Document Number: AN5216
Rev. 0, 12/2015

Dual BLDC Sensorless Motor Control Using MKV11x

1. Introduction

This application note describes the implementation of a sensorless motor control of two Brushless DC (BLDC) motors using a single Freescale 32-bit Kinetis MKV11 device. This document explains how to use the user interface for running two BLDC motors independently. This application note does not describe the sensorless motor control software and the BLDC motor-control theory in general, because that is described in *Three-Phase BLDC Sensorless Motor Control Application* (document DRM144).

Contents

1.1. Advantages and features of MKV11x device

The Kinetis KV11 subfamily represents a low-cost portfolio of ARM® Cortex®-M0+ MCUs with peripheral modules dedicated for dual BLDC motor-control applications. The typical application segment includes BLDC sensorless, PMSM sensorless FOC, ACIM V/Hz, and FOC motor-control applications. The basic block diagram of the MKV11 MCU is shown in Figure 1. The key features of the MKV11 MCU are:

- 75 MHz ARM Cortex-M0+ core
- Hardware divide peripheral module
- 4-channel eDMA
- Up to 128 KB flash memory
- Up to 16 KB SRAM
- CAN, IIC, SPI, 2 × UART communications ports
- 2 × 16-bit ADC modules
- 12-bit DAC module
- 2 × analog comparators with 6-bit DAC
- 2 × 6-channel FlexTimer module
- 2 × 2-channel FlexTimer with quadrature decoding feature
- Programmable delay block
- 32-bit CRC
- 1.71 V to 3.6 V, -40 °C to 105 °C

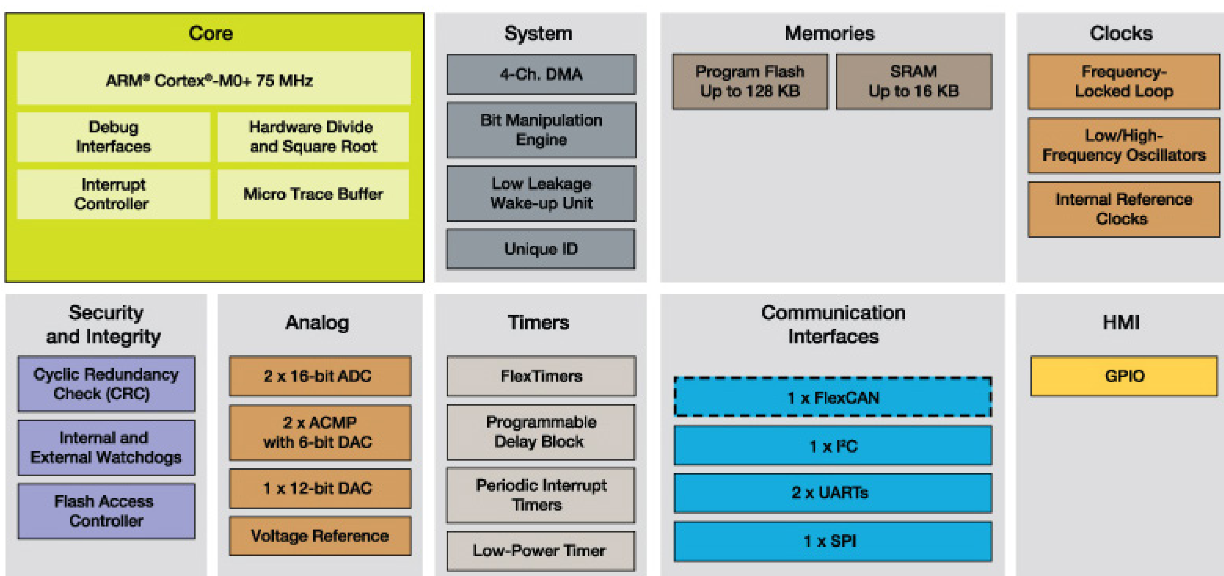

Figure 1. KV1x block diagram

2. Application concept

The application software enables easy porting of the application to other devices or platforms. The application software is divided into two main parts:

- Hardware-dependent code (dependent on hardware boards and MCU peripheral modules). This part of code contains CPU and peripheral modules' initialization and interrupt service routines handling.
- Hardware-independent code, which is a pure BLDC motor-control algorithm.

For the purpose of independent control of two motors, it is necessary for each motor to have its own data structure. All input/output variables for the control algorithm are stored in this structure. The first motor (M1) has the variables stored in the *gsM1_Drive* structure, and the second motor (M2) has the variables stored in the *gsM2_Drive* structure.

The independent control of two BLDC motors using one device is done in such way that the application uses two ADC interrupts (*ADC0_Isr* and *ADC1_Isr*), one ADC interrupt for each motor. The ADC interrupt *ADC0_Isr* carries out current measurement and Back-EMF voltage measurement. The fast control loop is called in the ADC interrupt, which processes the Back-EMF voltages and calculates the commutation period. The Back-EMF voltage is processed such that the integral of the Back-EMF voltage is calculated. If the integral of Back-EMF voltage reaches the threshold value, then the calculation of new commutation sector is performed. This is done in the *BldcCommutation* function. For detailed information about calculating the commutation sector and commutation period see *Three-Phase BLDC Sensorless Motor Control Application* (document DRM144).

Faults are checked after performing the fast control loop. When the fast control loop calculates new parameters for commutation, the *UpdateBldcHwModulesM1* function is called. This function performs a transition from the hardware-independent part of code (motor-control algorithm) to the hardware-dependent part of code. That means this function sets the MCU peripherals according to the calculated parameters from the fast control loop and slow control loop.

The *ADC1_Isr* is carried out in the same way as the *ADC0_Isr*, but it is used for the second motor. If the fast control loop calculates new parameters of commutation for the second motor, then the *UpdateBldcHwModulesM2* function is called. This function does the same thing as *UpdateBldcHwModulesM1*, but it sets the peripherals for the second motor.

The *ADC0* is triggered by the *PDB0* module, and the *ADC1* is triggered by the *PDB1* module. Both ADCs are triggered every PWM period. If the *PDB0* cannot trigger the *ADC0*, or the *PDB1* cannot trigger the *ADC1*, then the trigger error interrupt *PDB0_PDB1_Isr* is generated. This happens when the PDB trigger is generated while the ADC conversion is still in progress, and therefore it is not possible to start another ADC conversion (the period between the two PDB triggers is too short). The PDB module stops its operation if the trigger error flag is set. To clear the error flag, disable the PDB module, and then clear the error flag by writing zero into it. After that, enable the PDB module.

The *FTM2_Isr* is an important interrupt routine. This interrupt routine is used for both motors. The *FTM2_Isr* is used for slow control loop timing. First, the slow control loop for the first motor is called, and then the slow control loop for the second motor is called. The average speed is calculated in the slow control loop, together with the speed PI controller and torque PI controller, and thus the duty cycle for PWM is calculated. The slow control loop is executed every 1 ms.

The FlexTimer *FTM1* is used in this application for commutation timing during open-loop startup, and for timing alignment and calibration. The *FTM1_Isr* interrupt routine is used for these purposes. This interrupt routine is used for both motors, because both motors use the same FlexTimer (but different channels).

3. Application details

This section describes the MKV11x peripheral modules configuration. These peripheral modules are used in the application:

- Clock-distribution modules (MCG, SIM)
- Analog-to-digital converters (ADC0, ADC1)
- FlexTimer modules (FTM0, FTM1, FTM2, FTM3)
- Programmable delay blocks (PDB0, PDB1)
- Serial peripheral interface (SPI0)
- Universal asynchronous receiver/transmitter (UART0)
- CAN (FlexCan)

3.1. Clock distribution modules (MCG, SIM)

The MKV11x device uses two modules to configure and distribute the clock across peripheral modules (MCG and SIM). The MCG (Multipurpose Clock Generator) module provides several clock options for the MCU. The SIM (System Integration Module) provides system control and chip configuration. Here is the description of the modules' configuration:

- MCG:
  - Slow internal reference clock generator (32 768 Hz) is used as a source
  - The FLL is used to generate 72 MHz from the internal 32 768 Hz reference clock generator (FLL factor 2197)
- SIM:
  - Clock is enabled for all peripheral modules
  - System clock (core clock) divider is set to 1—system clock is set to 72 MHz
  - The bus/flash clock divider is set to 3—bus clock is set to 24 MHz

3.2. Analog-to-digital converters (ADC0, ADC1)

The ADC0 module measures the DC-bus current, DC-bus voltage, and phase voltages for the first motor. The ADC1 module measures the same current and voltages for the second motor. Both ADC modules have the same configuration:

- Input clock is set to 24 MHz
- High-speed mode (24 MHz)
- Short-sample (six ADCK cycles per sample time)
- Single-ended input conversion mode

- 12-bit resolution
- Hardware trigger by PDB
- Conversion-complete ISR, priority level 1 (high priority)

3.3. FlexTimer modules (FTM0, FTM1, FTM2, FTM3)

The FTM0 and FTM3 modules are used for the output PWM generation (FTM0 for the first motor and FTM3 for the second motor). The FTM1 module is used for commutation control during open-loop startup and application timing. The FTM2 module is used for slow control loop timing.

- FTM0 and FTM3—output PWM generation:
  - FTM is in enhanced mode
  - Input clock is set to system clock (72 MHz)
  - Output PWM frequency is set to 20 kHz
  - Center-aligned PWM configuration
  - Complementary PWM generation
  - Dead time is set to 875 ns
  - Duty cycle is updated using the LDOK feature
  - Output mask, inverting (swap), and software control registers are updated using software synchronization feature (commutation)
  - The FTM counter is reset on commutation
  - Trigger to PDB is enabled
- FTM1—commutation control:
  - FTM is in enhanced mode
  - Free-running timer
  - Input clock is set to 562 500 Hz (system clock/128)
  - Output-compare mode
  - ISR is on output compare, priority level 1 (high priority)
- FTM2—slow control loop:
  - FTM is in enhanced mode
  - Free-running timer
  - Input clock is set to 36 MHz (system clock/2)
  - Output-compare mode
  - ISR on output compare with a period of 1 ms, priority level 3 (low priority)

3.4. Programmable delay blocks (PDB0, PDB1)

The PDB modules are used to synchronize the ADC measurement time within the PWM period. The PDB0 module synchronizes ADC0, and the PDB1 module synchronizes ADC1. Both modules have the same configuration:

- Input clock is set to 24 MHz (bus clock/1)
- Double-buffered mode is selected for registers update
- Pre-trigger0 point is set to the middle of the PWM pulse
- Back-to-back mode is set for pre-trigger1 (second ADC measurement)
- FTM0 external trigger (EXTRIG) is selected as a source PDB0 trigger
- FTM3 external trigger (EXTRIG) is selected as a source PDB1 trigger
- Sequence Error ISR is enabled, priority level 1 (high priority)

3.5. Serial peripheral interface (SPI0)

SPI0 is used for MC33937 three-phase MOSFET pre-driver configuration and status reading. The Chip Select 0 (CS0) is used for pre-driver configuration of the first motor. The Chip Select 2 (CS2) is used for pre-driver configuration of the second motor. This is the module configuration:

- Baud rate is set to 1.28 MHz
- Falling-edge SCLK polarity
- Master SPI mode

3.6. Universal asynchronous receiver/transmitter (UART0)

UART0 is used for FreeMASTER communication between the MCU and PC. In this application, you can choose to communicate with FreeMASTER either over UART1 or CAN. This is the UART0 module configuration:

- Baud rate is set to 57600 bit/s
- The receiver and transmitter are enabled
- The other settings are set to default

3.7. CAN (FlexCAN)

FlexCAN is used for FreeMASTER communication between the MCU and PC. In this application, you can choose to communicate either over CAN or UART0. This is the FlexCAN module configuration:

- Bit rate is set to 500 kb/s
- Input clock is set to 72 MHz (system clock)
- The other settings are set to default

4. User interface

The application uses these two interfaces:

- The “Up/Down” buttons on the TWR-KV11Z75M Tower System module
- Remote control using FreeMASTER

4.1. Up/Down buttons

The BLDC motors start spinning immediately after powering the power stages and pressing one of the push-buttons:

- The “Up” button (SW1) increases the speed by 500 rpm. Both motors start spinning in the clockwise direction. Both motors spin at the same speed.
- The “Down” button (SW2) decreases the speed by 500 rpm. Both motors start spinning in the counter-clockwise direction. Both motors spin at the same speed.
- Further pressing of the buttons increases or decreases the required speed within the speed range from -5000 to 5000 rpm.
- If you press and hold both buttons for more than two seconds, then the demonstration mode switches on. The demonstration mode sets different speed and rotation direction for each motor. Each motor rotates with a different speed for a different time. The demonstration mode shows an independent control of two BLDC motors using one device. If you press and hold both buttons for more than two seconds or press the reset button, the demonstration mode switches off.

4.2. Remote control using FreeMASTER

Control the application using FreeMASTER (on-line debugging and monitoring tool running on the PC). FreeMASTER communicates with the application (MKV11) either over UART or over CAN.

4.2.1. Communicating over CAN interface

To achieve correct FreeMASTER operation and to communicate over CAN, perform these steps:

1. Use FreeMASTER version 2.0 (or higher).
2. Connect the CAN interface to the PC and to the TWR-SER board (connector J7).
3. Open FreeMASTER file *DUAL_BLDC_Sensorless.pmp.*
4. Go to “Project -> Options -> Comm” and set the “Plug-In Module” to “FreeMASTER-over-CAN Communication Plugin”. Click the “Configure” button and select the CAN interface. Set the “Bit Rate” to 500 kb. Click the “Test Connection” button. If the settings are correct, then this message pops up: “*Target is alive (responded to ping)*”. If the settings are not correct, then an error message pops up.
5. Toggle the communication on (by clicking the red “STOP” button). The communication between FreeMASTER and MKV11 starts. If not, “NOT CONNECTED” appears in the bottom-right corner. Toggle the communication off, and disconnect/connect the CAN device. Then toggle the communication back on.

NOTE

FreeMASTER supports these CAN interfaces: Vector, Vector-XL, IXXAT VCISDK, NICAN, GLinker, and ZLG. Use one of these CAN interfaces for a correct FreeMASTER functionality.

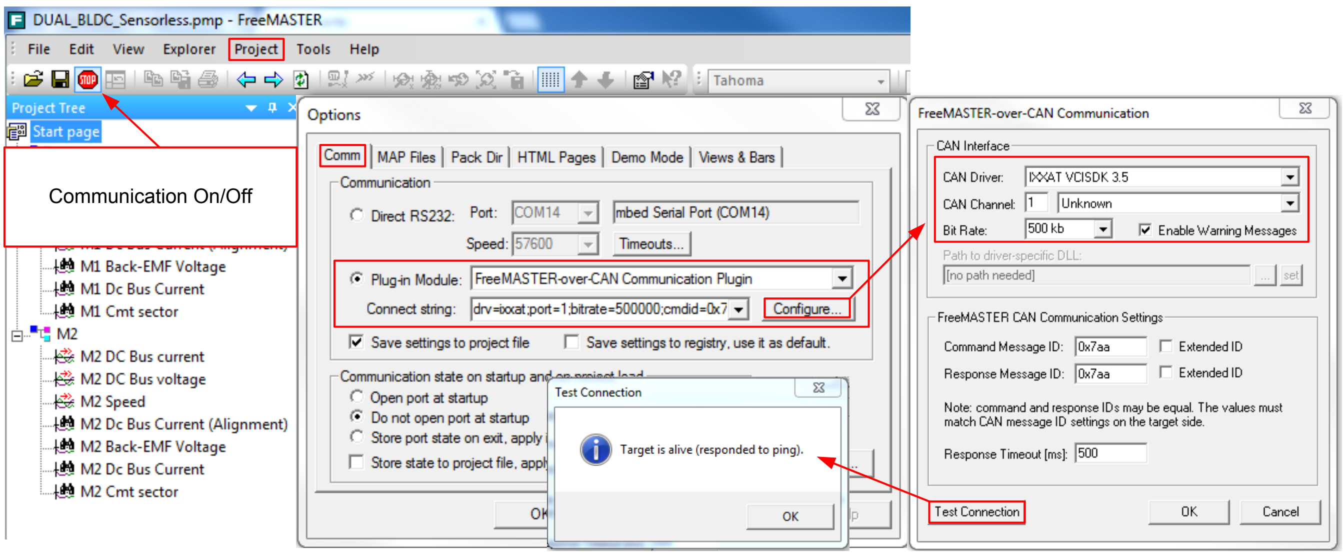

Figure 2. Setting communication over CAN in FreeMASTER

4.2.2. Communication over UART interface

To achieve a correct FreeMASTER operation and to select communication over UART, perform these steps:

1. Use FreeMASTER version 2.0 (or higher).
2. Connect the USB cable to the TWR-KV11Z75M board.
3. Open FreeMASTER file *DUAL_BLDC_Sensorless.pmp*.
4. Go to “Project -> Options -> Comm” and set the communication via Direct RS232. Select the virtual COM port to which the TWR-KV11Z75M board is connected. Set the communication speed to 57600 bit/s.
5. Toggle the communication on (clicking the red “STOP” button). The communication between FreeMASTER and MKV11 starts. If not, “NOT CONNECTED” appears in the bottom-right corner. Toggle the communication off, and disconnect/connect the USB cable. Then toggle the communication back on.

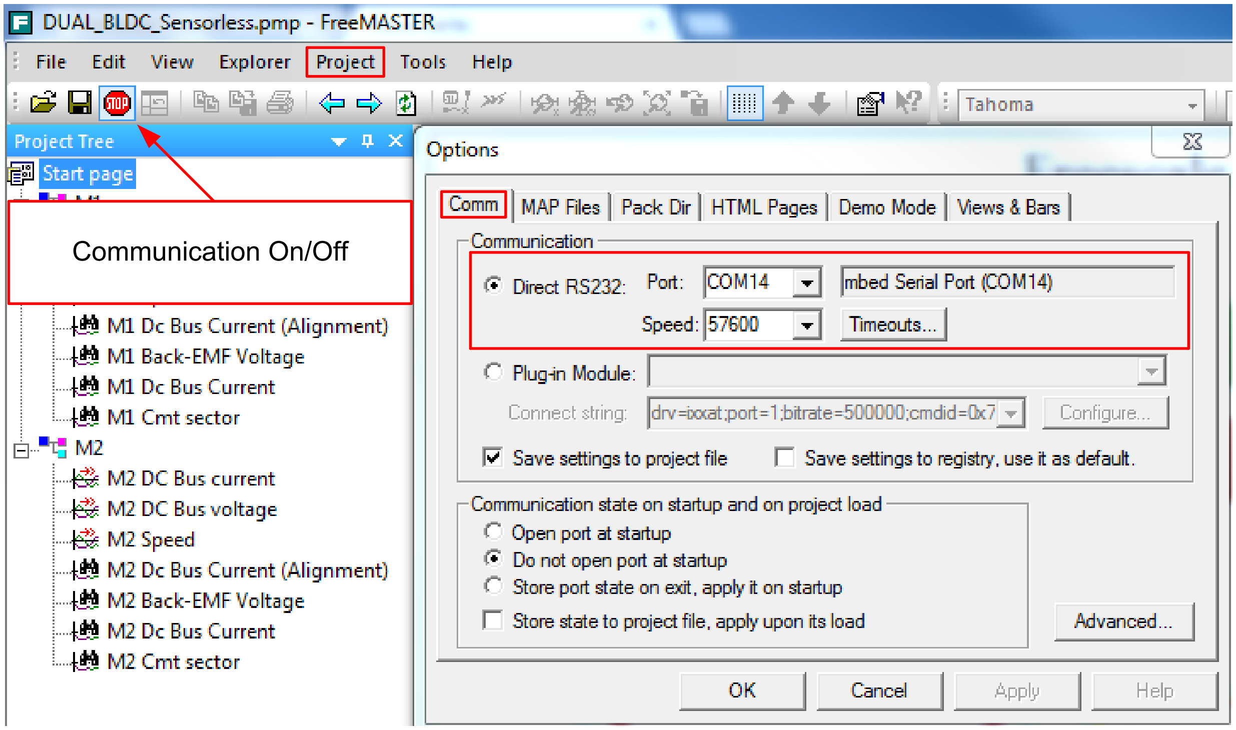

Figure 3. Setting communication over UART in FreeMASTER

NOTE

Make sure that the *freemaster_cfg.h* header file is set properly for the CAN or UART communication.

4.2.3. Description of FreeMASTER remote page

The FreeMASTER remote page is shown in Figure 4. The actual speed, actual DC-Bus current, and actual DC-Bus voltage meters for both motors are shown on this page. The meters for the first motor are on the left, and the meters for the second motor are on the right. The sliders for setting the required speed are under the speed meters. The sliders for setting the current limitation are between the DC-Bus current meters. The main application switch, application status label, and the button for clearing faults are between the speed meters.

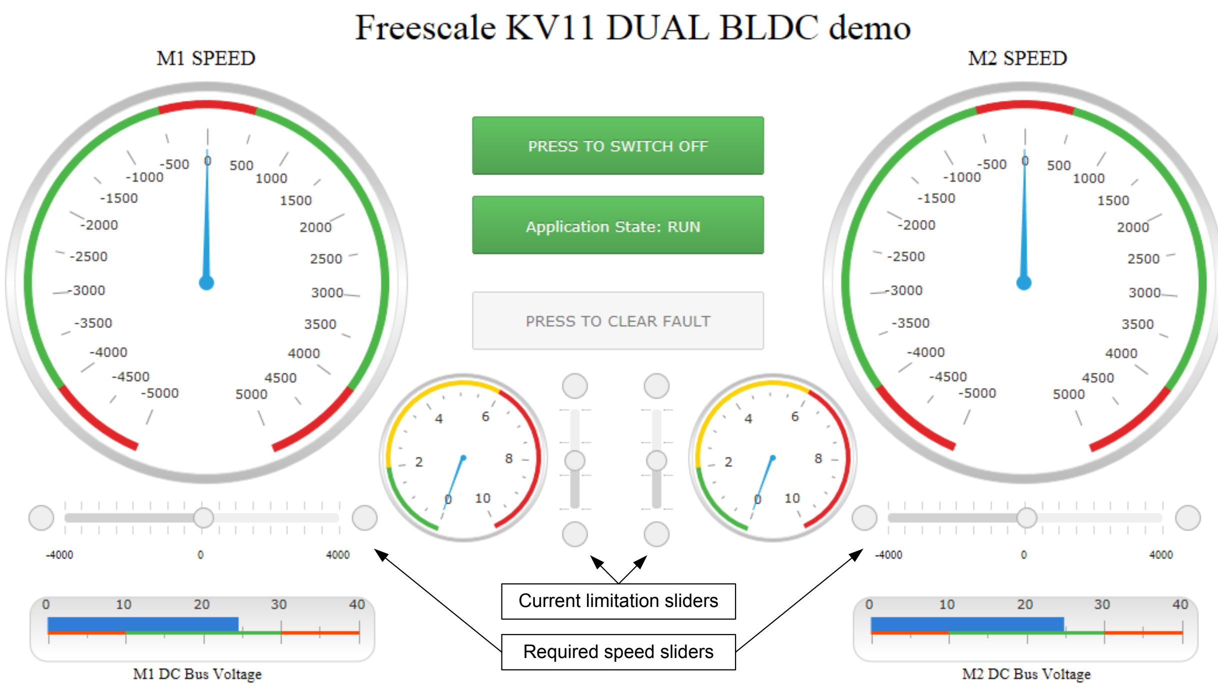

Figure 4. FreeMASTER remote page

When the power stages are powered and the communication is established, the application status label is green and displays “Application State: RUN”. This means that the application is on and ready for setting the required speeds. When the application is on, then the main application switch is green. To switch the application off immediately, click the main green application switch “PRESS TO SWITCH OFF”. If the application is off, then the application status label is red and displays “Application State: STOP” and the main application switch is red. To switch the application on, click the main red application switch “PRESS TO SWITCH ON”.

The button for clearing faults is disabled during normal operation. This button is enabled only when a fault occurs. These faults can occur: overcurrent, overvoltage, and undervoltage. When a fault occurs, the application switches off and the application state label displays the fault. To clear the fault, click the “PRESS TO CLEAR FAULT” button. When this application state label displays “Application State: STOP”, the button for clearing faults is disabled. Click the main application button to start the application again.

To set the required speed for the motors, slide the speed slider to the required value or click into the required value on the slider. You can also set the speed by clicking the small buttons on the slider edge. This increases or decreases the speed by 500 rpm.

To set the current limitation for the motors, slide the current limitation slider to the required value or click into the required value on the slider. You can also set the current limitation by clicking the small buttons on the slider edge. This increases or decreases the current limitation by 0.5 A. The default setting of current limitation is 3 A. If the actual current reaches the current-limitation value, then the current-limitation label shows up between the DC-Bus voltage meters, and the motor speed decreases until the actual DC-Bus current is below the limitation value of the current.

There are two sub-blocks in the project tree (M1 and M2). The sub-block M1 is for the first motor, and the sub-block M2 is for the second motor. These sub-blocks contain scopes and recorders for monitoring of speed, current, and voltages' characteristics. Monitor these characteristics by clicking one of the scopes or recorders. If you want to return to the remote page, click the "Start page" label on the top of the project tree. The structure of the project tree is shown in this figure:

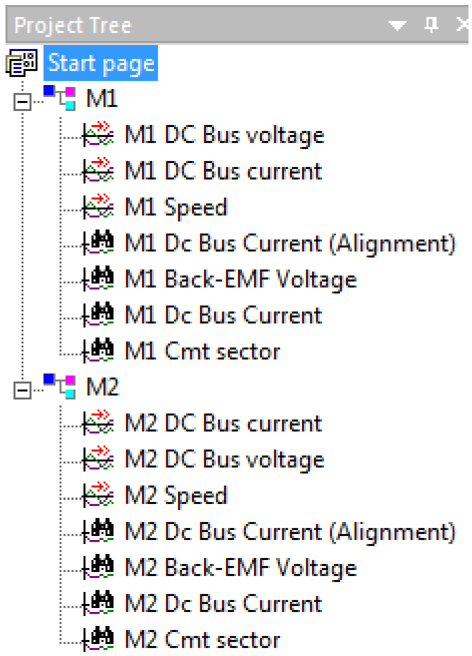

Figure 5. Structure of project tree in FreeMASTER

5. Hardware

This application contains these hardware modules:

- TWR-KV11Z75M board
- 2 × TWR-MC-LV3PH
- TWR-SER
- TWR-Elevator
- 2 × BLDC motor
- CAN interface

5.1. TWR-KV11Z75M Tower System module

The TWR-KV11Z75M Tower System module contains a Kinetis MKV11 MCU, dedicated for motor-control applications. This device is powerful enough for sensorless motor control of two BLDC motors. This device contains peripherals optimized for motor control. The block diagram of this device is shown in Figure 1. The TWR-KV11Z75M board is shown in this figure:

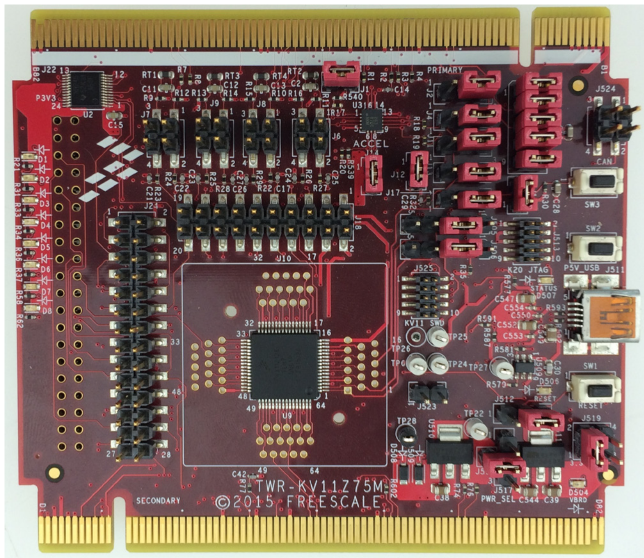

Figure 6. TWR-KV11Z75M Tower System module

Table 1. TWR-KV11Z75M jumper settings

Jumper	Position	Jumper	Position
J1	short	J12	short
J2	2-3	J13	2-3
J4	2-3	J14	short
J5	short	J17	2-3
J6	none	J505	2-3
J7	none	J506	2-3
J8	none	J512	2-3
J9	none	J519	1-2
J10	short	J523	none
J11	2-3	J524	none

5.2. TWR-MC-LV3PH Tower System module

The 3-Phase Low-Voltage Motor Control board (TWR-MC-LV3PH) is a peripheral Tower System module, interchangeable across the Tower development platform. The phase voltage and current feedback signals enable using a variety of algorithms to control 3-phase PMSM and BLDC motors. A high level of board protection (overcurrent, undervoltage, overtemperature, etc.) is provided by the MC33937 MOSFET pre-driver. The TWR-MC-LV3PH board is shown in the following figure.

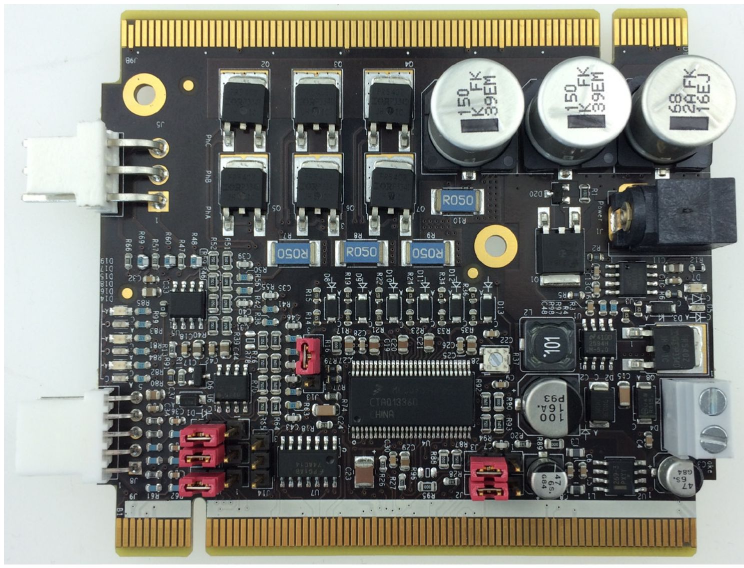

Figure 7. TWR-MC-LV3PH Tower System module

The features of the TWR-MC-LV3PH module are:

- Input power-supply voltage of 12-24 V DC
- Output current up to 8 A
- Power supply reverse polarity protection
- 3-phase bridge inverter (six MOSFETs)
- 3-phase MOSFET gate driver with overcurrent and undervoltage protection
- 3-phase and DC-Bus current-sensing shunts
- DC-Bus voltage sensing
- 3-phase back-EMF voltage-sensing circuitry
- Low-voltage on-board power supplies
- Encoder/Hall sensor sensing
- Motor power and signal connectors
- User LED, power-on LED, six PWM LEDs

Table 2. TWR-MC-LV3PH jumper settings

Jumper	Position
J2	2-3
J3	2-3
J10	2-3
J11	2-3
J12	2-3
J13	2-3
J14	none

5.3. TWR-SER Tower System module

The TWR-SER board is a peripheral Tower System module, interchangeable across the Tower development platform. This module has these features:

- USB Host, Device, and OTG with Mini-AB connector
- 10/100 Ethernet PHY with MII and RMII interface
- Ethernet connector with integrated magnetic and LEDs
- RS232 and RS485 transceivers and a single DB9 connector
- CAN transceiver with a 3-pin head

This application uses only the CAN transceiver from the TWR-SER module. If you configure the application to communicate over UART, then you do not need the TWR-SER module.
The communication over UART is realized via the OpenSDA feature, which is located directly on the TWR-KV11Z75M board. The TWR-SER board is shown in this figure:

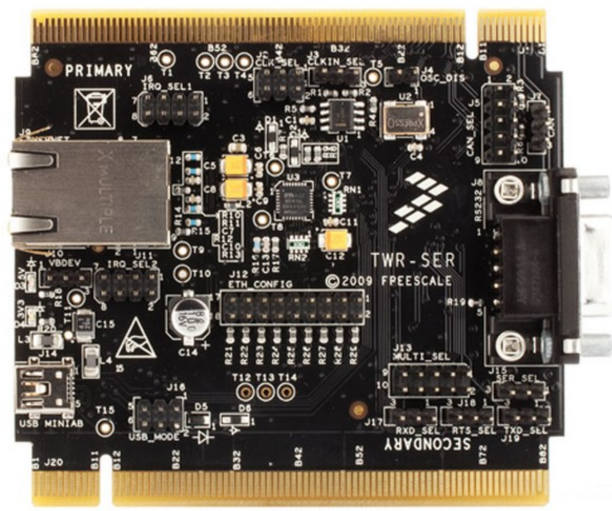

Figure 8. TWR-SER Tower System module

5.4. LINIX 45ZWN24 BLDC motor

This application uses two LINIX 45ZWN24 BLDC motors. The parameters of these motors are summarized in this table:

Table 3. Motor parameters

Parameter	Symbol	Value	Unit
Rated Voltage	V_t	24	V
Rated Speed @ V_t	—	4000	rpm
Rated Torque	T	0.0924	Nm
Rated Power	P	40	W
Continuous Current	I_{cs}	2.34	A
Number of Pole Pairs	PP	2	—

6. Acronyms and abbreviations

Table 4. Acronyms and abbreviations

Term	Meaning
ADC	Analog-to-Digital Converter
BEMF	Back-Electromotive Force
BLDC	Brushless DC Motor
CAN	Controller Area Network
CCW	Counter-Clockwise Direction
CMP	Comparator
CPU	Central Processing Unit
CW	Clockwise Direction
DAC	Digital-to-Analog Converter
DC	Direct Current
DMA	Direct Memory Access
DRM	Design Reference Manual
FOC	Field-Oriented Control
FTM	FlexTimer Module
GPIO	General-Purpose Input/Output
I^2C	Inter-Integrated Circuit interface
ISR	Interrupt Service Routine
KV11x	Freescale 32-bit ARM-based subfamily MCU
PDB	Programmable Delay Block
PWM	Pulse-Width Modulation
UART	Universal Asynchronous Receiver/Transmitter
SPI	Serial Peripheral Interface

7. References

These references are available on freescale.com:

- *K11 Sub-Family Reference Manual* (document KV11P64M75RM)
- *Three-Phase BLDC Sensorless Motor Control Application* (document DRM144)
- *Three-Phase BLDC Sensorless Control Using the MKV10x* (document AN4862)

8. Revision history

Table 5. Revision history

Revision number	Date	Substantive changes
0	12/2015	Initial release

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Document Number: AN5216
Rev. 0
12/2015