

Low Cost AC Motor Control Design Based on MSPM0 and Triac



Zoey Wei, Yahui Zhang

ABSTRACT

This application note implements cost efficiency in AC motor control using a low cost MCU MSPM0C1104 with an 8-pin package and Triac.

The main features of this design are listed below:

- Maximum output power 900W for 220V mains
- Input voltage range: 90-265 VAC, 50/60Hz
- Overcurrent protection
- Support soft start
- Support speed change
- A RUN LED to show the board is powering

Software and hardware schematic resources can be downloaded [here](#).

Table of Contents

1 Introduction	2
2 Block Diagram	3
2.1 Power Supply Module.....	3
2.2 Zero-Crossing Detection (ZCD).....	3
2.3 Current Detection.....	4
2.4 MSPM0 Microcontroller.....	5
2.5 TRIAC.....	5
3 Phase Angle Control	6
3.1 Operating Principle.....	6
3.2 Boundary Conditions and Special Cases.....	6
3.3 Implementation with MSPM0.....	7
4 Soft-Start Feature	8
5 Software	9
5.1 Software Workflow.....	9
5.2 Zero-Crossing Detection Implementation.....	10
5.3 Pulse Generation.....	10
5.4 Update Targeted Speed.....	10
5.5 Overcurrent Protection.....	10
5.6 User System Define.....	10
6 Results	12
6.1 Scope Waveforms.....	12
6.2 Speed Change.....	13
6.3 Soft Start.....	17
6.4 Temperature.....	18
7 Summary	19
8 References	19

Trademarks

All trademarks are the property of their respective owners.

1 Introduction

Modern household appliances including vacuum cleaners, blenders, and power tools increasingly require efficient speed control designs while maintaining simple switching mechanisms. Triac emerges as the most economical power switching components for direct 110/240V mains operation, making them preferred for cost-sensitive motor applications. When integrated with microcontroller units (MCUs), these systems gain intelligent capabilities including status monitoring, dynamic speed regulation, and overcurrent protection.

The TI MSPM0 microcontroller series delivers cost efficiency in this application with rich peripheral integration. This article describes how combining MSPM0 with a Triac enables a low-cost AC motor control.

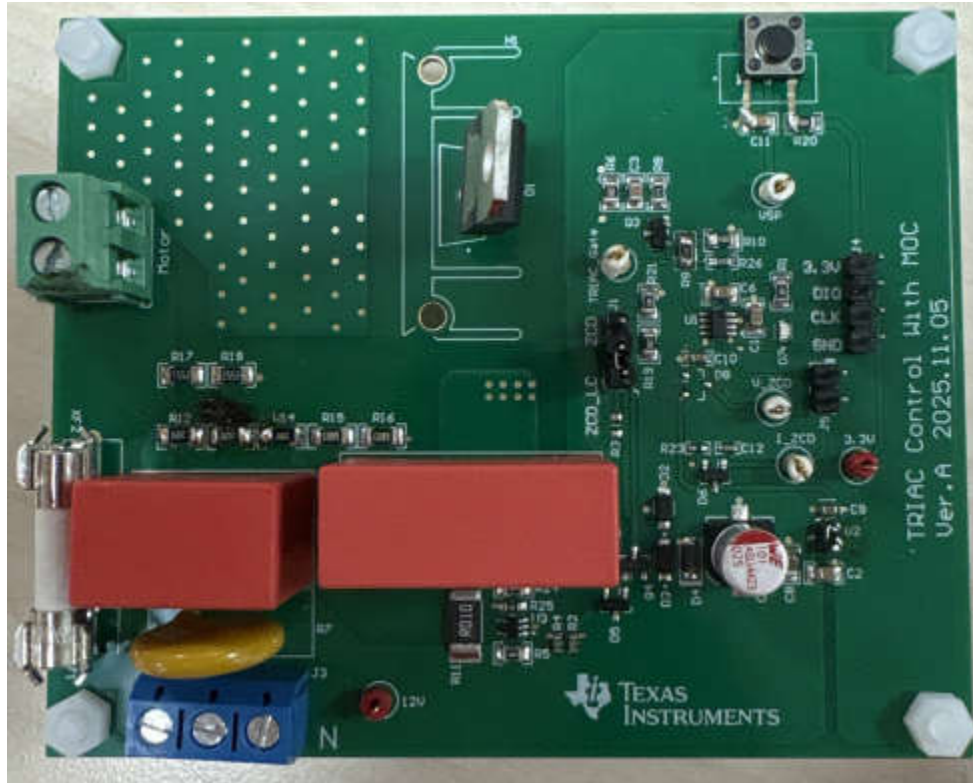


Figure 1-1. Evaluation Board

2 Block Diagram

Figure 2-1 shows six main functions of the system block diagram of the MSPM0-based Triac control design for AC motor applications. AC power is supplied to MSPM0 through a voltage module to generate 3.3V. MSPM0, as the core processing module, detects current and voltage and synchronously outputs a gate-driving signal in real time to drive Triac.

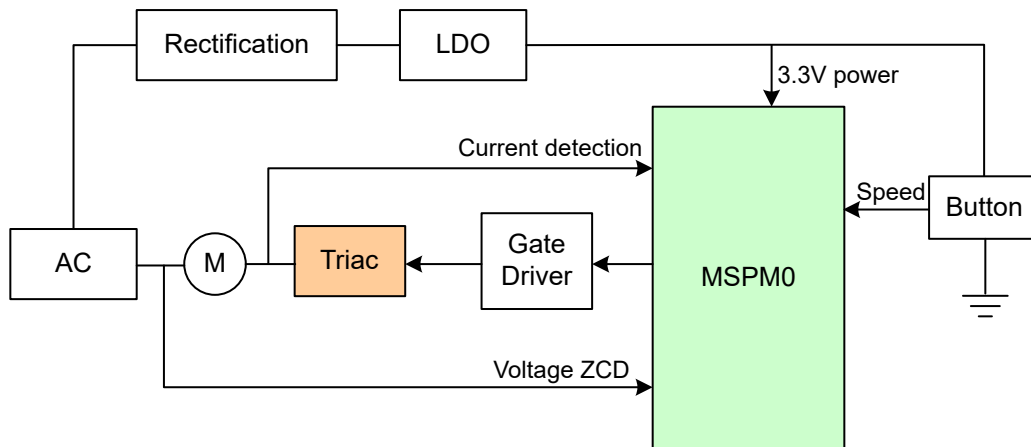


Figure 2-1. System Block Diagram

2.1 Power Supply Module

The AC mains input (90-265 VAC, 50/60 Hz) is converted to a stable 12V DC supply through a rectification circuit followed by a Zener diode. To power the MSPM0 and associated control circuitry, an LDO is used here to change 12V to 3.3V, verifying reliable operation across the wide input voltage range.

The design emphasizes simplicity, cost-effectiveness, and adequate isolation between high-voltage and low-voltage domains.

2.2 Zero-Crossing Detection (ZCD)

The ZCD circuit provides precise synchronization with the AC mains waveform, which is critical for accurate phase angle control. Proper zero-cross detection ensures the microcontroller can determine the firing angle for the TRIAC, enabling precise power control.

This design implements a voltage ZCD. AC current is limited by high value resistor, then connected to the transistor which converts the AC-referenced sinusoidal signal into a square wave synchronized with the mains zero-crossings, providing a clean digital 12V signal to MCU. From MCU side, there are two clamping diodes naturally limit the input signal from 12V to the safe 0-3.3V range for MCU interrupt generation.

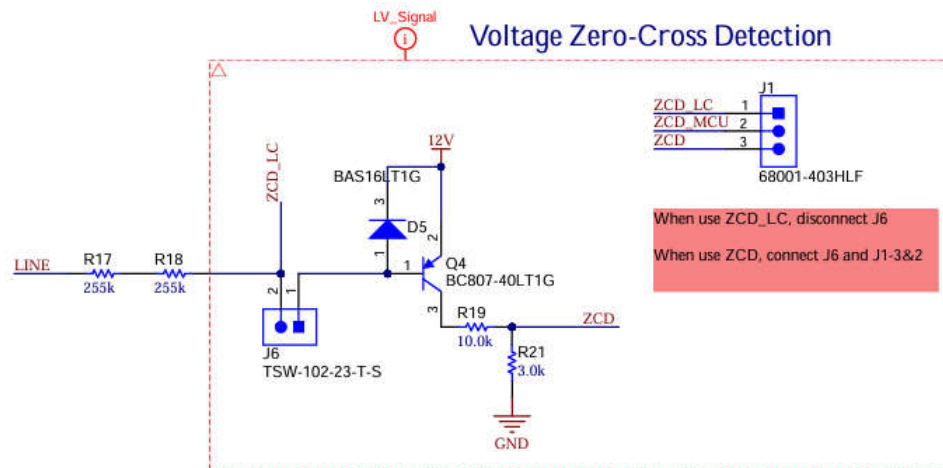


Figure 2-2. Voltage Zero-Cross Detection Schematic

As an alternative, the system supports current-based zero-crossing detection derived from the motor current sensing circuit. Because there is a predictable phase shift when the motor current crosses zero in inductive loads, this method provides reliable synchronization.

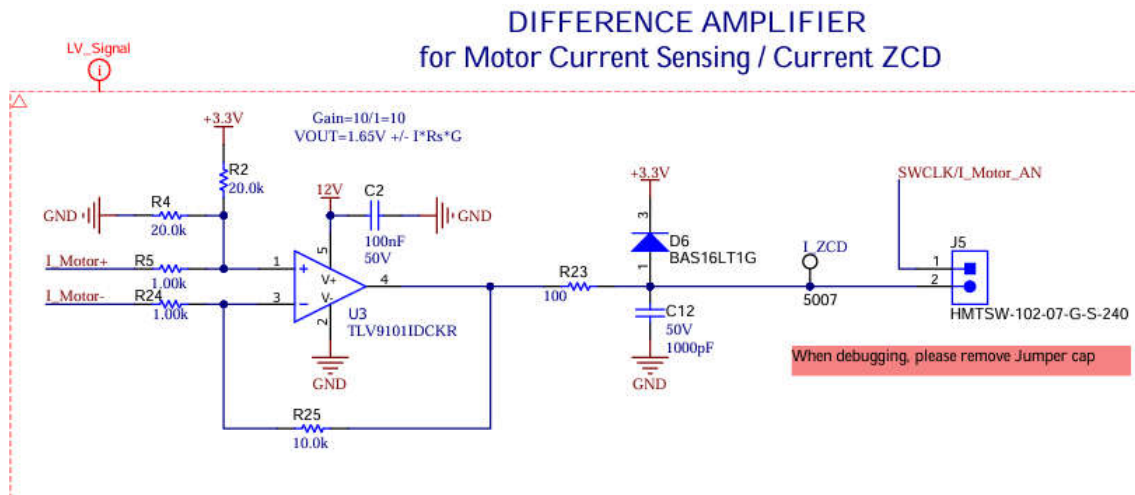


Figure 2-3. Current Zero-Cross Detection Schematic

2.3 Current Detection

In this design, voltage ZCD detection is used. Current ZCD circuitry is reused to monitor the current status in real time to prevent overload.

Using a low-value precision shunt resistor in the motor return path to generate a small differential voltage, which is converted by a differential amplifier voltage to a single-ended, ground-referenced signal for detection. This voltage is designed to range from 1.05V to 2.25V for MSPM0 ADC input.

MSPM0 continuously samples the current signal, comparing measured values against programmable protection thresholds. When current exceeds safe limits, the firmware immediately disables TRIAC triggering to protect the motor and power semiconductors.

2.4 MSPM0 Microcontroller

The MSPM0 microcontroller executes a real-time phase angle control algorithm based on user input, motor current feedback, and zero-crossing timing.

MSPM0 integrates all required peripherals to implement closed-loop motor control with minimal external components. TIMER is used to detect zero-crossing and judge the input voltage frequencies. GPIO is used for button capture to obtain the desired power level, and upon detecting a zero-crossing interrupt and voltage frequencies, the firmware calculates the required TRIAC firing delay based on the user setting. PWM generates precise timing delays from zero-crossing events to the TRIAC gate driver,

The MSPM0C series supports rich peripherals such as ADC, TIMER, GPIO, and so on with at least an eight pins package, which enables sophisticated motor control while maintaining cost efficiency.

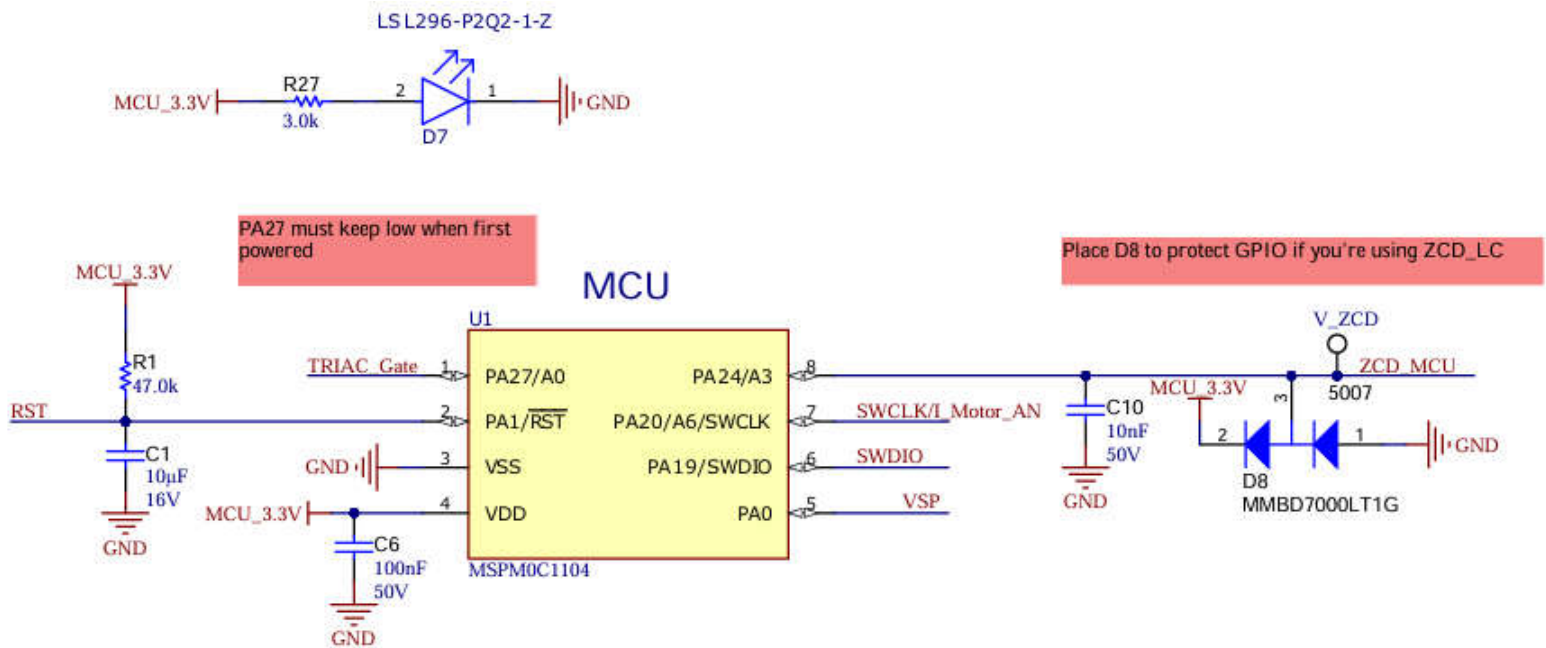


Figure 2-4. MCU Part Schematic

2.5 TRIAC

The TRIAC acts as a bidirectional AC switch connected in series with the universal motor, controlling power delivery through phase angle modulation. Once triggered by a gate pulse, the TRIAC latches into conduction and remains on until the load current naturally falls below the holding threshold at the AC zero-crossing. This self-commutating behavior simplifies the drive circuit, requiring only a brief trigger pulse rather than continuous gate drive.

The delay from zero-crossing to TRIAC firing determines the conduction angle and thus the RMS voltage and power delivered to the motor.

This design supports motor loads up to 900W at 220VAC, making the design preferred for vacuum cleaners, blenders, and similar household appliances.

3 Phase Angle Control

A button input allows users to adjust motor speed by changing the TRIAC firing angle. The phase angle control is the primary technique used to regulate AC motor speed in TRIAC-based applications.

3.1 Operating Principle

The AC line voltage forms a sinusoidal waveform with a period of 20ms at 50Hz (or 16.67ms at 60Hz), consisting of two half-cycles: positive and negative. At each zero-crossing point, the TRIAC is in off state. The MSPM0 microcontroller detects these zero-crossing events and generates a gate trigger pulse after a programmable delay time, known as the firing angle. Once triggered, the TRIAC conducts for the remainder of that half-cycle, allowing current to flow through the motor. The TRIAC automatically turns off when the AC current naturally crosses zero at the end of the half-cycle.

Firing angle determines the power delivered to the motor. The smaller the firing angle, the lower the delivered.

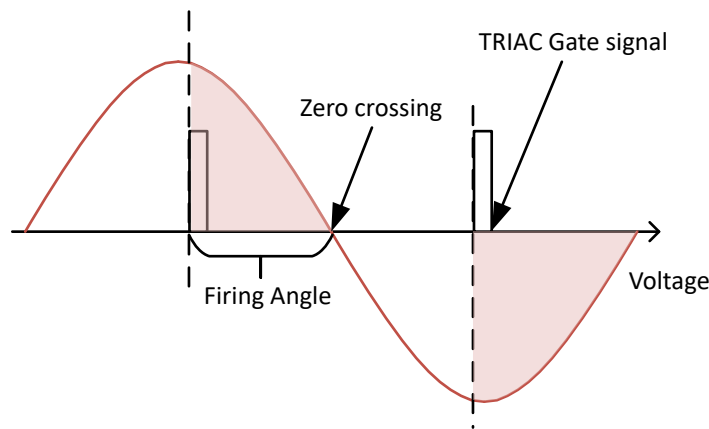


Figure 3-1. Phase Angle Control

3.2 Boundary Conditions and Special Cases

Universal motors used in vacuum cleaners and similar appliances are highly inductive loads. When driving an inductive load, the current lags the voltage by a phase angle determined by the inductance and resistance of the motor. This current lag means the TRIAC does not turn off at the voltage zero-crossing but continues to conduct for additional electrical degrees until the current passes through zero crossing.

If the firing angle is increased, approaching 180°, the TRIAC can be triggered before the current from the previous half-cycle has completely extinguished, as shown in Figure 3-2.

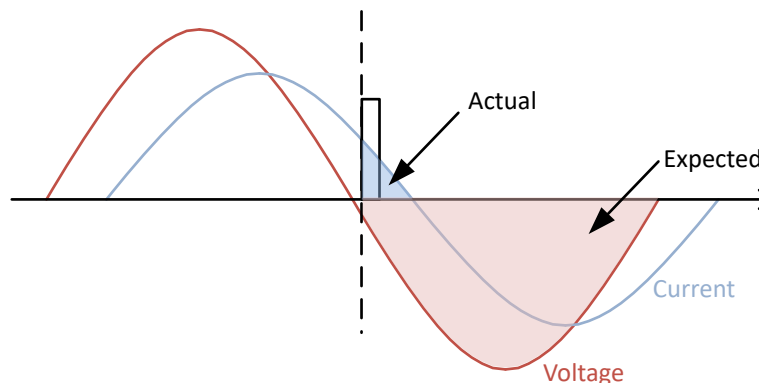


Figure 3-2. Unexpected Switch Off

This potential problem can be solved by two methods. One is extending the TRIAC gate pulse duration as the firing angle approaches. The gate pulse must have a sufficient duration to make sure that the gate pulse

spans the next current zero-crossing point, which verifies that the TRIAC does not switch off at the current zero-crossing point.

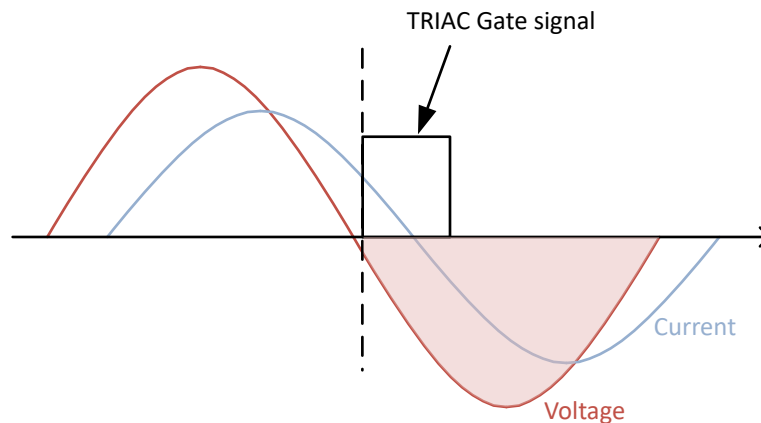


Figure 3-3. Gate Driving Using Long Pulse

Another simpler approach is to set boundaries for the firing angle within the firmware, thereby restricting operation to a specified range. The control scope of this method is not as extensive as that of the first approach. However, firmware development is relatively simpler. For the current design, this method was adopted.

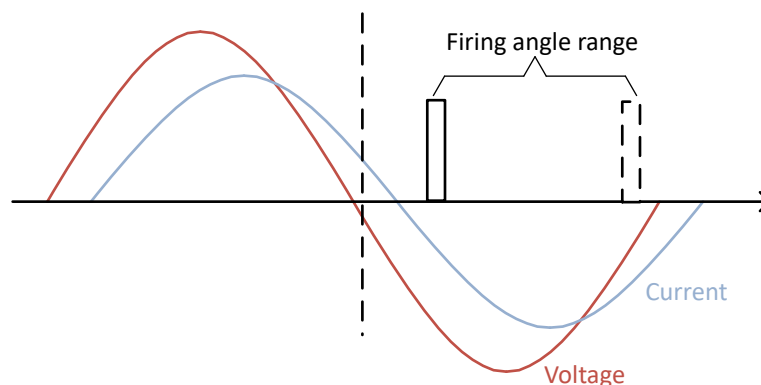


Figure 3-4. Firing Angle Range

3.3 Implementation with MSPM0

This reference design implements three discrete speed levels to provide simple and intuitive motor control:

- **Speed High:** 1000 μ s delay (approximately 162° firing angle) – Maximum power delivery for highest motor speed.
- **Speed Middle:** 5000 μ s delay (approximately 90° firing angle) – Medium power for moderate speed.
- **Speed Low:** 7000 μ s delay (approximately 54° firing angle) – Minimum power for lowest operating speed.
- The boundary setting for the firing angle is 30.6°.

The MSPM0 uses a high-resolution timer to precisely measure the AC period through zero-crossing detection. For each half-cycle, the microcontroller calculates the average period from the most recent two measurements to compensate for line frequency variations. Based on this period measurement and the selected speed mode, the control algorithm determines the delay time before generating a 1ms gate trigger pulse to turn on the TRIAC.

4 Soft-Start Feature

To prevent mechanical stress and reduce inrush current when the motor starts, this design implements a soft-start mechanism. Instead of immediately applying the target firing angle, the control algorithm gradually transitions from a large delay (low power) to the desired delay value. The transition occurs at a controlled rate defined by two parameters: UPDATE_RATE (every 2 zero-crossing cycles) and UPDATE_STEP (50 μ s per update). This creates a smooth acceleration ramp that enhances motor lifetime and reduces acoustic noise during startup.

For example, when transitioning from SpeedOff (10000 μ s) to SpeedHigh (1000 μ s), the delay decreases by 50 μ s every two cycles, resulting in a total transition time of approximately 3.6 seconds at 50Hz line frequency. This gradual power-up significantly reduces mechanical shock and improves system reliability.

5 Software

5.1 Software Workflow

The software consists of several main functions:

Table 5-1. Main API in firmware

API	Description
Triac_Init	Configures all peripherals and initializes control parameters
Triac_ControlLoop	Executes to perform all necessary calculations and trigger the Triac at the appropriate moment
Triac_updateCurDelay	Updates the firing angel
Triac_generatePulse	Generates gate driver pulse
CAPTURE_0_INST_IRQHandler	Timer interrupt handler Realizes the zero-cross detection
GPIO_IRQHandler	GPIO interrupt handler Detects button and change speed
ADC12_0_INST_IRQHandler	ADC interrupt handler Monitors current

Figure 5-1 is the software workflow.

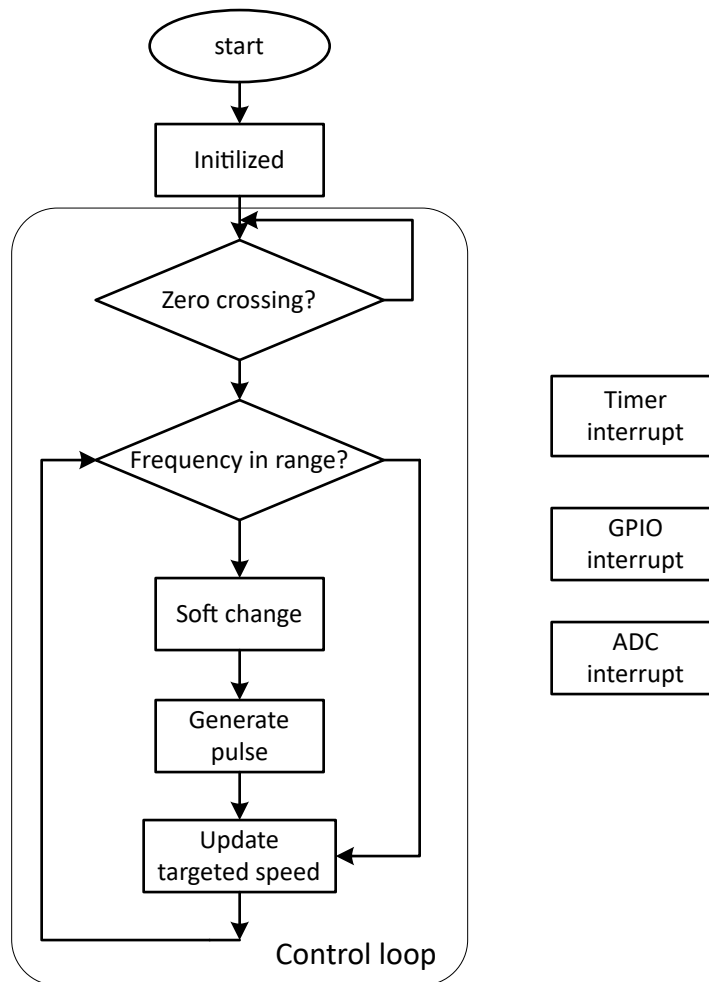


Figure 5-1. Software Workflow

5.2 Zero-Crossing Detection Implementation

The zero-crossing detection uses a timer operating in capture mode, configured to trigger on both rising and falling edges of the ZCD input signal. The interrupt handler (CAPTURE_0_INST_IRQHandler) processes three distinct events:

- **CC0_DN (First Half-Cycle):** Captures the timer value at the positive-to-negative zero crossing and calculates the first half-period duration
- **CC1_DN (Second Half-Cycle):** Captures the timer value at the negative-to-positive zero crossing, calculates the second half-period, and manually reloads the timer counter to maintain synchronization
- **ZERO Event:** If the timer reaches zero without detecting a zero-crossing, the system sets `gSynced = false` to indicate loss of synchronization, preventing erratic gate pulse generation

The captured period values are averaged (`periodAvg`) to provide a robust measurement that compensates for minor variations in line frequency and noise. This averaged value serves as the time base for calculating the precise firing angle delay.

5.3 Pulse Generation

The `Triac_generatePulse()` function uses a one-shot timer configured in PWM mode to create the gate trigger pulse. The timer is loaded with a value equal to `currentDelay + PULSE_WIDTH` (1ms), where `currentDelay` determines when the pulse starts and `PULSE_WIDTH` defines the pulse duration. The timer automatically stops after generating the pulse, verifying energy-efficient operation.

The pulse generation is conditional based on `pulseType`: when `currentDelay` exceeds `DELAY_MIN_POWER_TIME` (8300µs), `pulseType` is set to `noPulse` and no gate signal is generated, effectively turning off the motor. This threshold prevents operating the TRIAC at excessively small conduction angles where the motor can stall or draw excessive current.

5.4 Update Targeted Speed

The `GPIOA_IRQHandler` responds to button press events by incrementing the `speedMode` variable through four states: `SpeedOff`, `SpeedLow`, `SpeedMiddle`, and `SpeedHigh`. Each speed mode maps to a specific `targetDelay` value that determines the firing angle. The cyclic nature of this interface (wrapping from `SpeedHigh` back to `SpeedOff`) provides intuitive single-button control for users.

The soft-start mechanism in `Triac_updateCurDelay()` smoothly transitions `currentDelay` toward the new `targetDelay` by incrementing or decrementing at a controlled rate (`UPDATE_STEP` every `UPDATE_RATE` cycles). This prevents abrupt speed changes that can cause mechanical stress or acoustic noise.

5.5 Overcurrent Protection

The `ADC12_0_INST_IRQHandler` implements a hardware-based overcurrent protection mechanism using the MSPM0's integrated window comparator. During initialization, the ADC is configured with a high threshold corresponding to the maximum safe current level (`AC_CURRENT_LIMITATION = 3.2V` from the current sense circuit). When the sensed current exceeds this threshold, the window comparator generates an interrupt.

The ISR responds by immediately disabling all interrupts (`__disable_irq()`) to halt normal operation, clearing the period measurements to prevent further gate pulses, and triggering a breakpoint (`__BKPT(0)`) for debugging. In a production system, this breakpoint is replaced with a more sophisticated fault handling routine that can include status indication, automatic retry logic, or permanent shutdown depending on application requirements.

5.6 User System Define

The system configuration parameters are defined in the `system_define.h` header file. These macros allow users to customize the phase angle control behavior, timing parameters, and protection thresholds to match specific application requirements and AC line frequencies. All timing values are specified in microseconds (µs) unless otherwise noted.

The default timing parameters are designed for 50Hz AC mains (20ms period, 10ms half-cycle). For 60Hz AC mains operation (16.67ms period, 8.33ms half-cycle), the timing parameters must be adjusted proportionally to maintain equivalent phase angles and control characteristics.

Table 5-2. User Defined Macro list

Macro Name	Default Value	Description
PULSE_WIDTH	1000 μ s	Triac gate trigger pulse width. Defines the duration of the gate drive signal applied to the Triac. The pulse must be long enough to verify the load current reaches the Triac latching threshold, particularly important for inductive loads. A 1ms pulse provides reliable triggering across most operating conditions.
VOLTAGE_PERIOD_MAX	12000 μ s	Maximum acceptable AC half-cycle period. Serves as an upper boundary for period validation to detect abnormal line frequency conditions. For 50Hz operation, the nominal half-cycle is 10ms; the 12ms maximum allows 20% tolerance.
VOLTAGE_PERIOD_MIN	8000 μ s	Minimum acceptable AC half-cycle period. Serves as a lower boundary for period validation to reject spurious zero-crossing detections or noise-induced false triggers. The 8ms minimum provides 20% tolerance below the nominal 10ms half-cycle at 50Hz.
AC_INPUT_HALF_PERIOD	10000 μ s	Nominal AC input half-cycle period. At 50Hz, one complete cycle is 20ms, so each half-cycle is 10ms.
DELAY_SPEED_HIGH	1000 μ s	Firing delay for high-speed operation.
DELAY_SPEED_MIDDLE	5000 μ s	Firing delay for medium-speed operation.
DELAY_SPEED_LOW	7000 μ s	Firing delay for low-speed operation.
DELAY_MIN_POWER_TIME	8300 μ s	Minimum power threshold delay.
UPDATE_RATE	2 cycles	Delay update rate in zero-crossing cycles. Defines how frequently the currentDelay parameter is adjusted toward the targetDelay during speed transitions.
UPDATE_STEP	50 μ s	Delay adjustment increment per update. Each time UPDATE_RATE cycles have elapsed, the currentDelay is incremented or decremented by this step size toward the targetDelay.
AC_CURRENT_LIMITATION	3.2 V	Overcurrent protection threshold voltage.

6 Results

This section presents experimental results obtained from the MSPM0-based TRIAC control system operating with a universal motor load. The measurements validate the phase angle control implementation, demonstrate the soft-start functionality, and characterize the thermal performance under various operating conditions.

6.1 Scope Waveforms

The oscilloscope traces show the relationship between the zero-crossing detection signal (blue), TRIAC gate drive pulse (pink), and motor current (green), as shown in Figure 6-1. The gate pulse occurs after the programmed delay from the ZCD rising edge, and the motor current commences conduction following the gate pulse, and motor is switched off when next zero crossing point comes.

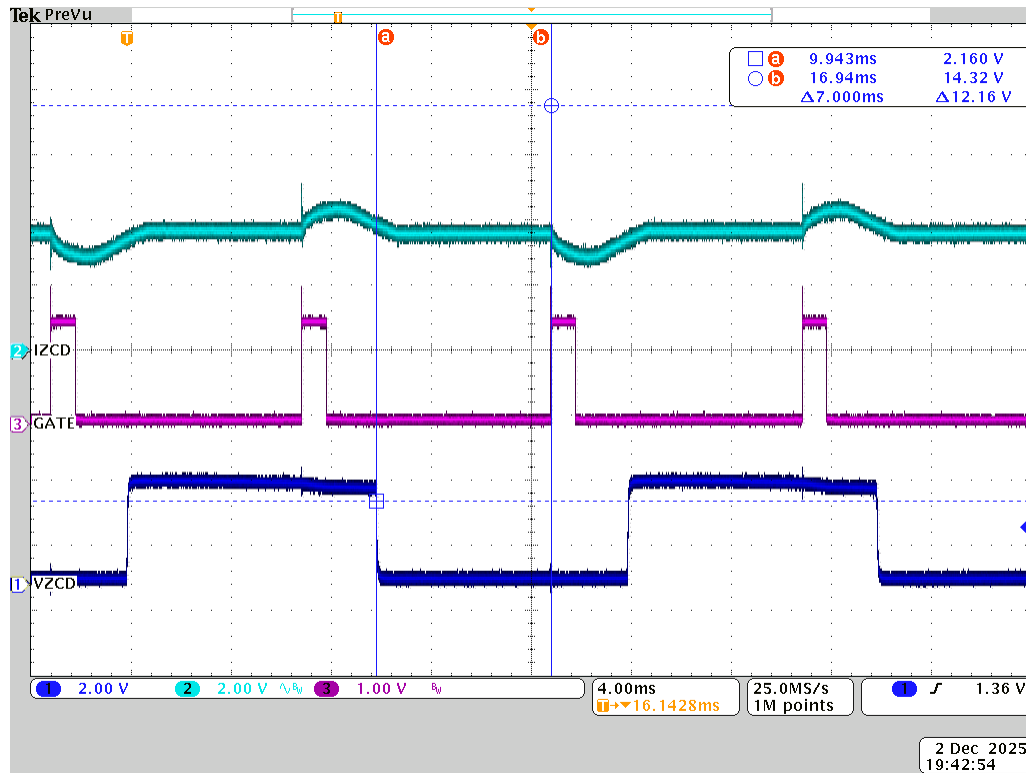


Figure 6-1. Signal at High Speed Mode (Blue - ZCD signal, Pink - Gate Driver Signal, Green - Motor Current)

6.2 Speed Change

Figure 6-2 through Figure 6-7 shows the gate signal, ZCD signal and motor current in different speed modes. As can be seen from the following figures, in low-speed mode, the phase angle is small, and the current sustaining time is short; conversely, in high-speed mode, the phase angle is large, and the current sustaining time is long.

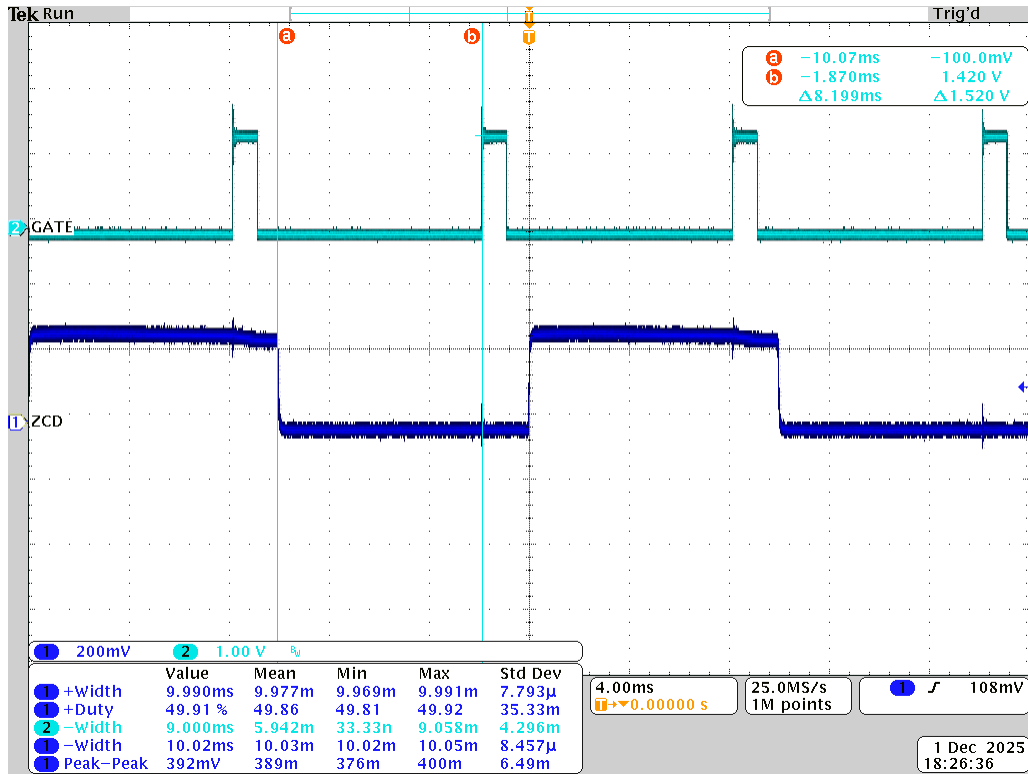


Figure 6-2. Signal in Speed Low Mode (Blue - ZCD signal, Green - Gate Driver)

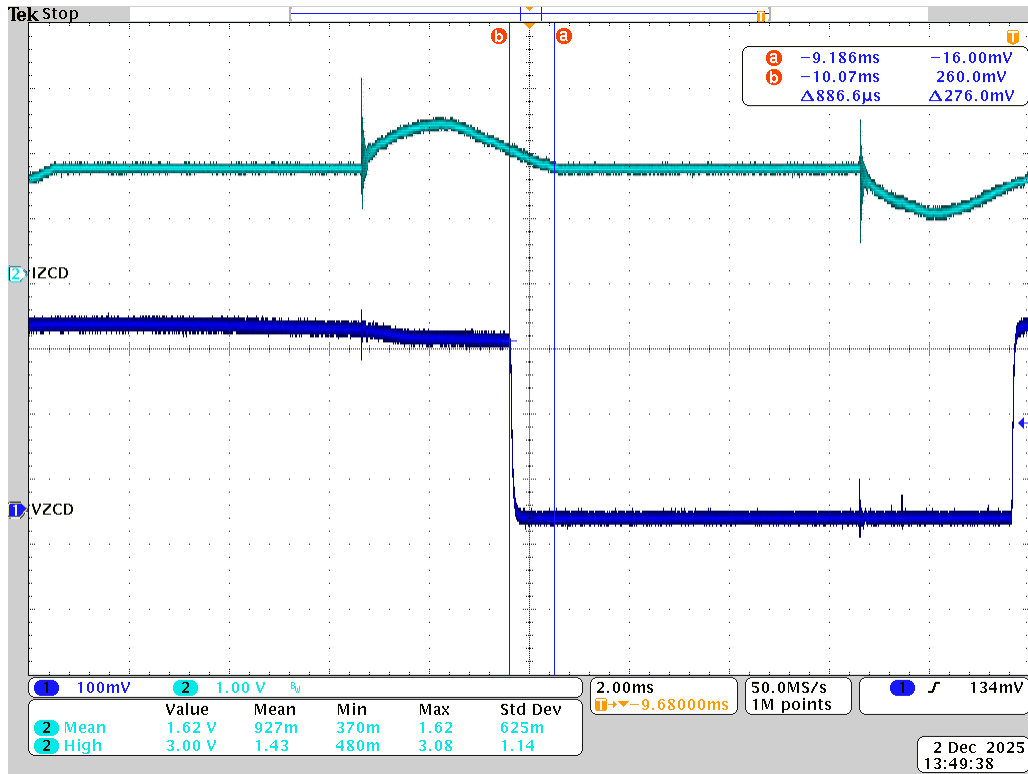


Figure 6-3. Signal in Speed Low Mode (Blue - ZCD signal, Green - Motor Current)

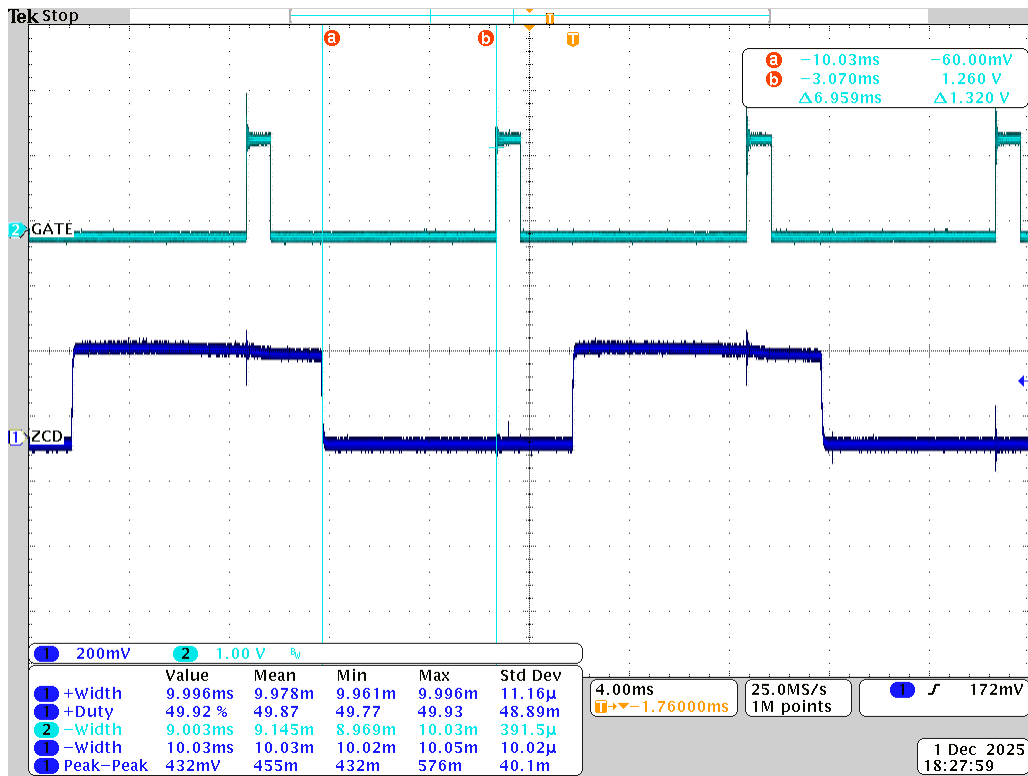


Figure 6-4. Signal in Speed Medium Mode (Blue - ZCD signal, Green - Gate Driver)

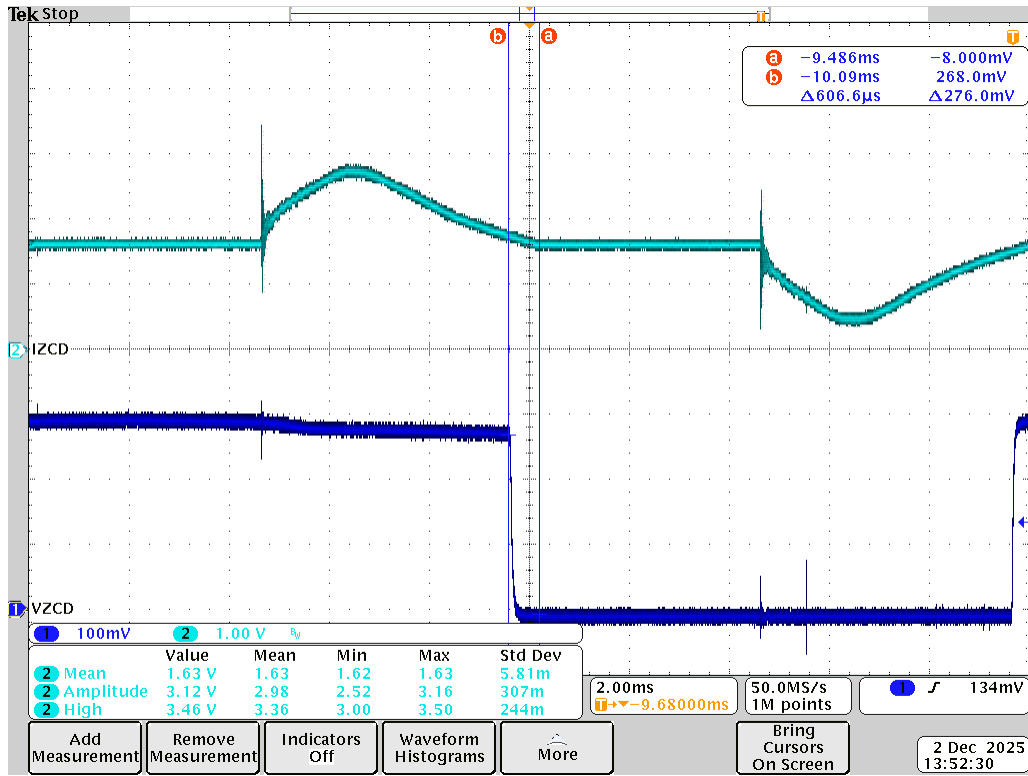


Figure 6-5. Signal in Speed Medium Mode (Blue - ZCD signal, Green - Motor Current)

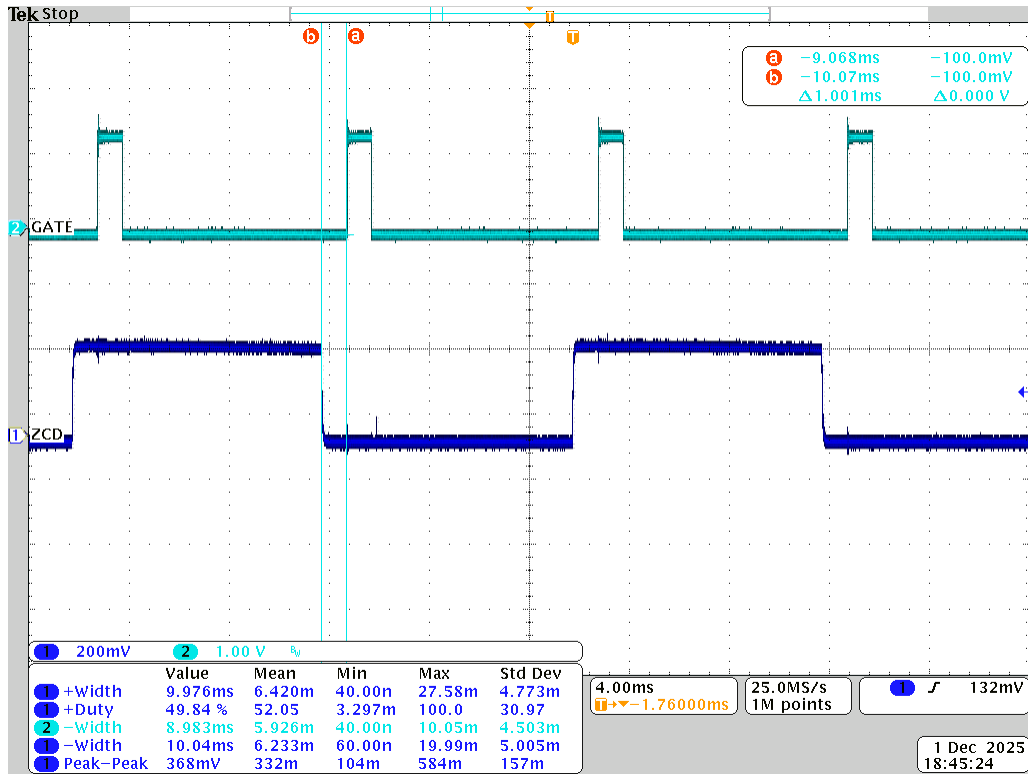


Figure 6-6. Signal in Speed Medium Mode (Blue - ZCD signal, Green - Gate Driver)

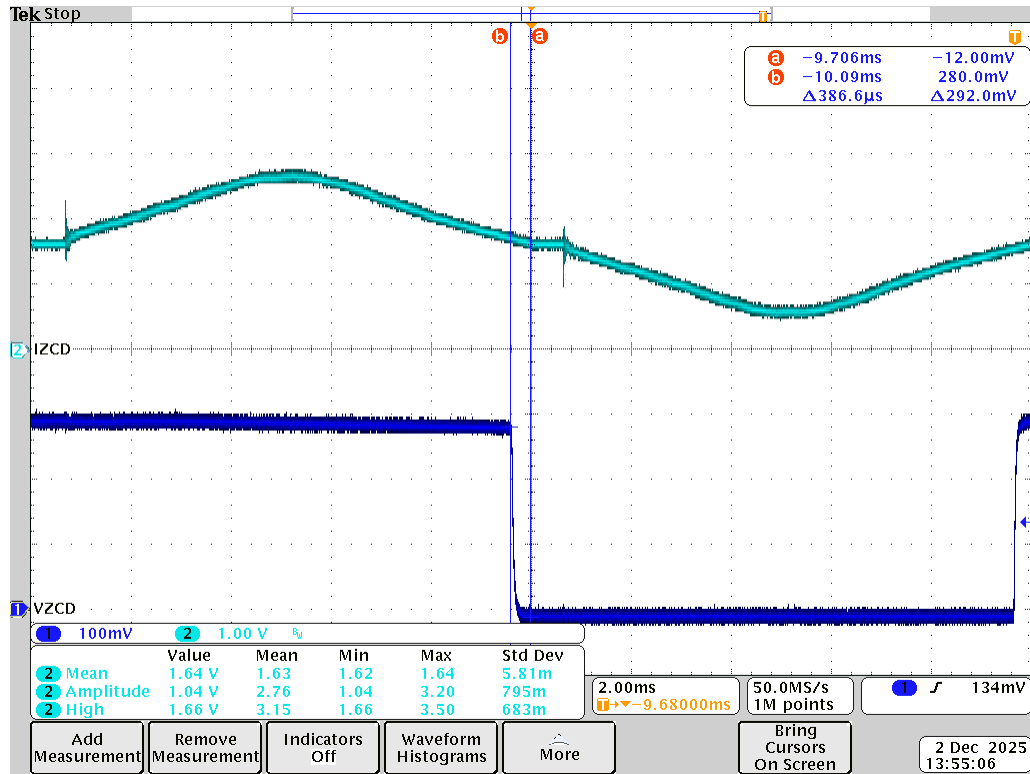


Figure 6-7. Signal in Speed Medium Mode (Blue - ZCD Signal, Green - Motor Current)

6.3 Soft Start

The firing angle progresses smoothly through intermediate values rather than jumping abruptly between the target speeds. This gradual transition is achieved by the UPDATE_RATE and UPDATE_STEP parameters, which incrementally adjust the firing delay every two AC cycles by 50µs. The smooth ramping prevents mechanical shock to the motor and driven load, reduces audible clicking or thumping noises, and minimizes current transients that can otherwise trip overcurrent protection or cause voltage flicker.

Figure 6-8 shows the output current behavior when changing speed from off to low mode. The motor current is increased smoothly as the motor accelerates.

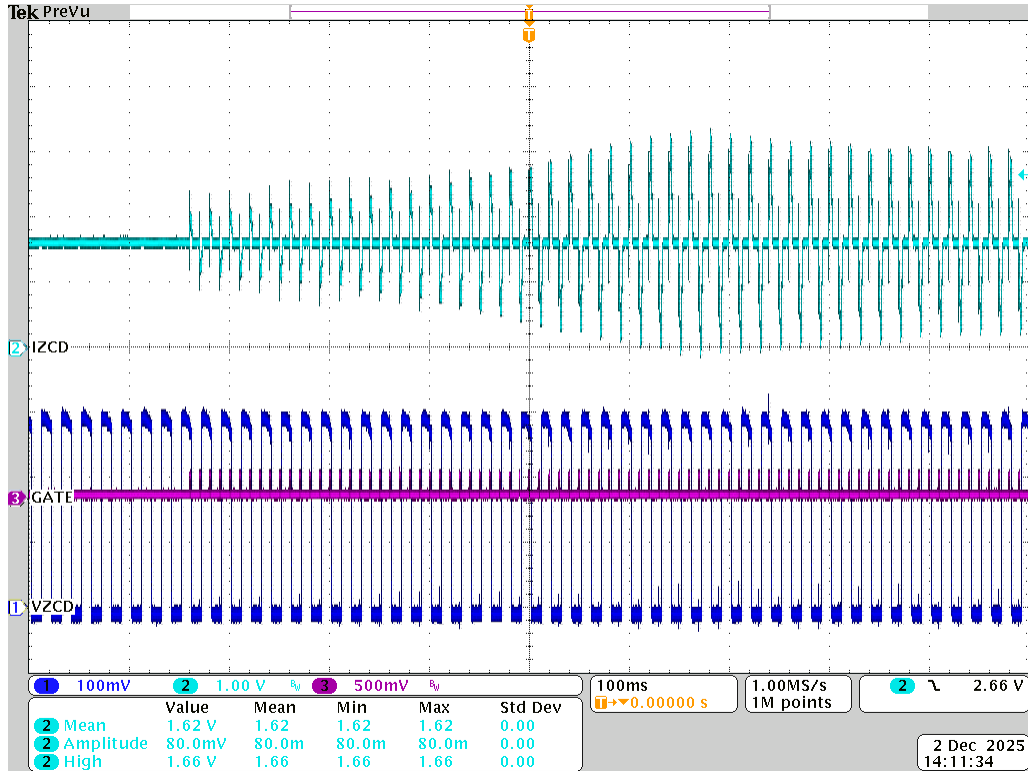


Figure 6-8. Soft Start

6.4 Temperature

Figure 6-9 presents the thermal measurement results for critical components. After running at high-speed mode for 10 minutes, the TRIAC can be up to 125°C due to conduction losses during current flow. While MSPM0C1104 MCU package temperature remains within 10°C of ambient under all operating conditions.

TI recommends providing proper heat dissipation for the TRIAC during actual production.

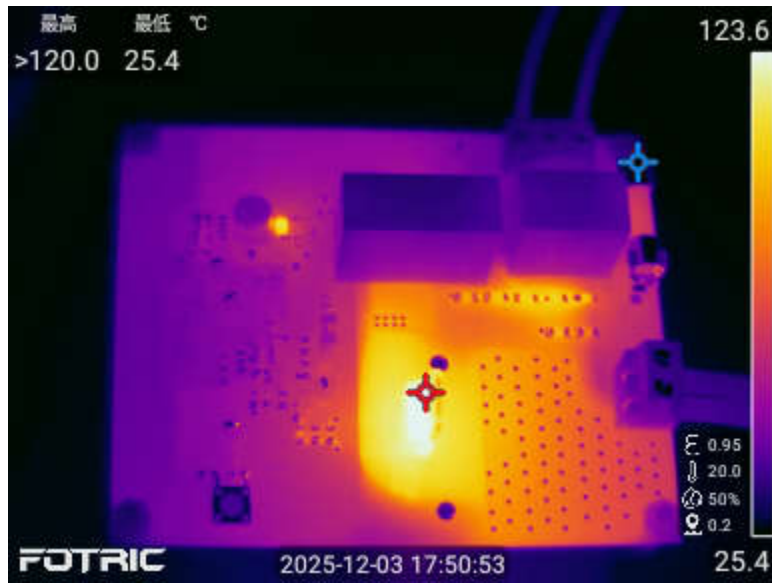


Figure 6-9. Temperature Test

7 Summary

This application note presents a complete TRIAC-based AC motor control design using the MSPM0C1104 microcontroller for household appliance applications. The design demonstrates that precise phase angle control, soft-start functionality, and overcurrent protection can be achieved with minimal external components through effective use of the integrated peripherals of the MSPM0 including high-resolution timers, ADC, and flexible GPIO.

8 References

1. Texas Instruments, [MSPM0C110x, MSPS003 Mixed-Signal Microcontrollers](#), datasheet.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you fully indemnify TI and its representatives against any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to [TI's Terms of Sale](#), [TI's General Quality Guidelines](#), or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

TI objects to and rejects any additional or different terms you may propose.

Copyright © 2026, Texas Instruments Incorporated

Last updated 10/2025