI control loops

Fig 12 shows a more detailed concept for speed control with FOC using the Quadrature encoder interface available on LPC15xx family of microcontrollers. The block schematic is divided into five main blocks.

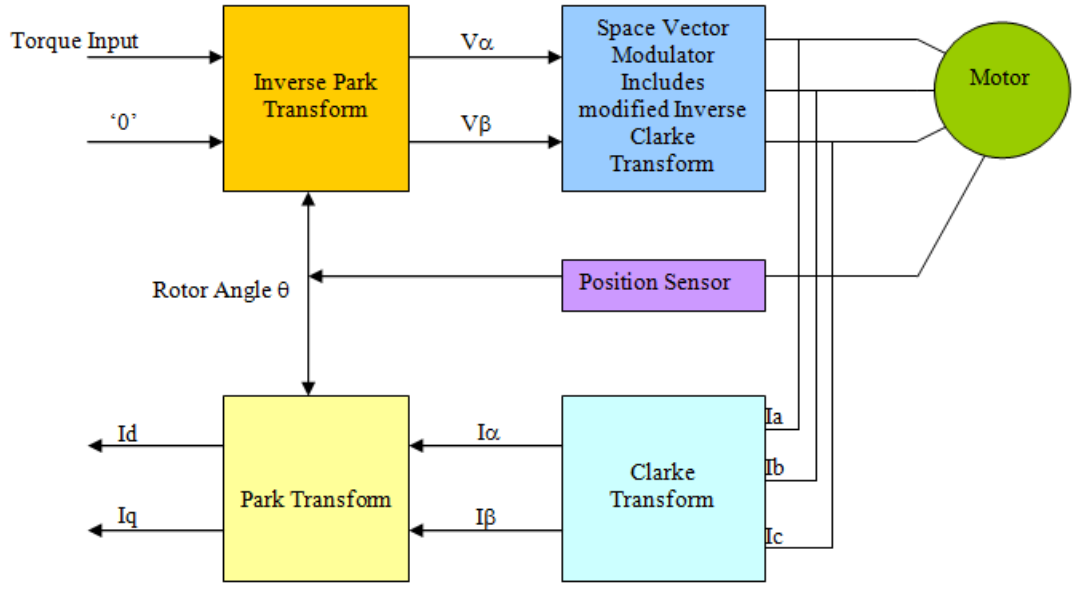

Fig 12. FOC motor control structure

Fig 13 shows the detailed concept for speed control with FOC. The block schematic is divided into smaller blocks.

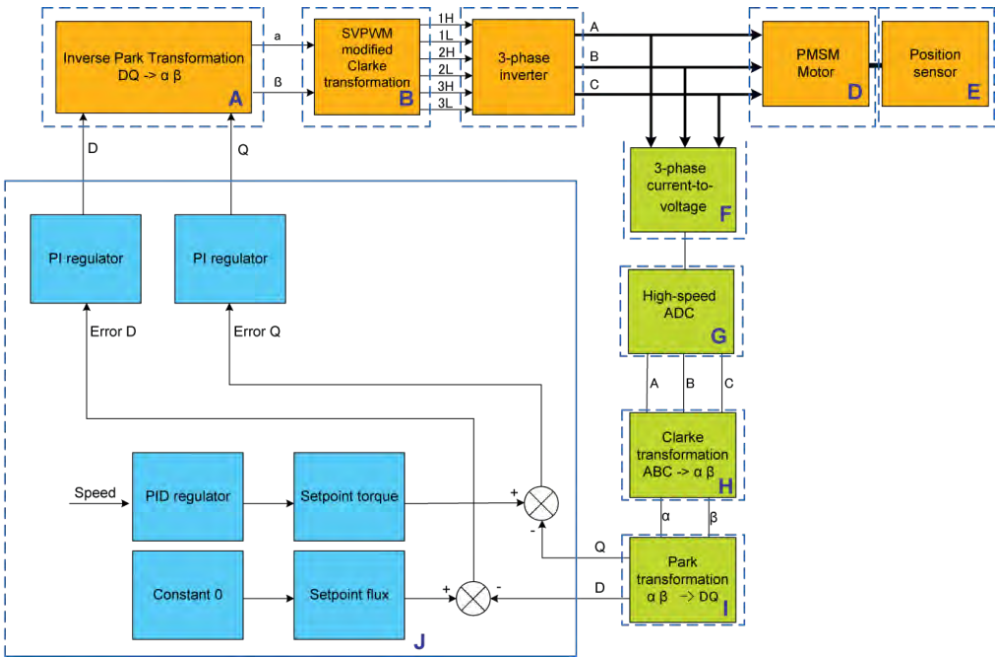

Fig 13. FOC block schematic

Table 3. Description of subsystems

Subsystem	Description	Function
A	Inverse Park transformation	Convert V_q and V_d setpoint to the static reference frame
B	Space Vector Pulse Width Modulation (SVPWM)	Generate PWM signals from SVPWM start for inverter according to reference voltage vector
C	3-phase inverter	Apply PWM signals to the PMSM motor
D	Brushless motor (PMSM)	Motor
E	Position sensor	Obtain rotor position. Needed for FOC transformations and speed calculation.
F	Phase current measurement	High Speed ADC for current measurements
G	Current vector reconstruction	Reconstruction of the current vector of the different phases.
H	Clarke transformation	Transformation from a three-phase to a two-phase system.
I	Park transformation	Transformation to a dynamic reference frame for time independent PI regulators.
J	PI regulation	Correct error between measured and reference user inputs.

3. Motor control setup

This section describes the LPC1549 LPCXpresso motor control kit hardware setup and firmware. The LPC1549 LPCXpresso motor control kit contains an LPC1549 LPCXpresso V2 Rev B board and an LPCXpresso motor control Rev B board.

The Motor Control Kits need to be correctly programmed before use, to avoid heating of the brake resistors. Please update the firmware on the LPC1549 LPCXpresso board in your kit, as described in this application note below, before attempting to connect the power adaptor to motor control board.

3.1 LPC1549 LPCXpresso motor controller kit serial port setup

The LPC1500 PMSM FOC Control zipfile contains the following items:

- Readme.txt
- AN11517
- LPC1500PMSMFOC_V1.1.zip
- LPC1500MotorControlGUI_V1.10

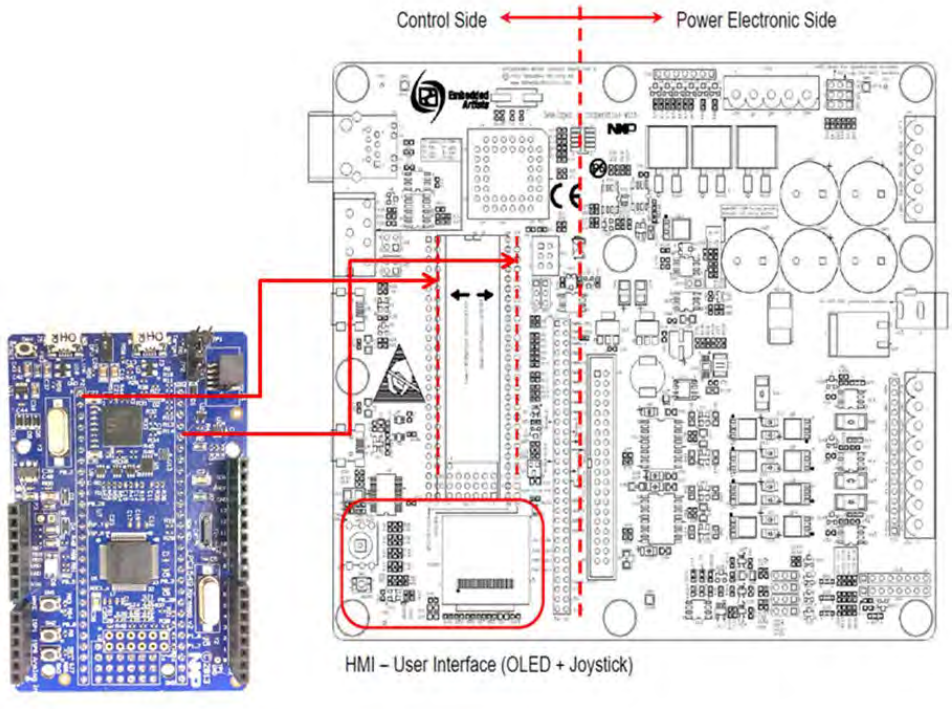

Fig 14. LPC1549 LPCXpresso board and LPC1549 LPCXpresso motor control board connection scheme

- Connect LPC1549 Xpresso board to motor control base board as shown in Fig 14. Please do not connect power adaptor to motor control board. You can open IAR project file inside LPC1500PMSMFOC_V1.1.zip file and create a binary file. If you want to flash target using debugger through SWD please follow below instructions. These steps are required to mitigate a problem seen with FTDI drivers.

To complete the demo setup make the below hardware connections to the LPCXpresso Motor Control kit.

1. Connect Micro USB cable to Micro USB connector (J3) of **LPC1549 Xpresso board.** This connector is used here to flash binary using USB ISP. Connect other end to PC. Put LPC1549 in ISP mode. To put device in USB ISP first press and hold ISP0 switch (SW3) and then press and release reset switch (SW4). ISP switches are shown in Fig 15.

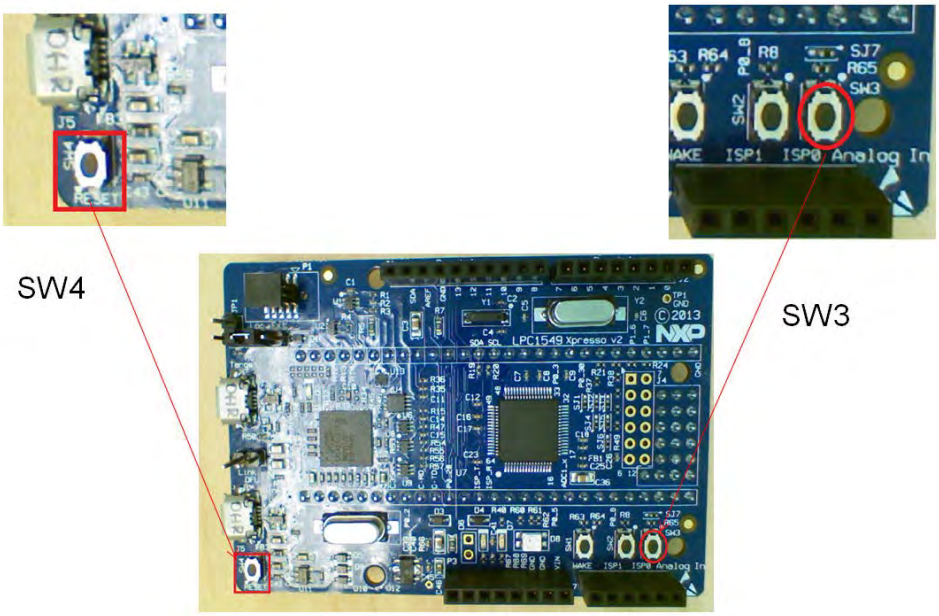

Fig 15. LPC1549 Xpresso board ISP Switches

2. Connect USB mini cable to Mini USB connector (J6) of LPCXpresso Motor Control board and connect other end to PC. It is used for serial communication with GUI.
3. This connection will show new VCOM com port on your PC as shown in Fig 16. To see COM port (USB to serial port) go to device manager and click on “Ports (COM & LPT)”. Here you will see the COM port. If you do not see com port Install FTDI drivers from FTDI_Drivers folder or download from http://www.ftdichip.com/Drivers/VCP.htm

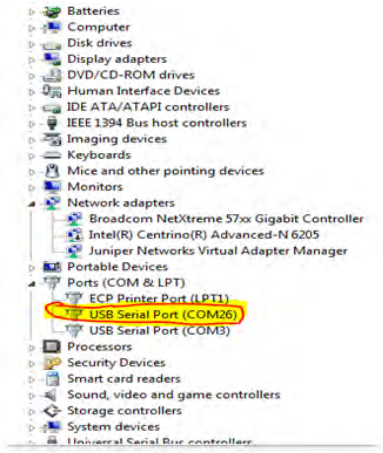

Fig 16. VCOM Port on PC

4. Unplug and Plug Mini USB connector (J6) on motor control board.
5. The PC mouse may not work properly. After installing drivers please follow below steps to mitigate this issue:
   a) Go to control panel and click on device manager.
   b) Check the com port associated with this motor control board (FTDI Com port).
   c) One way to do is plug and unplug USB mini cable and check the VCOM port (USB serial port).
   d) Right click on COM Port and go to port setting and click on advanced as shown.
   e) Uncheck serial enumerator in Miscellaneous Options as shown.

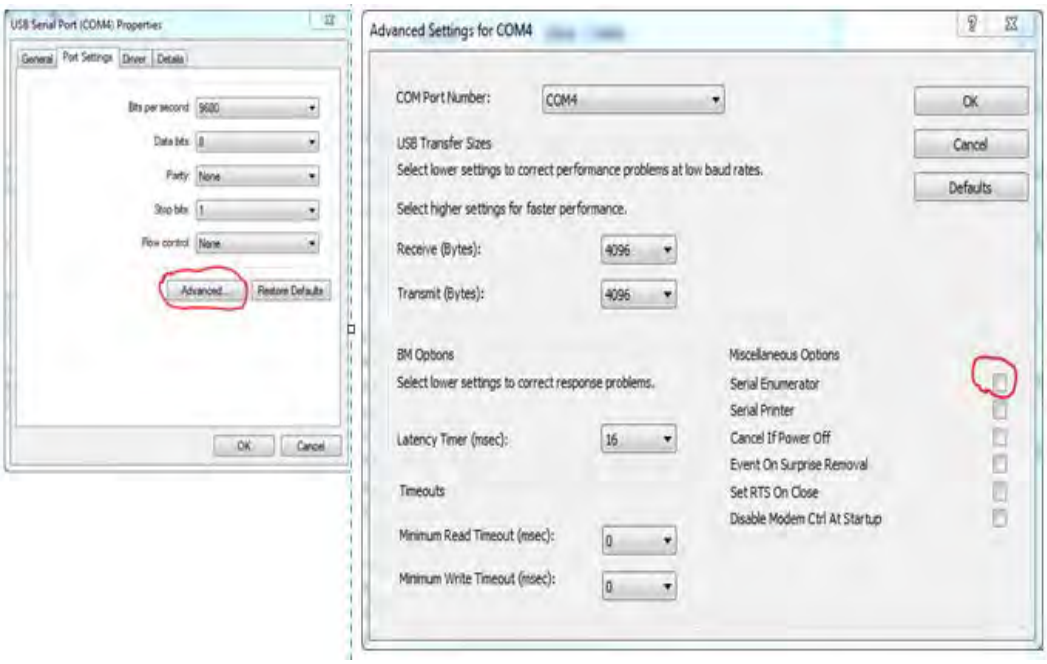

Fig 17. Com Port setting change

3.2 Flashing binary in LPC1549 Xpresso board

If you want to flash target using debugger, you can skip this sub section.

- **In LPC1549 Xpresso board** for USB ISP first press and hold ISP0 switch (SW3) and then press and release reset switch (SW4). After this release SW3. It will create a drive (may be D:) in PC.
- PC will show drive as shown in Fig 18.
- This drive contains firmware.bin file as shown here.

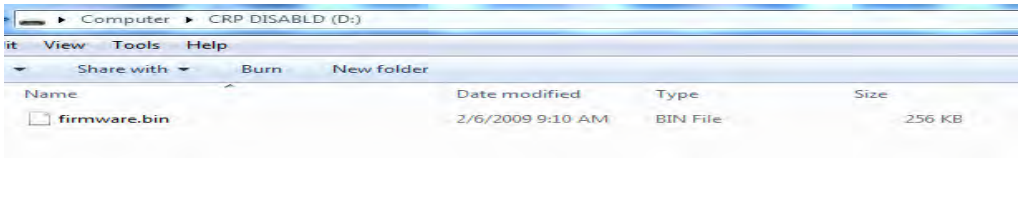

Fig 18. Flashing PMSM binary in LPC1549 Xpresso board

- Delete firmware.bin file.
- Copy binary file generated by you into this newly created drive.
- Reset board. You will see LED D8 blinking with green color light. If power supply is connected you will see blue and white color LED blinking, which should not be connected till this step. This is heart beat LED. If you do not see this please repeat procedure mentioned above.

- For more detailed information please refer to the LPCXpresso motor control kit user manual at

  http://www.nxp.com/documents/other/LPCXpresso_Motor_Control_Kit_UserManual.pdf

3.3 Flashing in LPC1549 Xpresso board Using debugger

You need to flash motor control firmware in LPC1549 Xpresso board before connecting power adaptor to avoid heating of brake resistors. Please open NM1.eww in IAR IDE build project and click on download and debug button to flash firmware into LPC1549 target board.

Fig 19 shows how an LPC1549 LPCXpresso board should be connected to the LPCXpresso Motor Control Rev B board. Jumper settings on the LPCXpresso Motor Control Board should be configured to their factory default states.

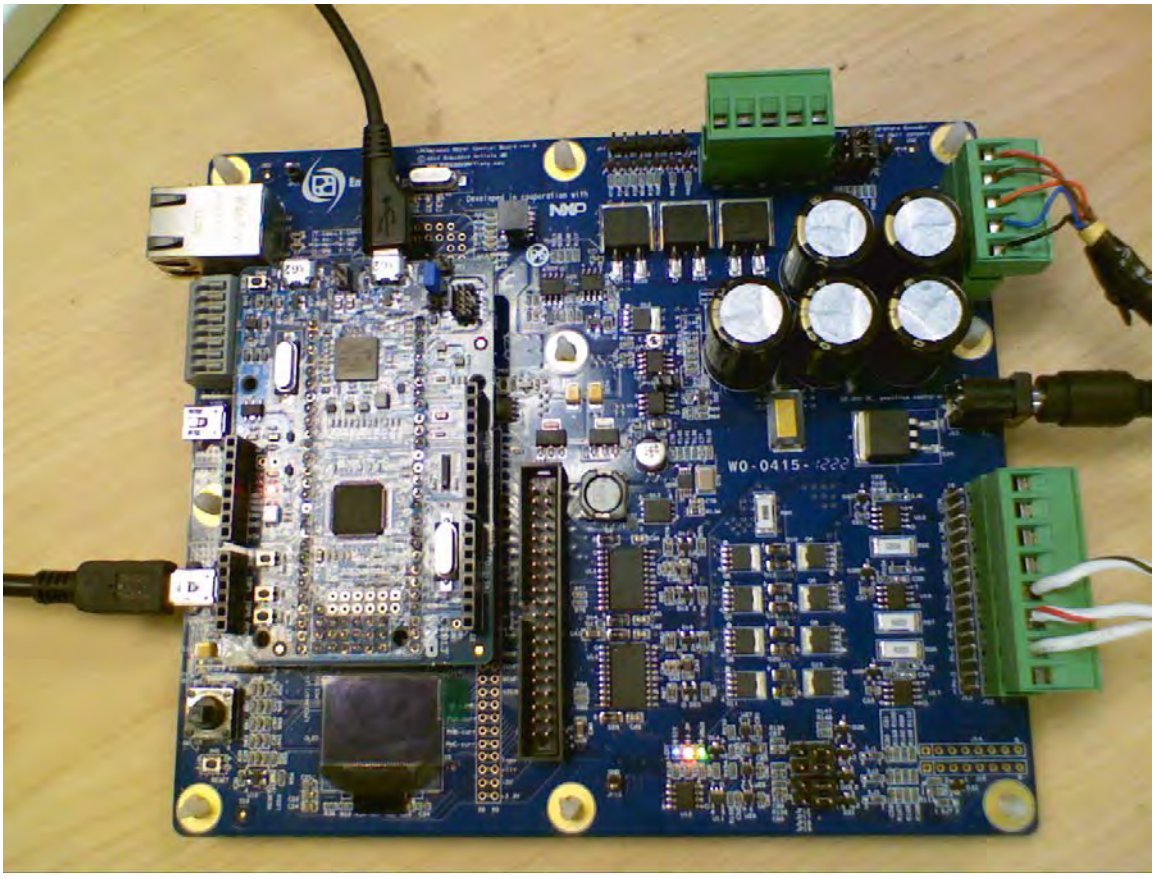

Fig 19. **LPC1549 LPCXpresso motor controller board**

Install motor control GUI from downloaded LPC1500 PMSM FOC Control_4.zip file. A short cut will be created on your desktop named “LPC1500 Motor Control”.

3.4 PMSM motor connection

The PMSM motor used in this application note is the motor from Teknic Inc.

Connect the PMSM motor as shown in Fig 20 and Fig 21.

Connect motor phase wires R, S and T to LPCXpresso Motor Control Board Phase Outputs A, B and C respectively.

Connect motor Quadrature Encoder Outputs A, B, and Index to the LPCXpresso Motor Control Board's Quadrature Encoder Inputs, A, B, and Index respectively.

Do not connect Hall sensors.

TEKNIC MOTOR (Wire Reference)							
PIN	COLOR	SIGNAL		PIN	COLOR	SIGNAL	
1*	DRAINx3	P DRAIN	Not used	9	**16AWG BLK**	PHASE R	Used
2*	N/A	N/A	Not used	10	**16AWG RED**	PHASE S	Used
3*	GRN	COMM S-T	Not used	11	**16AWG WHT**	PHASE S	Used
4*	GRN/WHT	COMM R-S	Not used	12	**RED**	5VDC IN	Used
5*	GRY/WHT	COMM T-R	Not used	13	**BRN**	ENC I	Used
6*	DRAINx1	E DRAIN	Not used	14	**ORN**	ENC B	Used
7	**BLK**	GND	Used	15	**BLU**	ENC A	Used
8*	BLU/WHT	ENC A~	Not used	16*	ORN/WHT	ENC B~	Not used

* Pins not used and are for reference only

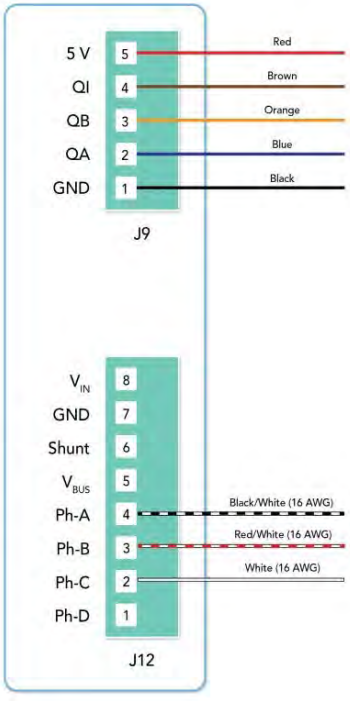

Fig 20. Teknic motor connections with LPCXpresso motor control board

After connecting PMSM motor, connect power supply adaptor to connector J15 of motor control board. See next slide for detail connection diagram. The colored LED on LPC1549 Xpresso board blinks, changing between blue and white color.

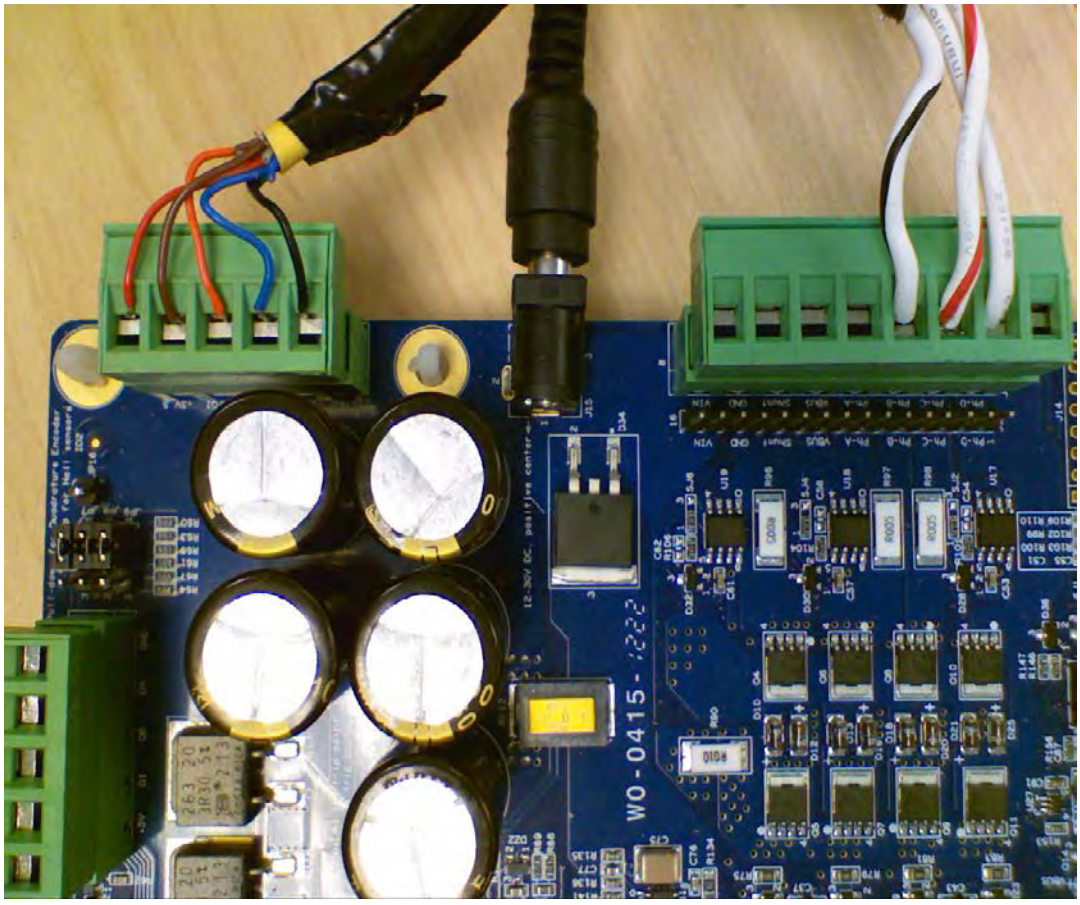

Fig 21. Teknic PMSM motor connections

Fig 22. PMSM motor control demo setup

3.5 PMSM motor control signal details

The PMSM FOC software performs initialization of the LPC1549 microcontroller. LPC1549 microcontroller resources which are used in this application note are listed below:

Table 4. LPC15xx resources

Item	Resource	Usage
1	UART0 connected to USB	RS232 Serial Console on USB
2	ADC0_0	Link Current or Input Current
3	ADC0_1	Phase C Current
4	ADC0_3	Phase B Current
5	ADC0_4	Phase A Current
6	ADC1_1	Phase A Terminal Voltage
7	ADC1_2	Phase B Terminal Voltage
8	ADC1_5	Phase C Terminal Voltage
9	SCT0_0	PWM Phase A Low
10	SCT0_1	PWM Phase A High
11	SCT0_2	PWM Phase B Low
12	SCT0_3	PWM Phase B High
13	SCT0_4	PWM Phase C Low
14	SCT0_6	PWM Phase C High
15	SCT0_7 Internally Mapped	ADC0 and ADC1 Conversion Sequence Trigger
16	PIO0_2	Quadrature encoder Sensor Input A
17	PIO0_30	Quadrature Encoder Sensor Input B
18	PIO0_17	Quadrature Encoder Sensor Index Input
19	DAC0	Output used for Analog Diagnostic
20	PIO0_22	Toggle Output used for Timing Diagnostic

4. PMSM FOC firmware

The firmware is provided in binary with this application note. This section outlines the operation of the PMSM FOC software. This software is intended to run on the LPC15xx family of microcontrollers.

This section addresses the following topics:

1) Rotor Angle Representation
2) Trigonometry Functions and Representation
3) PWM System Representation
4) FOC Motor Control Structure
5) FOC Measured Waveforms
6) FOC Algorithm Timing

This software makes use of floating point, fixed point, and lookup table techniques to enhance overall software performance with LPC15xx microcontrollers.

4.1 Rotor angle representation

The firmware represents the rotor angle as an integer from 0 to Motor count per pole pair (MOTOR_COUNTSPP). MOTOR_COUNTSPP is presently set to 1000. Thus 2π radians are equal to 360 degrees which means 1 degree is around 3 counts.

4.2 Phase current representation

Phase current is monitored with a 0.005 ohm shunt resistor connected on each phase and a 20x gain amplifier.

Current in Amps = ((ADC Value / 16) – ADC Offset) / 62.0

4.3 Processor and conversion timing

The processor runs at 72 MHz clock frequency. The ADC conversion system runs at 36 MHz.

4.4 Trigonometry functions and representation

The control firmware requires the following floating point trigonometric functions:

1) Sine
2) Cosine
3) Arc Tan

These functions are provided using lookup tables. The Sine and Cosine function share a single lookup table which spans angle in counts from 0 to MOTOR_COUNTSPP + (MOTOR_COUNTSPP / 4) + 10. The cosine function does a 90 degree (MOTOR_COUNTSPP / 4) offset of the input angle. The angle is represented as an integer. The sine or cosine value is provided as floating point table lookup.

The arc tan function performs a binary search on a table look-up. Arc tan covers 0 to 180 degrees or 0 to 500 counts. The sign of the numerator determines if the angle is in the first or second half of the circle.

All trigonometric functions perform straight table lookup without interpolation. Using 1000 counts to represent 360 degrees provides enough resolution to tolerate slight table lookup inaccuracy.

4.5 PWM system representation

The PWM system makes use of SCT0. Basic SCT0 clocking is setup such that the PWM frequency is 17.5 kHz. PWM modulation percentages from 0 to 100 percent makes use of PWM values from 0 to 1024. Full modulation is achieved with a PWM value of 1023. Zero modulation is achieved with a PWM value of 0.

SCT0 triggers ADC0 and ADC1 data conversions on every PWM cycle. FOC motor control is performed on every other PWM cycle. On first cycle the forward FOC calculations are made, and on second cycle the estimated angle calculations are made. In this manner we spread out the processing load to run the FOC control algorithm.

4.6 FOC measured waveforms

The scope trace in shown in Fig 23 shows various measured waveforms.

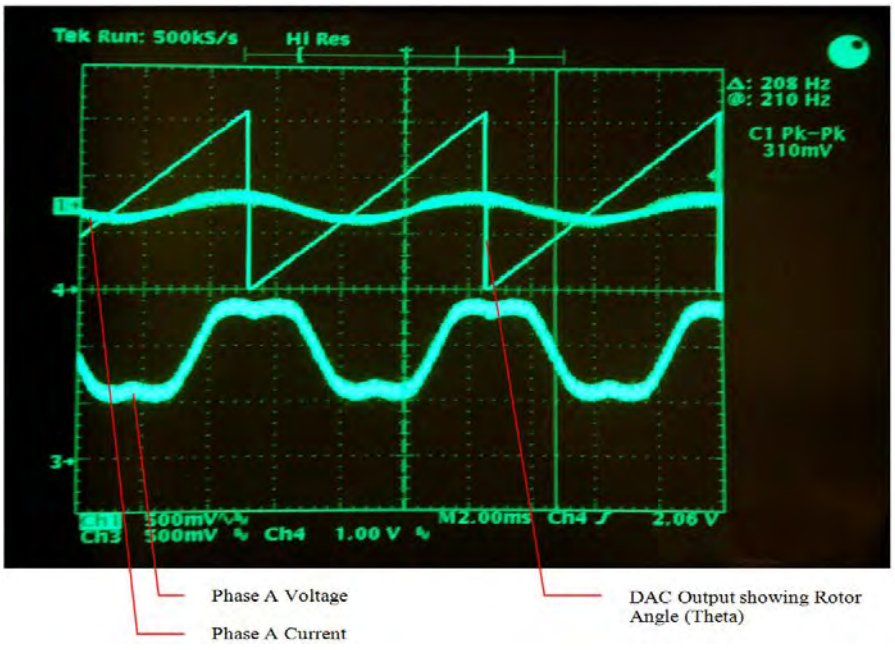

Fig 23. Motor operating with a properly set rotor offset angle

In Fig 23 the scope trace shows motor operating with a properly set rotor offset angle. Observe the symmetry of the voltage waveform.

In Fig 24 the scope trace shows a motor operating with an incorrectly set rotor offset angle. Observe the discontinuities in the voltage waveform.

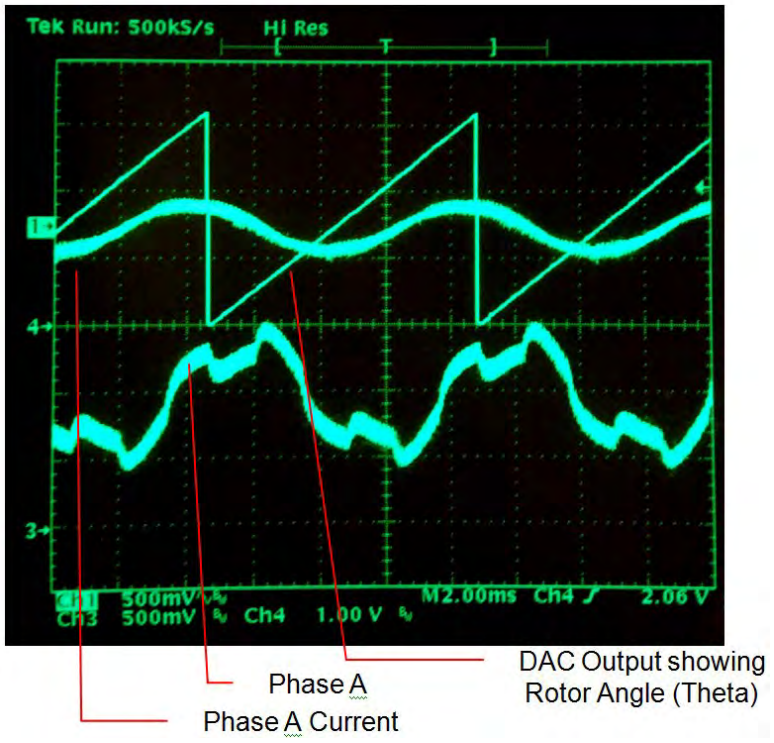

Fig 24. Motor operating with an incorrectly set rotor offset angle

4.7 FOC algorithm timing

The FOC algorithm has been implemented with a mixture of floating point, fixed point and table lookup techniques. These have been used in an attempt to make the code both flexible and for rapid execution when running on a microcontroller which does not have intrinsic floating point features.

The FOC algorithm itself requires approximately 26 usec to run. That includes Inverse Park, SVM, Clark, and Park transforms.

4.8 FOC program statistics

The FOC algorithm has been implemented with a mixture of floating point, fixed point and table lookup techniques. These have been used in an attempt to make the code both flexible and for rapid execution. Code size statistics are shown below:

Table 5. Code size statistics

Item	Description	Value
1	Approx. Flash ROM Code Size including all libraries	25.7 kB
2	Approx. SRAM Storage Size including all libraries	14.0 kB
3	Approx. FOC with PI routines execution time	26uSec Max per PWM Event
4	Approx. Processor Loading	50%

Note: The above statistics include the floating point library and the printf library. Code size and execution times may be greatly reduced by substituting fixed point for floating point operations.

5. PMSM firmware tuning

5.1 Tuning guidelines

This section provides some guidelines for tuning the PMSM firmware for use with a particular motor. This chapter is written assuming that you have experience with PI Controllers, PMSM Motors, BLDC Motors and motor concepts. Motor tuning requires time and patience. This chapter contains some useful information and guidelines. Actual motor tuning may require modification of this firmware and experimentation with the various firmware parameters.

We highly recommend using a motor with a quadrature position sensor with resolution of at least 1000 counts per magnetic 360 degree revolution.

The motor can be tuned either by using a GUI or by directly communicating with a terminal program such as Tera Term.

5.2 Graphical user interface

The graphical user interface is capable of configuring different FOC parameters and displaying various signals. The purpose of the tool is to create a debug environment for developing FOC. Graphics user interface contains a common interface and two tabs

The following sections describe the controls and displays available on the GUI.

Please note that the firmware works with GUI if the firmware is compiled with GUI_Interface=1.

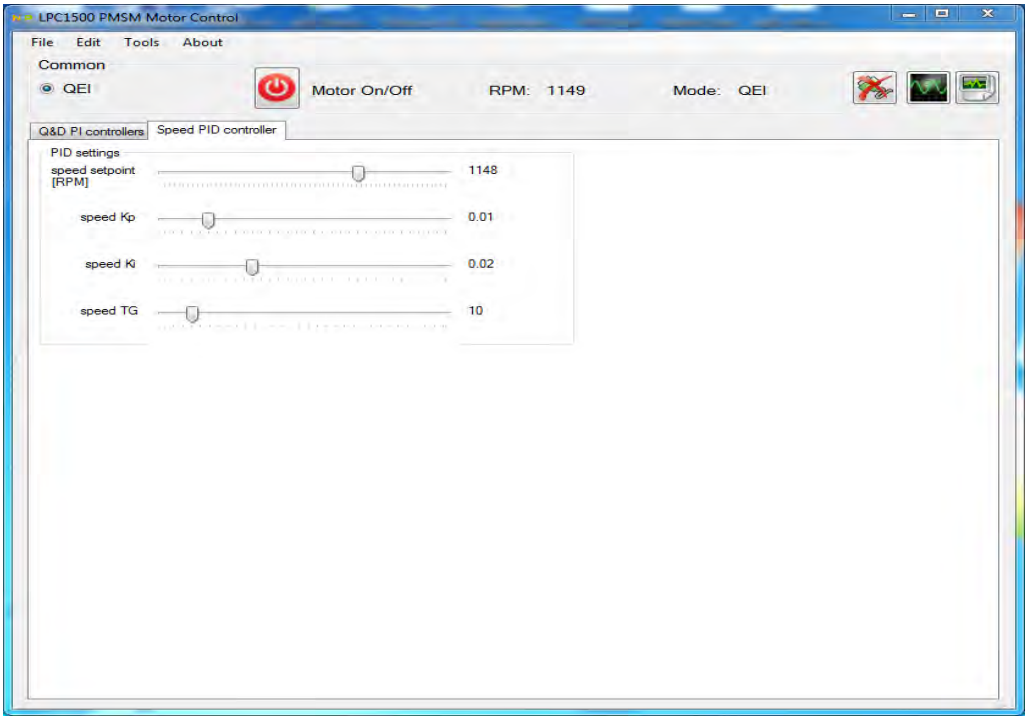

Fig 25. GUI interface

5.2.1 Common group

The common group contains the following control and displays.

Opens the COM port. Settings can be changed under menu edit → COM settings.

When the COM is opened, the button view changes to 'disconnect'. This closes the COM port.

Opens a new scope window. The number of scopes is unlimited. This feature is also available under menu Tools → Add scope. See section 5.2.5.

Opens the datalogger window. This feature is also available under menu Tools → Datalogger. See section 5.2.6.

Other than the above four Widgets, there is a Motor On/Off control and a radio button to switch modes from sensored QEI to sensorless mode.

5.2.2 Q&D PI controllers tab

On the first tab, the parameters of the Q&D PI controllers are displayed. The PID and PIQ parameters are internal torque production parameters of the FOC algorithm. Start with the values provided as the default. PID is the PI Algorithm for the D-axis. PIQ is the PI Algorithm for the Q-axis. Varying these parameters will directly affect torque production, motor efficiency and performance with respect to changing load.

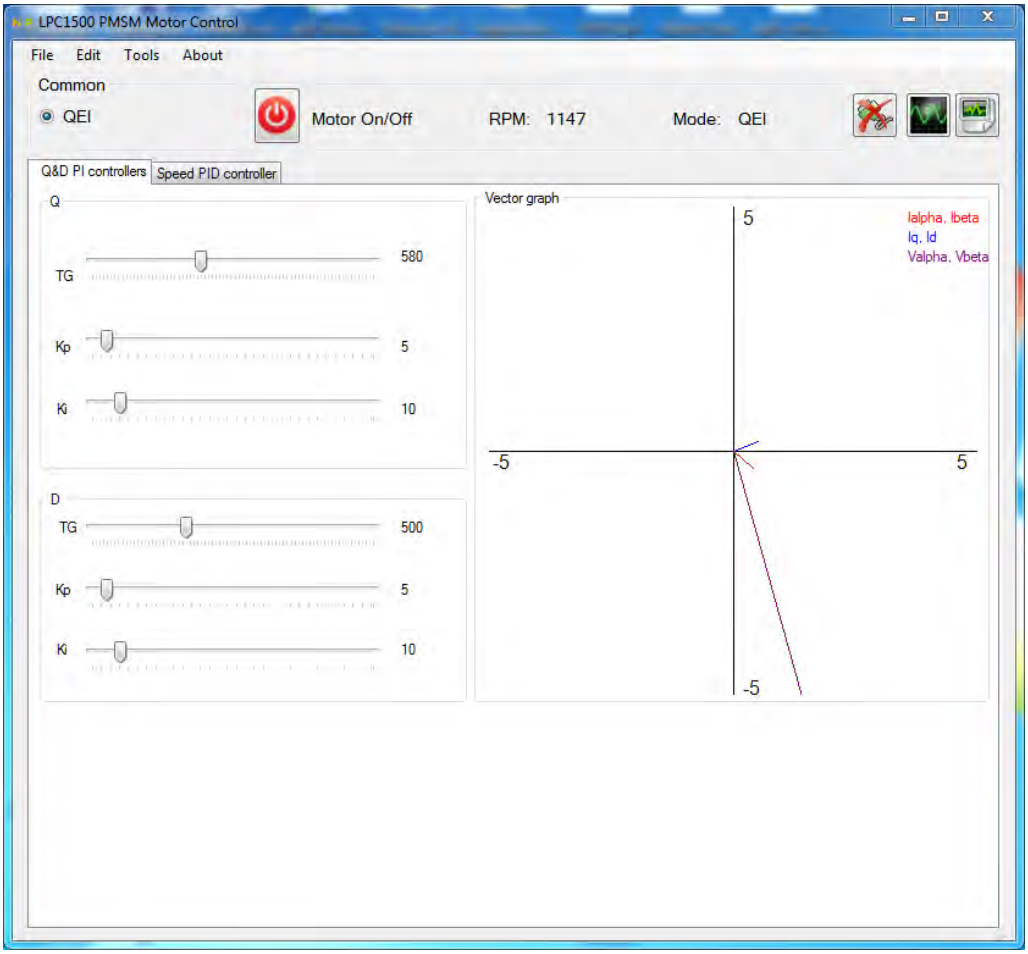

Fig 26. Q&D PI controller tab

Using the sliders can modify the values of K_p and K_i and TG for Q and D. The value is instantly transmitted to the demo kit when the COM connection is opened. The vector graph displays the current vectors $I_{\alpha\text{-}\beta}$, $I_{q\text{-}d}$ and $V_{\alpha\text{-}\beta}$. The red current vector $I_{\alpha\text{-}\beta}$ is controlled perpendicular to the rotor angle by the magnitude and angle of the purple voltage vector.

5.2.3 Speed PID controller tab

On the second Speed PID controller tab, the parameters of the speed PID controller can be configured. The RPM parameters affect the constant velocity PI Algorithm. A crude means by which tuning may be accomplished (for any PI system) is to set the 'IGAIN' parameter to zero thereby turning off the integrator. Increase the 'PGAIN' parameter until the system oscillates. Divide the value of the PGAIN parameter just at oscillation by half and use that as the actual PGAIN figure. Now increase the IGAIN parameter to smoothen the operation and achieve the desired performance making little or no change to the PGAIN parameter. Use the TGAIN parameter to vary the integrator time period. TGAIN is actually 1/TGAIN representing the integrator fraction used in the PI algorithm.

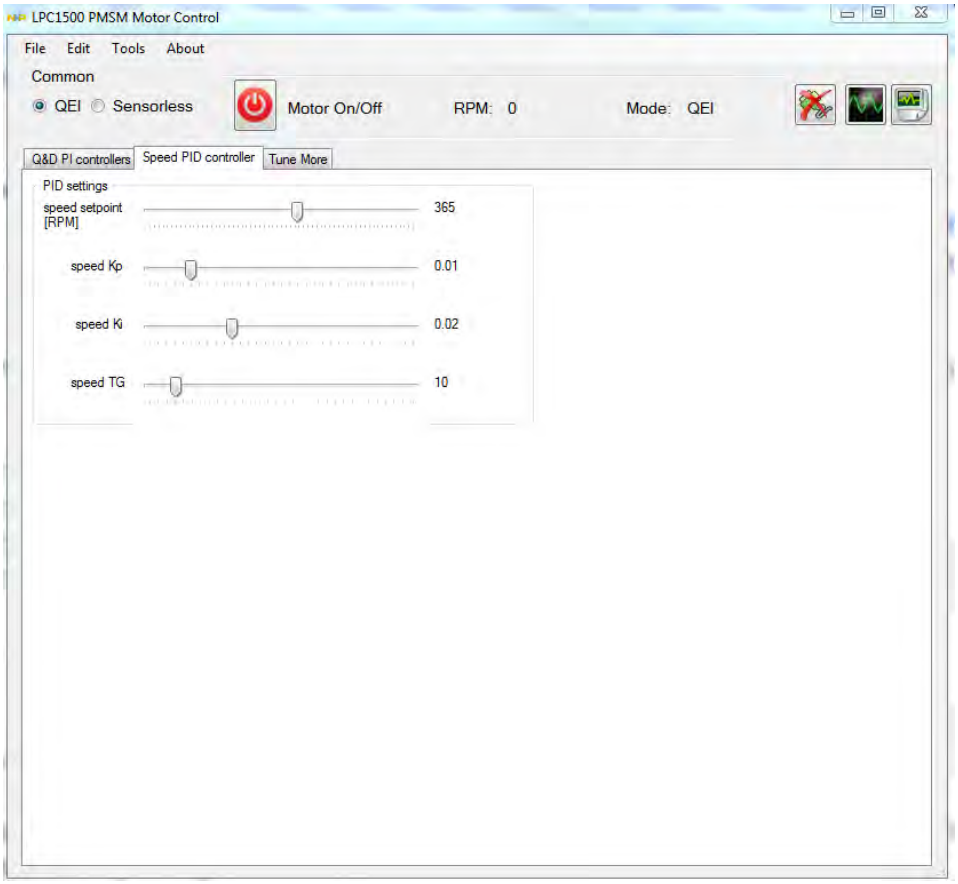

Fig 27. Speed PID controller tab

5.2.4 File open

Under menu file the options are given to open motor parameters. All the PI(D) controller values and the parameters under the virtual motor tab can be loaded to an XML motor data file.

5.2.5 Scope window

A scope window can display variables that are constantly transmitted by the demoboard. A second channel is available by clicking the Channel 2 tab. By default it is set to <no signal>. The range of the output view is configurable by manually entering the maximum and minimum value. Auto ranging is also possible.

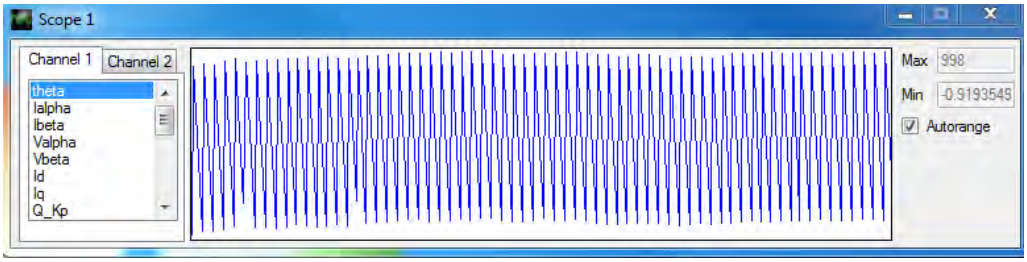

Fig 28. Scope window

5.2.6 Datalogger

To export the received signals to a text file and do further analysis in a spreadsheet program, a datalogger is available. Adding signal blocks and defining a column separator character can configure the output.

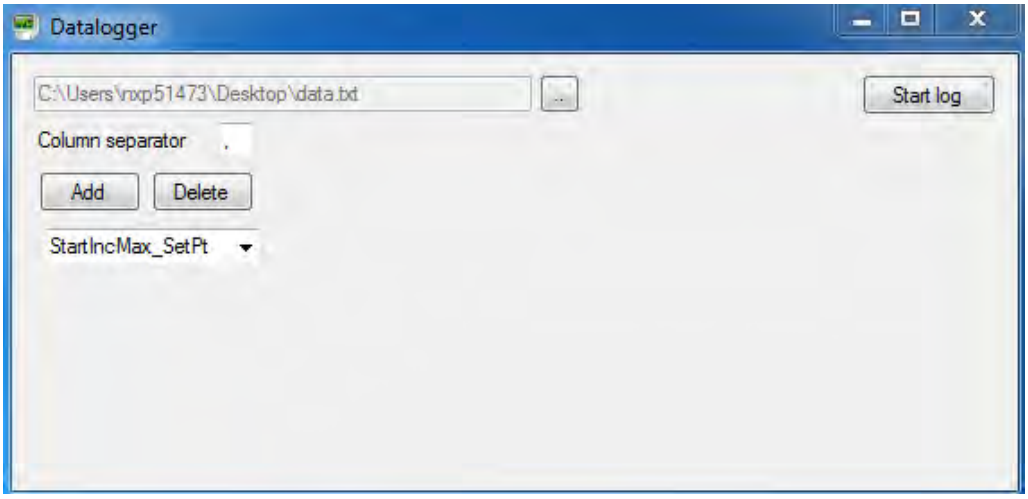

Fig 29. Datalogger

5.3 Tuning using Tera Term

To tune firmware using this method build firmware after defining GUI_Interface=1 to GUI_Interface=0 in IAR IDE as shown in Fig 30.

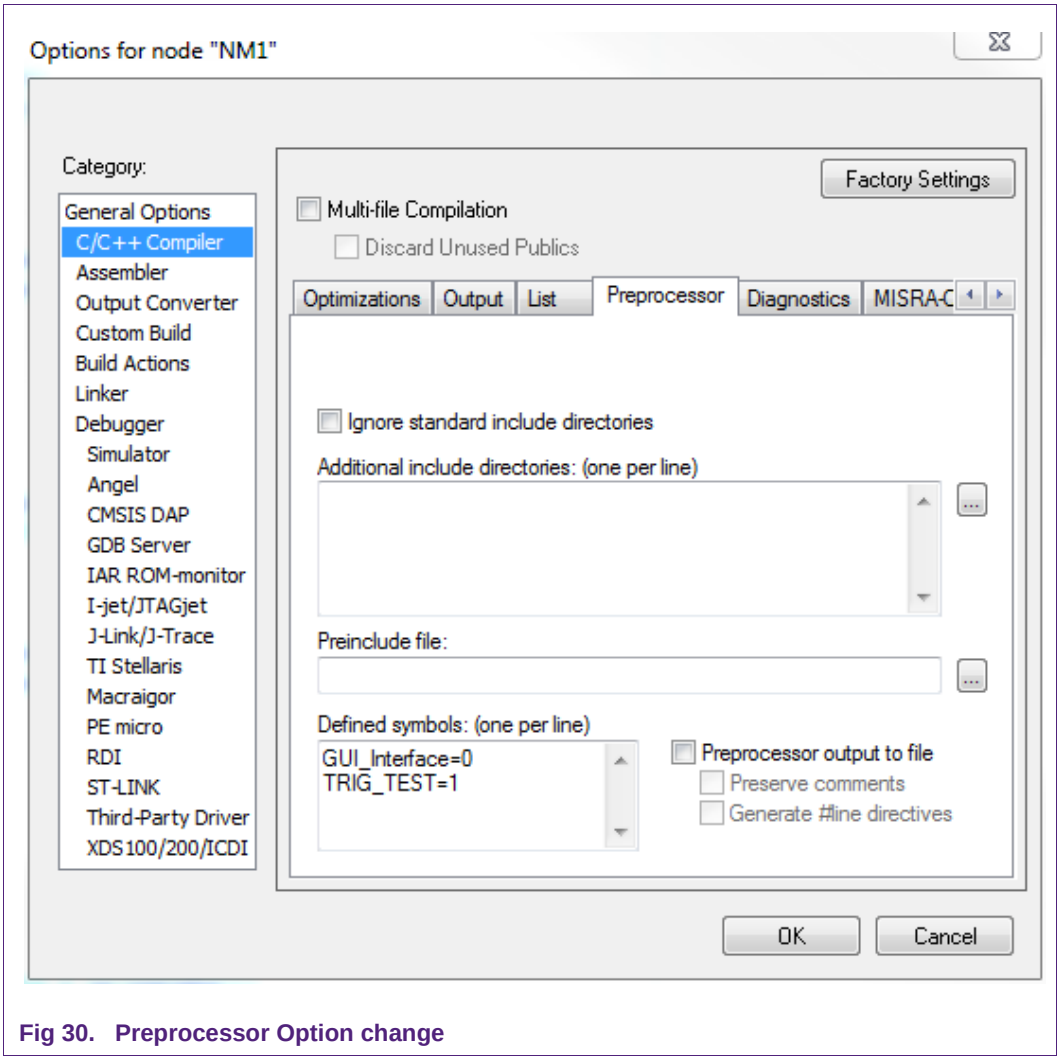

Fig 30. Preprocessor Option change

This chapter outlines the operation of the LPC1500 PMSM control software. Once the hardware is setup as described in the previous chapter, power on the Xpresso Motor Controller Board.

Start your favorite the host computer's terminal emulation program (we are using Tera Term), with baud rate 15200 bits/second, no parity, 1 stop bit. Reset the board. The NXP motor control firmware will sign on.

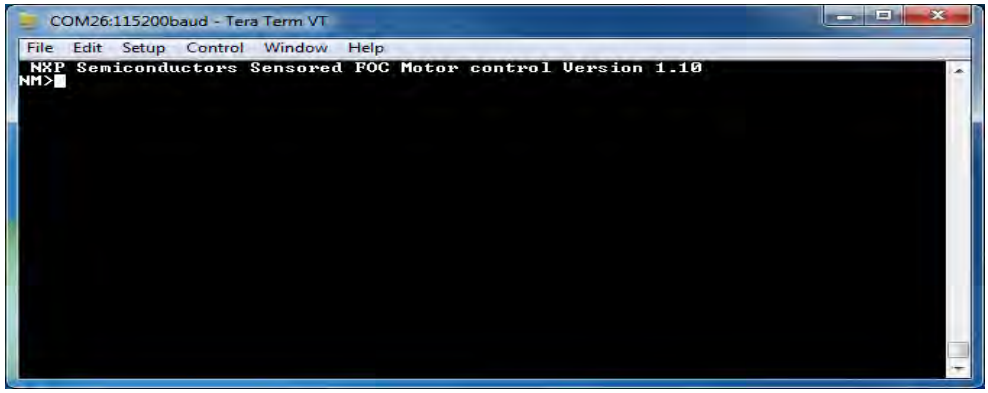

Fig 31. Terminal Window at sign On

Type question mark (?) after prompt (>) and press enter key, you will see all available commands as shown in the below screen.

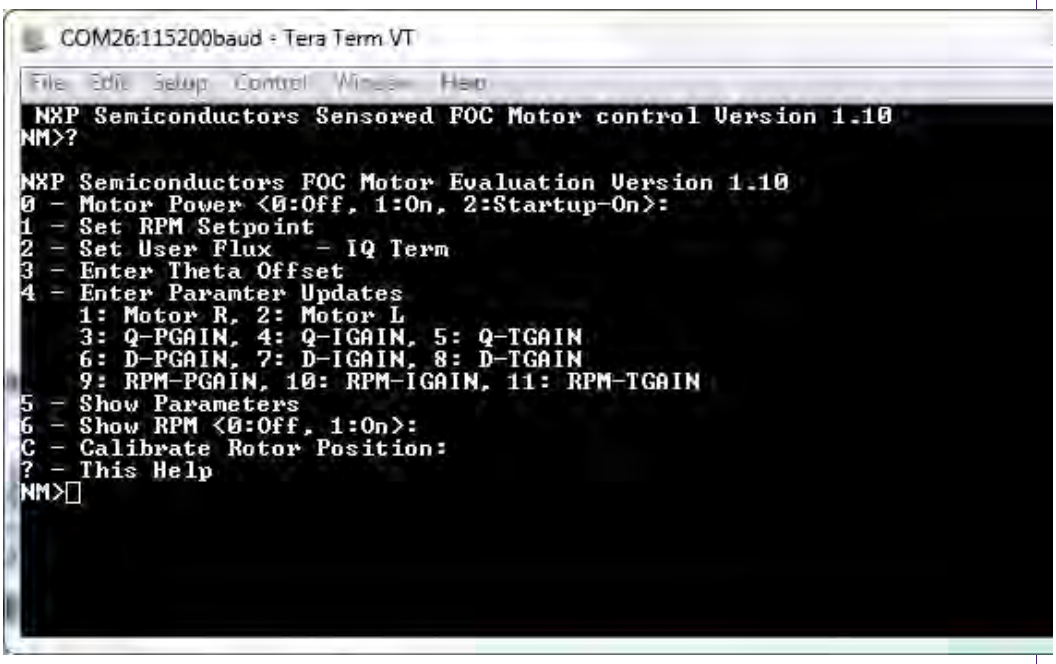

Fig 32. Available commands

Below commands are available.

5.3.1 Command 0: Motor Power Off / Start / On Command

This command applies power to the Motor. At start-up, the NM1 software disables the motor power drivers. The motor is free to turn.

Type ‘0’ <space> ‘0’ <CR> to disable the motor.

Type ‘0’ <space> ‘1’ <CR> to power the motor without running the startup sequence.

Type ‘0’ <space> ‘2’ <CR> to power up the motor running the startup sequence first. RPM value will be the startup RPM parameter.

This command simply starts the motor drivers. The motor will not actually run without inputting a non-zero rpm value.

5.3.2 Command 1: Set RPM command

This command sets the RPM for motor operation. Valid entries are integers from 0 to 3000. This command provides the setpoint for a PI-controlled constant velocity routine. This command sets the constant velocity in RPM.

For a typical low-speed motor startup, enter the following:

Type ‘1’ <space> ‘200’ <CR>

At power up, the default value is 0.

5.3.3 Command 2: Set User Flux Command

This command sets the flux value for motor operation. Valid entries are integers from 0 to 1023. The User flux Command provides input to the FOC algorithm’s flux control input.

Typical PMSM motor operation should have this value fixed at 0. This value may be modified to perform field weakening operation.

To perform field weakening, enter the following:

Type ‘2’ <space> ‘-50’ <CR>

The flux command sets the startup sequence flux as well as the running torque. At power up, the default torque value is 0. In general, you should not have to modify this value.

5.3.4 Command 3: Set Theta Offset Command

This command causes a constant offset to the rotor position angle. At high speed, adding a small theta offset can improve motor operating efficiency.

To set the Theta Offset, enter the following:

Type ‘3’ <space> ‘20’ <CR>

The above command adds 20 counts to the rotor angle position, thereby advancing the timing. The default Theta Offset value is 0.

Note: Theta offset is enumerated in counts. 1000 counts is equal to 360 degrees of offset – or a complete magnetic revolution. Thus 20 counts provides (20/1000) * 360 = 7.2 Degrees of offset.

Note: The Calibrate Rotor Position command calculates the optimum rotor offset value and sets that value automatically. The Show Parameters command displays the rotor angle offset.

5.3.5 Command 4: Enter Operating Parameters Command

This command sets the operating parameters. Eight operating parameters may be entered. These parameters control the internal PI routines for Q and D current as well as Motor resistance and inductance setpoints.

The operating parameters are express as floating point numbers.

To set the Resistance Value, enter the following:

Type ‘4’ <space> ‘1’ <space> ‘1.45’ <CR>

The above command sets the resistance value to 1.45 ohms. The default Resistance value is 0.72 Ohms. The default value may be adjusted using a software define.

The Teknic motor seems to run best with values from 0.72 to 1.45.

The NXP motor seems to run best with values from 3.0 to 10.0.

5.3.6 Command 5: Show Operating Parameters Command

This command displays the current operating parameters. An example of the output of this command is shown below.

Type '5'

1: Motor R = 1.4500

2: Motor L = 0.2000

3: Q PGAIN = 5.0000

4: Q IGAIN = 10.0000

5: Q TGAIN = 500.0000

6: D PGAIN = 5.0000

7: D IGAIN = 10.0000

8: D TGAIN = 500.0000

9: RPM PGAIN = 0.0100

10: RPM IGAIN = 0.0200

11: RPM TGAIN = 10.0000

Rotor Angle Offset Est0 = -54

5.3.7 Command 6: Show RPM Display Off/On Command

This command turns the periodic RPM display Off or On. When On, the motor's RPM is displayed periodically on the console. RPM is based upon the number of motor poles and the integrated rotor position.

To set the RPM Display On, enter the following:

Type '6' <space> '1' <CR>

To set the RPM Display Off, enter the following:

Type '6' <space> '0' <CR>

The RPM Display defaults to Off.

An example is shown below:

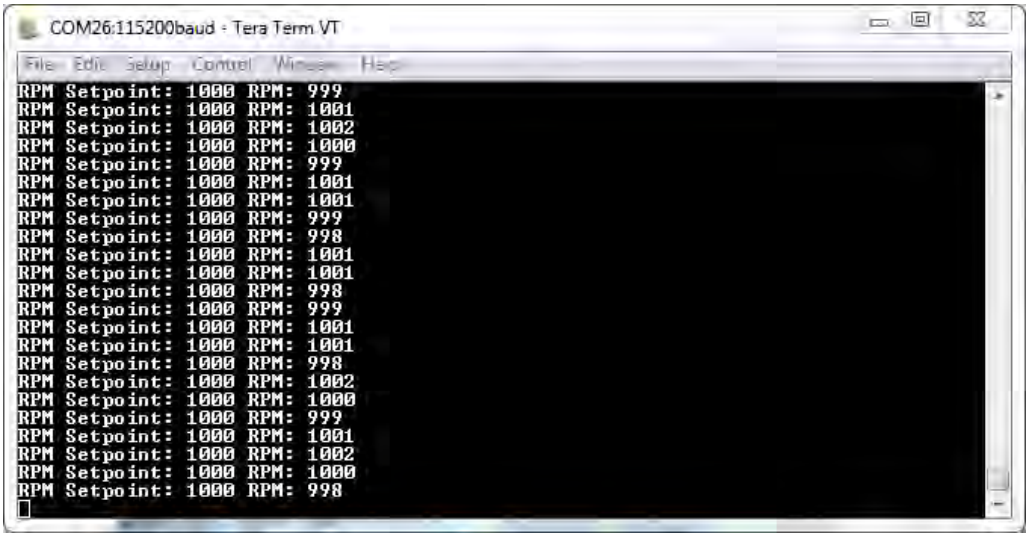

Fig 33. Continuous RPM Display

5.3.8 Command C: Calibrate Rotor Position Command

This command calculates the proper rotor offset. The routine spins the motor, then locks the rotor to 0 degrees, then calculates the offset and then releases the motor.

Before running this command – manually rotate the motor through at least two complete revolutions.

To calibrate the rotor position, enter the following:

Type ‘C’ <CR>

Note: Once this routine completes you may display the rotor offset using the show parameters command.

6. Legal information

6.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

6.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

6.3 Trademarks

Notice: All referenced brands, product names, service names and trademarks are property of their respective owners.

7. Contents

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 2 June 2014
Document identifier: AN11517