AN14805

MCX A34x Triple-Motor and Interleaved PFC Control using an HVP Board

Rev. 2.0 — 18 December 2025

Application note

Document information

Information	Content
Keywords	AN14805, MCX A, MCX A34x, PMSM, PFC, FOC, FRDM-MC-LVPMSM, HVP-MC3PH
Abstract	This document describes the implementation of interleaved PFC and 3MC FOC for triple 3-phase Permanent Magnet Synchronous Motors (PMSMs) using the NXP MCX A34x MCU.

1 Introduction

This document describes the implementation of interleaved Power Factor Correction (PFC) and 3MC Field-Oriented Control (FOC) for triple 3-phase permanent magnet synchronous motors (PMSMs) using the NXP MCX A34x microcontroller (MCU).

The MCX A34x MCU uses a single Arm Cortex-M33 core, which operates at speeds of up to 180 MHz. It provides multiple high-speed connectivity options with serial peripherals, while supporting low-power consumption. It also provides timers and analog interfaces.

The 3-phase PMSM control and power factor correction demo comprises the following hardware components:

- One HVP-MC3PH motherboard (supports high-voltage PMSM control of motor 1)
- One MCX A34x 3MC controller (daughter) card (X-MCXA346-3MC)
- One dual-motor adapter board
- Two FRDM-MC-LVPMSM driver boards (support low-voltage PMSM control of motor 2 and motor 3)
- Three 3-phase PMSM motors

The MCX A34x MCU samples the currents and voltages of the motors through its Analog-to-Digital Converter (ADC) module. Then, it generates pulse width modulation (PWM) based on the sensorless Field-Oriented Control (FOC) algorithm to drive the motors. At the same time, the Universal Asynchronous Receiver/Transmitter (UART) module is used to communicate with the FreeMASTER for command dispatch, variable observation, and some other functions that can be debugged easily. Finally, precise position control and smooth speed regulation can be achieved for two of the three motors. The corresponding demo code can be found on NXP Application Code Hub.

The motor control software and the following PMSM control concepts are described in *Sensorless PMSM Field-Oriented Control* (DRM148):

- Motor mathematical model
- Coordinate transformation
- Space Vector Pulse Width Modulation (SVPWM)
- Back Electromagnetic Field (BEMF) observer based sensorless algorithm

The following documents describe the MCX A34x functions available in NXP Real-Time Control Embedded Software Libraries (RTCESL):

- *GMCLIB User's Guide* (CM33FGMCLIBUG)
- *AMCLIB User's Guide* (CM33FAMCLIBUG)

The current document describes the MCX A34x MCU features, hardware platforms used for the demo, demo setup, peripheral settings, CPU loading performance, and memory usage.

2 Demo block diagrams

Figure 1 shows the high-level system block diagram of the 3-phase PMSM control and power factor correction demo.

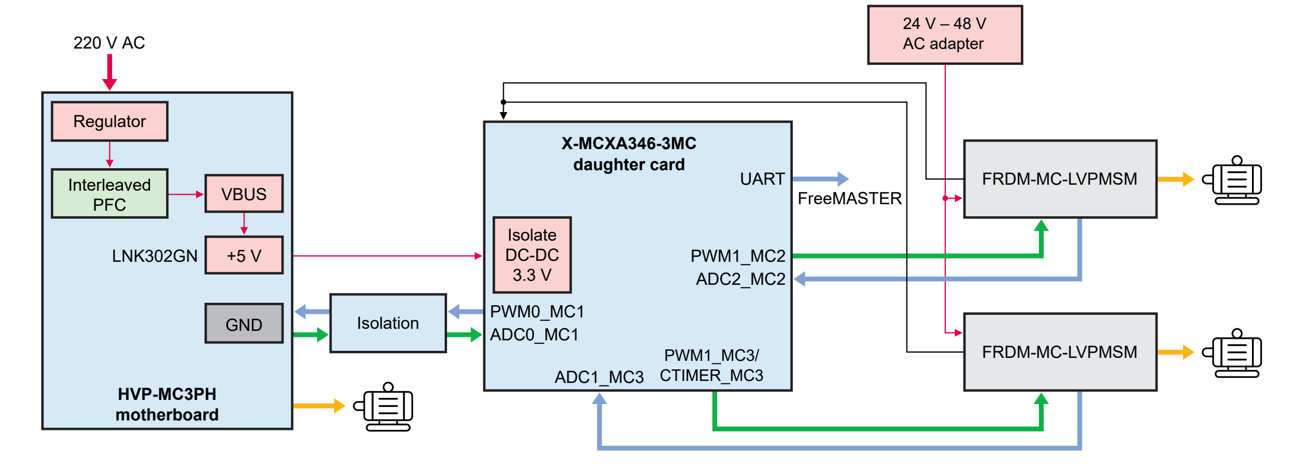

Figure 1. Demo high-level system block diagram

Figure 2 shows the LVPMSM dual-motor control architecture of the 3-phase PMSM control and power factor correction demo.

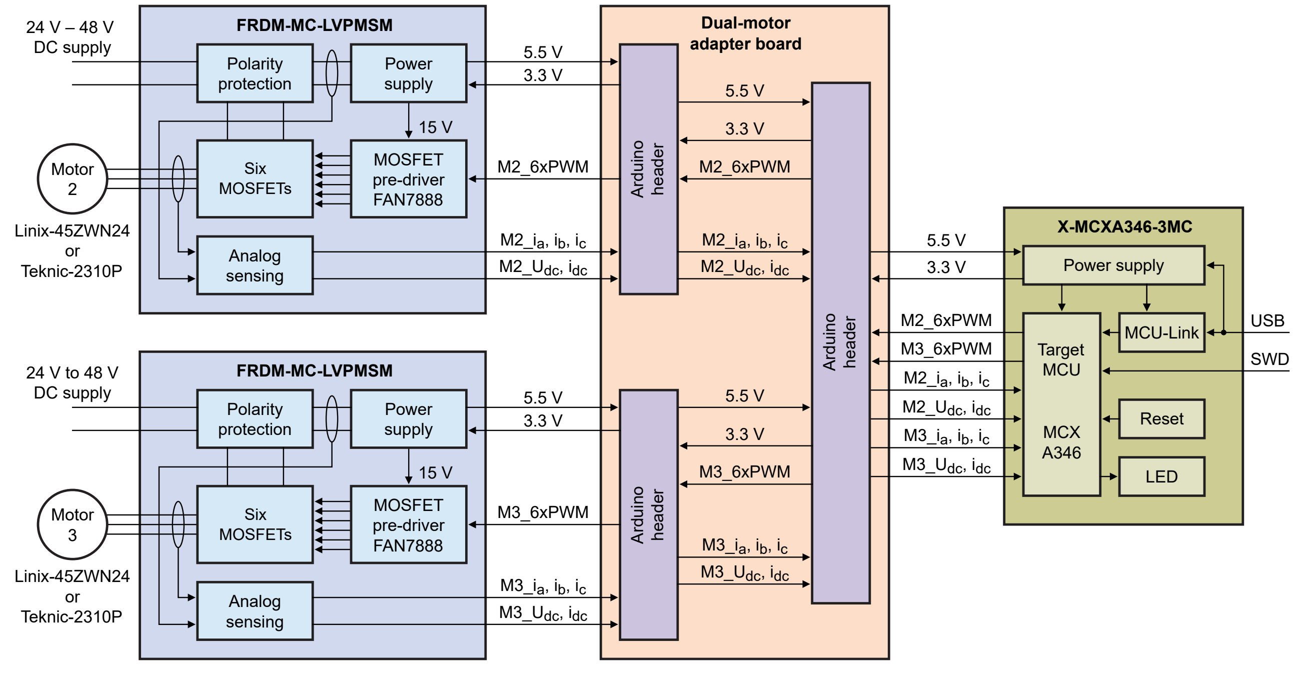

Figure 2. LVPMSM dual-motor control architecture diagram

A FRDM-MC-LVPMSM board can accept a DC power supply in the range 24 V – 48 V. The board has a reverse polarity protection circuitry for protection from power faults. The FRDM-MC-LVPMSM board has an output supply of 5.5 V that is used to provide power to the MCX A34x 3MC controller card (not populated by default). The output current reaches up to 5 A RMS. The FRDM-MC-LVPMSM board has a 3-phase MOSFET gate driver with over-current and under-voltage protection. The 3-phase MOSFET gate driver connects to the six MOSFETs on the board that act as a 3-phase bridge inverter. The board supports measurement of analog quantities, such as 3-phase motor current, DC bus voltage, and DC bus current.

The MCX A34x 3MC controller card is a design and evaluation platform based on the NXP MCX A34x MCU. The board is compatible with the Arduino UNO RI boards. It can be used with a wide range of development tools, including NXP MCUXpresso IDE, IAR Embedded Workbench, and Arm Keil MDK. The board is lead-free and RoHS-compliant.

The dual-motor adapter board is a flexible motor control adapter board with three Arduino headers. The three Arduino headers are used to connect two FRDM-MC-LVPMSM boards and one MCX A34x 3MC controller card.

For debugging the MCX A34x MCU, the MCX A34x 3MC controller card uses an onboard MCU-Link debugger, which is based on another MCU.

3 MCU motor control features and peripheral settings

The subsections that follow describe the motor control related features of the MCX A34x MCU and the required peripheral settings for the 3-phase PMSM control and PFC demo.

3.1 Clock generation (SCG)

The system clock generator (SCG) module of the MCX A34x MCU generates and controls the clocks of various modules in the design. This module uses the available clock sources to generate the clock roots.

The clock source used in the motor control application is the FRO180M (FIRC), which can boot up to 180 MHz at most. The clock frequency for the Arm core and eFlexPWM is 180 MHz.

Table 1 shows the clock source configuration for motor control peripherals.

Table 1. Clock source configuration for motor control peripherals

Module on MCU	Clock source	Divided frequency
main_clock	180 MHz	FRO180M
Core clock	180 MHz	fro_hf
eFlexPWM	180 MHz	main_clk
ADC	60 MHz	fro_hf / 3
LPTMR	12 MHz	fro_12m
CTIMER	180 MHz	fro_hf
CMP	180 MHz	fro_hf
LPUART	12 MHz	fro_12m

3.2 PWM generation (eFlexPWM)

Each of the two eFlexPWM modules of the MCX A34x MCU contains four PWM submodules. Each submodule is set up to control a single half-bridge power stage. Fault channel support is provided. This PWM module can generate various switching patterns, including highly sophisticated waveforms. It can enable the generation of 3-phase PWM signals connected to the MOSFET H-bridge via pre-drivers.

Each eFlexPWM module has a 16-bit counter that counts only in the upward direction. The counter counts to the VAL1 value and then resets to the INIT value. During the counting process, the counter value is compared with the value in the VAL2/VAL3 register to control the high and low switching of the output level.

Figure 3 illustrates the three motors + interleaved PFC use case.

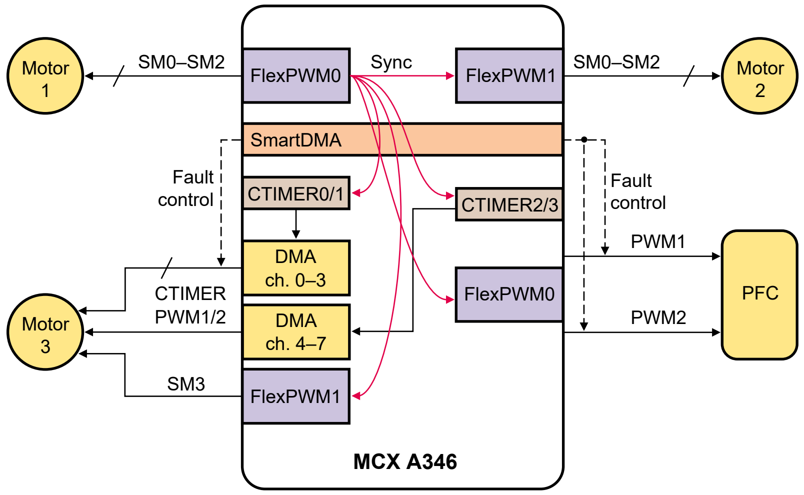

Figure 3. Three motors + interleaved PFC use case

The three PWM submodules of FlexPWM0 that are used to control motor 1 in this demo are configured as follows:

- FlexPWM0 submodule 0 configuration:
  - The IPBus clock source is 180 MHz.
  - The output frequency of the submodule waveform is 8 kHz with 125 µs period.
  - The INIT register is set to -11250, and the VAL1 register is set to 11249.
  - Complementary mode with 1 µs dead time
  - During every PWM cycle, this PWM submodule sends a master sync signal to other submodules for synchronization.
  - Trigger zero signal (TRIG0) from VAL4 (-11190) provides sample synchronization with the ADC0 module via the INPUTMUX module.
- FlexPWM0 submodule 1 / submodule 2 configuration:
  - FlexPWM0 submodule 0 is the clock source for this submodule.
  - The output frequency of the submodule waveform is 8 kHz with 125 µs period.
  - The INIT register is set to -11250, and the VAL1 register is set to 11249.
  - Complementary mode with 1 µs dead time
  - During every PWM cycle, a reload signal is received from submodule 0 for synchronization.
  - Trigger zero signal (TRIG0) from VAL4 (-3690) provides sample synchronization with the ADC1 module via the INPUTMUX module.
  - Trigger one signal (TRIG1) from VAL5 (-3750) provides external force for FlexPWM1 via the INPUTMUX module.

The submodule 3 of FlexPWM0 that is used to control PFC in this demo is configured as follows:

- FlexPWM0 submodule 0 is the clock source for this submodule.
- The output frequency of the submodule waveform is 48 kHz with 20.083 µs period.
- The INIT register is set to -1875, and the VAL1 register is set to 1874.
- Local sync is used for initialization.
- Trigger one signal (TRIG1) from VAL1 (1874) provides the trigger signal for AOI0 via the INPUTMUX module.

The three PWM submodules of FlexPWM1 that are used to control motor 2 in this demo are configured as follows:

- FlexPWM1 submodule 0 configuration:
  - The IPBus clock source is 180 MHz.
  - The output frequency of the submodule waveform is 8 kHz with 125 μs period.
  - The INIT register is set to -11250, and the VAL1 register is set to 11249.
  - Complementary mode with 0.5 μs dead time
  - The force signal from FlexPWM0 causes the counter to initialize.
  - During every PWM cycle, this PWM submodule sends a sync signal to other submodules for synchronization.
- FlexPWM1 submodule 1 / submodule 2 configuration:
  - FlexPWM1 submodule 0 is the clock source for this submodule.
  - The output frequency of the submodule waveform is 8 kHz with 125 μs period.
  - The INIT register is set to -11250, and the VAL1 register is set to 11249.
  - Complementary mode with 0.5 μs dead time
  - The force signal from FlexPWM0 causes the counter to initialize.

The submodule 3 of FlexPWM1 that is used to control motor 3 phase A in this demo is configured as follows:

- The output frequency of the submodule waveform is 8 kHz with 125 μs period.
- The INIT register is set to -11250, and the VAL1 register is set to 11249.
- Complementary mode with 0.5 μs dead time
- The external sync signal from FlexPWM0 causes the counter to initialize.

3.3 Counters/timers (CTIMER)

This demo uses CTIMER for controlling motor 3:

- CTIMER0/1 match 0/1 is used to trigger Enhanced Direct Memory Access (eDMA) for PWM of motor 3 phase B
- CTIMER0/1 match 3 is set to 22490, and it is used for a period
- CTIMER2/3 match 0/1 is used to trigger eDMA for PWM of motor 3 phase C
- CTIMER2/3 match 3 is set to 22490, and it is used for a period
- CTIMER4 match 0 is selected as the trigger input for sampling I_{pfc1} and U_{DC}
- CTIMER4 match 1 is selected as the trigger input for sampling I_{pfc2} and U_{AC}

Figure 4 shows the hardware timing and synchronization on the MCX A34x MCU.

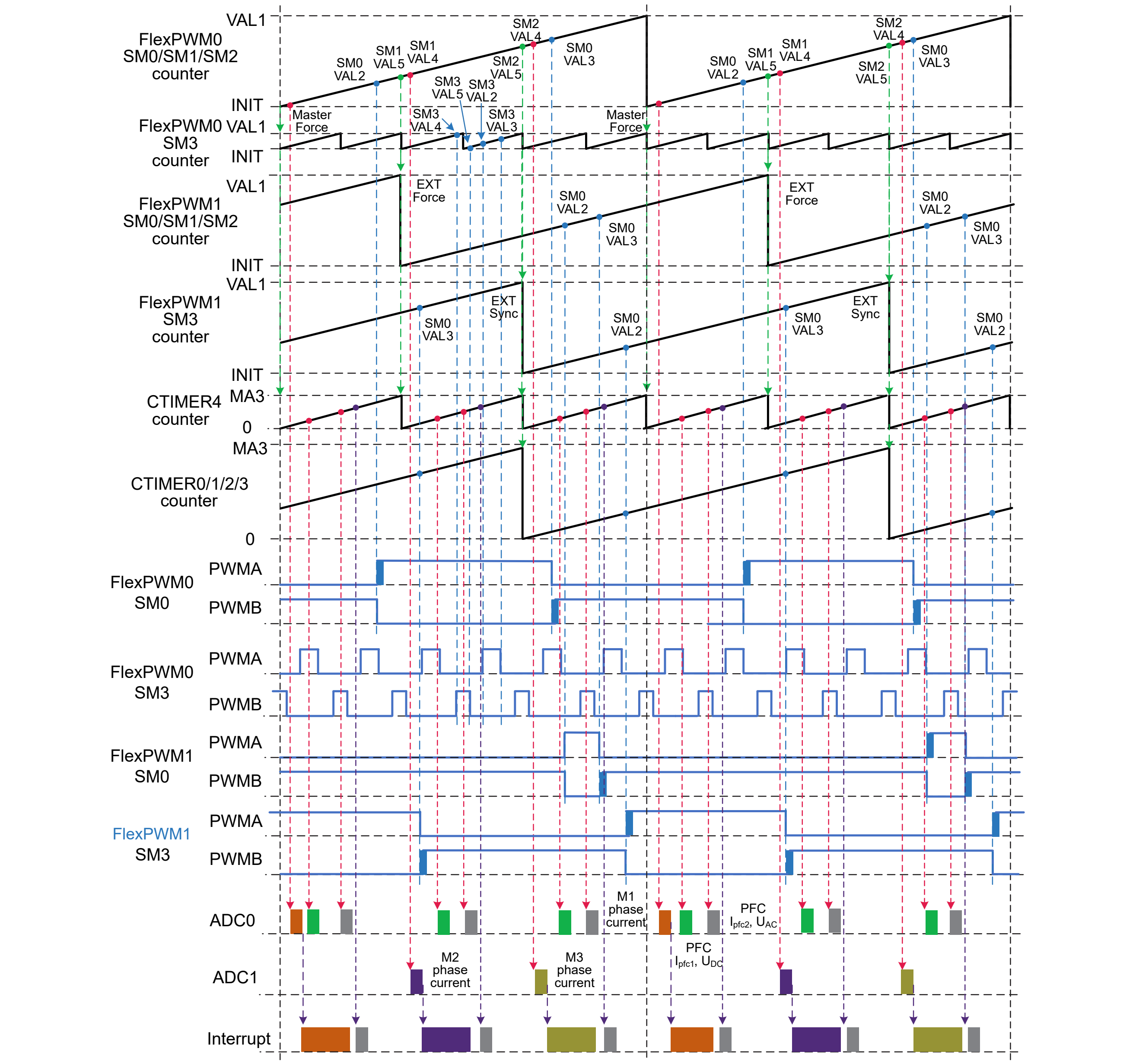

Figure 4. Hardware timing and synchronization on MCX A34x

In Figure 4, the top signal shows the FlexPWM0 submodule 0 counter (SM0 counter). The dead time is inserted at the rising edge of both the top and bottom PWMs. The submodule generates the master reload at every PWM cycle.

When the offset bus current sample conversion completes, the counter enters the ADC ISR (ADC interrupt). The FOC calculation is done in this interrupt.

3.4 Peripheral trigger control (INPUTMUX)

The Input Multiplexing (INPUTMUX) module provides signal routing options for internal peripherals. Some peripheral inputs are multiplexed to multiple input sources. The input sources can be external pins, interrupts, output signals of other peripherals, or other internal signals, as follows:

- PWM0_SM0_OUT_TRIG0 is selected as the trigger input for ADC0 TRIG0.
- PWM0_SM1_OUT_TRIG0 is selected as the trigger input for ADC1 TRIG0.
- PWM0_SM2_OUT_TRIG0 is selected as the trigger input for ADC1 TRIG1.
- CTIMER4_MATCH0 is selected as the trigger input for ADC0 TRIG1.
- CTIMER4_MATCH1 is selected as the trigger input for ADC0 TRIG2.
- PWM0_SM1_OUT_TRIG1 is selected as the trigger input for PWM1 external force.
- PWM0_SM2_OUT_TRIG1 is selected as the trigger input for PWM1 external sync.
- PWM0_SM3_OUT_TRIG1 is selected as the trigger input for AOI0 input 0.
- AOI0_OUT0 is selected as the trigger input for CTIMER4 trigger input.
- PWM1_SM3_OUT_TRIG0 is selected as the trigger input for CTIMER0 trigger input.
- PWM1_SM3_OUT_TRIG0 is selected as the trigger input for CTIMER1 trigger input.
- PWM1_SM3_OUT_TRIG0 is selected as the trigger input for CTIMER2 trigger input.
- PWM1_SM3_OUT_TRIG0 is selected as the trigger input for CTIMER3 trigger input.
- CMP0_OUT is selected as the trigger input for PWM1 FAULT0.
- CMP1_OUT is selected as the trigger input for PWM1 FAULT2.
- TRIG_IN0 is selected as the trigger input for PWM0 FAULT0.
- TRIG_IN1 is selected as the trigger input for PWM0 FAULT1.
- CMP1_OUT is selected as the trigger input for SmartDMA0.

3.5 Analog sensing (ADC0 and ADC1)

This section contains the following subsections:

- Section 3.5.1 "Analog sensing with ADC0"
- Section 3.5.2 "Analog sensing with ADC1"

3.5.1 Analog sensing with ADC0

ADC0 is used to sample 2-phase current and DC/AC voltage for interleaved PFC. It is also used to sample phase current for motor 1 FOC. ADC0 operates in 12-bit mode with single-ended conversion and hardware trigger selected.

The command buffers CMD1–CMD3 are used to sample the 3-phase current of motor 1. The command buffers CMD4–CMD7 are used to sample PFC control.

In ADC0 IRQ service routing, the motor 1 fast loop function is executed (8 kHz).

3.5.2 Analog sensing with ADC1

ADC1 is used to sample phase current for motor 2 and motor 3. It operates in 12-bit mode with single-ended conversion and hardware trigger selected.

The command buffers CMD1–CMD4 are used to sample the 3-phase current and DC bus voltage for motor 2. The command buffers CMD5–CMD7 are used to sample phase current and DC bus voltage for motor 3.

In ADC1 IRQ service routing, the motor 2 and motor 3 fast loop function is executed (8 kHz).

3.6 Slow loop interrupt generation (LPTMR0)

LPTMR0 is used to generate slow loop interrupt and perform demo speed simulation for motor 1, motor 2, and motor 3.

Usually, the slow loop interrupt is 8 times slower than the fast loop. The speed-loop frequency is set in the M1_SPEED_LOOP_FREQ macro and it equals to 1000 Hz. Set the match value register MATCH0 according to the slow loop frequency. When the counter reaches MATCH0, it resets to 0 and sends an interrupt request.

3.7 Hardware protection (CMP0 and CMP1)

This section contains the following subsections:

- Section 3.7.1 "Using CMP0 for hardware protection"
- Section 3.7.2 "Using CMP1 for hardware protection"

3.7.1 Using CMP0 for hardware protection

The CMP0 module provides a circuit to compare the motor 2 DC bus current sample signal with the internal 8-bit DAC output value. It generates a trigger to lock the FlexPWM1 output if the DC bus current is high.

3.7.2 Using CMP1 for hardware protection

The CMP1 module provides a circuit to compare the motor 3 DC bus current sample signal with the internal 8-bit DAC output value. It generates a trigger to lock the FlexPWM1 output if the DC bus current is high.

3.8 Math acceleration (MAU)

The MCX A34x MCU provides one Math Acceleration Unit (MAU) module. The MAU module can accelerate mathematical operations, including sine, cosine, arctangent, square root, reciprocal, and reciprocal square root. The acceleration provided by MAU can be beneficial for applications, such as motor control and sensor data analysis.

3.9 FreeMASTER communication (LPUART)

The Low-Power Universal Asynchronous Receiver/Transmitter (LPUART) module of the MCX A34x MCU is used for FreeMASTER communication between the MCU board and the host computer. For FreeMASTER communication, use these configuration settings:

- The baud rate is set to 115200 bit/s.
- Both the receiver and transmitter are enabled.

4 Set up and run demo

This section demonstrates how to set up and run the 3-phase PMSM control and power factor correction demo. The demo uses the following three motors:

- Two low-voltage motors (LINIX 45ZWN24-40)
- One high-voltage motor

The motor parameters and software structure for a low-voltage motor are described in *MCUXpresso SDK Field-Oriented Control (FOC) of 3-Phase PMSM and BLDC Motors* user guide (PMSMFRDMMCXA153).

The motor parameters for a high-voltage motor are described in Table 2.

Table 2. High-voltage motor parameters

Characteristic	Symbol	Value	Unit
Rated voltage	Vt	380	V
Rated speed	-	1500	RPM

Table 2. High-voltage motor parameters*...continued*

Characteristic	Symbol	Value	Unit
Rated torque	T	3	Nm
Rated power	P	1100	W
Continuous current	Ics	2.5	A
Number of pole pairs	pp	4	-

4.1 Prepare demo setup

Follow these steps to prepare the demo setup:

1. Connect one of the two FRDM-MC-LVPMSM boards to the Arduino connectors J1_M2 – J4_M2 of the dual-motor adapter board.
2. Connect the other FRDM-MC-LVPMSM board to the Arduino connectors J1_M3 – J4_M3 of the dual-motor adapter board.
3. Connect the MCX A34x 3MC controller card to the dual-motor adapter board through ribbon cables.
   Note: *Hardware design files of the MCX A34x 3MC controller card are available on nxp.com as application note software for this application note.*
4. Connect the 3-phase wire of one of the two low-voltage motors (LINIX 45ZWN24-40) to the connector J7 on one FRDM-MC-LVPMSM board according to the phase sequence given below:
   - White wire – phase A
   - Blue wire – phase B
   - Green wire – phase C
5. Similarly, connect the 3-phase wire of the other low-voltage motor to the connector J7 on the other FRDM-MC-LVPMSM board.
6. Supply power to each FRDM-MC-LVPMSM board with a separate 24 V adaptor.
7. Connect the MCU-Link USB connector J2 of the MCX A34x 3MC controller card to the host computer through a mini-USB cable.
   Note: *You are recommended not to program the controller card when plugging it into the HVP motherboard and when it is in a power-on state.*
8. On the host computer, start MCUXpresso IDE.
9. Compile the program code.
10. Download the latest debug firmware (CMSIS-DAP or J-Link, based on the existing firmware in the onboard debugger) by clicking the **Debug** button in the IDE.
11. Plug the controller card into the HVP motherboard.
12. Connect the 3-phase wire of the high-voltage motor to the HVP motherboard.
13. Connect the connector J4 of the HVP motherboard to the FreeMASTER tool for motherboard debugging through a USB Type-B cable.
14. Supply power to the HVP motherboard.

Figure 5 shows the demo setup with boards and motors connected.

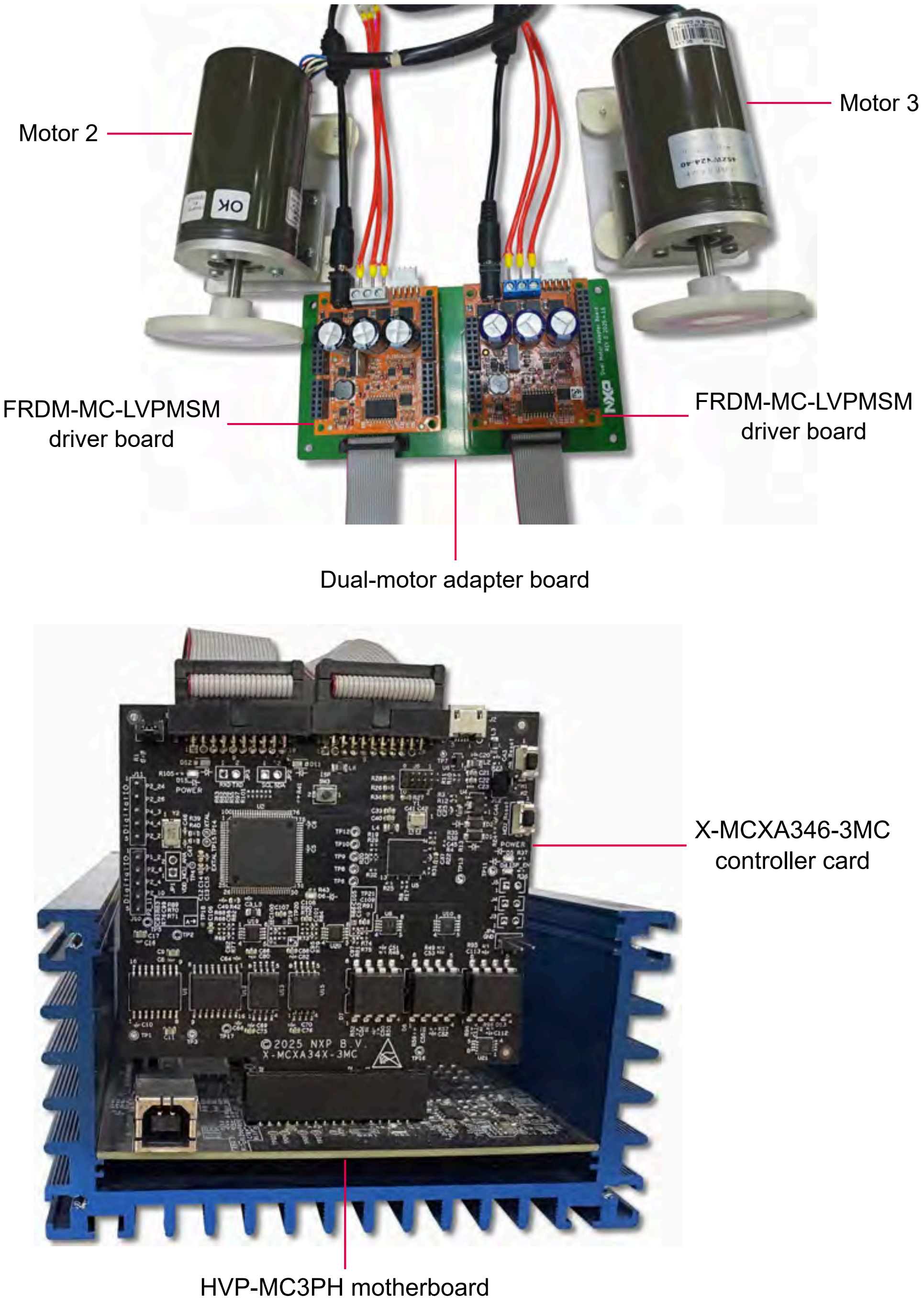
Motor 2
Motor 3
FRDM-MC-LVPMSM driver board
FRDM-MC-LVPMSM driver board
Dual-motor adapter board
X-MCXA346-3MC controller card
HVP-MC3PH motherboard

Figure 5. Demo connections

4.2 Set up demo project in MCUXpresso IDE

The demo project is available on NXP Application Code Hub. You can clone the project code from the MCUXpresso IDE itself using the following steps:

1. Start the MCUXpresso IDE.
2. Click the "Import from Application Code Hub" link in the **Quickstart Panel**. The **Application Code Hub** wizard starts with the **Import projects from Application Code Hub** page.

3. Find the demo project by performing a search operation or by applying filtering.
4. Click the project tile to open the project description page.
5. Click "GitHub link" at the top of the page to copy the GitHub link of the demo project.
6. Click the **Next** button. The **Branch Selection** page of the wizard appears.
7. Select the **main** branch and click the **Next** button. The **Local Destination** page of the wizard appears.
8. Specify the path to the local folder where you want to save the project files in the **Directory** field of the **Destination** group.
9. Click the **Next** button. The **Select a wizard to use for importing projects** page of the wizard appears.
10. Select the **Import existing Eclipse projects** option. The MCUXpresso IDE clones the repository to the local folder path that you specified.
11. After cloning completes, click the **Next** button. The **Import Projects** page of the wizard appears.
12. Select the project in the repository (only one project in the repository).
13. Click the **Finish** button.

4.3 Run demo

Use the FreeMASTER project in the demo code package to control the motor, change rotor speed, and observe the speed or other values.

5 CPU loading performance

Table 3 shows the CPU loading performance and memory usage for the 3-phase PMSM control and power factor correction demo.

Table 3. CPU loading performance and memory usage

Debug configuration	Flash usage	SRAM usage	SRAMX usage	Fast loop loading	Slow loop loading	Total loading
All code in flash	65.9 KB	30.3 KB	0	65.38%	2.98%	68.36%
Partial code in SRAM/SRAMX	66.3 KB	35.2 KB	4.4 KB	30.09%	1.14%	31.23%

6 References

For more details on the MCX A34x MCU, refer to *MCX A345 and A346 Reference Manual* (MCXAP144M240F60RM) and *MCXA345/346 Data Sheet* (MCXAP144M240F60).

For more details on the MCX A34x functions in NXP RTCESL, refer to *GMCLIB User's Guide* (CM33FGMCLIBUG) and *AMCLIB User's Guide* (CM33FAMCLIBUG).

For more details on the sensorless PMSM FOC, refer to *Sensorless PMSM Field-Oriented Control* (DRM148).

For more details on the FRDM-MC-LVPMSM board, refer to NXP FRDM Development Platform for Low-Voltage, 3-Phase PMSM Motor Control.

For more details on motor parameters and software structure for a low-voltage motor, refer to *MCUXpresso SDK Field-Oriented Control (FOC) of 3-Phase PMSM and BLDC Motors* (PMSMFRDMMCXA153).

For more details on the HVP-MC3PH board, refer to High-voltage development platform.

7 Acronyms

Table 4 lists the acronyms used in this document.

Table 4. Acronyms

Acronym	Description
3MC	Three Motor Control
AC	Alternating Current
ADC	Analog-to-Digital Converter
BEMF	Back Electromagnetic Field
CMSIS	Common Microcontroller Software Interface Standard
CPU	Central Processing Unit
DAC	Digital-to-Analog Converter
DAP	Debug Access Port
DC	Direct Current
DNP	Do Not Populate
eDMA	Enhanced Direct Memory Access
eFlexPWM	Enhanced Flex Pulse Width Modulator
FIRC	Fast Internal Reference Clock
FOC	Field-Oriented Control
GND	Ground
GPIO	General-Purpose Input/Output
I^2C	Inter-Integrated Circuit
INPUTMUX	Input Multiplexing
I/O	Input/Output
IDE	Integrated Development Environment
ISP	In-System Programming
ISR	Interrupt Service Routine
LED	Light-Emitting Diode
LPUART	Low-Power Universal Asynchronous Receiver/Transmitter
MAU	Math Acceleration Unit
MCU	Microcontroller Unit
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
PFC	Power Factor Correction
PMSM	Permanent Magnet Synchronous Motor
PWM	Pulse Width Modulation/Modulator
RMS	Root Mean Square
RoHS	Restriction of Hazardous Substances
RTCESL	Real-Time Control Embedded Software Libraries
SCL	Serial Clock
SDA	Serial Data
SPI	Serial Peripheral Interface

Table 4. Acronyms*...continued*

Acronym	Description
SRAM	Static Random-Access Memory
SVPWM	Space Vector Pulse Width Modulation
SWD	Serial Wire Debug
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus

8 Revision history

Table 5 summarizes the revisions to this document.

Table 5. Revision history

Document ID	Release date	Description
AN14805 v.2.0	18 December 2025	• Changed document title to indicate triple-motor control • Updated Application Code Hub link in Section 1 and Section 4.2 to point to the triple-motor control demo • Updated Section 3.2 • Added Section 3.3 • Updated Section 3.4
AN14805 v.1.0	13 October 2025	Initial public release

Legal information

Definitions

Draft — A draft status on a document indicates that 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 in a draft version of a document and shall have no liability for the consequences of use of such information.

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.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

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.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

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

NXP — wordmark and logo are trademarks of NXP B.V.

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile — are trademarks and/or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

IAR — is a trademark of IAR Systems AB.

J-Link — is a trademark of SEGGER Microcontroller GmbH.

Contents

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

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

Document feedback

Date of release: 18 December 2025
Document identifier: AN14805