AN14708

HVAC Outdoor Unit Controller Solution based on MCX A34x

Rev. 2.0 — 11 July 2025

Application note

Document information

Information	Content
Keywords	AN14708, HVAC, MCX A34x, PMSM, FOC, PFC, compressor, fan
Abstract	This application note provides a solution for simultaneously implementing sensorless Field Orientation Control (FOC) of 2 Permanent Magnet Synchronous Motors (PMSM) and a single-phase Power Factor Correction (PFC) converter using a single chip.

1 Introduction

To improve integration and control board space utilization and save hardware costs, more and more air conditioning manufacturers are using a 3-in-1 control solution of a fan, a compressor, and a PFC in the outdoor air conditioner units. This application note provides a high-performance and cost effective solution for simultaneously implementing sensorless Field Orientation Control (FOC) of 2 Permanent Magnet Synchronous Motors (PMSM) and a single-phase Power Factor Correction (PFC) converter using a single MCX A34x chip. This control system is named 3in1 system for short in this document.

2 Application features and components

The system is designed to drive 2 sensorless 3-phase PMSMs (compressor and fan) and single-phase Power Factor Correction (PFC). The application features are as follows:

Compressor control:

- Sensorless 3-phase PMSM speed vector (FOC) with a 6.25-kHz control frequency
- Torque compensation
- Harmonic suppression
- Field-weakening control
- Single-shunt sampling and phase current recognition
- Align and open-loop startup
- 1-kHz speed loop control
- 3 pole pairs, 1200 W of power, 10.12-A peak current
- Mechanical speed range: 240-3600 RPM
- Maximum acceleration of 15000 RPM/s

Fan control:

- Sensorless 3-phase PMSM speed vector (FOC) with a 16-kHz control frequency
- On-the-fly startup when the rotor is rotating
- 1-kHz speed loop control
- 5 pole pairs, 25 W of rated power, 0.5-A peak current
- Mechanical speed range: 100-900 RPM
- Acceleration of 100 RPM/s

PFC:

- Single-shunt for current sampling
- 64-kHz PWM
- 32-kHz fast current loop control
- 5-kHz slow voltage loop control
- 20-A peak AC current input

There are also the following components available:

- Serial communication based on SCI-to-USB, connected to NXP FreeMASTER debugging tools
- Over-current protection for motor control and PFC
- AC input frequency range detection
- Functional safety (IEC60730 Class B)

3 NXP MCX A34x advantages and features

The MCX A34x features a Cortex-M33 core with a frequency of up to 180 MHz, primarily designed for industrial and consumer IoT applications. It also integrates specialized peripheral modules, such as Pulse Width Modulators (PWM), Analog-to-Digital Converters (ADC), timers, communication peripherals (UART, SPI, IIC), decoders (QDC), and on-chip Flash and RAM storage. With these advanced peripherals and features, the MCX A346 MCU can achieve vector (FOC) control and single-phase power factor correction (PFC) control of 2 sensorless PMSMs using a single chip.

MCX A34x mainly includes the following peripheral modules and functions:

- Arm Cortex-M33, 180-MHz with 741.6 CoreMark (4.12 CoreMark/MHz)
- Up to 1 MB of Flash with 8-kB cache (0-wait state) and 256 kB of SRAM (8 kB of RAM with ECC)
- MAU (Math Accelerate Unit) supports trigonometric functions (sine, cosine, and arctan algorithms), reciprocal, square, and square root, which can greatly boost the motor-control algorithm execution speed
- 2x FlexPWM, each with 4 submodules, providing 16 complementary outputs of PWM
- 2x Quadrature Encoder/Decoder (eQDC)
- 2x AOI (AND/OR/Invert) module with support up to 4 output triggers
- 4x 16-bit ADC, up to 3.2 Msps in the 16-bit mode, and 4 Msps in the 12-bit mode
- 1x 12-bit DAC, up to 1 Mbit/s
- 3 x high-speed comparators with 8 input pins and an 8-bit DAC as internal reference
- 4x OPAMP without PGA
- 2x LPSPI, 4x LPI2C, 6x LPUART
- 1x FlexCAN with FD
- 5x 32-bit standard general-purpose asynchronous timers/counters, which support up to 4 capture inputs and 4 compare outputs, PWM mode, and an external count input
- 1x SmartDMA, able to emulate different serial/parallel interfaces (parallel LCD, camera, I2S, I2C), key scan, DMA, PWM, and fulfill special customer requirements

Figure 1 is the block diagram of the MCX A34x MCU:

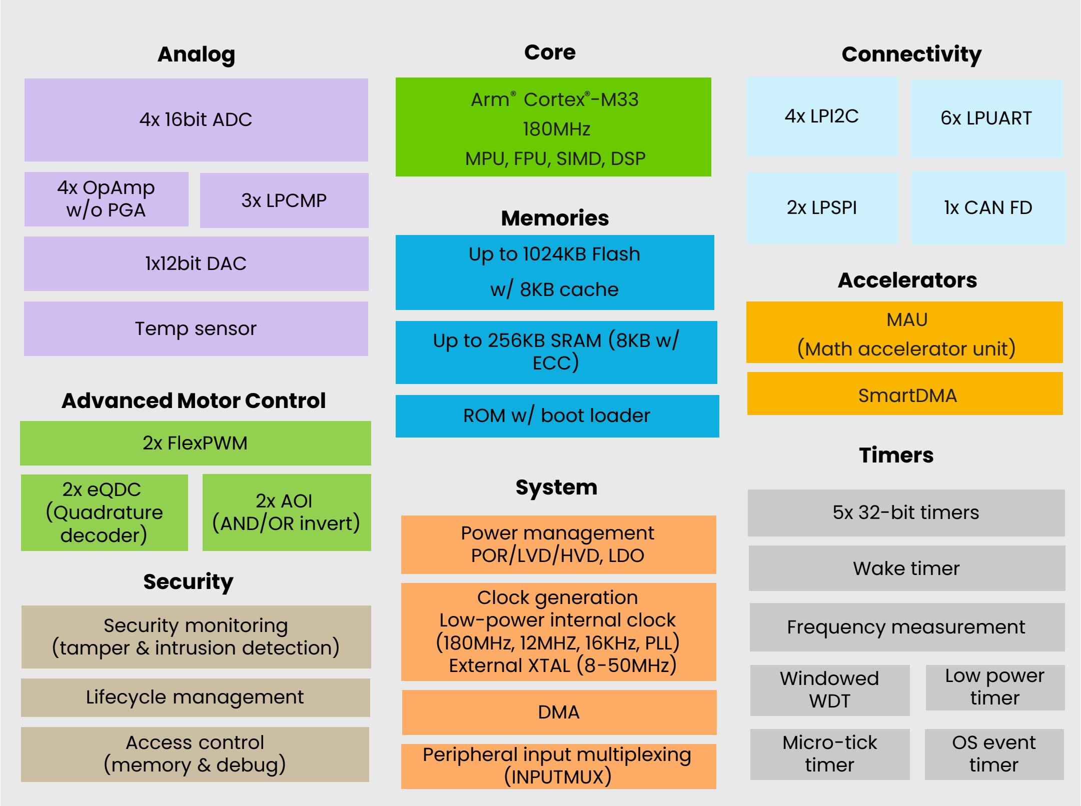


Figure 1. Block diagram

4 System concept

This section describes the system concept.

4.1 Dual motors and PFC control system

The block diagram of a dual-motor-control system with a digital PFC converter is shown in Figure 2. This is a typical architecture for an air conditioning outdoor unit application system: one motor driver drives the compressor, the other motor driver drives the fan, and the PFC converter controls the DC-Bus voltage.

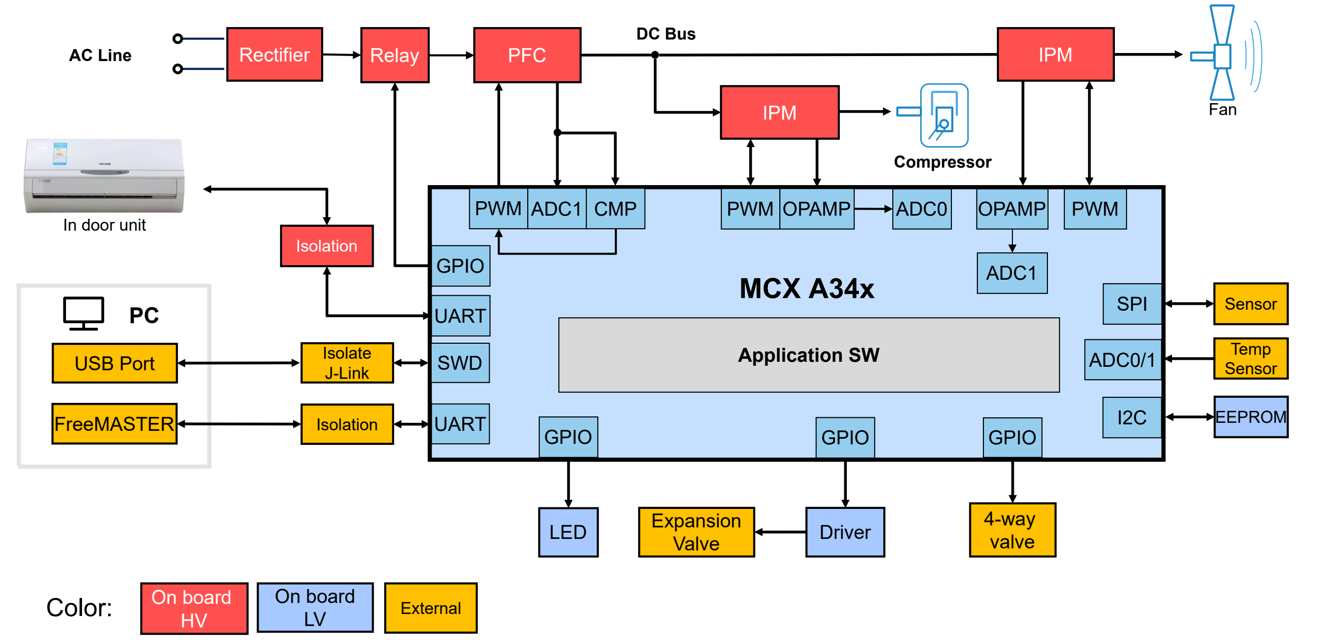


Figure 2. HVAC 3in1 system architecture

4.2 Compressor control

The software algorithms used to control the compressor and fan are basically the same. The basic control algorithm uses NXP's embedded motor software library (RTCESL). The compressor control is different from the general motor-control technology. Based on the load characteristics of the compressor, we use special control algorithms to improve the control performance. The following sections introduce them separately.

4.2.1 Startup process

The motor inside the compressor drives a crankshaft, which in turn drives a piston moving in a reciprocating motion. The refrigerant is compressed in this motion. Since the high-side pressure is much higher than the low-side pressure, there’s a significant load torque change in 1 mechanical revolution. When the motor is running at a high speed, this periodic load torque change is not a big problem because the load change in a very short period does not lead to significant speed variation. When the compressor has been idle for a significant period of time, the pressure difference between the high side and the low side still exists, and it comes to 0 over time. When there is a large residual pressure difference, the loading can be either large or small at the very moment of startup because the exact rotor and piston position are not known. Therefore, we do not know whether the piston is to move against the pressure or the other way around at this very moment of startup, which makes the startup challenging. In practical use, when a compressor is stopped, it is not started immediately, even if there is a valid speed command, unless a couple of minutes (usually 1~3 min) have passed by. When the ambient temperature of a compressor is high, the residual pressure difference can be still big, which makes the startup challenging. The startup method mentioned here uses a fast-converging flux observer at an open-loop startup, which greatly shortens the startup time.

As the rotor position observer does not give a reliable feedback at very low speeds, the motor must reach a certain speed in the open-loop mode. The start-up process expects similar start-up conditions for each startup. The method consists of a generated rotating field with the Q current profile that spins the rotor according to the generated speed.

The startup process is divided into 4 parts, described in the following subsections.

4.2.1.1 Alignment

To start the motor, align the rotor to a known place to ensure the maximum start-up torque. At the beginning of the alignment phase, a voltage is applied to the d-axis for a period of time. After the alignment phase is completed, the rotor position is aligned to 0 degrees. Figure 3 shows the control diagram.

4.2.1.2 Open-loop startup

After alignment, the rotor is placed at position 0. A current vector is constructed at position 0 and starts at the specified ramp. In this process, the electrical torque increases as the angle between the current vector and the rotor flux increases. There is no loss of synchronization when the angle between the current vector and the rotor flux stays below 90°. It needs a large current to pull up a heavy load in this fashion. In this stage, the d-axis position is generated by the integral of the ramping speed. The speed stops increasing when it reaches the catch-up speed and starts to merge. The observer starts to work when the current vector speed is higher than a half of the catch-up speed.

4.2.1.3 Merge

At the end of the open-loop startup stage, as soon as the current vector speed reaches the catch-up speed, it is assumed that the rotor has reached the catch-up speed and the observer is working properly. It is time to switch from the manually given rotor position PosExt to the observed rotor position PosEst.

There must be a gap between PosExt and PosEst when the current vector speed reaches the catch-up speed. It is not recommended to switch from PosExt to PosEst abruptly, because it can cause undesired current spikes. A new position, PosMerged, makes the transition smooth:

- PosMerged = PosExt + Coeff * (PosEst − PosExt)

This Coeff variable determines how much of the estimated position and speed information is merged with the generated ones. At the beginning, it is 0 % for the estimated and 100 % for the predicted. Their sum is always 1. Coeff increases from 0 % to 100 % within 83 fast loops. It takes 13.3 ms to move from the manually given position (PosExt) to the observed position (PosEst).

4.2.1.4 Close-loop spin

At the end of the merge stage, when Coeff reaches 1, the observed position is used for FOC and the speed loop control is enabled. The actual q-axis current is set to the integral part of the speed controller to make sure that there is no step change on the q-axis current reference. Now the speed loop and the current loops are fully engaged, and the rotor position and speed feedback come from the observer.

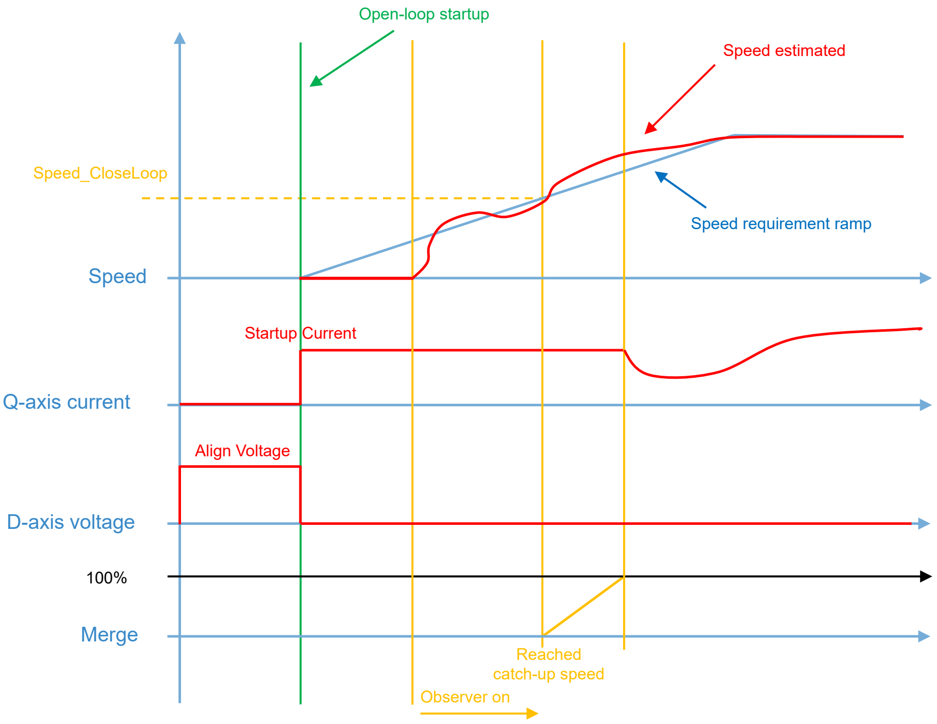


Figure 3. Compressor startup process

4.2.2 Field-weakening control

The motors can also operate in the speed and torque ranges where the DC-Bus voltage is not high enough over the BEMF of the motor. This condition is given by the difference of the Q voltage reserve (difference between the Q voltage limit and the applied Q voltage) and the Q current error. If the difference is negative, the Field PI Controller starts reducing the required D current. The field-weakening algorithm executes to reduce the permanent magnet magnetic flux and lower the actual BEMF. Having lowered the BEMF, the current can be delivered to the motor phase from the DC-Bus again. The speed increases but the capability of the maximum torque is proportionally reduced. This method of field weakening was developed and later patented by Freescale (WO/2009/138821).

4.2.3 Torque compensation

Due to cost considerations, single-rotor compressors are most widely used in household air conditioning. The compression process has a "compress" phase and a "release" phase. The torque requirement is different in these 2 phases. The conventional closed loop feedback architecture, as used in the speed loop, does not solve this issue due to the delay of the PI regulator. The solution adds a feedforward torque component to compensate for the compression process.

The mechanical angle is tracked by monitoring the feedback speed and checking the differences with the target speed. If the real speed continues to be lower than target speed, it must be in the "compress" phase and add some torque for compensation. If the real speed is higher than the target speed, it must be in the "release" phase and reduce some torque for compensation.

Figure 4 shows the diagram of the torque compensation.

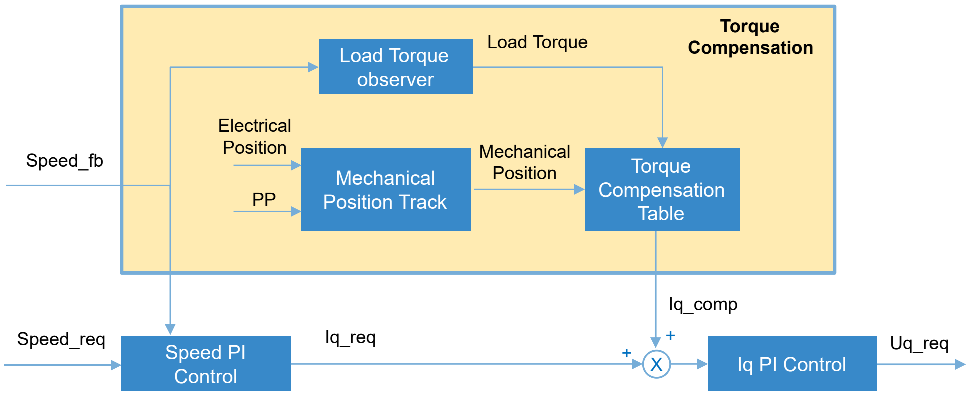


Figure 4. Torque compensation structure diagram

4.2.4 Harmonic suppression

Due to the PWM dead time and the PWM phase shift caused by the single-shunt sampling strategy, combined with the 5th / 7th harmonic in the back electromotive force of the compressor, the harmonic is generated in the phase current when using conventional motor control algorithms for the FOC control.

From the perspective of the field-oriented control, these harmonics at the phases manifest as AC disturbances in the D/Q voltages. The 2 current loops try to suppress these disturbances, but their ability to do so depends on the current loop bandwidth and the magnitude of the disturbances. Since both the magnitude and frequency go up with the motor speed, even high-bandwidth current loops cannot keep up.

Due to the use of single-resistor sampling in this scheme and the control frequency of the compressor current loop being only 6.25 kHz. The bandwidth of the current loop is set to 200-300 Hz. The harmonic frequency of the current on the d/q axis is 6 times the fundamental frequency.

Therefore, the highest frequency observed is given by the following equation:

- N_max / 60 * PP *6 = 3600 RPM / 60 * 3 * 6 = 1080 Hz

This value is higher than the current-loop bandwidth.

Therefore, relying solely on the regulating effect of current loop PI control cannot eliminate these current harmonics.

The phase-current waveform of the compressor running at 1500 RPM using a conventional FOC algorithm is shown in Figure 5.

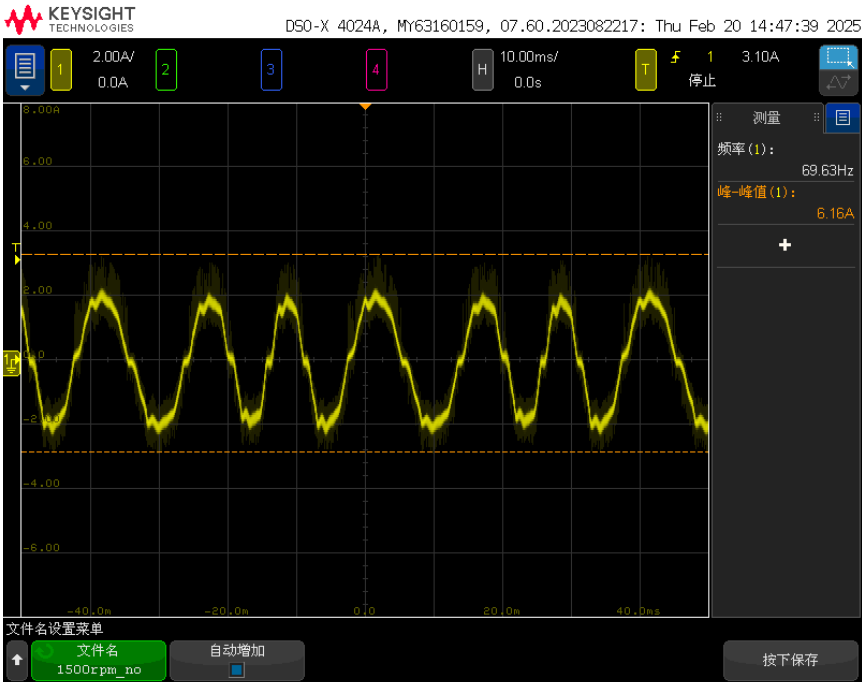


Figure 5. Phase current before harmonic suppression

To suppress the current harmonics, we have added a harmonic suppression function in the d-axis current control. After the harmonic suppression is turned on, the 6th harmonic component in the d-axis current is significantly reduced, thereby reducing electromagnetic noise and improving EMI performance. The waveform is shown in the Figure 6.

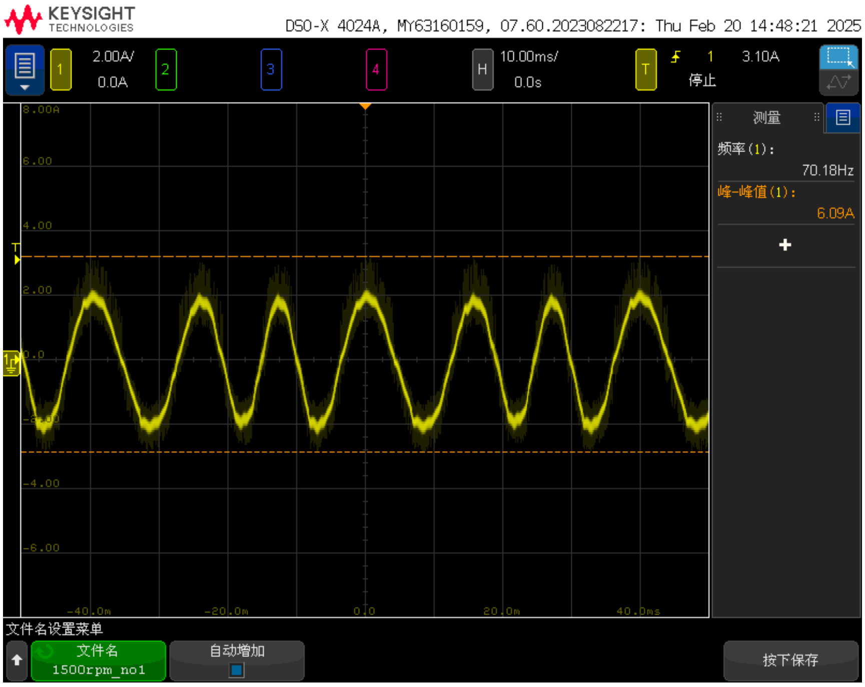


Figure 6. Phase current after harmonic suppression

4.3 Fan control

The on-the-fly startup and anti-wind startup functions are necessary features for fan applications. Most fans with larger blades are prone to passive rotation due to air movement caused by airflow. In this case, it is necessary to directly enter the closed-loop operation during the motor rotation process. Compared to the traditional 3-step startup method, it eliminates the process of alignment and open-loop startup and shortens the startup time.

Therefore, we used a special flux observer with a very low convergence speed threshold, which allows it to achieve a 0 speed in the closed-loop startup under light-load conditions.

4.3.1 Zero-speed startup

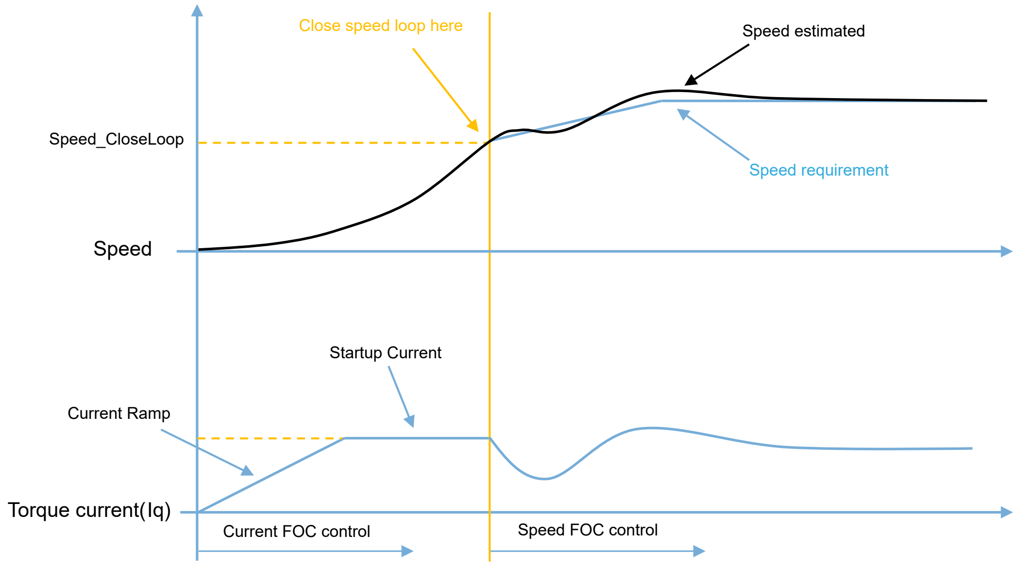


Figure 7. Zero-speed startup process for a fan

Figure 7 shows the process of a 0-speed closed-loop startup of the fan. When receiving the motor speed command, the flux observer starts working directly and controls the Id to 0. Iq gradually increases to the startup current through a ramp. At this point, the motor starts to accelerate slowly. After the motor speed exceeds a certain threshold (Speed_CloseLoop), the close-loop speed control is enabled, and it enters the speed FOC control, the motor spins according to the speed requirement.

4.3.2 On-the-fly startup

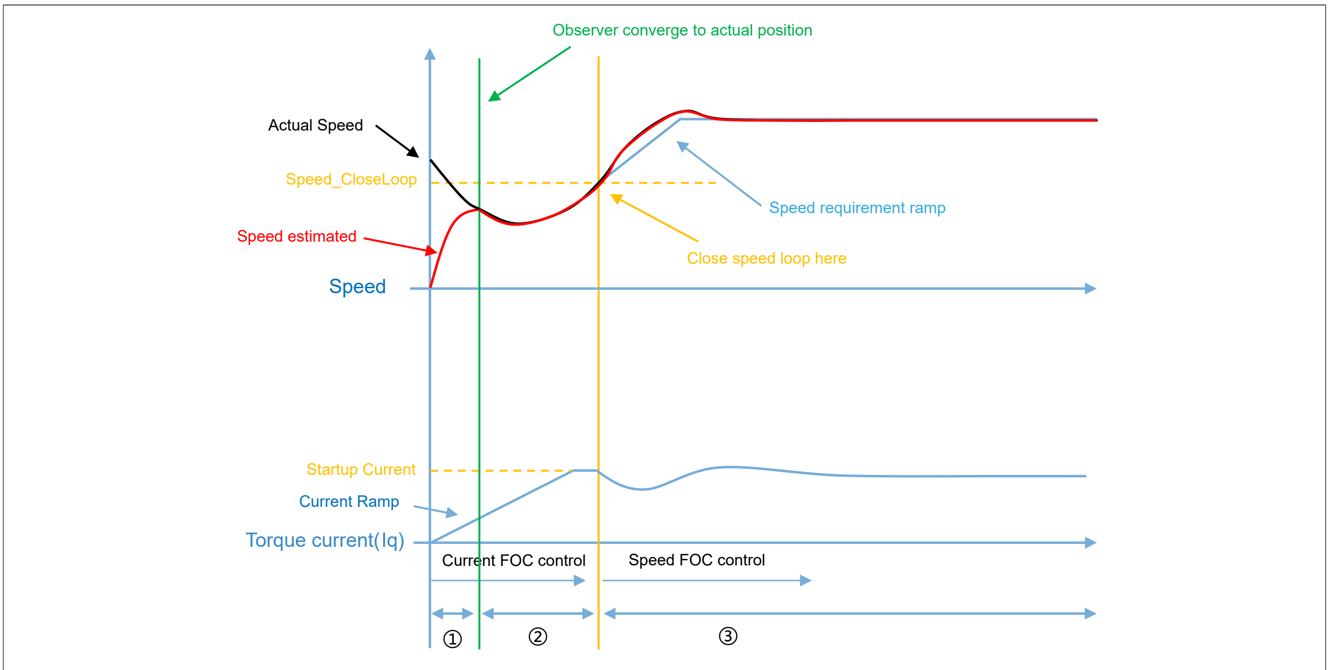


Figure 8. On-the-fly startup process for fan

The process of starting with a tailwind is similar to the 0-speed starting process in using a current loop to directly start in a closed loop and then switching to the speed loop after reaching a certain speed. It is divided into 3 stages, as shown in Figure 8.

1. It this stage, the observer operates directly in a closed-loop manner, with the d-axis current controlled to 0 and the q-axis current gradually increasing in a ramp manner. During this stage, the estimated rotor position and the velocity output by the observer gradually converge to the actual values.
2. It this stage, the q-axis current continues to gradually increase in a ramp manner until it reaches Startup_Current, and the motor speed increases accordingly. It determines the motor speed and when the speed reaches Speed_CloseLoop, it closes the speed loop and enters the third stage.
3. At this stage, it enters the speed FOC control mode and the motor speed follows the ramp change of the speed command.

4.3.3 Anti-wind startup

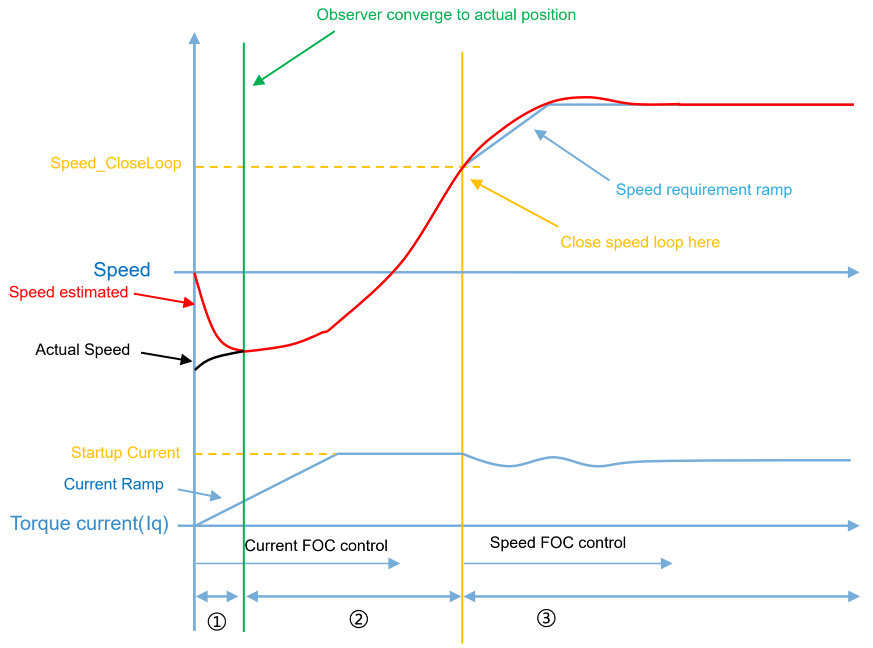


Figure 9. Anti-wind startup process for fan

The process of starting against the wind is also divided into 3 stages:

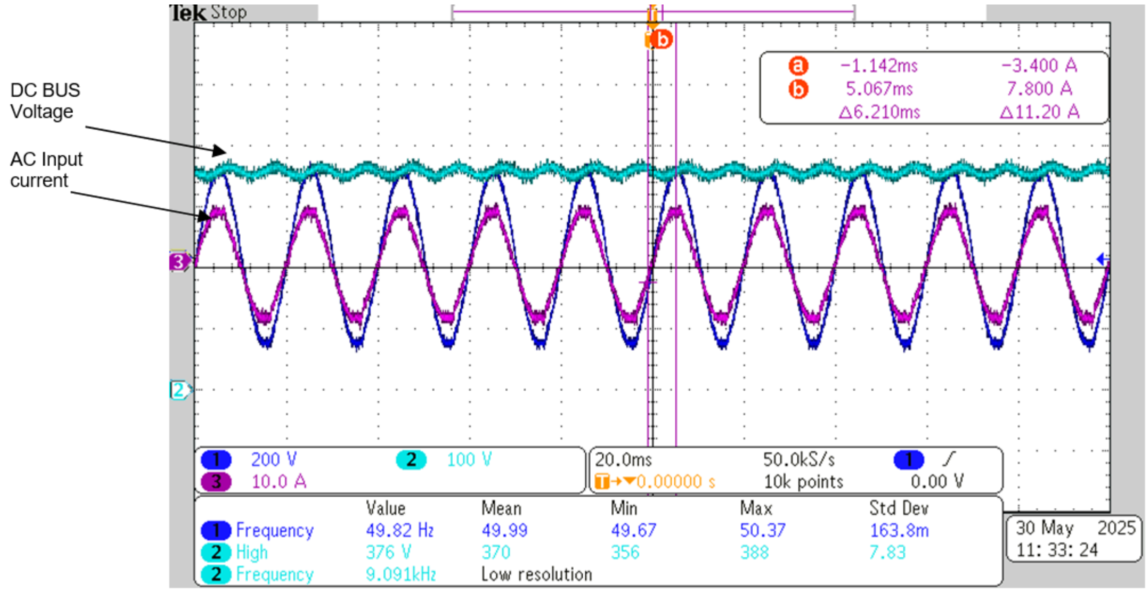
1. The initial speed of the motor is opposite to the given speed direction and the q-axis current gradually increases in a ramp manner. During this stage, the rotor position and speed output by the observer gradually converge to the actual value.
2. The q-axis current continues to increase until it reaches the Startup_Current value, at which the motor rotor gradually decelerates and crosses the 0 speed to begin forward acceleration. When the speed reaches Speed_CloseLoop, the closed-loop speed control is engaged.
3. At this stage, it enters the speed FOC control mode and the motor speed follows the ramp change of the speed command.

4.4 PFC control

The PFC circuit uses a 300-μH inductor (L = 300 μH). The PFC circuit defaults to 360 V when enabled (under a full load state, the waveform is shown in Figure 10) but it can be adjusted to 380 V to increase the load power to 1500 W.

The following 2 control loops are used for the PFC control:

- The outer loop controls the DC-Bus voltage.
- The inner loop controls the inductor current, which makes the current sinusoidal and maintains the same phase with the input voltage.



```
ⓐ  -1.142ms     -3.400 A
ⓑ  5.067ms      7.800 A
   Δ6.210ms     Δ11.20 A

1  200 V      2  100 V      20.0ms            50.0kS/s     1  ∫
3  10.0 A                   T→▼0.00000 s      10k points   0.00 V

              Value      Mean    Min     Max     Std Dev
1 Frequency   49.82 Hz   49.99   49.67   50.37   163.8m
2 High        376 V      370     356     388     7.83
2 Frequency   9.091kHz   Low resolution

30 May 2025
11: 33: 24
```

Figure 10. PFC waveforms under 1200 W

4.4.1 AC input voltage peak value and phase detection

The current control of the PFC requires the real-time AC input peak voltage value and phase information to compensate for the controller and get the current reference.

The AC input voltage peak value is detected in CTIMER0_IRQHandler, which is updated in a 5-kHz intervals. The actual DC-Bus voltage is also updated in this ISR. When a voltage increase is detected, the algorithm starts tracking, and the maximum value detected is deemed as the peak value when the voltage starts to fall. The PFC cannot be enabled until 8 AC input voltage peaks are detected.

The AC input voltage phase (0-crossing) is detected in FLEXPWM1_SUBMODULE1_IRQHandler, which is calculated in a 32-kHz frequency. During the AC input peak voltage detection, the program starts to detect a rising threshold after a voltage-rising trend is confirmed, and the program starts to detect a falling threshold after a voltage-falling trend is confirmed. The rising and falling thresholds have the same value. The time interval between the adjacent detected falling thresholds is a half of the AC input period. The rising voltage detection point is used to calculate the AC input 0-crossing point. The sinusoidal reference for the current can be calculated with the detected 0-crossing point and the AC input period.

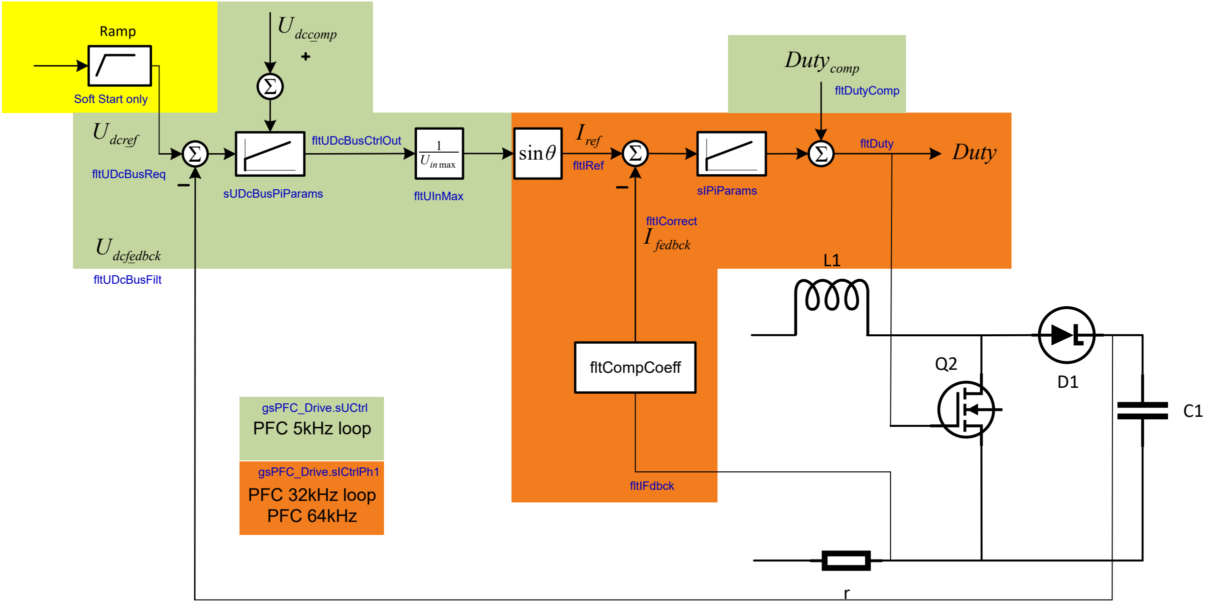


Figure 11. PFC control diagram

5 Peripheral configuration

This section describes the key peripheral configuration.

5.1 Timing control

To achieve control of 2 motors and 1 PFC using 1 chip, PWM synchronous timing control is very important. In this solution, the PWM frequencies of the fan and the PFC are set to an integer multiple of each other to prevent ADC sampling-sequence conflicts. The compressor PWM frequency is set to a noninteger multiple of the fan PWM frequency to improve the EMI performance.

For comprehensive consideration of the operating noise and the switch losses of the fan, the PWM frequency of the fan is set to 16 kHz. The PWM frequency of the PFC must be an integer multiple of the PWM frequency of the fan. Increasing the PWM frequency of the PFC can reduce the size of the filtering inductor and save costs, but it increases the losses of the switching tube and the CPU load. After a comprehensive consideration, 64 kHz is chosen as the PWM frequency. The fan algorithm and the PFC algorithm are calculated every 2 PWM cycles, so the control frequency of the fan is 8 kHz and the PFC control frequency is 32 kHz.

In Figure 12, the PFC control algorithm is arranged between the 2 motor-control algorithms. This arrangement ensures that the PFC control and the fan do not interfere with each other. Considering the large amount of algorithm computation in the fan motor control, it is likely that all calculations for this control cannot be completed before the next PFC control interrupt arrives. How to fully utilize the time gap is mentioned above. The system sets the PFC interrupt to have a higher priority than the fan control interrupt. In this way, without interrupting the PFC operation, the motor can use this time gap to complete calculations before the corresponding PWM reload signal arrives.

To avoid the impact of spikes generated by PFC switches on motor-current sampling, the current sampling point of the motor always aligns with the midpoint of the PFC PWM (as shown in Figure 12). To achieve this goal, the whole PWM is set to the center alignment mode. Considering the actual hardware delay, the system sets the PWM synchronization of the 2 motors to be completed through the hardware synchronization, with a specific reference to the timing synchronization design of the motor-control algorithm. The PWM synchronization of the

PFC is achieved by setting a PWM comparison interrupt after the first motor PWM is running and starting the PFC PWM during the interrupt.

The PWM timing of the 2 motors and the PFC is shown in Figure 12:

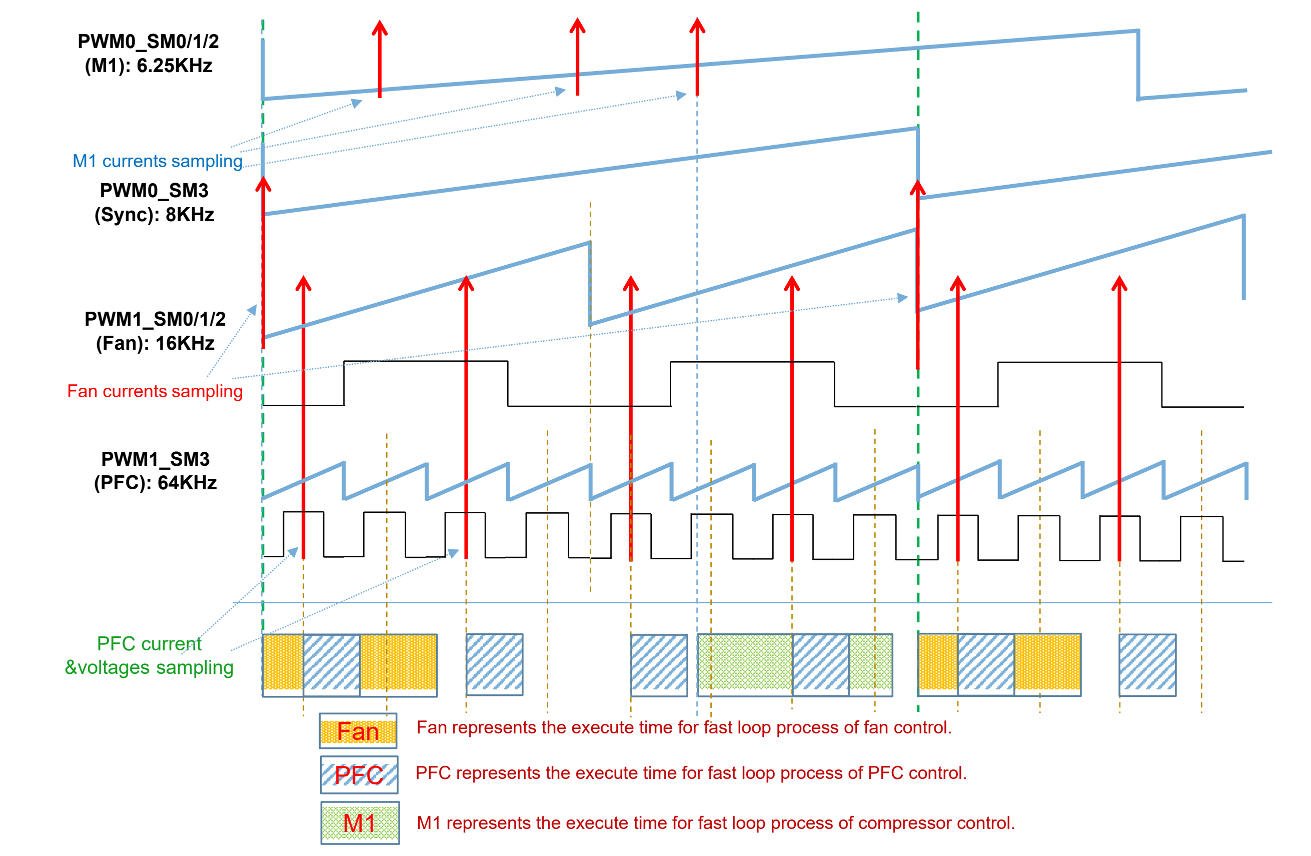

PWM0_SM0/1/2
(M1): 6.25KHz
M1 currents sampling
PWM0_SM3
(Sync): 8KHz
PWM1_SM0/1/2
(Fan): 16KHz
Fan currents sampling
PWM1_SM3
(PFC): 64KHz
PFC current
&voltages sampling
Fan
Fan represents the execute time for fast loop process of fan control.
PFC
PFC represents the execute time for fast loop process of PFC control.
M1
M1 represents the execute time for fast loop process of compressor control.

Figure 12. Control timing overview

The PWM frequency of the compressor is 6.25 kHz, which is output by the SM0 / SM1 / SM2 submodules of PWM0. Due to the use of the single-resistor sampling technology, it is necessary to sample the bus current 3 times in the first half of the PWM cycle, and then obtain the 3-phase current of the motor through the phase current reconstruction algorithm.

The SM3 of PWM0 is set to 8 kHz to provide a synchronization time reference for the fan and the PFC. At the time of reloading the PWM0 SM3 counter, force the clocks of the 4 submodules of PWM1 to achieve clock synchronization between the fan and the PFC.

In this application, both the fan and the PFC undergo the ADC sampling and control every 2 PWM cycles.

All software algorithms in this application are interrupt driven, and interrupts are required to perform algorithm calculations after the ADC sampling sequence is completed. Both the fan and the compressor control utilize the ADC conversion to complete interrupts, while the PFC uses the PWM comparison interrupts. Based on the control frequency and computational complexity, the interrupt priority is set to the PFC being higher than the fan and compressor.

5.2 Key peripheral configuration

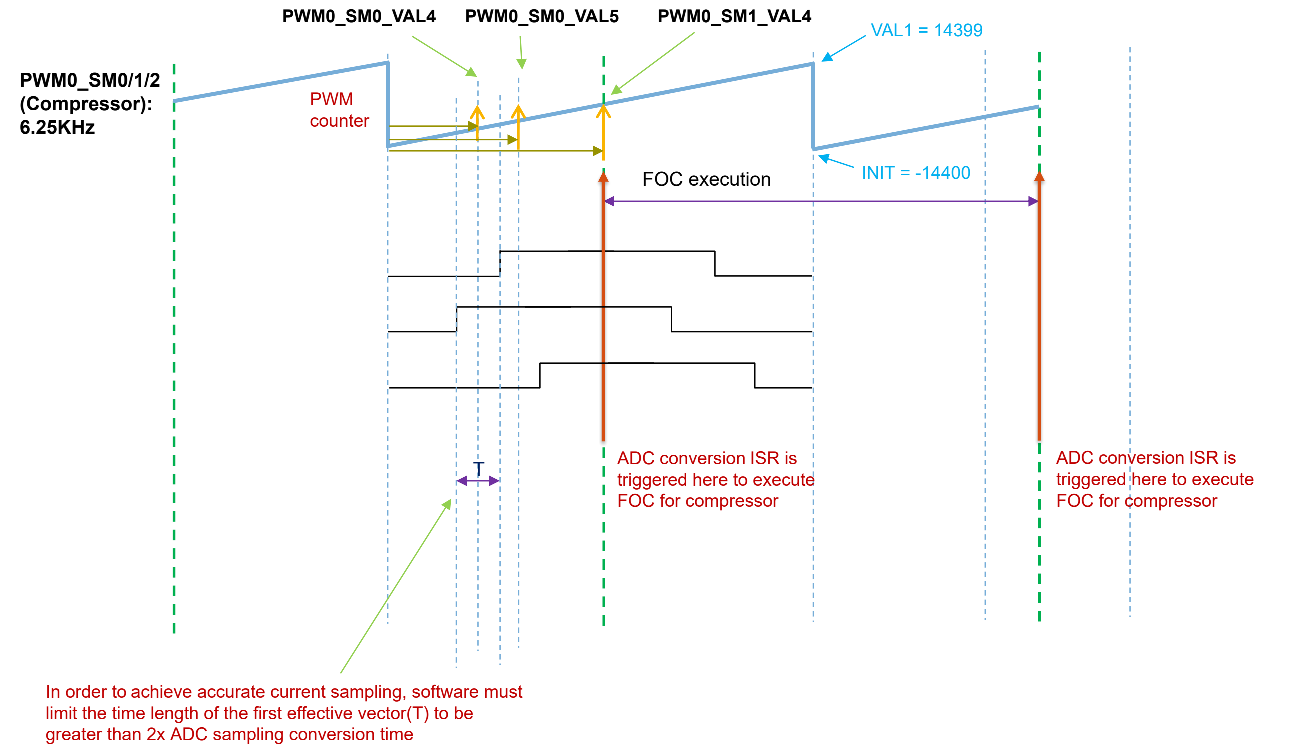
This section describes the key peripheral configuration.

5.2.1 FlexPWM and ADC

The clock of the FlexPWM module is 180 MHz, which is the same as the core clock.

- FlexPWM0: Submodules (SM) 0-2 are for compressor control, and SM3 is for timing synchronization. SM0-2 operates at 6.25 kHz, so the PWM modulo is 180 MHz / 6.25 kHz = 28800. Here, we set the values of the Init register and the VAL1 register to be equal in size and opposite in sign to achieve a center-aligned PWM.

Use the SM0_VAL4 / VAL5 registers and the SM1_VAL4 register to trigger the DC-Bus current sampling and enter the ADC conversion completion interrupt for FOC operation after the 3rd sampling conversion is completed.


PWM0_SM0_VAL4
PWM0_SM0_VAL5
PWM0_SM1_VAL4
VAL1 = 14399
PWM0_SM0/1/2 (Compressor): 6.25KHz
PWM counter
FOC execution
INIT = -14400
T
ADC conversion ISR is triggered here to execute FOC for compressor
ADC conversion ISR is triggered here to execute FOC for compressor

In order to achieve accurate current sampling, software must limit the time length of the first effective vector(T) to be greater than 2x ADC sampling conversion time

Figure 13. Timing control for 1-shunt sampling

PWM0 SM3 operates at 8 kHz, the PWM0_SM3_VAL4 register is used to synchronize the timing of the 4 submodules of PWM1, while triggering ADC1 to sequentially sample the 2-phase currents of the fan. The value of PWM0_SM3_VAL4 should be -180 MHz / 8 kHz / 2 = -11250.

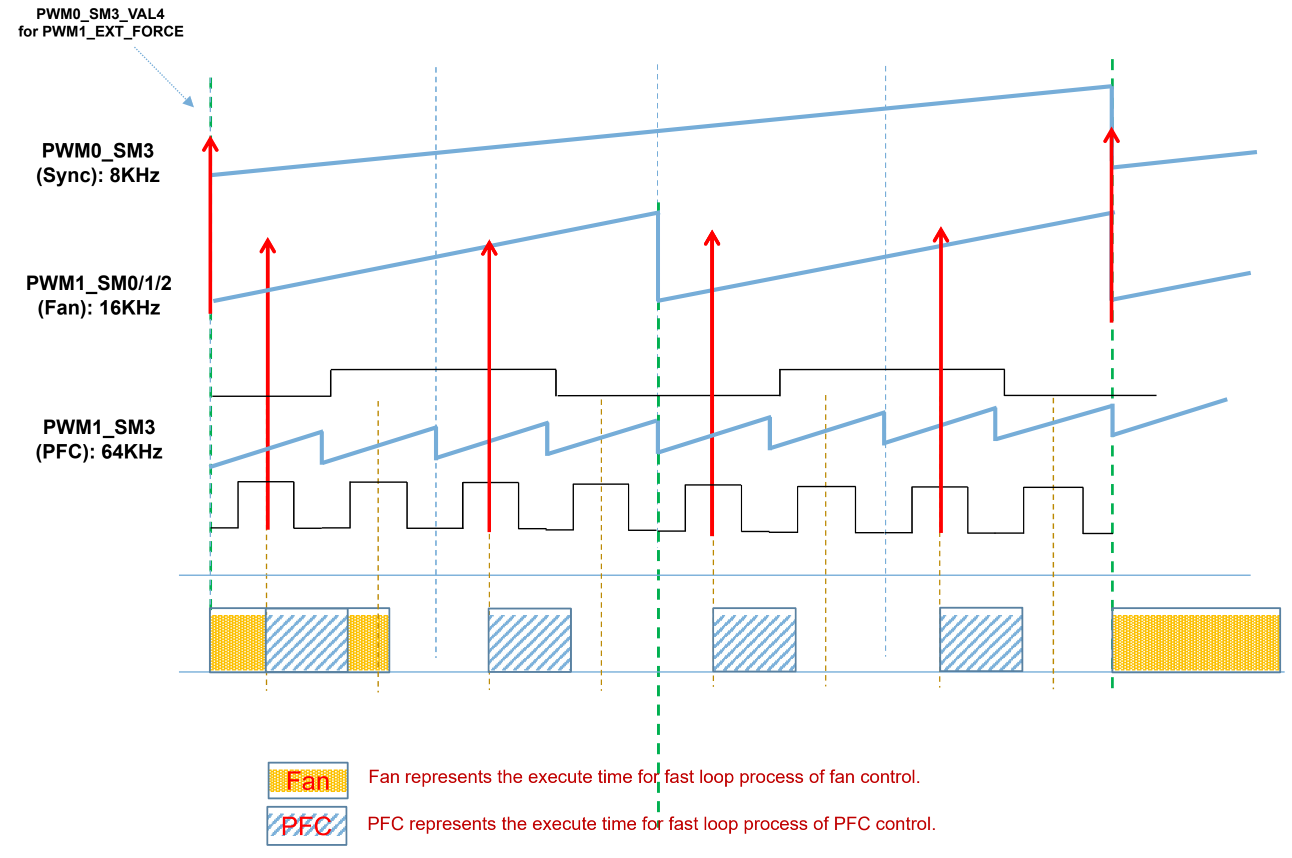

PWM0_SM3_VAL4
for PWM1_EXT_FORCE
PWM0_SM3
(Sync): 8KHz
PWM1_SM0/1/2
(Fan): 16KHz
PWM1_SM3
(PFC): 64KHz
Fan
Fan represents the execute time for fast loop process of fan control.
PFC
PFC represents the execute time for fast loop process of PFC control.

Figure 14. Synchronous PWM between fan and PFC

- FlexPWM1: Submodules (SM) 0-2 are for fan control, and SM3 is for PFC control. SM0-2 work at 16 kHz, so the PWM modulo is 180 MHz / 16 kHz = 11250, SM0_2_INIT = -5625, SM0-2_VAL1 = 5624. SM3 works at 64 kHz, so the PWM modulo is 180 MHz / 64 kHz = 2812.SM3_INIT = -1406, SM3_VAL1 = 1405.

The VAL4 and VAL5 registers of PWM1_SM0 are used to trigger the ADC1 sampling sequence to sample the DC-Bus voltage, DC-Bus current, and the voltage of the AC side. Due to the ADC module having only 1 conversion completion interrupt, it is used for the fan control. PWM comparison interrupts must be used here to execute PFC control algorithms.

The sampling point of the PFC must be at the center of the PWM, so the SM0_VAL4 register is set to -180 MHz / 16 kHz * 3 / 8 = -4219, SM0_VAL5 register is set to 180 MHz / 16 kHz / 8 = 1406. Due to the time required for the ADC sampling and conversion, it is necessary to wait for a period to trigger an interrupt and read the ADC conversion result. The delay time is set to 540 cycles, so the SM1_VAL4 register is set to -4219 + 540 = -3679 and the SM1_VAL5 register is set to 1406 + 540 = 1946.

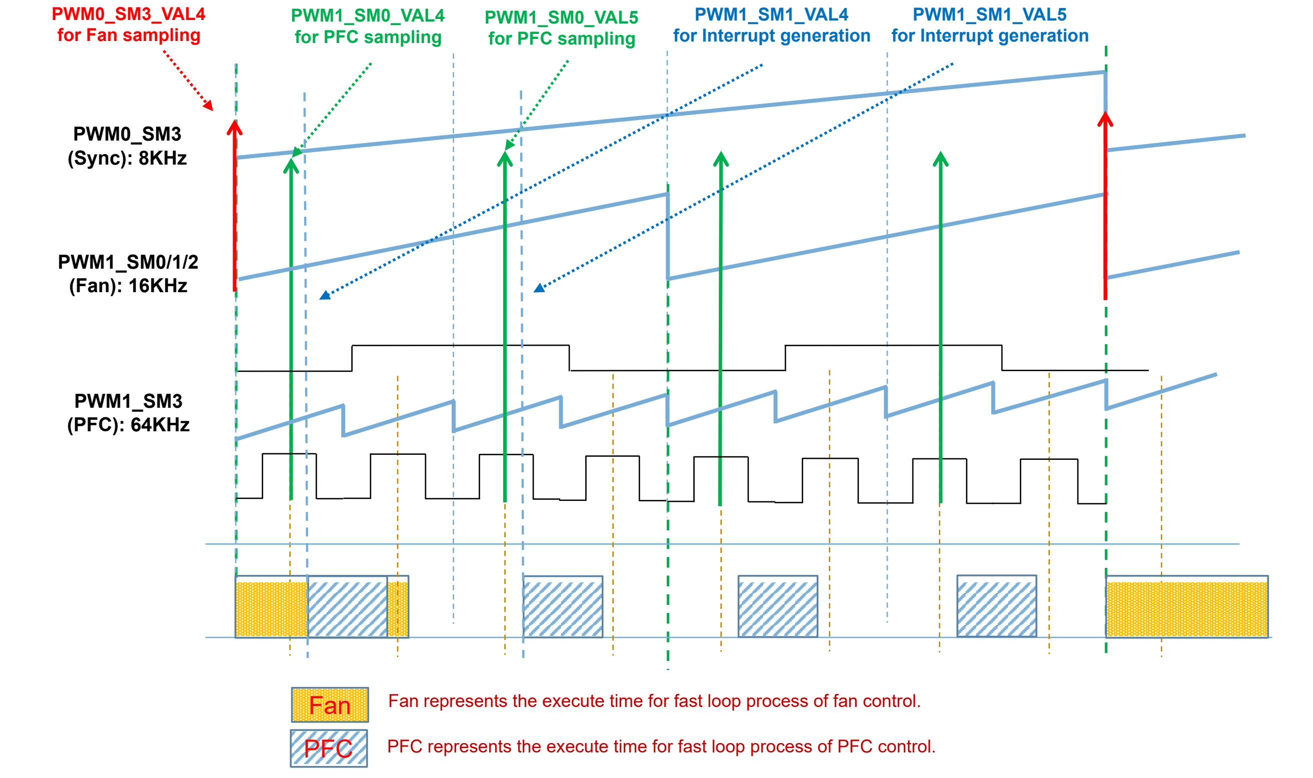
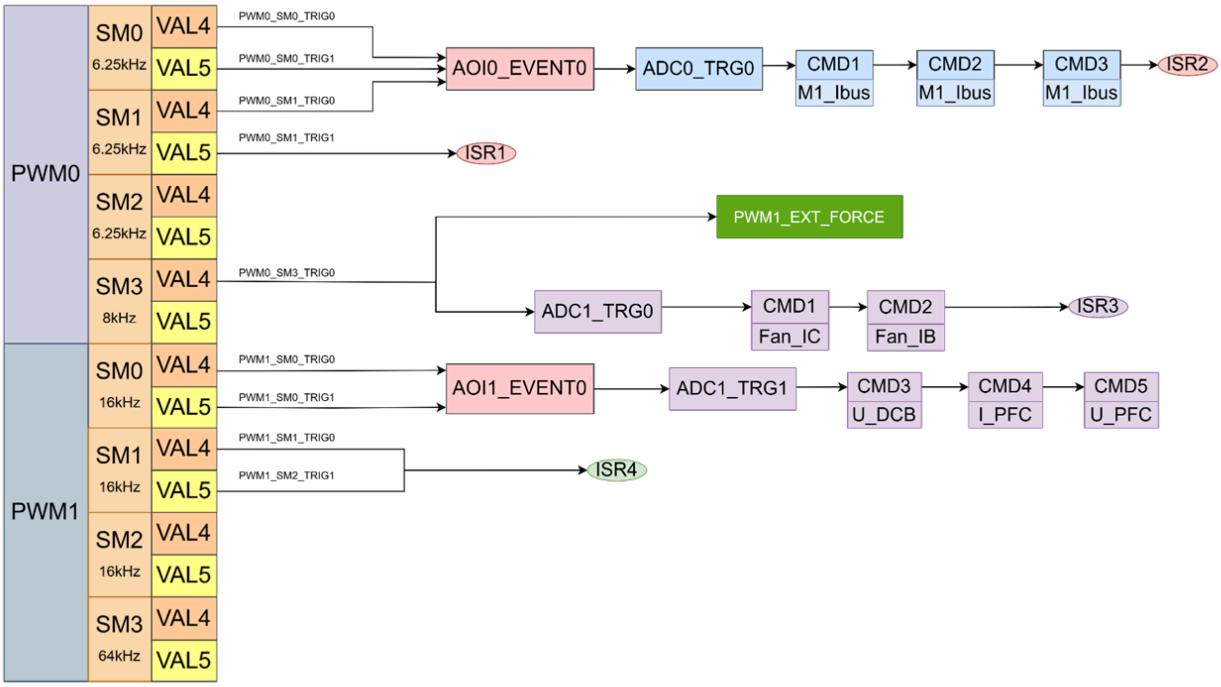


Figure 15. ADC sampling for fan and PFC



ISR1 PWM0_SM1 compare interrupt to enable ADC0/ADC1 trigger for the start point of whole control. Disable itself once executed.

ISR2 ADC0 conversion trigger completion interrupt for compressor control.

ISR3 ADC1 conversion trigger completion interrupt for fan control.

ISR4 PWM1_SM1 compare interrupt for PFC control.

The Fan_IB and Fan_IC are sampled in series, and the Fan_IA is calculated based on these two phase currents.
Due to different trigger sources of the ADC module share the same interrupt vector. In order to control interrupt priority, ISR3/ISR4 are used to control the fan and PFC respectively.

Figure 16. Signal multiplex

CTimer is used to generate a 5-kHz periodic interrupt for the PFC voltage loop and a 1-kHz periodic interrupt for the speed loop control of compressor and fan.

6 Hardware design

This section describes the hardware design.

6.1 Overview of the HVAC 3-in-1 control board

In this application, the HVAC 3-in-1 control board includes the power circuit and the NXP MCX A346 control circuit along with peripheral circuits, capable of controlling the fan, compressor, and PFC simultaneously.
The fan driver uses Silan IPM (U4) with a maximum output current of up to 6 A at a room temperature. The compressor driver uses MITSUBISHI ELECTRIC IPM (U5) with a maximum output current of up to 15 A at a room temperature. The power board is designed with an active power factor correction circuit and the AC power input is designed to meet the 220 V / 50 Hz range of the Chinese power supply. In addition, the power board also includes the following interfaces for system expansion applications: SWD interface for debugging, interface for serial communication, interface for SPI communication, 4-way valve control interface, electronic expansion valve control interface, ADC interface for temperature detection, and so on. Figure 17 shows the picture of the HVAC 3-in-1 control board system.

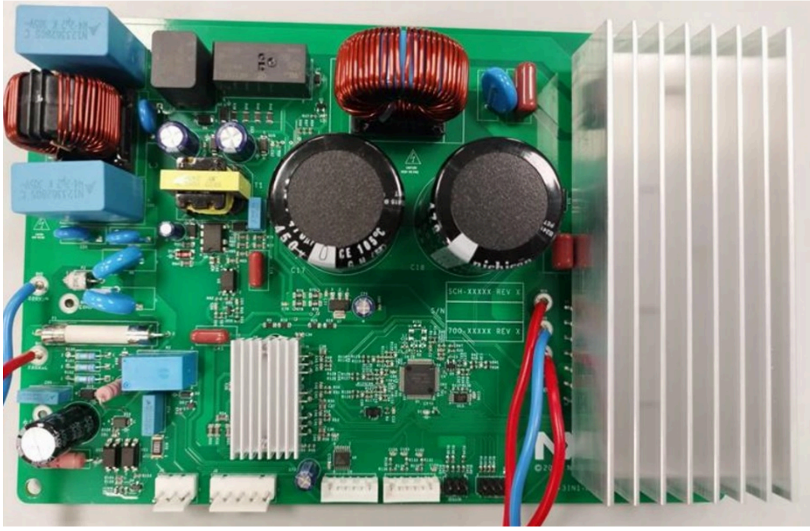


Figure 17. HVAC 3-in-1 control board

6.2 Motor-control board specifications and parameters

- Low-cost and high-performance MCX A346 MCU is single controller to control 2 motors and 1 fan
- Input voltage reference: 220 V AC, 50 Hz
- Active single-phase PFC converter
- Fan and compressor 3-phase motor variable frequency controller, each converter driver includes the following components:
  - IPM with overcurrent and undervoltage protection
  - Motor phase current detection
  - IPM fault signal detection
- DC-Bus voltage detection
- UART serial communication interface
- ADC interface used for temperature detection
- LED used for status indication
- Onboard DC-DC power supply, +15 V, +12 V, +3.3 V

6.3 Hardware description

Figure 18 shows the system block diagram of the HVAC 3-in-1 control board. The MCX A346 functional block in the figure represents the control system which is the core of the entire HVAC 3-in-1 control board, using the NXP MCXA346VLH (64-pin LQFP package) MCU controller.

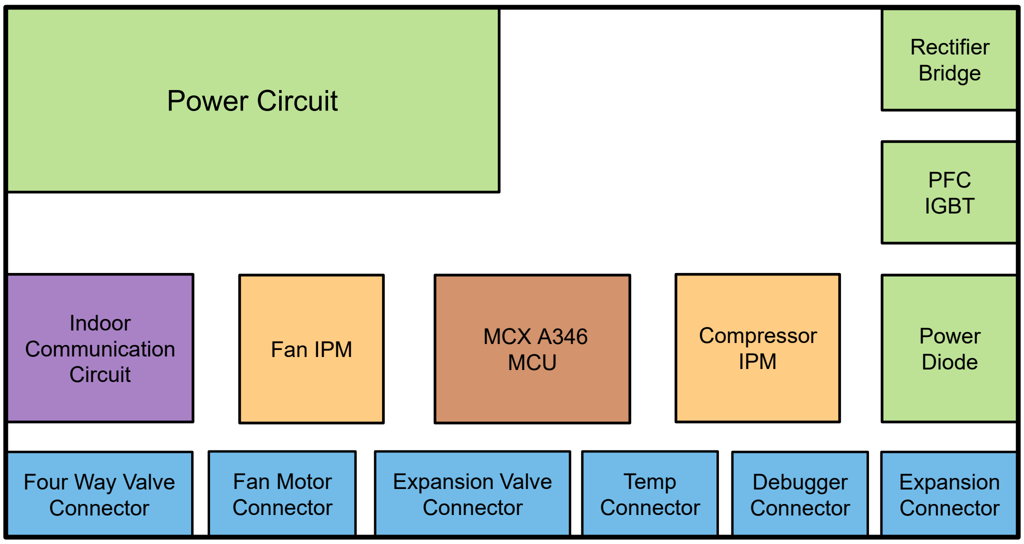


Figure 18. Hardware system block diagram

6.3.1 Input filter circuit

The input filter on the board can reduce the high-frequency noise interference of the PFC power stage, achieving good EMI performance. Figure 19 shows the schematic diagram of the common mode filter.

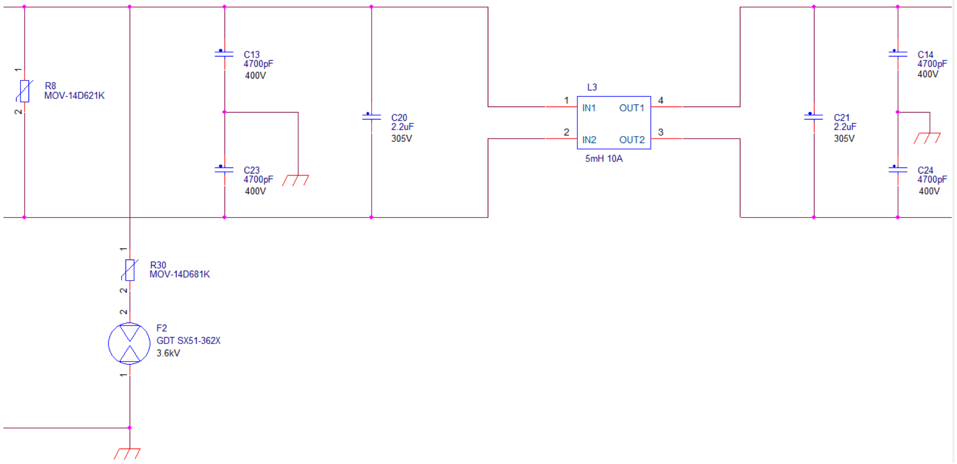


Figure 19. Input filter circuit

6.3.2 Single-phase active PFC (Power Factor Correction) circuit

The PFC circuit can effectively reduce the ripple of voltage and current on the DC side and improve the power factor as well as energy efficiency. Figure 20 is the schematic diagram of the PFC topology.

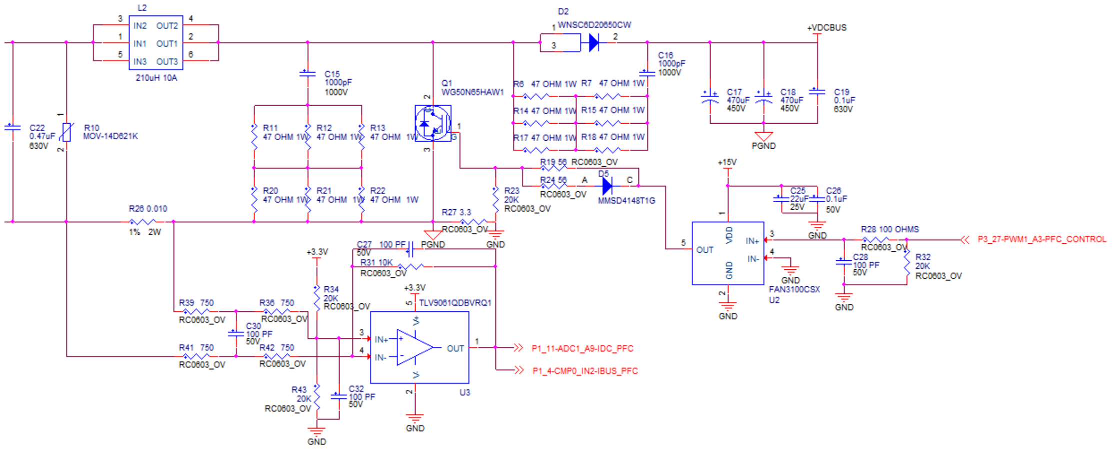


Figure 20. Single-phase active PFC circuit

The active PFC power circuit on the motor-control power board is controlled by the MCX A346. By detecting the AC input rectified voltage, the DC-Bus voltage, and the DC-Bus current, it realizes the MCU control of a constant DC-Bus voltage output. The PWM control signal from the MCX A346 is amplified by the predriver to enhance its drive signal capacity, thereby controlling the IGBT. The driver chip is powered by an onboard +15 V supply. The R26 precision-sampling resistor is used for the PFC current sampling and the voltage drop across the resistor is amplified by a high-precision operational amplifier before being fed into the ADC port of the MCU.

6.3.3 On-board auxiliary power supply

The C17 / C18 DC-Bus capacitors store electrical energy, which is used to power the entire system control board, while also supplying the onboard DC-DC auxiliary power supply, generating +15 V for the IPM / IGBT driver chips, +12 V for relays and the electronic expansion valve drive circuit, and +3.3 V for the digital / analog control circuits. A voltage divider circuit can generate a +1.65 V bias voltage for the motor-current-sensing circuit. Figure 21 shows the schematic of the onboard auxiliary power supply.

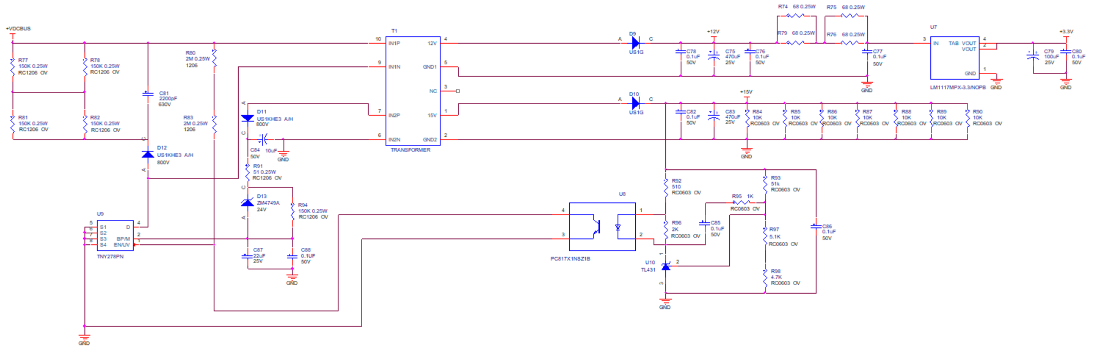


Figure 21. Auxiliary power circuit

6.3.4 3-phase motor driver

The Intelligent Power Module (IPM) is suitable for the direct drive of motors. The IPM internally includes an IGBT inverter bridge array, an IGBT gate driver, 3 bootstrap capacitor charging diodes, under-voltage protection and over-current protection circuits, and so on. Figure 22 shows the fan IPM module and its peripheral control circuit.

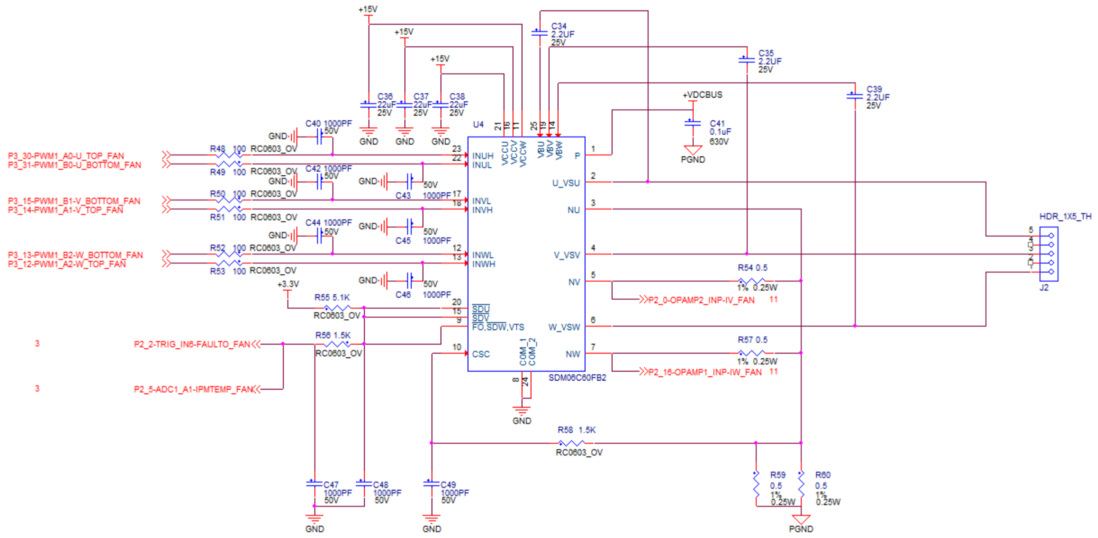


Figure 22. Fan IPM driver circuit

The DC-Bus + VDCBUS provide the energy for motor control to the IPM. +15 V provides a DC power source for the IPM's own operation, powering the internal drivers and the internal analog-digital circuits. The 6 input signals are PWM control signals from the MCX A346, which is based on the +3.3 V standard. The 6 input signals are connected to the gate of the 3-phase IGBT through the internal drivers, and the 6 input signals along with their 6 corresponding IGBTs can be controlled independently. The 3-phase output is connected to the 3-phase terminals of the controlled fan motor. The fan control uses 2-shunt sampling, and the currents of phase V and phase W of the fan are sampled through precision-sampling resistors, as shown in the circuit topology in Figure 22. The voltage drop across the sampling resistors is amplified by an on-chip high-precision operational amplifier, which is connected to the ADC port of the MCU. At the same time, using the internal comparator of the MCU, a corresponding high or low level is output to indicate whether the IPM is overloaded, and this signal is connected to the CSC port of the IPM to control the overload protection of the IPM. When the CSC port level exceeds the typical value of 0.48 V, the internal control circuit of the IPM will turn off all 6 IGBTs and send a fault signal to the MCX A346 through the FO port. After the MCX A346 receives the IPM FO fault signal, it prohibits the output of 6-channel PWM control signals to achieve system protection. The overcurrent protection circuit must ensure the fastest response speed to guarantee the safe operation of the system.

The compressor current sampling used single-shunt sampling method, with the current of the lower bridge Arm bus sampled through a sampling resistor. The 3-phase current of the compressor is obtained through the reconstruction of phase currents. Figure 23 shows the IPM module of the compressor and its peripheral control circuit.

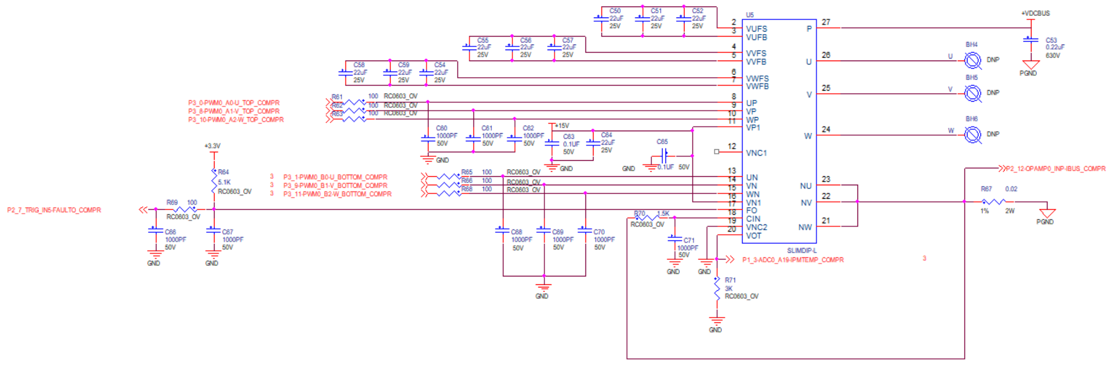


Figure 23. Compressor IPM driver circuit

6.3.5 Analog differential amplifier circuit

The analog amplifier circuit achieves the amplification of the small-signal voltage in the current-sampling circuit. A high-precision operational amplifier, powered by +3.3 V and +1.65 V bias levels, is used to detect bidirectional currents. Figure 24, Figure 25, and Figure 26 show the signal amplification conditioning circuits for the fan phase V, phase W, and the bus current of the compressor lower bridge Arm (respectively) by using MCX A346 on-chip amplifiers.

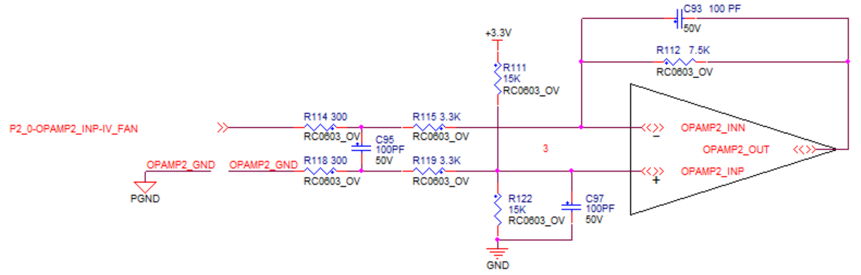


Figure 24. Fan V signal amplification-conditioning circuit

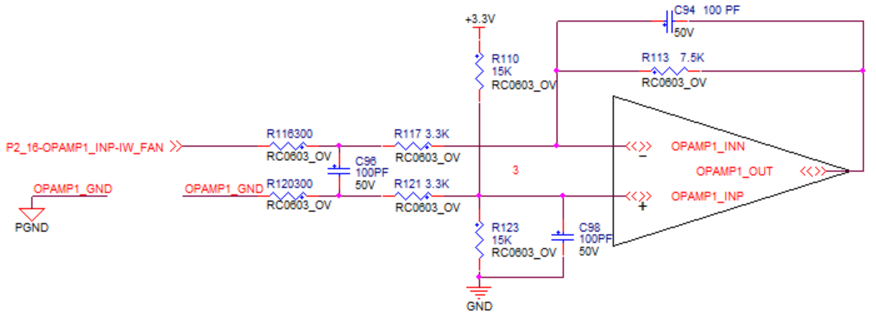


Figure 25. Fan W confidence signal amplification-conditioning circuit

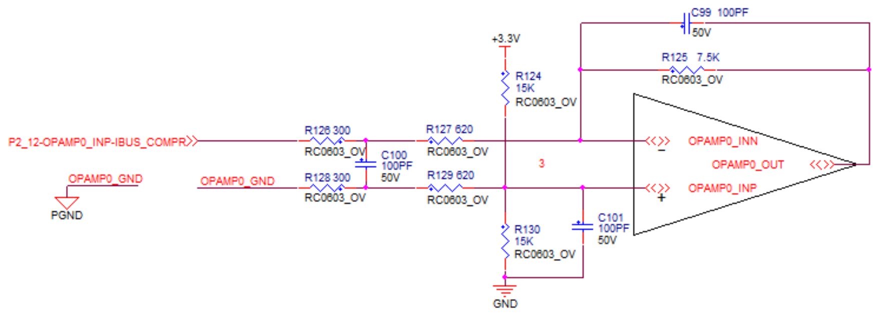


Figure 26. Compressor bus signal amplification-conditioning circuit

Taking phase V of the fan as an example, the R114/R118 input resistors and the C95 capacitor form a high-frequency input filter to eliminate high-frequency noise in the circuit. The values of resistors R111~R112, R114~R115, R118~R119, and R122 determine the amplification factor of the amplifier. The OPAMP is an internal operational amplifier of the MCX A346, whose output connects to the ADC input channel directly via the same MCU pin. The analysis of the current signal conditioning circuit for phase W of the fan and the compressor is consistent with that of phase V of the fan.

6.3.6 UART serial communication and SWD debugging circuit

The MCX A346 has UART controllers for serial communication with external systems. To facilitate communication with a PC, the HVAC 3-in-1 board is designed with serial communication corresponding interfaces J5-3 and J5-5, allowing communication through the connection of an external serial debugging tool. If electrical isolation is required, the UART communication must use an external isolator. Additionally, the board has reserved SWD download debugging ports J5-2 and J5-4, which can be directly connected to an external debugger for program modification and debugging. Figure 27 shows the circuits for serial communication and SWD debugging.

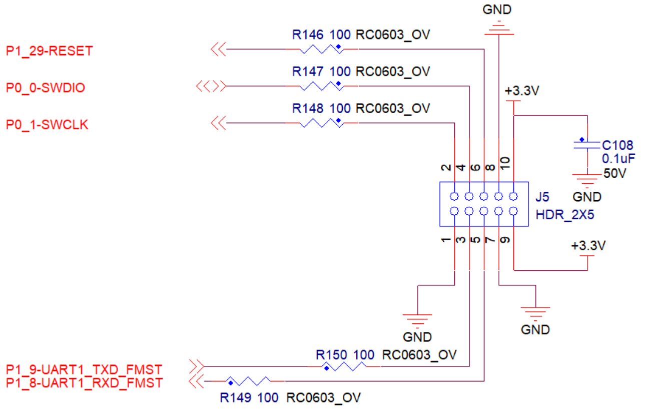


Figure 27. Serial communication and SWD debugging circuit

6.3.7 Indoor unit serial communication circuit

The power board is designed with an interface for serial communication with the indoor unit, using an optocoupler chip for electrical isolation. This serial communication interface reuses the power line for communication without additional wires. Figure 28 shows the communication circuit of the outdoor unit.

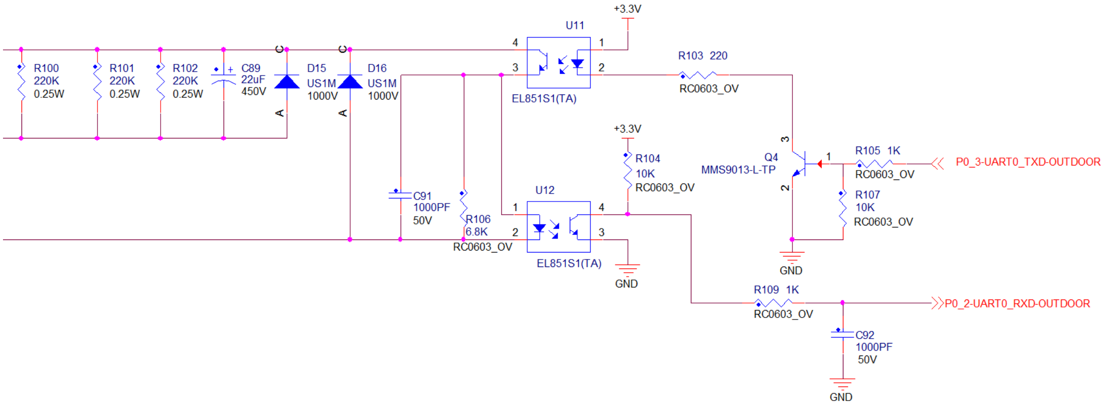


Figure 28. Indoor unit serial communication circuit

7 Software design

This section describes the software design.

7.1 Code architecture

The PMSM and the PFC control are interrupt-driven. There are 4 Interrupt Service Routines (ISRs) in the system for the fast-loop control.

- FLEXPWM1_SUBMODULE1_IRQHandler: It is generated by PWM1_SM1_VAL4 and PWM1_SM1_VAL5 compare match signals with the highest priority, which is 32 kHz. The input AC voltage phase detection and PFC current loop are realized in this ISR.
- ADC1_IRQHandler: It is generated by the ADC1 conversion finished signal with the 2nd priority, which is 8 kHz. The fast-loop control for the fan is realized in this ISR.
- ADC0_IRQHandler: It is generated by the ADC0 conversion finished signal with the 3rd priority, which is 6.25 kHz. The fast-loop control for the compressor is realized in this ISR.
- CTIMER0_IRQHandler: It is generated by the CTimer0 with the lowest priority. The slow-loop control for the compressor, fan, and AC input peak detection are realized in this ISR. A simple state machine is implemented to control the behavior of the DC-Bus voltage controller depending on the loading.

7.2 State machines

There are 2 state machines for motor control (main and substate machine) and the same for the PFC. The 2 motor-control state machines share the same code structures for the main state machine.

7.2.1 Main state machine

There are 4 states in the main state machine, as shown in Figure 29. It first enters the Init state after powering on the MCU and performs the software variable initialization here. After the initialization is completed, it automatically enters the Stop state. After setting the speed command or turning on the switch variable, the software enters the Run state. Similarly, when the speed command (fltSpeedCmd) is set to 0 or the switch variable is turned off, it will return to the Stop state.

At any time during the Run and Stop states, if any fault is detected (such as overcurrent, undervoltage, overspeed, and so on), the fault flag is set and the state machine enters the Fault state. When the fault flag is cleared and a delay duration has passed, it returns to the Stop state again.

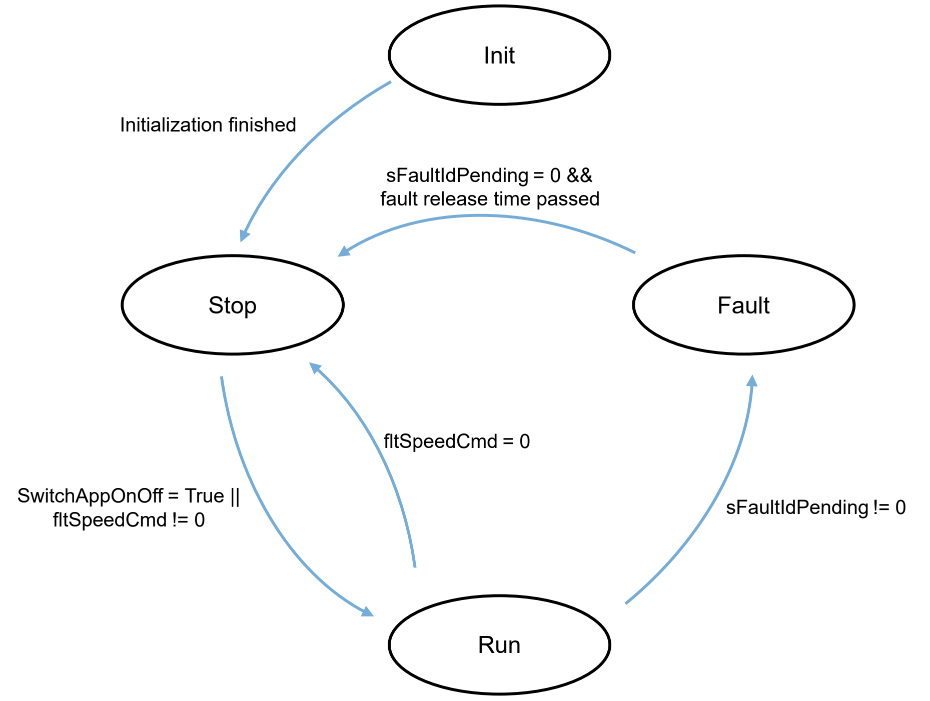


Figure 29. Main state machine

7.2.2 Substate machine of compressor

The substate machine is enabled only in the Run state of the main state machine. The details and switching conditions are shown in Figure 30. The traditional 3-step open-loop starting method is used in the compressor control.

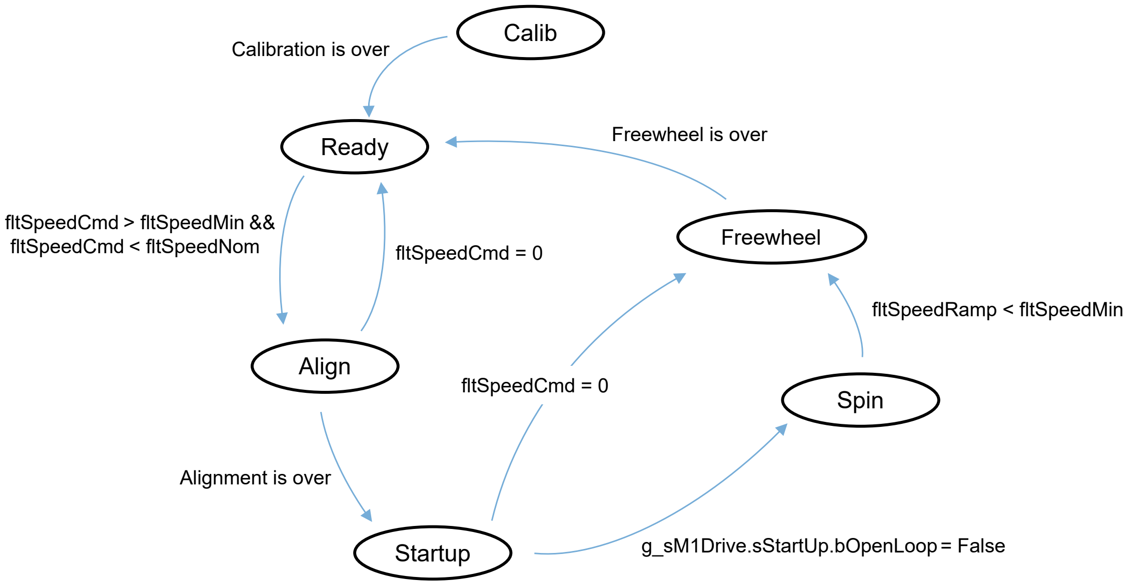


Figure 30. Substate machine of compressor

7.2.3 Substate machine of fan

The substate machine for a fan is different from the compressor. The align state has been removed to realize the close-loop and on-the-fly startup functions, and there are some modifications on other states.

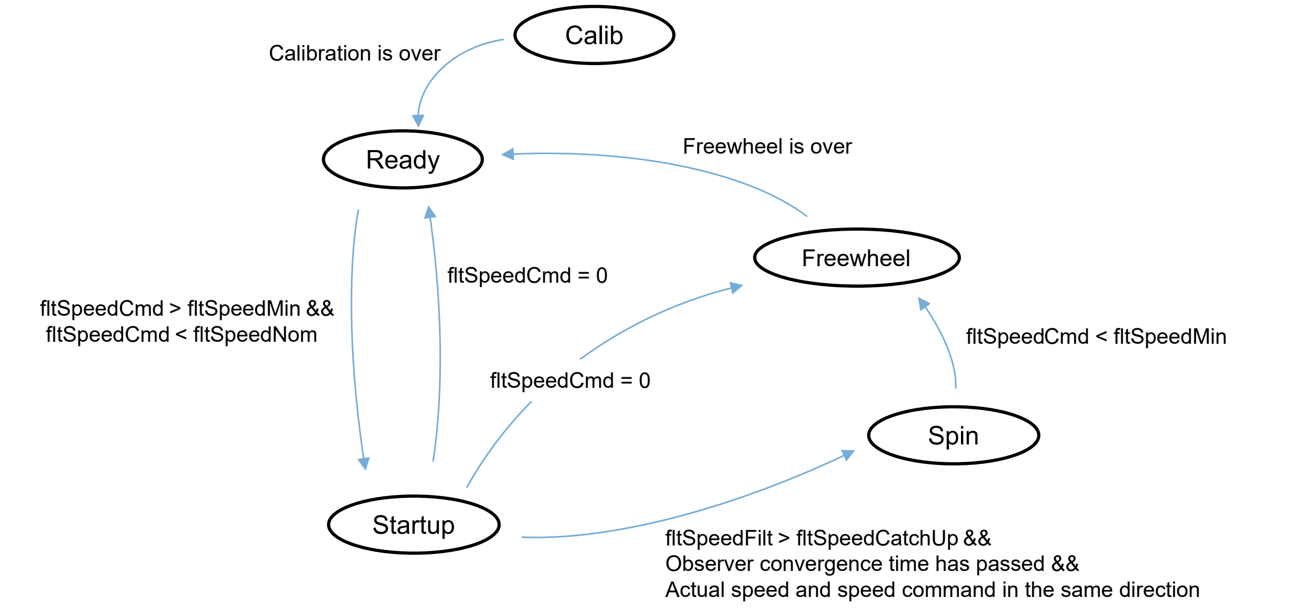

Calib
Calibration is over
Ready
Freewheel is over
Freewheel
fltSpeedCmd = 0
fltSpeedCmd > fltSpeedMin &&
fltSpeedCmd < fltSpeedNom
fltSpeedCmd < fltSpeedMin
fltSpeedCmd = 0
Spin
Startup
fltSpeedFilt > fltSpeedCatchUp &&
Observer convergence time has passed &&
Actual speed and speed command in the same direction

Figure 31. Substate machine of fan

7.2.4 State machine of PFC

The main Finite state machine for the PFC is implemented in FLEXPWM1_SUBMODULE1_IRQHandler with a frequency of 32 kHz. The current loop is realized in the Run state within this ISR, which is called PFC 32-kHz loop in Figure 11. A substate machine for the PFC voltage controller setting is implemented in CTIMER0_IRQHandler with a frequency of 5 kHz. The voltage loop is realized directly within this ISR, which is called PFC 5-kHz loop in Figure 11. The substate machine is enabled only in the Run state of the main state machine.

7.2.4.1 ADC result-ready ISR and PFC main state machine

Within FLEXPWM1_SUBMODULE1_IRQHandler, execute the following tasks in a sequence:

1. Update the 2-phase MOSFET currents and the AC input voltage.
2. Analyze the AC input voltage phase to get the 0-crossing point and the period. The sinθ in Figure 11 is calculated based on this information.
3. Execute the main Finite state machine for the PFC, as shown in Figure 32.

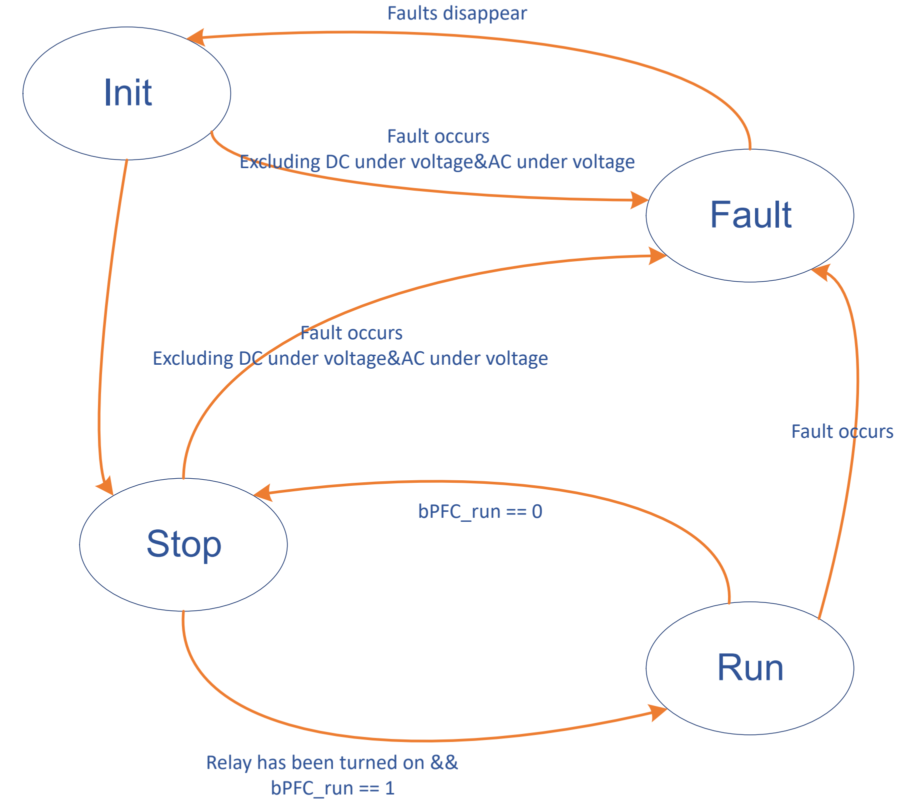

Faults disappear
Init
Fault occurs
Excluding DC under voltage&AC under voltage
Fault
Fault occurs
Excluding DC under voltage&AC under voltage
Fault occurs
Stop
bPFC_run == 0
Run
Relay has been turned on &&
bPFC_run == 1

Figure 32. Main state machine for PFC

Init: This is the first state after the power-on reset. All variables are initialized and the current offsets are calibrated within 200 ms. It goes to the Stop state after the calibration is done.

Stop: The PFC circuit is turned off in this state. It monitors the AC input and DC-Bus voltages to determine if the DC-Bus relay should be turned on or off. After the relay is turned on, it monitors the value of bPFC_RUN to start the PFC control. The variable value can be modified through FreeMASTER.

Run: The state machine goes to this state when the relay is turned on and bPFC_RUN is 1. The current loop is implemented in this state and the output duty is compensated depending on the Discontinuous Current Mode (DCM) or the Continuous Current Mode (CCM).

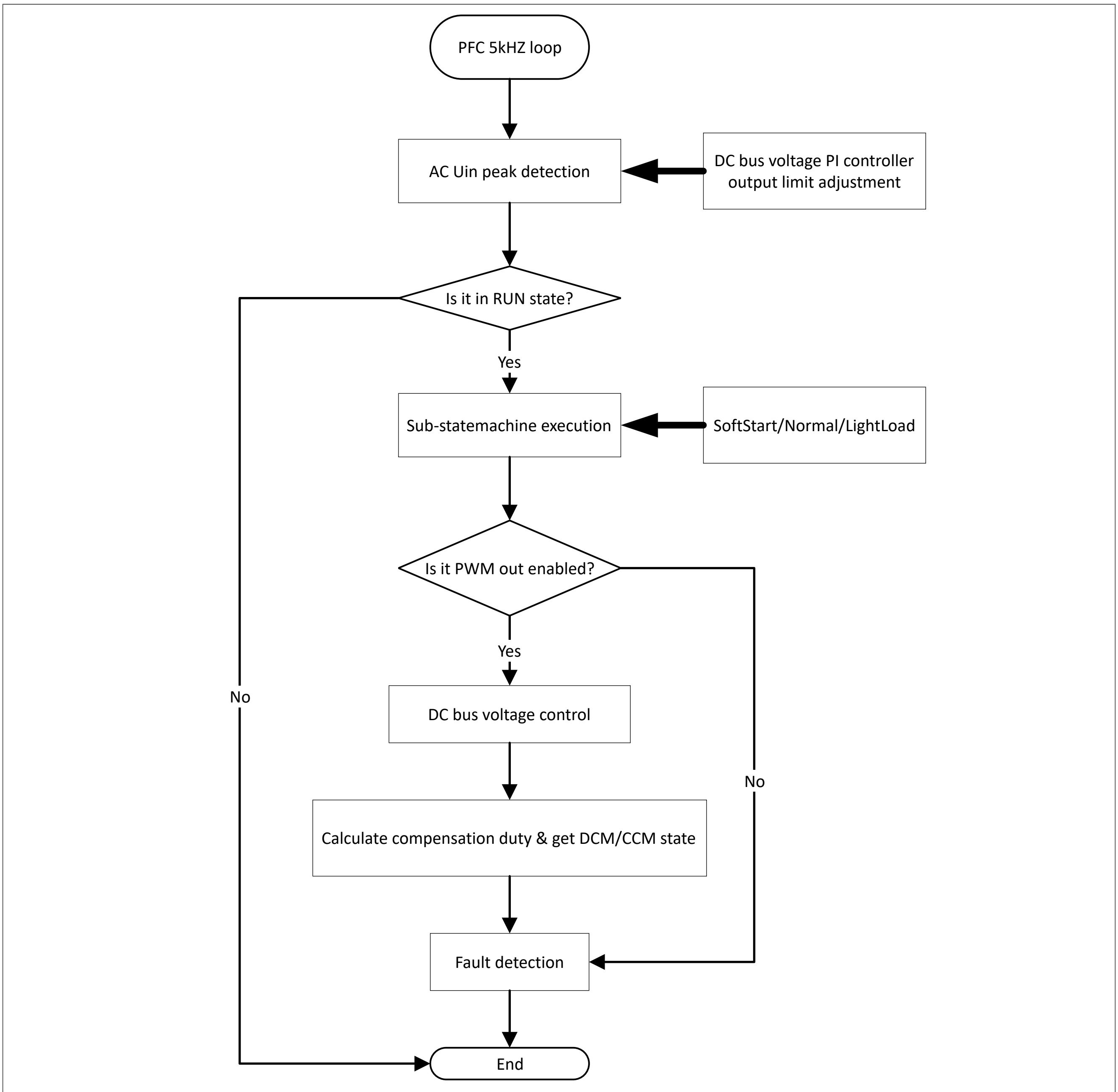

PFC 5kHZ loop
AC Uin peak detection
DC bus voltage PI controller output limit adjustment
Is it in RUN state?
Yes
No
Sub-statemachine execution
SoftStart/Normal/LightLoad
Is it PWM out enabled?
Yes
No
DC bus voltage control
Calculate compensation duty & get DCM/CCM state
Fault detection
End

Figure 33. Slow loop for PFC control

Fault: When there is over- / under-voltage on the DC-Bus or the AC input peak, over-current, or over / under AC input frequency, the state machine goes to this state and the Init state after the faults disappears. The DC-Bus under-voltage and AC input peak under-voltage protection are not enabled when the PFC control is not enabled, as shown in Figure 34.

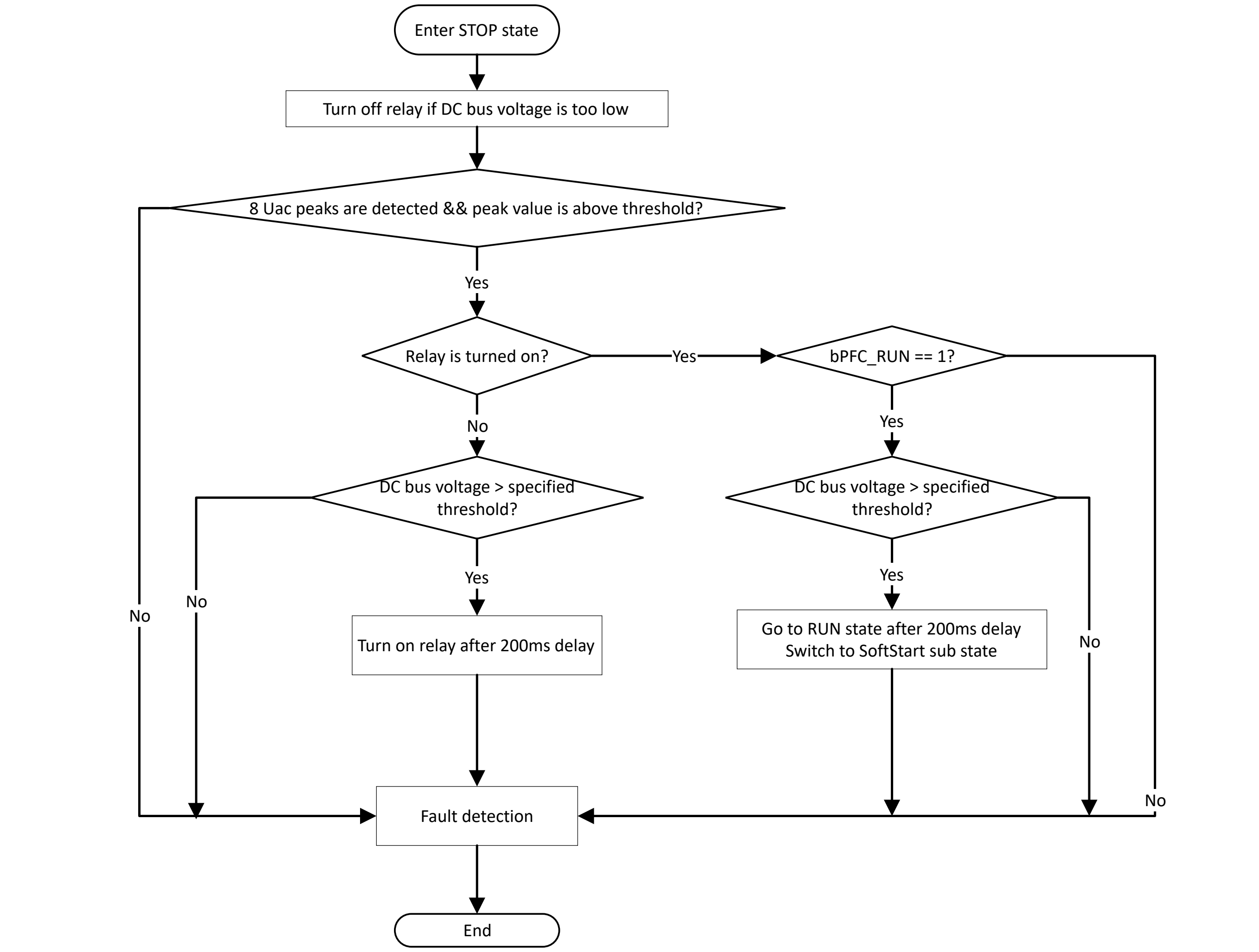

Enter STOP state
Turn off relay if DC bus voltage is too low
8 Uac peaks are detected && peak value is above threshold?
Yes
No
Relay is turned on?
Yes
bPFC_RUN == 1?
No
Yes
DC bus voltage > specified threshold?
No
Yes
Turn on relay after 200ms delay
DC bus voltage > specified threshold?
Yes
Go to RUN state after 200ms delay
Switch to SoftStart sub state
No
No
Fault detection
End

Figure 34. STOP state flowchart for PFC control

7.2.4.2 CTimer0 periodic ISR and PFC substate machine

Within CTIMER0_IRQHandler, execute the following tasks in a sequence:

1. Update the feedback DC-Bus voltage.
2. Detect the AC input voltage peak value.
3. When the main state machine is in the Run state, execute the substate machine, which configures the voltage loop controller settings considering the DC-Bus loading change.

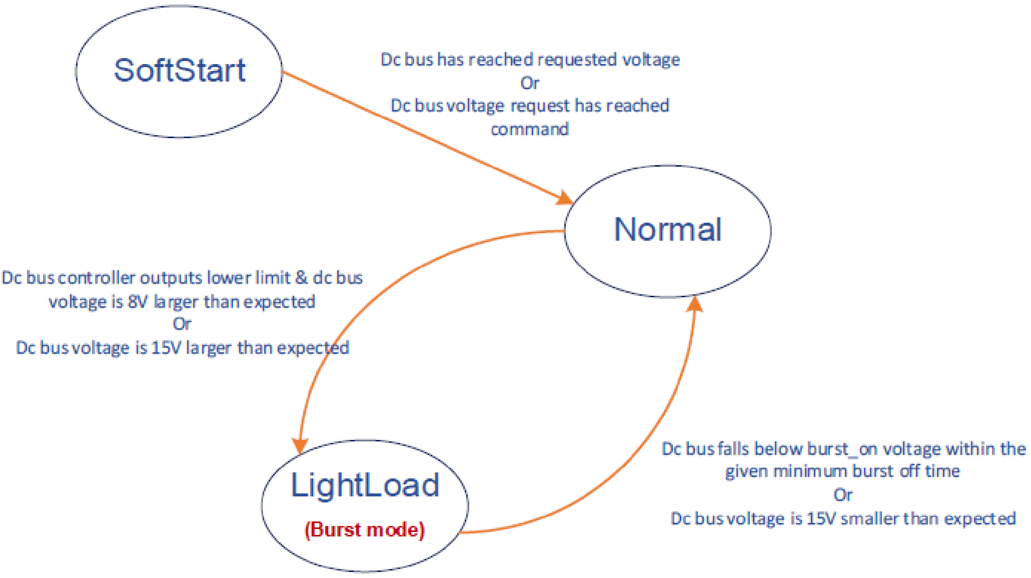

SoftStart
Dc bus has reached requested voltage
Or
Dc bus voltage request has reached command
Normal
Dc bus controller outputs lower limit & dc bus voltage is 8V larger than expected
Or
Dc bus voltage is 15V larger than expected
LightLoad
(Burst mode)
Dc bus falls below burst_on voltage within the given minimum burst off time
Or
Dc bus voltage is 15V smaller than expected

Figure 35. PFC state machine for slow loop

A. SoftStart: The PFC control is just started. The voltage PI controller output lower limit is set to 0, considering that the DC-Bus voltage increases fast through the rectifier bridge when the loading is light. A ramp is implemented here for the DC-Bus voltage request. It goes to the Normal state when the actual DC-Bus voltage has reached the command voltage during the request voltage ramping (loading is light) or the request voltage has reached the command, but the actual DC-Bus voltage has not reached the command yet (loading is heavy). The voltage controller output is used as a current amplitude reference for the current loop.

B. Normal: The DC-Bus loading is relatively high. The voltage PI controller output lower limit is no longer 0 and the voltage controller output is used as a current-amplitude reference for the current loop. When the DC-Bus voltage overshoots a lot or when the voltage controller always outputs the lower limit value, the substate goes to the LightLoad state.

C. LightLoad: The DC-Bus loading is relatively low. This state is also called the Burst mode. The voltage PI controller output lower limit is no longer 0. Because the loading is low, disable the PWM output when the DC-Bus is higher than PFC_U_DCB_BUSRT_OFF, which is called Burst Off, and enable the PWM output when the DC-Bus drops below PFC_U_DCB_BURST_ON, which is called Burst On. During the Burst On stage, only the voltage controller output lower limit is used as a current-amplitude reference. The state goes back to Normal when the loading increases.

7.3 Project file structure

The total number of source (*.c) and header files (*.h) in the project is larger. Therefore, only the key project files are described in detail, and the rest is described in groups.

The main project folder is divided into the following directories:

- \board - initialization and configuration files for the hardware board
- \CMSIS - Cortex MCU software interface standard files
- \device - Software Development Kit
- \drivers - peripheral driver files
- \freemaster - FreeMASTER software support files
- \MCXA346 - compiler necessary files
- \motor_control\freemaster - FreeMASTER GUI files

- \motor_control\pfc - PFC algorithm and state machine files
- \motor_control\pmsm - motor-control algorithm and state machine files
- \rtcesl - Real-Time Control Embedded Software Motor Control and Power Conversion Libraries
- \source - main files and algorithm library files
- \startup - MCU start-up files
- \utilities - files for UART debugging

The files in the folders are as follows:

- pmsm_control.c - the structure definitions and subroutines dedicated for the execution of the motor-control algorithm (FOC control algorithm, startup process, speed control loop)
- m1_sm_snsless.c, m2_sm_snsless.c and PFC_statemachine.c - the substate machine for compressor, fan, and PFC
- m1_pmsm_appconfig.h and m2_pmsm_appconfig.h - the motor-control parameters generated by the FreeMASTER MCAT tools
- mc_periph_init.c - the initialization code for peripherals

7.4 Peripheral usage

Table 1. Peripheral usage

Module	Channels	Used	Purpose
ADC0	24	1+4	Current sampling for compressor and temperature
ADC1	17	5+1	Current sampling for fan and PFC and temperature
FlexPWM0	8	8	PWM output for compressor and sync
FlexPWM1	8	8	PWM output for fan and PFC
LPCMP	3	1	PFC over-current protection
UART	4	2	Communication with Free MASTER and indoor unit
OPAMP	3	3	Current amplifier for fan and compressor
CTimer	3	1	Slow-loop control
AOI	8	2	Current-sampling trigger of fan and compressor

8 Revision history

Table 2. Revision history

Document ID	Release date	Description
AN14708 v.2.0	11 July 2025	• Initial public release
AN14708 v.1.0	8 July 2025	• Initial version

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.

CoreMark — is a registered trademark of SPEC.

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: 11 July 2025
Document identifier: AN14708