AN11523

BLDC motor control using the LPC15xx SCTimer/PWM

Rev. 3.1 — 17 April 2014 Application note

Document information

Info	Content
Keywords	SCTimer/PWM, BLDC, motor control, LPC15xx
Abstract	This application note describes an implementation of brushless DC motor control using the SCTimer/PWM.

Revision history

Rev	Date	Description
3.1	20140417	• Changed number for motor control kit from OM13067 to OM13068.
3	20140414	• Updated Keil IDE; added LPCXpresso IDE.
2.1	20140312	• Updated code headers.
2	20140303	• Updated Fig 6, Fig 7, Fig 9, Fig 10, and Fig 12. • Updated Section 4.1, paragraph 5.
1.1	20140226	• Minor updates.
1	20140221	• Initial version.

Contact information

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

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

1. Introduction

The LPC15xx are ARM Cortex-M3 based microcontrollers that operate at CPU frequencies of up to 72 MHz. The LPC15xx devices include up to 256 kB of flash memory, 32 kB of ROM, a 4 kB EEPROM and up to 36 kB of SRAM.

The LPC15xx peripherals include one full-speed USB 2.0 device, two SPI interfaces, three USARTs, one Fast-mode Plus I^2C-bus interface, one C_CAN module, a PWM/timer subsystem with four configurable, multi-purpose State Configurable Timers (SCTimer/PWM) with input pre-processing unit, a Real-time clock module with independent power supply and dedicated oscillator, two 12-channel/12-bit, 2 Msamples/sec ADCs, one 12-bit, 500 kSamples/sec DAC, four voltage comparators with internal voltage reference, and a temperature sensor. A DMA engine can service most peripherals.

This application note demonstrates how to implement six-step commutation or brushless DC motor control on the LPC15xx family of microcontrollers. The LPC15xx devices are equipped with an SCTimer/PWM unit which reduces CPU utilization during motor control while also reducing development time. The SCTimer/PWM is highly integrated among the analog peripherals. The switch matrix supports flexible assignment of the SCTimer/PWM functions to the pins.

This application note also intends to be a reference and starting point for motor control system developers using LPC15xx family of microcontrollers.

2. Brushless DC motor principle

Brushless DC motors consist of a permanent magnet rotor with a three-phase stator winding. As the name implies, BLDC motors do not use brushes for commutation; instead, they are electronically commutated. Typically, three Hall sensors are used to detect the rotor position and commutation is based on these sensor inputs.

In a brushless DC motor, the electromagnets do not move; instead, the permanent magnets rotate and the three-phase stator windings remain static. This solves the problem of transferring current to a moving rotor. In order to do this, the brush-commutator assembly is replaced by an intelligent electronic “controller”. The controller performs the same power distribution as found in a brushed DC motor, but it uses a solid-state circuit rather than a commutator/brush system.

The torque generated by the motor depends on the applied current, which is the result of the applied voltage. The achieved speed will be linked to the acceleration of the motor up to the condition at which the delivered torque equals the load torque. The effective current (and thus the torque) will drop at increasing speed, due to the BEMF generated by the motor, and this effect will be reflected in a voltage dependent speed.

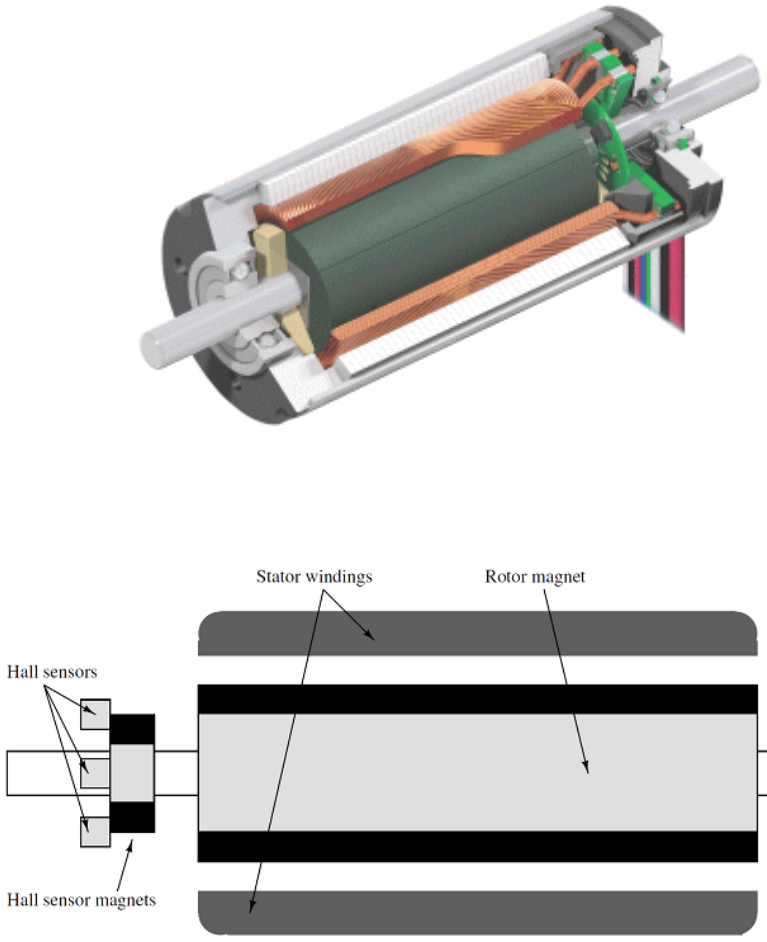

Fig 1. BLDC motor

2.1 Electrical commutation

A BLDC motor is driven by voltage strokes coupled with the given rotor position. These voltage strokes must be properly applied to the active phases of the three-phase winding system so that the angle between the stator flux and the rotor flux is kept close to 90° to maximize torque. Therefore, the controller needs some means of determining the rotor's orientation/position (relative to the stator coils).

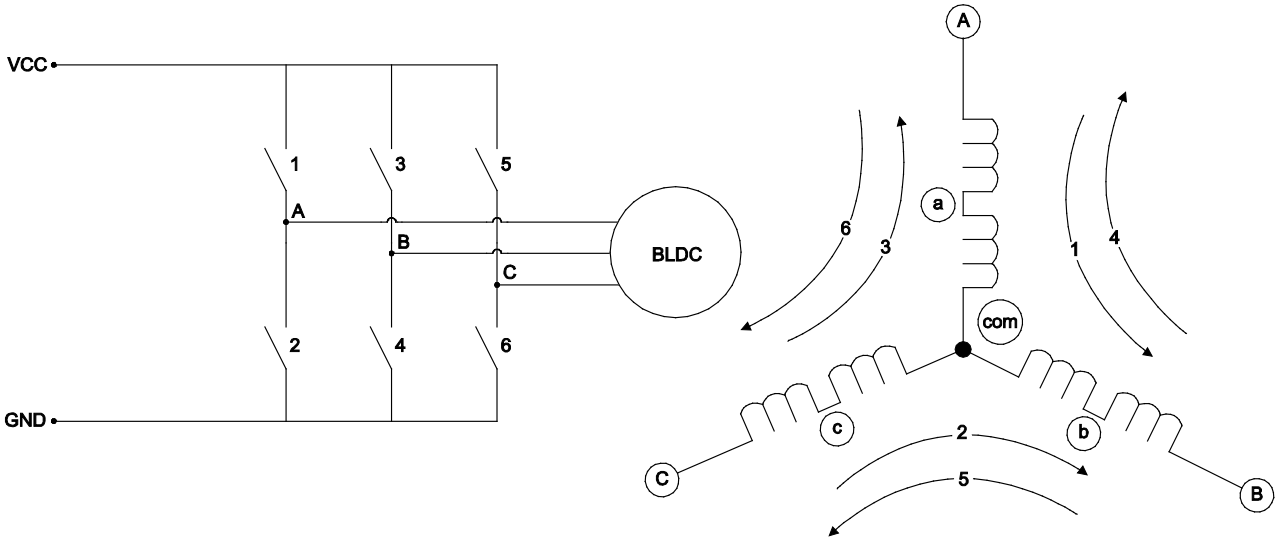

Fig 2. Principle of activation for BLDC motor windings

Fig 2 depicts a systematic implementation on how to drive the motor coils for a motor rotation. The current direction through the coils determines the orientation of the stator flux. By sequentially driving or pulling the current through the coils the rotor will be either pulled or pushed. A BLDC motor is wound in such a way that the current direction in the stator coils will cause an electrical revolution by applying it in six steps. As shown in Fig 2 each phase driver is pushing or pulling current through its phase in two consecutive steps. These steps are shown in Fig 3 and Fig 4.

Sequence number	Switching interval	Phase current A	Phase current B	Phase current C	Switch closed
0	0^o – 60^o	+	-	OFF	1 , 4
1	60^o – 120^o	+	OFF	-	1 , 6
2	120^o – 180^o	OFF	+	-	3 , 6
3	180^o – 240^o	-	+	OFF	3 , 2
4	240^o – 300^o	-	OFF	+	5 , 2
5	300^o – 360^o	OFF	-	+	5 , 4

Fig 3. Example of commutation steps

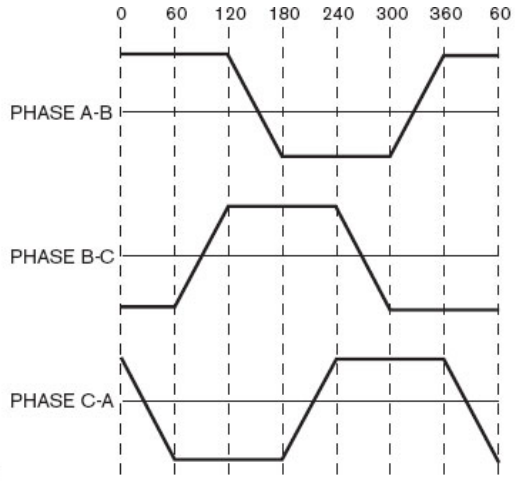

Fig 4. Six-step/trapezoidal commutation

The above switching sequence is called trapezoidal commutation. Fig 5 shows the relation between the different naming definitions for the same type of motor control (six-step commutation over Hall A/Hall B/Hall C sensor edges, block commutation ia / ib / ic and trapezoidal commutation Ea/ Eb/ Ec.

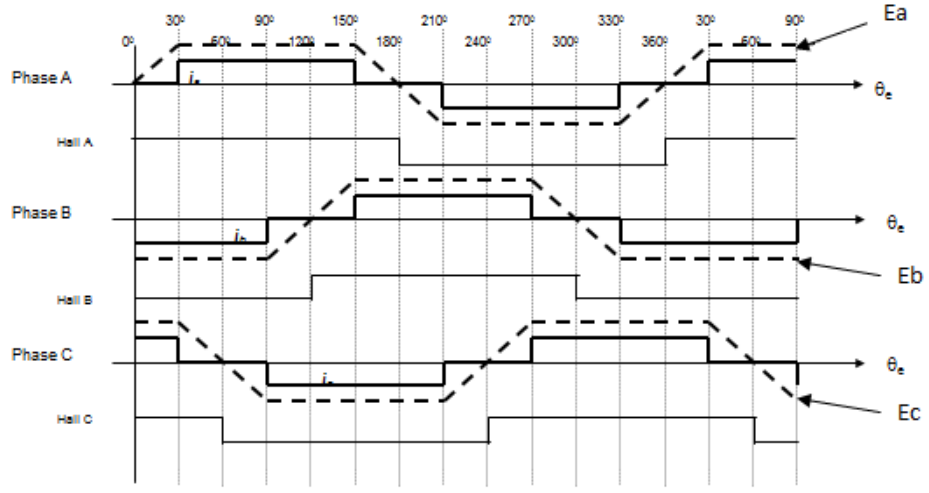

Fig 5. Example of switching sequence with sensors and phases

2.2 Revolution speed control

Varying the voltage across the motor can simply control the rotor speed. This can be achieved by Pulse Width Modulation (PWM) of the phase voltage. By increasing or decreasing the duty-cycle, more or less current per commutation step will flow through the stator coils. This affects the stator flux and flux density, which changes the force between the rotor and stator.

This means that the rotation speed is determined by the load of the rotor, the current during each phase and the voltage applied.

2.3 Position feedback

The rotor feedback position can be accomplished through a couple of techniques. The most commonly used technique is the digital Hall sensor feedback, but other techniques include using an encoder or even eliminating sensors entirely. This application note will only focus on the Hall sensor feedback and will not explore sensorless operation.

2.3.1 Hall sensor feedback

The Hall sensors are placed such that they generate an edge at each switching interval. This makes it feasible to determine the current rotor orientation, in order to activate each phase in the right sequence.

3. SCTimer/PWM

The SCTimer/PWM combines the features of a timer and a state machine making possible to develop 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 gets 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.

For example, in comparison with a classic timer or PWM generation block, it is possible to define a more complex sequence of events, having the SCTimer/PWM evolve autonomously through the state machine over time, in response to the time-based or I/O defined events. This typically implies that less CPU interventions are needed, since the SCTimer/PWM is able to handle most of the control sequence in hardware. At constant speed, when PID is not used there is no CPU intervention. The CPU could be used for any other function during this period.

Additionally, the SCTimer/PWM can define events which evaluate either the level or the edges of an input signal. In some scenarios this task would be too expensive in terms of CPU resources to be performed in software, by periodically polling the input line with the CPU in order to detect changes.

Even defining an interrupt to detect the signal change on the input might introduce an excessive latency at the system level, for performing the required response to the event. The SCTimer/PWM efficiently solves this issue, by guaranteeing jitter free response to an input signal fast.

Notice that the SCTimer/PWM can be clocked at the CPU speed, which implies (at 72 MHz) a resolution up to 13.8 ns. Higher average resolution is achieved on the match registers by using a dithering mechanism. At the start of each new SCTimer/PWM counter cycle, the dither engine determines which matches are to be delayed by one clock during the coming counter cycle. This results in a unique dither pattern for each match register.

For more details, please refer to the device user manual.

4. Application requirements

For this application note the following high level features are provided:

1. One dedicated time-based activation (deactivation) point for each of the power line pairs (bridge phases). In any sector, the circuit needs to drive a specific pair of PWM phases at any time.
2. The ability to detect crossing of sectors during the motor rotation, in order to change the pair of PWM phases which needs to be active. This can be done by observing the Hall sensor inputs, so a total of three inputs for the Hall sensors are required.
3. Provide six digital output lines, one per each PWM phase which needs to be generated.

4.1 BLDC motor unidirectional

Fig 6 shows the SCTimer/PWM state machine configuration for controlling the brushless DC motor in one direction:

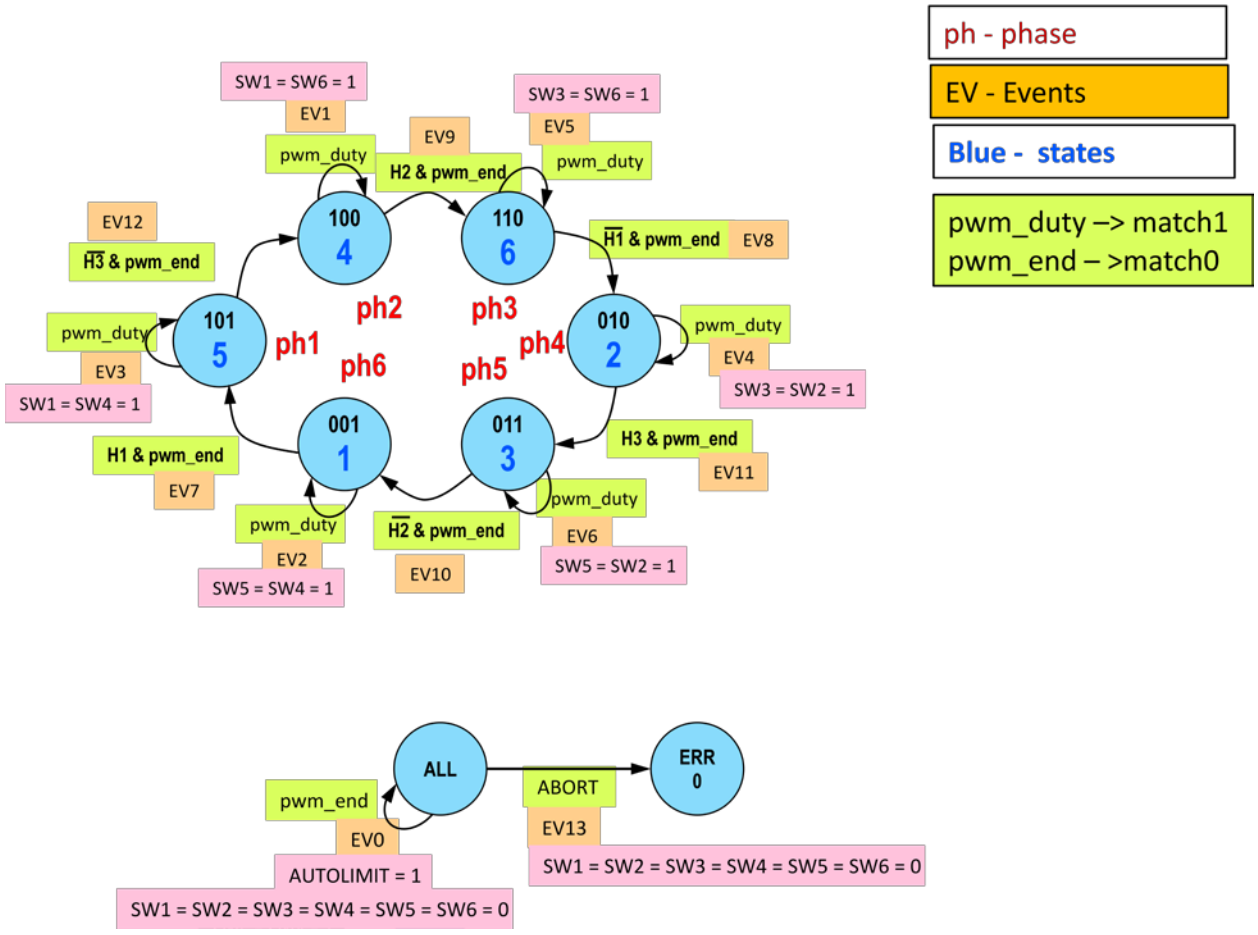

Fig 6. State diagram for one direction

PHASE	HALL			STATE	EVENT	DRIVE	
	H1	H2	H3				
1	1	0	1	5	3	SW1	SW4
2	1	0	0	4	1	SW1	SW6
3	1	1	0	6	5	SW3	SW6
4	0	1	0	2	4	SW3	SW2
5	0	1	1	3	6	SW5	SW2
6	0	0	1	1	2	SW5	SW4

Fig 7. State and event assignments

In the state diagram (Fig 6), the SCTimer/PWM input/output signals are shown in green, the SCTimer/PWM events are represented by arrows and the event names are shown in orange, the effects of the output and signals are shown in pink. When an event is active, some actions on the defined signals are performed. Fig 7 shows the state assignments and event assignments for each Hall sensor signal. Fig 8 shows the various events, states and actions during each event.

Event	Condition		State	Action	New state
0	pwm_end	match 0	ALL	SW1/6 = 0 AUTOLIMIT	No change
1	pwm_duty	match 1	4	SW1 = SW6 = 1	1 (no change)
2	pwm_duty	match 1	1	SW5 = SW4 = 1	2 (no change)
3	pwm_duty	match 1	5	SW1 = SW4 = 1	3 (no change)
4	pwm_duty	match 1	2	SW3 = SW2 = 1	4 (no change)
5	pwm_duty	match 1	6	SW3 = SW6 = 1	5 (no change)
6	pwm_duty	match 1	3	SW5 = SW2 = 1	6 (no change)
7	H1 && pwm_end	SCT0_IN0 && match 0	1	-	State + 4
8	!H1 && pwm_end	!SCT0_IN0 && match 0	6	-	State - 4
9	H2 && pwm_end	SCT0_IN1 && match 0	4	-	State + 2
10	!H2 && pwm_end	!SCT0_IN1 && match 0	3	-	State - 2
11	H3 && pwm_end	SCT0_IN2 && match 0	2	-	State + 1
12	!H3 && pwm_end	!SCT0_IN2 && match 0	5	-	State - 1
13	ABORT	!SCT0_IN3	ALL	SW1/6 = 0 HALT = 1	0 (error state)
14					
15					

Fig 8. Events, state transitions and switching actions

The states of the associated state machine are represented by bubbles and the state assignment is included in the circle. The ALL state is the default entry state of the 32-bit SCTimer/PWM. The state machine is responsible for managing the generation of the PWM signals and to monitor the Hall sensor inputs to detect sector crossings.

The Hall sensors are physically connected on the motor in such a way that there is always just one sensor level changing, when moving from one sector to the next. The motor speed regulation can be done by the application by changing the match register values as appropriate, in order to modify the timing of each event (thus its duty cycle).

The SCTimer/PWM starts counting. When the time based event ‘EV0’ is triggered in state ALL, all the switches SW1 to SW6 are open (logic 0). The match1 reload register is loaded with the duty cycle value and is used to manage the generation of the PWM signal, along with the match0 reload register which is loaded with the PWM period value (pwm_end). Assume the motor is initially at state1 (001) and event EV2 is triggered. At the end of PWM period (pwm_end) , the event (*H1&&pwm end*) is evaluated. When the level on Hall sensor H1 goes from low to high, the state changes from state1(001) to state5(101). In state5(101), event EV3 is triggered. At the end of PWM period , event (!*H3&pwm_end)* is evaluated. When the level on Hall sensor H3 goes from high to low , the state changes to state4(100) and event EV1 is triggered in this state. At the end of PWM period , event *(H2& pwm_end)* is evaluated and when the Hall sensor H2 goes from low to high, the state changes to state6(110) and the event EV5 is triggered in this state. In this way , the states of the SCTimer/PWM change based on the Hall sensor inputs.

In this fashion, the system will autonomously switch between the states and keep the motor rotating, without any software intervention. The speed of rotation will be proportional to the duty cycle resulting from the match values being programmed by the application at runtime.

Each event can modify its counter STATE value. If more than one event associated with the same counter occurs in a given clock cycle, only the state change specified for the highest-numbered event among them takes place.

Note: at the beginning the application will have to detect the level of the Hall sensors while the motor is stopped and preset the SCTimer/PWM state machine to the associated state (sector). Having done this, when started the SCTimer/PWM will begin to drive the appropriate PWM signals in order to make the motor rotate.

4.2 BLDC motor bidirectional

Fig 9 shows the SCTimer/PWM state machine configuration for controlling the brushless DC motor for bidirectional motion.

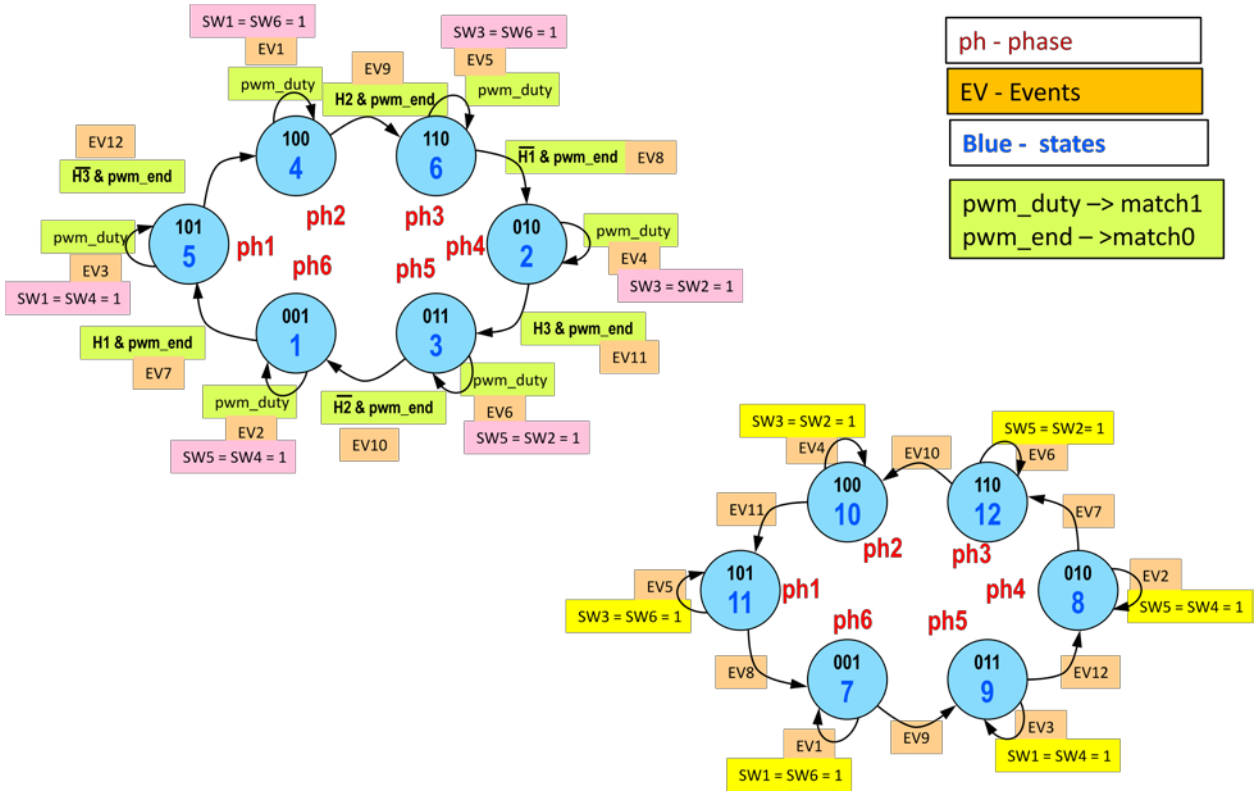

Fig 9. State diagram for bidirection

PHASE	HALL			STATE	EVENT	DRIVE	
	H1	H2	H3				
1	1	0	1	5 & 11	3	SW1	SW4
2	1	0	0	4 & 10	1	SW1	SW6
3	1	1	0	6 & 12	5	SW3	SW6
4	0	1	0	2 & 8	4	SW3	SW2
5	0	1	1	3 & 9	6	SW5	SW2
6	0	0	1	1 & 7	2	SW5	SW4

Fig 10. State and event assignments

For the bidirectional motion, six new states are introduced but no new events are defined. The operation is similar to the unidirectional motion except that an event is triggered in two states. These new states are defined for operating the motor in reverse direction. See Fig 11 for the state transitions and actions taking place for different events.

Event	Condition		State	Action	New state
0	pwm_end	match 0	ALL	SW1/6 = 0 AUTOLIMIT	No change
1	pwm_duty	match 1	4 & 7	SW1 = SW6 = 1	1 (no change)
2	pwm_duty	match 1	1 & 8	SW5 = SW4 = 1	2 (no change)
3	pwm_duty	match 1	5 & 9	SW1 = SW4 = 1	3 (no change)
4	pwm_duty	match 1	2 & 10	SW3 = SW2 = 1	4 (no change)
5	pwm_duty	match 1	6 & 11	SW3 = SW6 = 1	5 (no change)
6	pwm_duty	match 1	3 & 12	SW5 = SW2 = 1	6 (no change)
7	H1 && pwm_end	SCT0_IN0 && match 0	1 & 8	-	State + 4
8	!H1 && pwm_end	!SCT0_IN0 && match 0	6 & 11	-	State - 4
9	H2 && pwm_end	SCT0_IN1 && match 0	4 & 7	-	State + 2
10	!H2 && pwm_end	!SCT0_IN1 && match 0	3 & 12	-	State - 2
11	H3 && pwm_end	SCT0_IN2 && match 0	2 & 10	-	State + 1
12	!H3 && pwm_end	!SCT0_IN2 && match 0	5 & 9	-	State - 1
13	ABORT	!SCT0_IN3	ALL	SW1/6 = 0 HALT = 1	0 (error state)
14					
15					

Fig 11. Events, state transitions and switching actions

When the counter reaches the end of PWM period (*pwm_end*) the event *'H1 && pwm_end'* is evaluated and a transition is made to state 5 if it is in state 1 or the transition is made to state 12 if the current state is 8. In this way the new 6 states help in making the motor rotate in both the directions.

5. Application setup

5.1 Hardware configuration

The block diagram for the test application is shown in Fig 12.

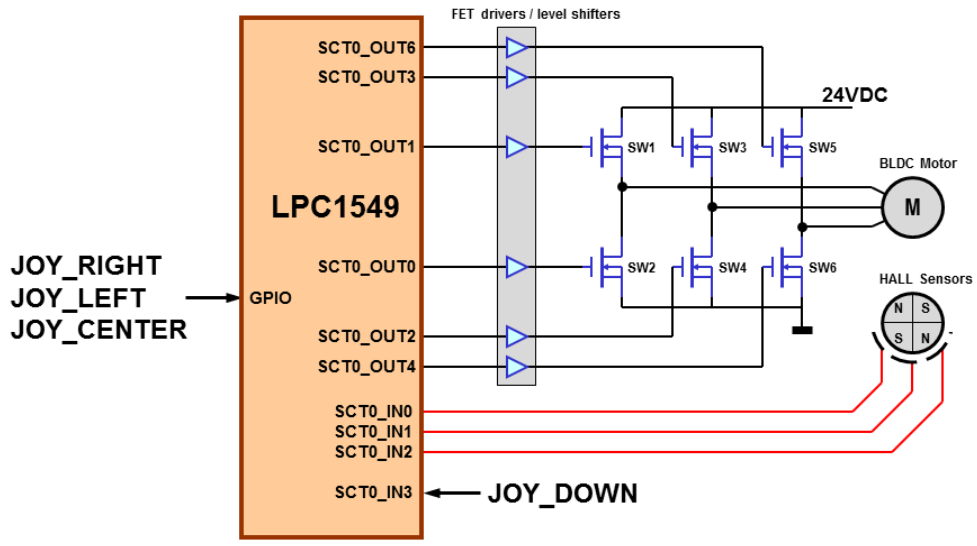

Fig 12. BLDC block diagram for the test application

For the test, the LPC1549 Motor Control Kit (OM13068) has been used:

http://www.nxp.com/demoboard/OM13068.html

The kit also includes a BLDC motor with a Hall sensor mounted.

The board generating the digital control signals for the application is an LPCXpresso board equipped with an LPC1549 device (OM13056).

http://www.nxp.com/demoboard/OM13056.html

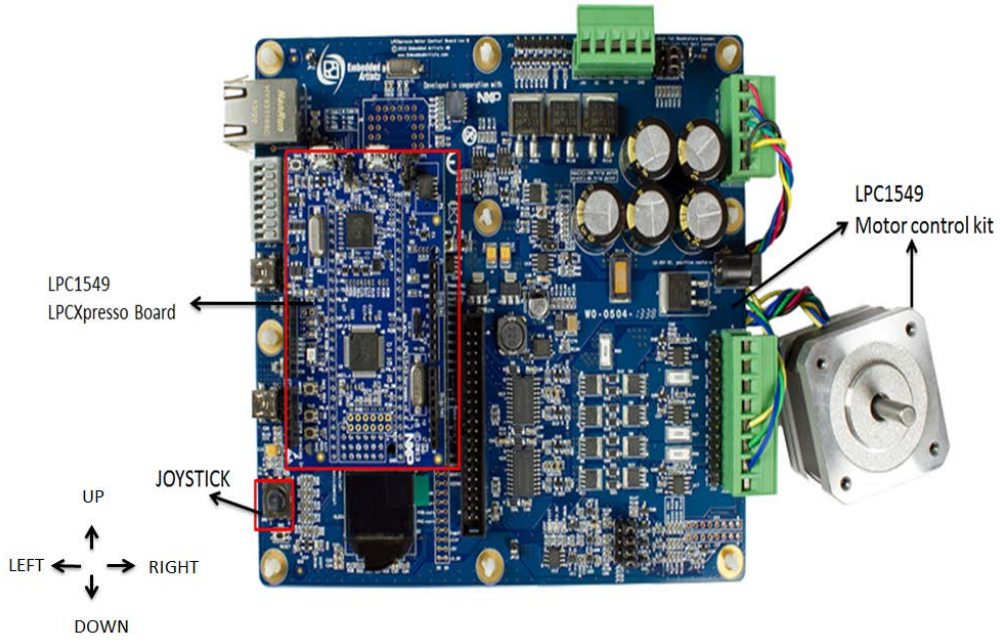

Fig 13. LPC1549 motor control kit with LPCXpresso Board

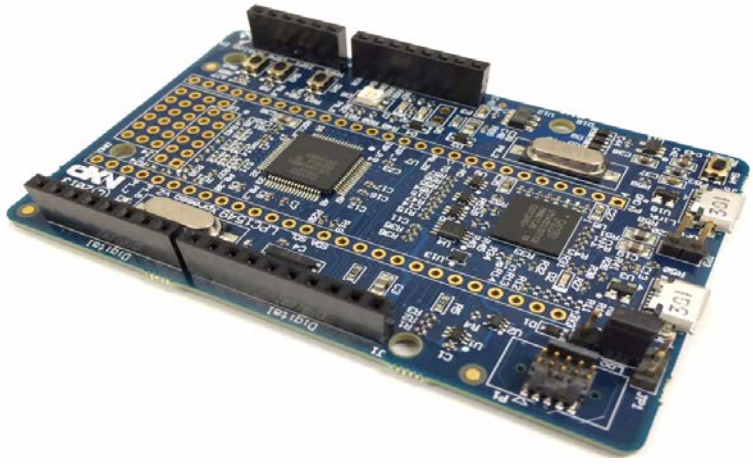

Fig 14. LPC1549 LPCXpresso board

Fig 15 lists the required signals from the LPC1549 device.

Function	Pin Name	Signal	Purpose
SCT0_OUT0	P0_6	Q2	Motor phase A low side FET drive
SCT0_OUT1	P0_29	Q1	Motor phase A high side FET drive
SCT0_OUT2	P0_26	Q4	Motor phase B low side FET drive
SCT0_OUT3	P0_0	Q3	Motor phase B high side FET drive
SCT0_OUT4	P0_1	Q6	Motor phase C low side FET drive
SCT0_OUT6	P0_24	Q5	Motor phase C high side FET drive
SCT_IN0	P0_2	HA	Hall sensor A input
SCT_IN1	P0_30	HB	Hall sensor B input
SCT_IN2	P0_17	HC	Hall sensor C input
JOYSTICK RIGHT	P1_8	JOY_R	Increase motor speed
JOYSTICK LEFT	P1_7	JOY_L	Decrease motor speed
JOYSTICK CENTER	P1_5	JOY_C	Stop the motor

Fig 15. List of signals used in LPC microcontroller

On building and downloading the code onto the device, a startup message appears on the OLED. The joystick on the motor control kit is used to control the speed and direction of the motor. When the motor is at rest, a status message “Motor status: Stop” is displayed on the OLED. On pressing the joystick in the up direction, the motor starts to run in the forward direction and the message on the OLED changes to “Motor status: Running”. To increase the speed of the motor, press the joystick in the up direction and to decrease the speed, press the joystick in the down direction. On pressing the joystick at the center, the motor comes to a halt and the display message on the OLED changes. Further, on pressing the joystick in the down direction, the motor begins to run in the reverse direction. To increase the speed, press the joystick in the down direction and to decrease the speed, press the joystick in the up direction. On pressing the left side of the joystick control, the motor operation is aborted.

6. Conclusion

This application note provides an example implementation for the BLDC motor control in the LPC15xx family using the SCTimer/PWM.

The SCTimer/PWM is highly flexible as it is based on states and events. It provides all the required hardware resources needed to accommodate the application requirements. The SCTimer/PWM ensures deterministic performance as the switching of states takes place in the hardware.

7. Legal information

7.1 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.

7.2 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.

7.3 Trademarks

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

8. List of figures

9. Contents

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

Date of release: 17 April 2014
Document identifier: AN11523