Design Guide: TIDA-010257

10kW Vienna Rectifier-Based, Three-Phase Power Factor Correction Reference Design

Description

The Vienna rectifier power topology is used in high-power, three-phase power factor correction applications such as appliances, electric vehicle (EV) chargers, and telecom rectifiers. Control design of the rectifier can be complex. This design guide illustrates a method to control the power stage using the C2000™ microcontroller (MCU). The design also enables monitoring and control of a Vienna rectifier based on GUI. The hardware and software available with this design helps accelerate the time to market.

Resources

TIDA-010257	Design Folder
TMS320F2800137, TMCS1123	Product Folder
AMC1350, UCC5350	Product Folder
UCC28750, LM25180	Product Folder
ISOTMP35, ISO6721	Product Folder
TPS54202, TLV76133	Product Folder

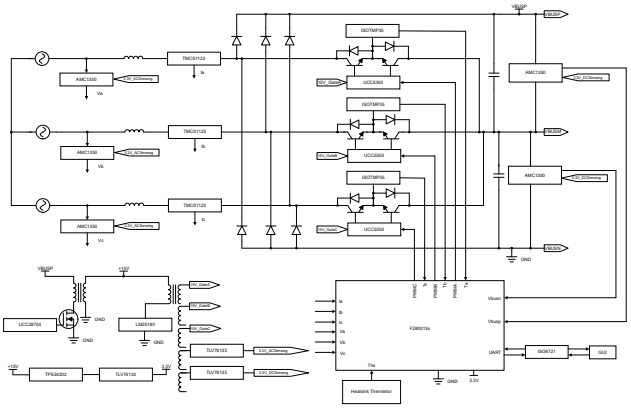
Ask our TI E2E™ support experts

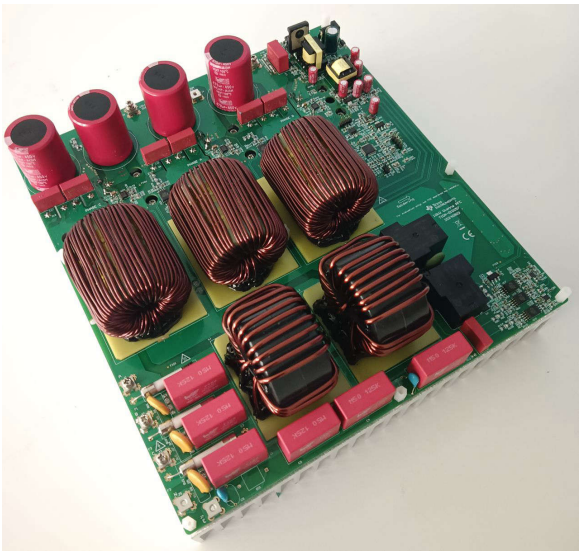
Features

- Three-phase input 380V, 400V_{L-L} 50/60Hz, output 650V DC nominal, 10kW
- 40kHz pulse width modulation (PWM) switching
- > 98% peak efficiency
- Less than 1.5% total harmonic distortion (THD) at full load
- Software available for F280013x
- Monitoring and control of Vienna rectifier based on UART GUI

Applications

- Air Conditioner outdoor unit
- Heat pump
- EV charging infrastructure
- Telecom rectifiers
- Factory automation and control





1 System Description

Three-phase power is used by equipment operating at high power in industrial applications. To improve grid power quality and reduce the harmonic currents drawn, power factor correction is needed as many of the forward loads are DC. For example, commercial air conditioner, EV charger.

Though many topologies exist for active three-phase power factor conversion, a Vienna rectifier is popular due to the operation in continuous conduction mode (CCM), inherent multilevel switching (three level), and reduced voltage stress on the power devices. Traditionally, hysteresis-based controllers have been used for Vienna rectifiers. Only recently have sine triangle-based PWM been shown to work for Vienna rectifier control. This control can be quite challenging to design. Several variants of Vienna rectifiers exist, Figure 1-1 shows the variant of the Vienna rectifier chosen in this design along with the key voltages and currents being sensed.

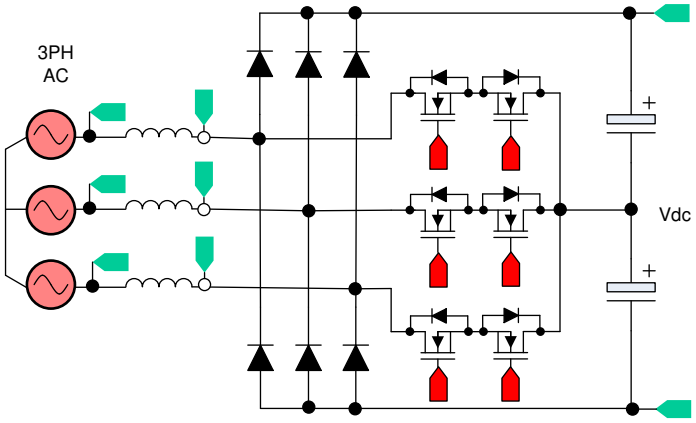


Figure 1-1. Vienna Rectifier Variant Implemented

A Y-connection Vienna rectifier is implemented in this design guide. With this design, the purpose is to provide an example of how to control a Vienna rectifier and how to tune the different loops using the C2000 MCU.

1.1 Terminology

PWM	Pulse Width Modulation
FET, MOSFET	Metal Oxide Semiconductor Field Effect Transistor
IGBT	Insulated Gate Bipolar Transistor
HVAC	Heating, Ventilation, and Air Conditioning
RMS	Root Mean Square
PLL	Phase Locked Loop

1.2 Key System Specifications

Table 1-1 details the three-phase Vienna rectifier key power specifications.

Table 1-1. Key System Specifications

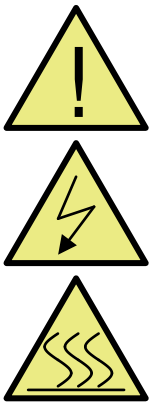
PARAMETER	SPECIFICATION
Input voltage (V_{IN})	• AC 208V_{RMS} V_{L-L} or 120V_{RMS} L-N , 60Hz or • AC 380V_{RMS} V_{L-L} or 220V_{RMS} L-N , 50Hz
Input current (I_{IN})	• 16A RMS maximum
Output voltage (V_{OUT})	• 650V DC bus nominal
Output current (I_{OUT})	• Absolute RMS maximum 16A, pulse maximum 29A
Power rating	• 10kW at three-phase 380V_{RMS}
Current THD	• < 1.5% at 10kW load • < 5% under 2.5kW load
Efficiency	Peak 98%
Primary filter inductor	355µH under 31A bias current
Output capacitance	940µF
PWM switching frequency	40kHz

WARNING

TI intends this reference design to be operated in a ***lab environment only and does not consider the board to be a finished product*** for general consumer use.

TI Intends this reference design to be used only by ***qualified engineers and technicians*** familiar with risks associated with handling high-voltage electrical and mechanical components, systems, and subsystems.

There are ***accessible high voltages present on the board*** . The board operates at voltages and currents that can cause shock, fire, or injury if not properly handled or applied. Use the equipment with necessary caution and appropriate safeguards to avoid injuring yourself or damaging property.



CAUTION

Do not leave reference design powered when unattended.

High voltage! There are ***accessible high voltages present on the board*** . Electric shock is possible. The board operates at voltages and currents that can cause shock, fire, or injury if not properly handled. Use the equipment with necessary caution and appropriate safeguards to avoid injuring yourself or damaging property. For safety, use of isolated test equipment with overvoltage and overcurrent protection is highly recommended.

TI considers it the user's responsibility to confirm that the voltages and isolation requirements are identified and understood before energizing the board or simulation. ***When energized, do not touch the reference design or components connected to the reference design*** .

Hot surface! Contact may cause burns. Do not touch!

Some components can reach high temperatures >55°C when the board is powered on. The user must not touch the board at any point during operation or immediately after operating, as high temperatures can be present.

2 System Overview

2.1 Block Diagram

Figure 2-1 shows the block diagram of the Vienna rectifier chosen in this design along with the key voltages and currents being sensed.

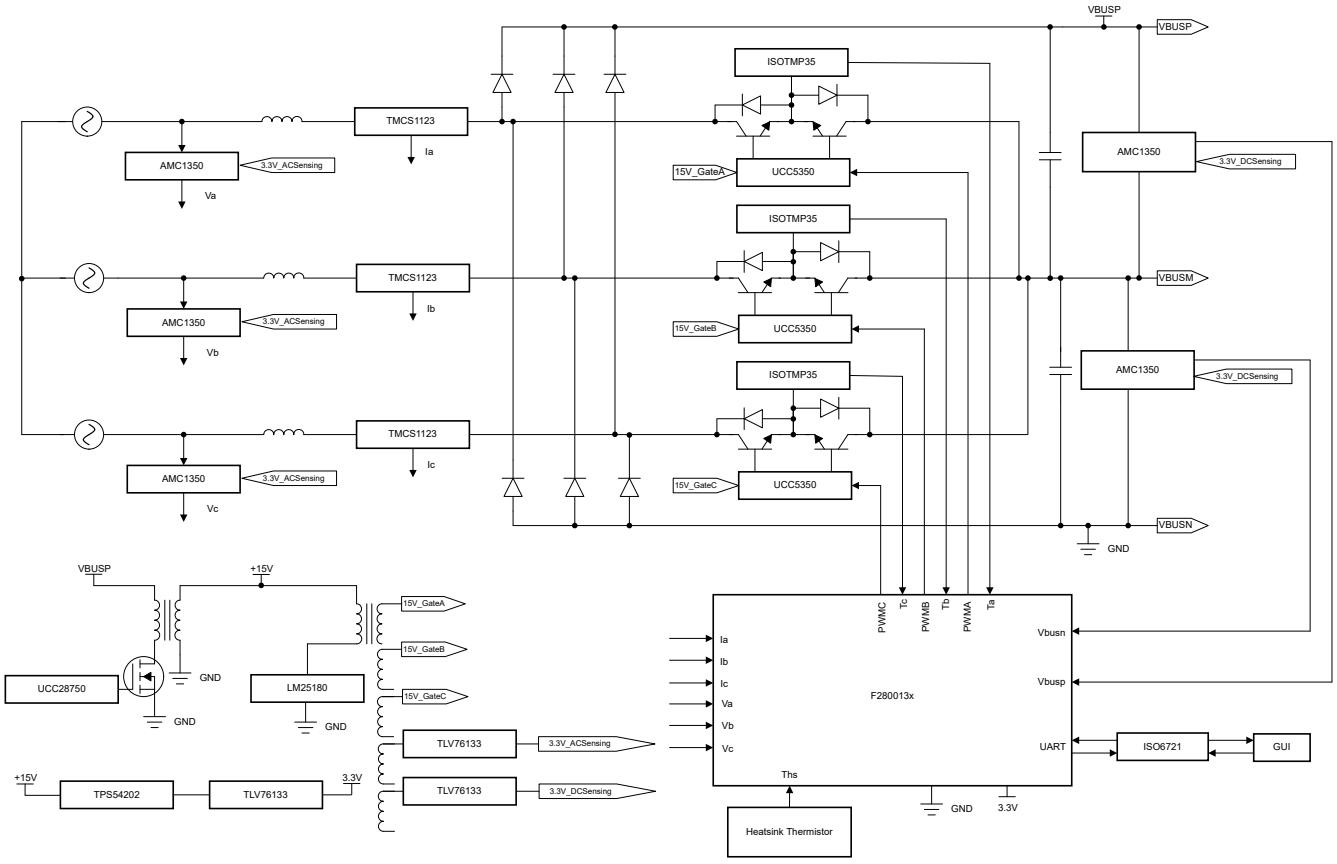


Figure 2-1. Block Diagram

2.2 Design Considerations

2.2.1 Control System Design Theory

This section discusses the control system design theory

2.2.1.1 PWM Modulation

Figure 2-2 shows a simplified, single-phase diagram of the Vienna rectifier. To control this rectifier, the duty cycle is controlled to regulate the voltage v_{xiN} directly. That is, if the software variable *Duty* is set to 1, v_{xiN} becomes the highest voltage possible by never turning the Q1 and Q2 switches on and letting the inductor connect to the DC bus though the bridge diode. Similarly when *Duty* is set to 0, PWM is modulated such that Q1 and Q2 always conduct, making v_{xiN} connect to the midpoint of the DC bus (which is zero), which creates the lowest possible voltage for the switching cycle.

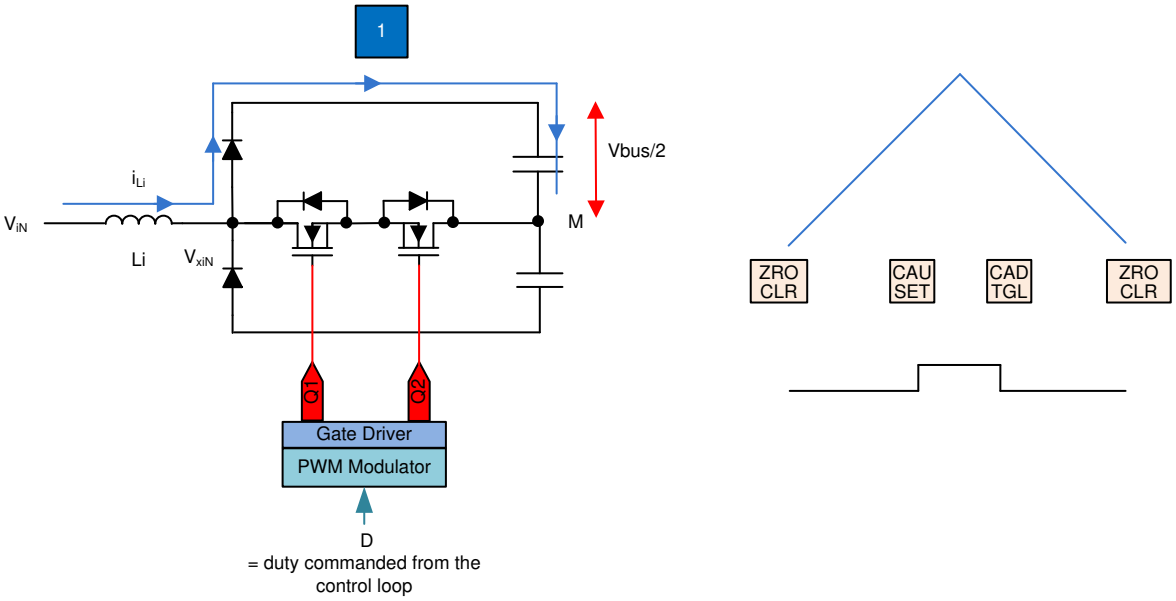


Figure 2-2. Single Phase Diagram of Vienna Rectifier

Figure 2-3 shows the detailed PWM configuration.

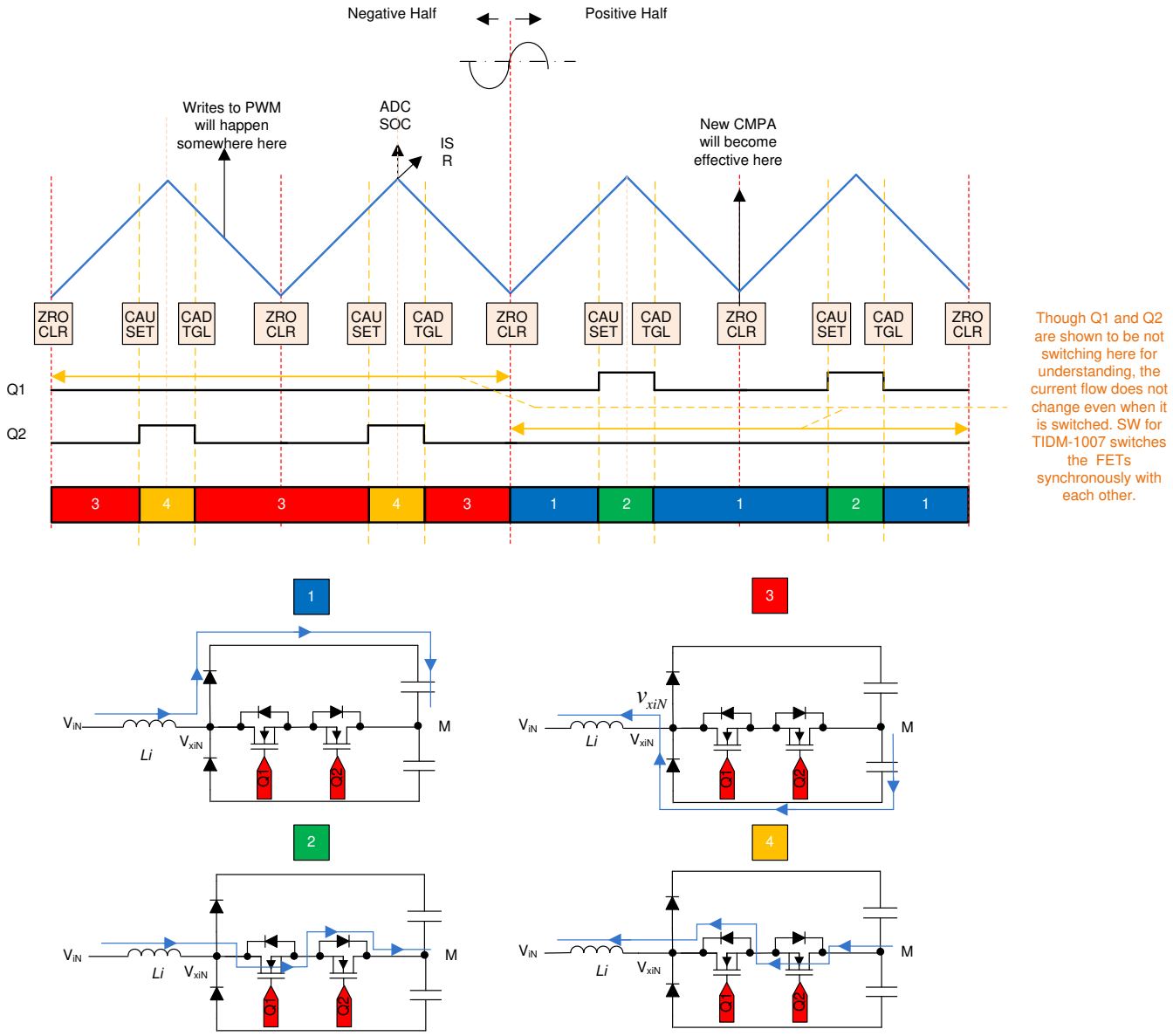


Figure 2-3. Vienna Rectifier Detailed PWM Modulation Scheme

2.2.1.2 Current Loop Model

To understand the current loop model, first closely look at the inductor current. In Figure 2-4 the duty cycle D is provided to the PWM modulator, which is connected to the switches Q1 and Q2. Remembering this, see Equation 1:

$$V_{xiN} = D \times \frac{V_{bus}}{2} \quad (1)$$

Note

When D is set to 1, all the switches are *off*, and when D is 0, all switches are *on*, which connect the inductor to the point to M.

To modulate the current through the inductor, the voltage v_{xiN} is regulated using the duty cycle control of the Q1 and Q2 switches. Assuming the direction of current is positive in the direction from the AC line into the rectifier and using the DC bus feedforward, the input AC voltage feedforward along with the assumption that the grid is fairly stiff. The current loop can be simplified as shown in Figure 2-4, and the current loop plant model can be written as in Equation 2.

$$H_{p\_i} = \frac{i_{Li}^*}{D} = \frac{1}{K_{v\_gain}} \times K_{i\_gain} \times K_{i\_fltr} \times G_d \times \frac{1}{Z_i} \quad (2)$$

where

- K_{v_gain} is the inverse of the maximum voltage sensed that is $1/V_{max\_sense}$ for the bus and the AC input. Assumed the AC voltage maximum sense and the DC Bus voltage maximum sense are equal.
- K_{i_gain} is the inverse of the maximum AC current sensed.
- K_{i_fltr} is the response of the RC filter connected from the current sensor to the ADC pin.
- G_d is the digital delay associated with the PWM update and digital control.
- i_{Li}^* is the current command, i_{Li} is the actual inductor current.
- $V_{bus}/2$ is the voltage across one of the output bus capacitor.
- Z_i is the impedance of the inductor which includes inductance L_i and resistance R_i.
- H_{p_i} is the current loop plant as seen by the digital controller G_i.
- v_{iN} is the instantaneous AC voltage at the input.

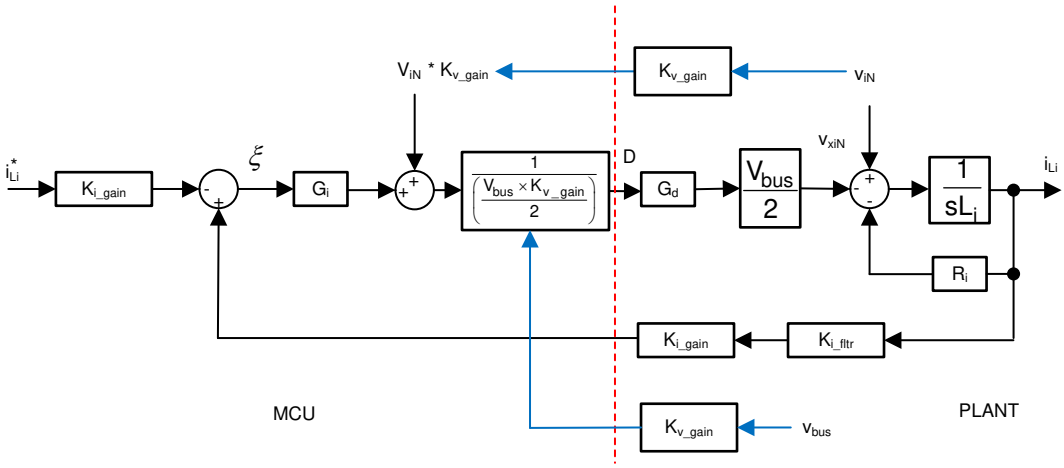


Figure 2-4. Current Loop Control Model

Note

The negative sign on the reference is in place because the current loop is thought to be regulating the voltage v_{xiN}. To increase the current, v_{xiN} must be reduced and thus, the opposite sign for reference and feedback in Figure 2-4. This current loop model is used to tune the current compensator. A simple proportional controller is used for the current loop. The gain of the proportional gain is adjusted to make sure the system is stable.

2.2.1.3 DC Bus Regulation Loop

The DC bus regulation loop is assumed to be providing the power reference. This loop is divided by the square of the line voltages RMS to provide the conductance, which is further multiplied by the line voltage to give the instantaneous current command.

A small signal model of the DC bus regulation loop is developed by linearizing Equation 3 around the operating point.

$$i_{DC}v_{bus} = 3nv_{Nrms}i_{Nrms} \Longrightarrow \hat{i}_{DC} = 3n\frac{\bar{V}_{Nrms}}{\bar{V}_{bus}}i_{Li} \tag{3}$$

For resistive load, the bus voltage and current relate as shown in Equation 4.

$$\hat{V}_{bus} = \frac{R_L}{1 + sR_LC_o}\hat{i}_{DC} \tag{4}$$

The DC voltage regulation loop control model can be drawn, as shown in Figure 2-5. An additional Vbus feedforward is applied to make the control loop independent of the bus voltage, and, thus, the plant model for the bus control can be written as shown in Equation 5.

$$H_{p\_bus} = H_{load} \times N \times K_{i\_gain} \times K_{v\_gain} \times K_{v\_flt} \tag{5}$$

where

- H_{p_bus} is the voltage loop plant as seen by the digital controller G_v.
- Output of the G_v is the power reference P_o *
- v_{bus} * is the voltage command and voltage reference, v_{bus} is the actual bus voltage.
- C_o is the output capacitor, R_L is the load resistance.

Using Figure 2-5, a proportional integrator (PI) compensator is designed for the voltage loop. The bandwidth of this loop is kept low as the loop is in conflict with the THD below steady state.

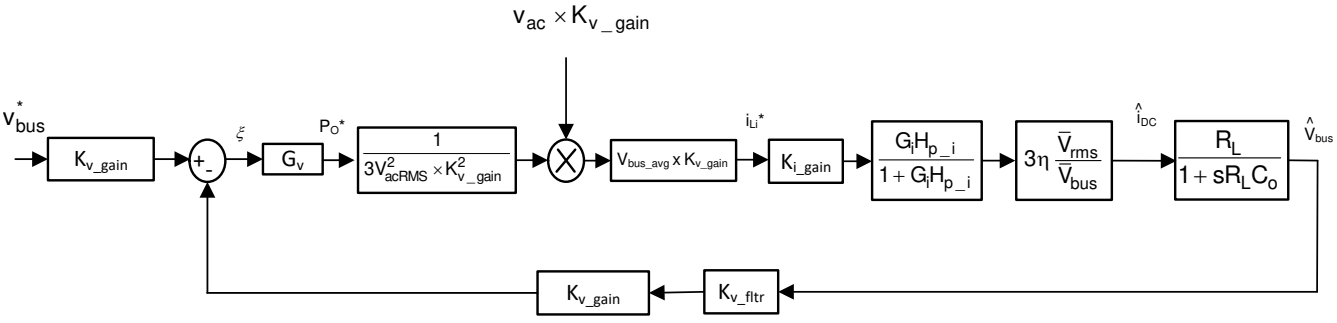


Figure 2-5. DC Voltage Loop Control Model

Additionally, a non-linear PI loop is used to reduce the transients in case of step load changes. Figure 2-6 shows the structure of the non-linear PI loop implemented on this design

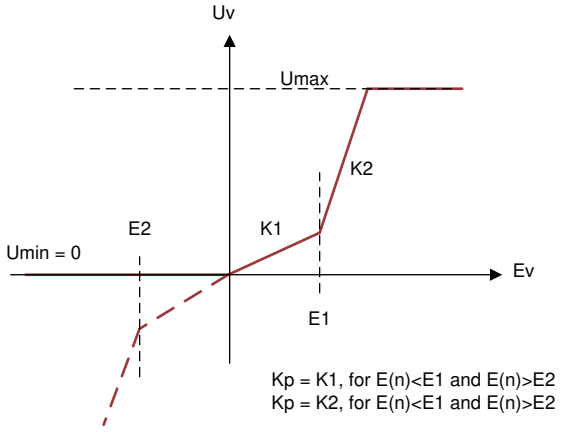


Figure 2-6. Non-Linear PI Loop for Voltage Controller

2.2.1.4 DC Voltage Balance Controller

A split capacitor is used for the output voltage bus in the Vienna rectifier. The voltages across these capacitors do not necessarily stay balanced naturally, hence a DC balance controller loop is added. This loop modulates an offset, which is added to the duty cycle thus, modulating the current through the midpoint to balance the voltages on the split capacitor.

A simple proportional gain is used for the DC bus balance controller, with the output of the balance loop given as in Equation 6.

$$G_{s\_out} = \left(V_{bus\_PM} - V_{bus\_MN}\right) \times G_{s\_gain\_Kp} \quad (6)$$

2.3 Highlighted Products

2.3.1 TMS320F280013x

The TMS320F280013x (F280013x) is a member of the C2000™ real-time microcontroller family of scalable, ultra-low latency devices designed for efficiency in power electronics. The real-time control subsystem is based on TI's 32-bit C28x DSP core, which provides 120MHz of signal processing performance for floating- or fixed-point code running from either on-chip flash or SRAM. The C28x CPU is further boosted by the Trigonometric Math Unit (TMU), speeding up common algorithms key to real-time control systems. The F280013x supports up to 256KB (128KW) of flash memory. Up to 36KB (18KW) of on-chip SRAM is also available to supplement the flash memory. High-performance analog blocks are integrated into the F280013x real-time microcontroller (MCU) and are closely coupled with the processing and PWM units to provide a best-in-class real-time signal chain performance. Fourteen PWM channels enable control of various power stages from a three-phase inverter to power-factor correction and other advanced multilevel power topologies.

2.3.2 UCC5350

The UCC53x0 is a family of single-channel, isolated gate drivers designed to drive MOSFETs, IGBTs, SiC MOSFETs, and GaN FETs (UCC5350SBD). The UCC53x0S provides a split output that controls the rise and fall times individually. The UCC53x0M connects the gate of the transistor to an internal clamp to prevent false turn-on caused by Miller current. The UCC53x0E has a UVLO2 referenced to GND2 to get a true UVLO reading. The UCC53x0 is available in a 4mm SOIC-8 (D) or 8.5mm SOIC-8 (DWV) package and can support isolation voltage up to $3kV_{RMS}$ and $5kV_{RMS}$, respectively. With these various options the UCC53x0 family is a good fit for motor drives and industrial power supplies. Compared to an optocoupler, the UCC53x0 family has lower part-to-part skew, lower propagation delay, higher operating temperature, and higher CMTI.

2.3.3 AMC1350

The AMC1350 is a precision, isolated amplifier with an output separated from the input circuitry by an isolation barrier that is highly resistant to magnetic interference. This barrier is certified to provide reinforced galvanic isolation of up to $5kV_{RMS}$ according to VDE V 0884-11 and UL1577, and supports a working voltage of up to $1.5kV_{RMS}$. The isolation barrier separates parts of the system that operate on different common-mode voltage levels and protects the low-voltage side from potentially harmful voltages and damage. The high-impedance input of the AMC1350 is optimized for connection to high-impedance resistive dividers or other voltage signal sources with high output resistance. The excellent accuracy and low temperature drift supports accurate AC and DC voltage sensing in DC/DC converters, frequency inverters, AC motor, and servo-drive applications over the extended industrial temperature range from –40°C to +125°C.

2.3.4 TMCS1123

The TMCS1123 is a galvanically isolated Hall-effect current sensor with industry leading isolation and accuracy. An output voltage proportional to the input current is provided with excellent linearity and low drift at all sensitivity options. Precision signal conditioning circuitry with built-in drift compensation is capable of less than 1.4% maximum sensitivity error over temperature and lifetime with no system level calibration, or less than 1% maximum sensitivity error including both lifetime and temperature drift with a one-time calibration at room temperature.

AC or DC input current flows through an internal conductor generating a magnetic field measured by integrated on-chip Hall-effect sensors. Coreless construction eliminates the need for magnetic concentrators. Differential Hall sensors reject interference from stray external magnetic fields. Low conductor resistance increases measurable current ranges up to ±96A while minimizing power loss and easing thermal dissipation requirements. Insulation capable of withstanding $5kV_{RMS}$, coupled with minimum 8.1mm creepage and clearance provide up to $1.3kV_{DC}$ reliable lifetime reinforced working voltage. Integrated shielding enables excellent common-mode rejection and transient immunity.

Fixed sensitivity allows the device to operate from a single 3V to 5.5V power supply, eliminating ratiometry errors and improving supply noise rejection.

2.3.5 UCC28750

UCC28750 is a highly-integrated current-mode, continuous-conduction-capable, PWM controller optimized for high-performance, low standby power, and cost-effective offline flyback converter applications using an

optocoupler. During light-load conditions, the device enters frequency foldback and burst modes, improving light-load efficiency. The burst-mode algorithm used in UCC28750 controls the minimum effective switching frequency to prevent audible noise during light-load conditions. Frequency dithering improves EMI performance, and is active in normal, frequency foldback, and power boost operations.

UCC28750 provides protection features that enable a robust converter design with minimal external components. Output over-power protection (OPP) and cycle-by-cycle overcurrent limit protect the load and power-stage components from electrical stress. Overvoltage and under-voltage lockout (OVLO and UVLO) prevent switching in unwanted input conditions. The FLT pin provides line brown-out sensing and protection, or external overtemperature and overvoltage protection, depending on the device variant. The FLT pin is also used to disable the device through external control by pulling the pin to ground, regardless of the device variant.

2.3.6 LM25180

The LM25180 is a primary-side regulated (PSR) flyback converter with high efficiency over a wide input voltage range of 4.5V to 42V. The isolated output voltage is sampled from the primary-side flyback voltage, eliminating the need for an optocoupler, voltage reference, or third winding from the transformer for output voltage regulation.

The high level of integration results in a simple, reliable, and high-density design with only one component crossing the isolation barrier. Boundary conduction mode (BCM) switching enables a compact magnetic design and better than ±1.5% load and line regulation performance. An integrated 65V power MOSFET provides output power up to 7W with enhanced headroom for line transients.

2.3.7 ISOTMP35

The TI ISOTMP35 is the first isolated temperature sensor IC in the industry, combining an integrated isolation barrier, up to 3000V_{RMS} withstand voltage, with an analog temperature sensor featuring a 10mV/°C slope from –40°C to 150°C. This integration enables the sensor to be colocated with high-voltage heat sources (for example: HV FETs, IGBTs, or HV contactors) without requiring expensive isolation circuitry. The direct contact with the high-voltage heat source also provides greater accuracy and faster thermal response compared with approaches where the sensor is placed further away to meet isolation requirements. Operating from a non-isolated 2.3V to 5.5V supply, the ISOTMP35 allows easy integration into applications where sub-regulated power is not available on the high-voltage plane. The integrated isolation barrier satisfies UL 1577 requirements. The surface mount package (7-pin SOIC) provides excellent heat flow from the heat source to the embedded thermal sensor, minimizing thermal mass and providing more accurate heat-source measurement. This reduces the need for time-consuming thermal modeling and improves system design margin by reducing mechanical variations due to manufacturing and assembly. The ISOTMP35 class-AB output driver provides a strong 500µA maximum output to drive capacitive loads up to 1000pF and is designed to directly interface with analog-to-digital converter (ADC) sample and hold inputs.

2.3.8 TLV76133

The TLV761 is a linear voltage regulator that improves the functionality of a traditional x1117 regulator (TLV1117 or LM1117) with tighter output accuracy and low quiescent current (I_Q) to lower the standby power consumption. The TLV761 is pin-to-pin compatible with other fixed SOT-223, TO-252 regulators. The TLV761 input voltage range is from 2.5V to 16V and provides an output voltage range from 0.8V to 13V to support a wide variety of applications. The wide bandwidth PSRR performance of the TLV761 is typically greater than 60dB at 1kHz and 40dB at 1MHz, which helps attenuate the switching frequency of an upstream DC/DC converter and minimizes post regulator filtering. Additionally, the TLV761 has an internal soft-start feature to reduce inrush current during start-up, which can help save space and cost in a design by minimizing input capacitance. The TLV761 features a foldback current limit that limits the power dissipation of the device during high-load current faults or shorting events.

2.3.9 TLV9062

The TLV9061 (single), TLV9062 (dual), and TLV9064 (quad) are single-, dual-, and quad- low-voltage, 1.8V to 5.5V) operational amplifiers, op amps) with rail-to-rail input and output swing capabilities. These devices are highly cost-effective options for applications where low-voltage operation, a small footprint, and high capacitive load drive are required. Although the capacitive load drive of the TLV906x is 100pF, the resistive open-loop

output impedance makes stabilizing with higher capacitive loads simpler. These op amps are designed specifically for low-voltage operation (1.8V to 5.5V), with performance specifications similar to the OPAx316 and TLVx316 devices.

The TLV906xS devices include a shutdown mode that allow the amplifiers to switch into standby mode with typical current consumption less than 1µA. The TLV906xS family helps simplify system design, because the family is unity-gain stable, integrates the RFI and EMI rejection filter, and provides no phase reversal in overdrive condition. Micro-size packages, such as X2SON and X2QFN, are offered for all the channel variants (single, dual, and quad), along with industry-standard packages, such as SOIC, MSOP, SOT-23, and TSSOP.

2.4 Hardware Design

2.4.1 Inductor Design

Input inductor (L_i) filters out the switching frequency harmonics. Inductor design, amongst other factors, depends on calculation of the current ripple and choosing a material for the core that can tolerate the calculated current ripple. Figure 2-7 shows one switching cycle waveform of the inverter output voltage vi with respect to the inductor current.

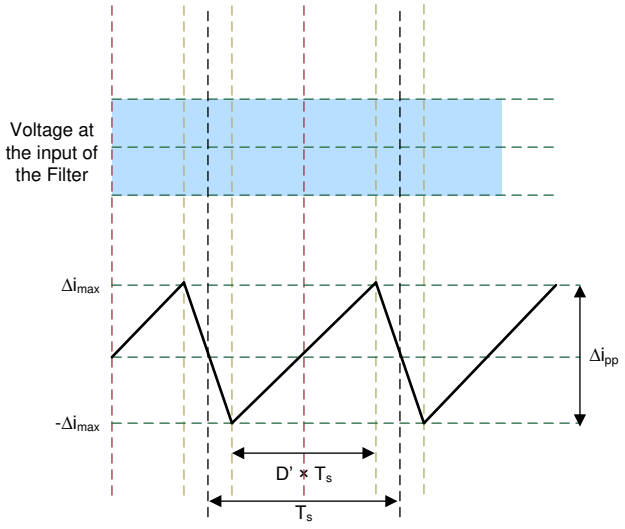


Figure 2-7. Current Ripple Calculation

The voltage across the inductor is given by V = L_i(di/dt). For the Vienna rectifier, see Equation 7.

$$\left(\frac{V_{bus}}{2} - V_{in}\right) = L_i \times \frac{\Delta i_{pp}}{D' \times T_s} \tag{7}$$

where

- $T_s = 1/F_{sw}$ is the switching period
- D' is the duty cycle for which the switches are ON

For control design, D is assumed to be the voltage at the other terminal of the inductor and is related to D' by D' = 1 – D. Rearranging the current ripple at any instant in the AC waveform is given as Equation 8.

$$\Delta i_{pp} = \frac{D' \times T_s \times \left(\frac{V_{bus}}{2} - V_{in}\right)}{L_i} \tag{8}$$

Now assuming modulation index to be m_a the duty cycle can be given as D' = m_a*sin(ωt) and assuming that V_{in} = D' × (V_{bus}/2), Equation 9 can be derived.

$$\Delta i_{pp} = \frac{\frac{V_{bus}}{2} \times T_s \times m_a \times \sin(\omega t) \times (1 - m_a \sin(\omega t))}{L_i} \quad (9)$$

From Equation 9, it is clear that the peak ripple is a factor where the input AC is in the sinusoidal waveform.

To get the maximum value differentiating the equation with respect to time, use Equation 10.

$$\frac{d(\Delta i_{pp})}{dt} = K\{\cos(\omega t)(1 - m_a \sin(\omega t)) - m_a \sin(\omega t) \times \cos(\omega t)\} = 0 \quad (10)$$

Which gives the maximum ripple exists at $\sin(\omega t)=1/(2 \times m_a)$, Substituting this value, Equation 11 is derived.

$$\Delta i_{pp_{max}} = \frac{\frac{V_{bus}}{2} \times T_s}{4 \times L_i} \Rightarrow L_i = \frac{\frac{V_{bus}}{2}}{4 \times F_{sw} \times \Delta i_{pp_{max}}} \quad (11)$$

With these values in mind, an appropriate core can be selected along with an inductor designed to meet this inductance value.

2.4.2 Bus Capacitor Selection

The bus capacitor is responsible for removing the ripple on the DC voltage that can be caused by the draw of sinusoidal currents. The capacitor value and the DC bus ripple are related by Equation 12.

$$C = \left(\frac{1}{3}\right) \frac{P_{ac}}{4 \times f \times \left(V^2 - (V - \Delta V)^2\right)} \quad (12)$$

This equation is used to select the minimum DC bus capacitance value.

Note

The calculation in Equation 12 can yield to an over design of the capacitor. The capacitor sizing is more dependent on the load and the nature of current drawn. For three-phase PFC the power ripple is fairly small as the input always has a path to the output. Only use Equation 12 for reference and be aware that this yields to over design.

2.4.3 Input AC Voltage Sensing

First a virtual neutral is constructed by using a resistor network connected in Y along with some capacitance for stability. In this design, the controller is kept on the cold side, thus, an isolated amplifier AMC1350 is used to process the VL-N' voltages as shown in Figure 2-8. As the AMC1350 is designed considering low-impedance sources for current-sensing applications, the input differential resistance plays a non-linear role in the total gain calculation. Therefore, a final calibration during build level one must be done, and the maximum AC voltage range must be adjusted according to the calibration.

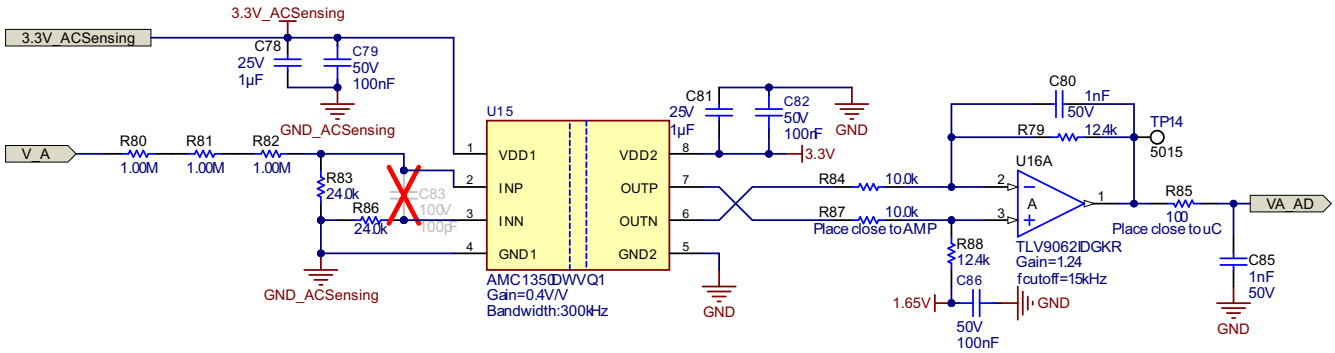


Figure 2-8. Input AC Voltage Sensing

2.4.4 Output DCBUS Voltage Sensing

Similarly, the DCBUS voltage, which is split between two capacitors, is sensed using AMC1350 and TLV9062 as shown in Figure 2-9. Because accuracy is important in the DCBUS voltage sensing, a further step of calibration with offset and gain adjustment needs to be carried out.

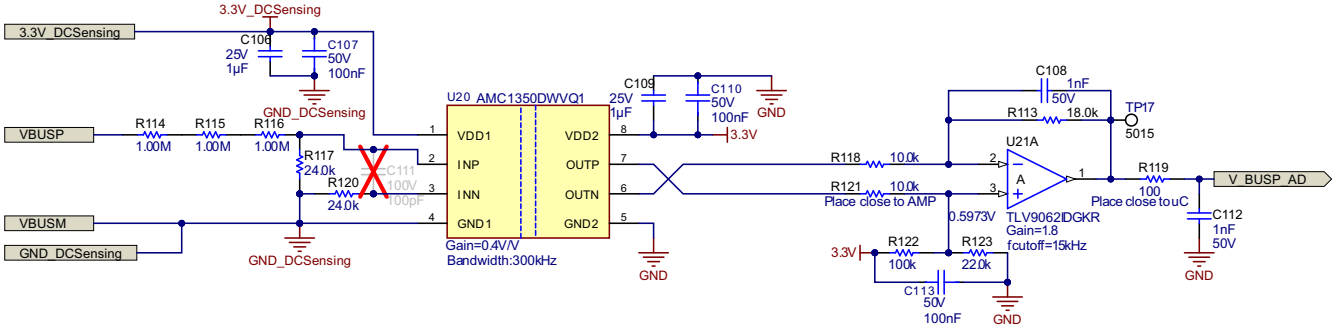


Figure 2-9. Bus Voltage Sensing

2.4.5 Auxiliary Power Supply

An auxiliary power supply is designed to provide 15V voltage rails. This power supply is controlled by U7 UCC28750 and a high-voltage MOSFET Q8. The 3.3V voltage rails are created with an LDO U11 TLV76133 followed after with a DC/DC buck supply. This setup helps reduce the voltage drop and heat on the LDO, and the LDO also provides a clean and stable 3.3V, which can be used as voltage reference for ADC. Figure 2-10 shows this circuit.

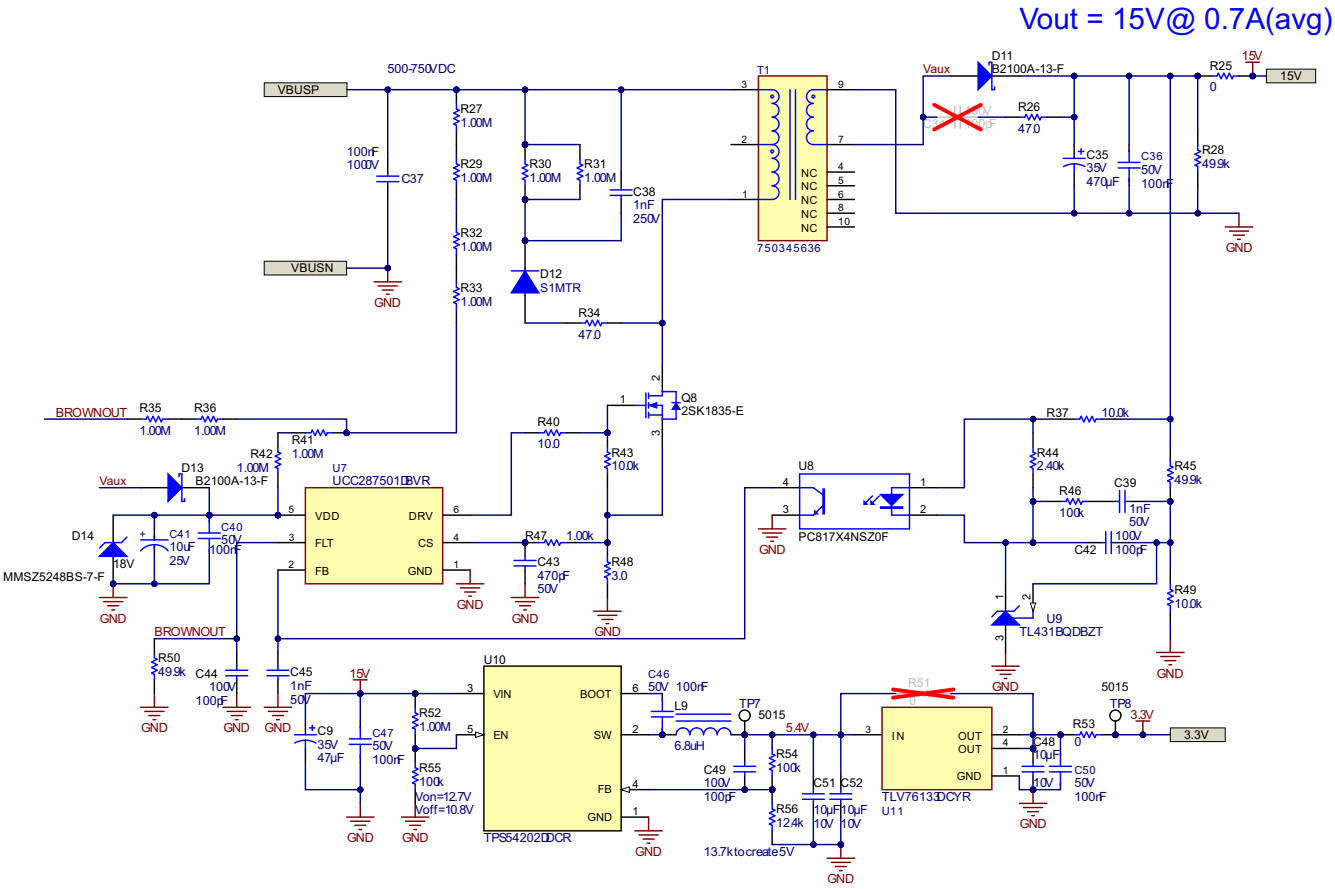


Figure 2-10. Auxiliary Power Supply

2.4.6 Isolated Power Supply

This reference design needs five isolated low-voltage rails to support AC input voltage sensing, DC output voltage sensing, and for IGBT gate drivers to work. A simple low-voltage flyback power supply based on LM25180 is designed to create these isolated voltage rails. Figure 2-11 shows this circuit.

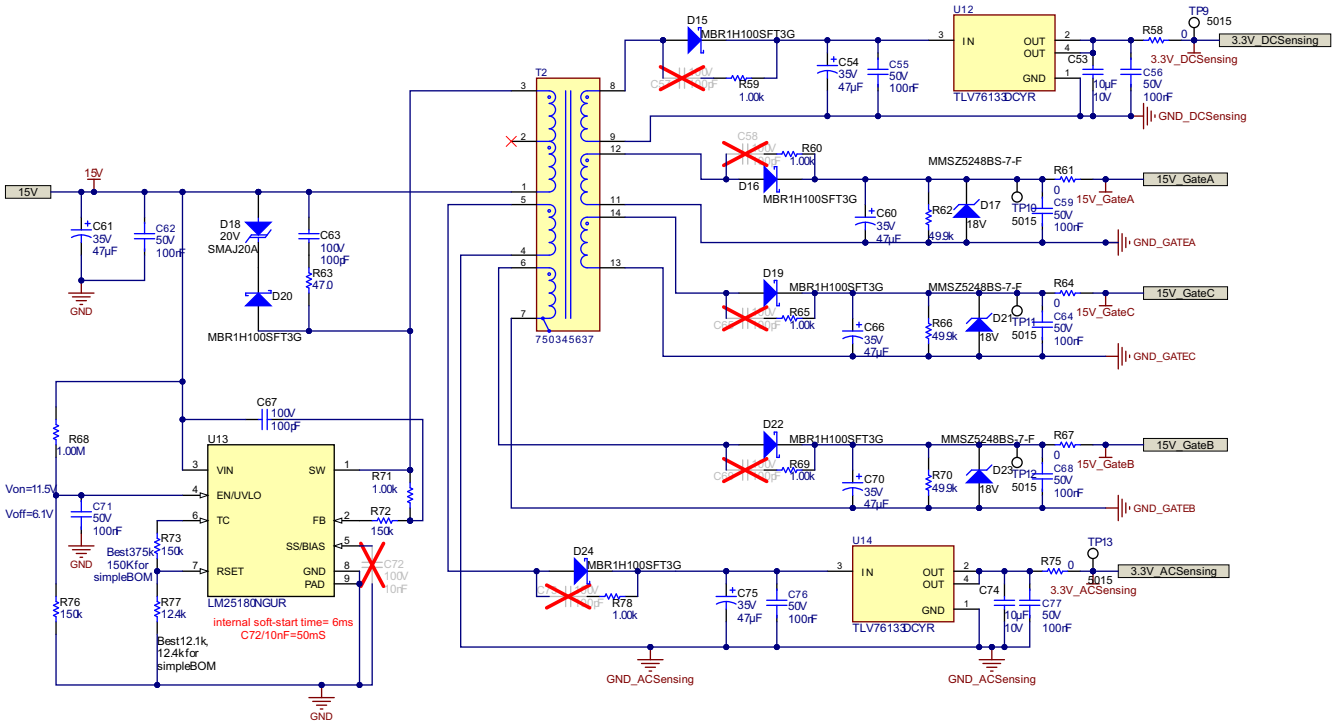


Figure 2-11. Isolated Power Supply

2.4.7 Inductor Current Sensing

The TMCS1123 Hall-effect sensor is used to sense the current through the inductor. The Hall-effect sensor has a built-in 1.65V offset, which ADC can measure directly. Figure 2-12 shows the circuit.

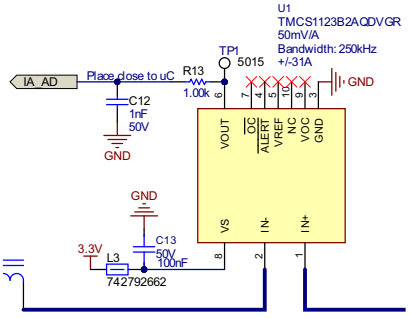


Figure 2-12. Current Sense Using Hall-Effect Sensor

2.4.8 Gate Driver

Figure 2-13 shows the UCC5350MC device being used to drive IGBT on this reference design. This gate driver has 10A source and sink current, and a Miller clamp. High-current driving can speed up IGBT rise and fall time, and also reduce power loss during on-off transition.

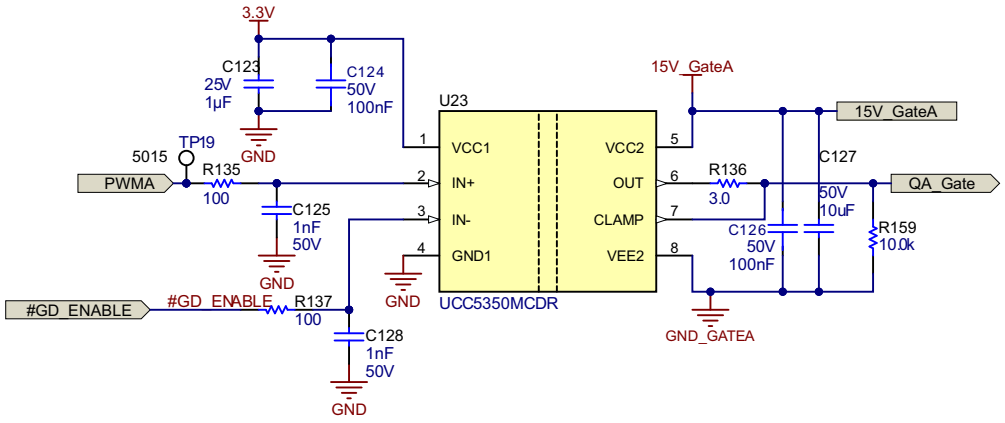


Figure 2-13. Gate Driver

2.4.9 Isolated Temperature Sensing

Figure 2-14 shows the ISOTMP35 isolated temperature sensor which is used to sense the temperature of power devices. The TSENSE pad can be on the same copper pours with the power devices because the pad is isolated. Ambient temperature is also easy to calculate with the output voltage, since this voltage is linear.

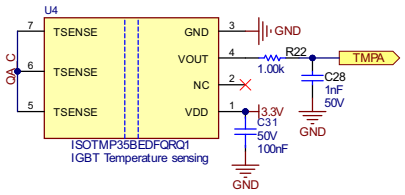


Figure 2-14. Isolated Temperature Sensing

2.4.10 Overcurrent, Overvoltage Protection (CMPSS)

Most power electronics converters need protection from an overcurrent and overvoltage event. For this design, multiple comparators are required, and references for the trip must be generated, as shown in Figure 2-15.

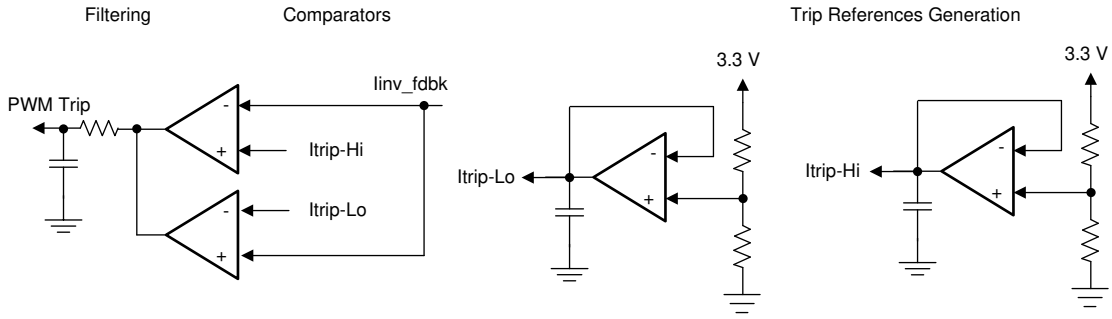


Figure 2-15. Trip Generation for PWM Using Comparators and Reference Generators

All this circuitry is avoided when using the C2000 MCUs such as TMS320F280013x, which have on-chip window comparators as part of the CMPSS that are internally connected to the PWM module and can enable fast tripping of the PWM. This device saves board space and is cost efficient in the end application because extra components can be avoided using on-chip resources, as shown in Figure 2-16.

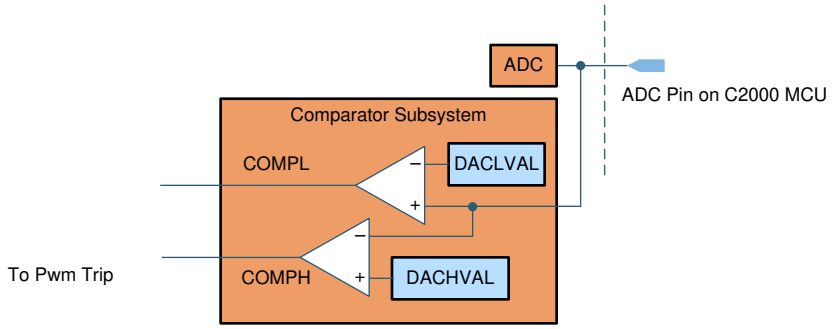


Figure 2-16. Comparator Subsystem (CMPSS) Used for Overcurrent and Overvoltage Protection

3 Hardware, Software, Testing Requirements, and Test Results

This section covers hardware, software and test results.

3.1 Hardware Requirements

3.1.1 Getting Started Hardware

This section details the necessary equipment, test setup, and procedure instructions for the reference design board and software testing and validation.

3.1.1.1 Board Overview

The reference board has functional groups that enable a complete three-phase Vienna PFC system. The following list show the function blocks on the board. Figure 3-1 shows the top view of the board and different blocks. Table 3-1 shows the key connectors and functions.

- AC input line filters and relays
- Boost Inductors
- Bridge and IGBTs
- Auxiliary power supply and isolated power supply
- DSP, F2800137
- Voltage sensing and gate drivers
- IGBT gate drivers

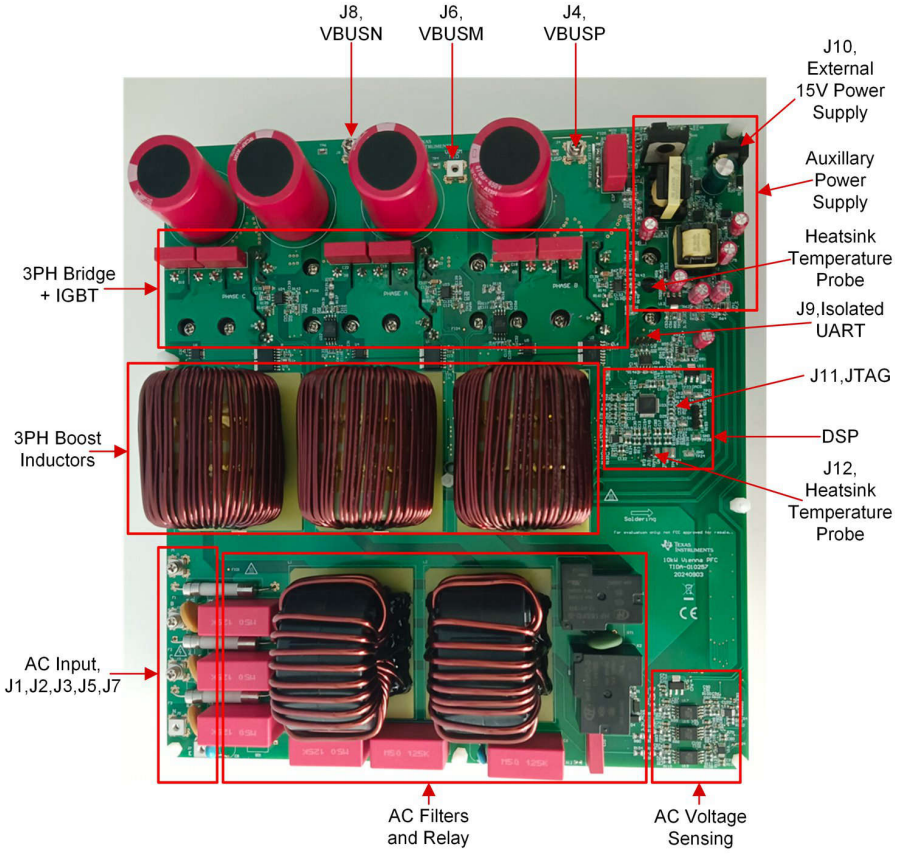


Figure 3-1. Board Overview

Table 3-1. Key Connectors and Function

CONNECTOR NAME	FUNCTION
J1, J2, J3	Lines of three phase AC input
J5	Neutral of three phase AC input
J7	Ground of three phase AC input
J4	Output DCBUS positive
J6	Output DCBUS middle
J8	Output DCBUS negative
J9	Isolated universal asynchronous receiver-transmitter (UART) for GUI
J10	15V, 1A external power supply
J11	cJTAG for debugging
J12	Heat sink thermistor

3.1.1.2 Test Equipment

Make sure the following equipment is ready for testing:

1. 10kW AC power source range from 0VAC to 400VAC L to L, or grid power from 208VAC to 400VAC L to L
2. 10kW DC load, load can be changed.
3. 15V, 1A power supply adapter
4. USB to UART adapter and cable
5. Multimeters
6. Three-phase power analyzer
7. Digital oscilloscope

3.2 Software Requirements

3.2.1 Getting Started GUI

Source code for this reference design is provided so the designer can debug firmware directly, as explained in Section 3.2.2. However, software debugging needs more time. To speed up development, a UART-based GUI software is provided to help quickly control the board and observe the board working status. This section introduces how to use the UART GUI.

When connecting the host PC to this reference board at J9, the UART port needs to be supplied with the external 5V or 3.3V voltage rail since J9 is an isolated UART port.

3.2.1.1 Test Setup

Figure 3-2 shows the hardware connection for testing with the GUI. Set up the hardware using the following steps:

1. Connect GND, TX, RX, and 3.3V or 5V to VCC_ISO at J9 to the host PC through a USB-to-UART adapter.
2. Connect the AC input cable to J1, J2, J3. J5 and J7 can be connected or left disconnected.
3. Connect the DC output cable to J4 and J8, set no load.
4. Connect the multimeter, oscilloscope probes, and other measurement equipment to probe or analyze various signals and parameters.
5. Run GUI software on the laptop.

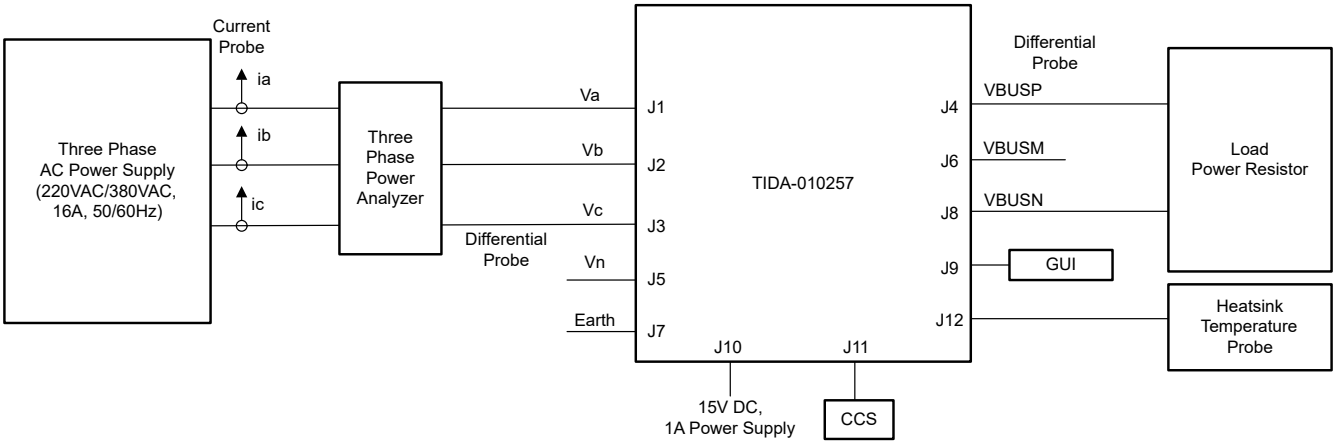


Figure 3-2. Hardware Setup for GUI

3.2.1.2 Overview of a GUI Software

Any UART terminal GUI software can be used for the communication between the host PC and the reference design board. Figure 3-3 shows a GUI software, this software has the display window to show the reported working status from the reference design board, the GUI software can also send commands to the board in HEX mode. The supported baud rate is 115200.

The reported board working status includes AC input voltage, DC output voltage, AC input current, AC input power factors, power devices temperatures, heat sink temperature, and power on time. Those data are in ASCII mode.

The commands are:

- 0x11 is the command to start PFC
- 0x22 is the command to stop PFC
- 0x33 is the command to clear errors

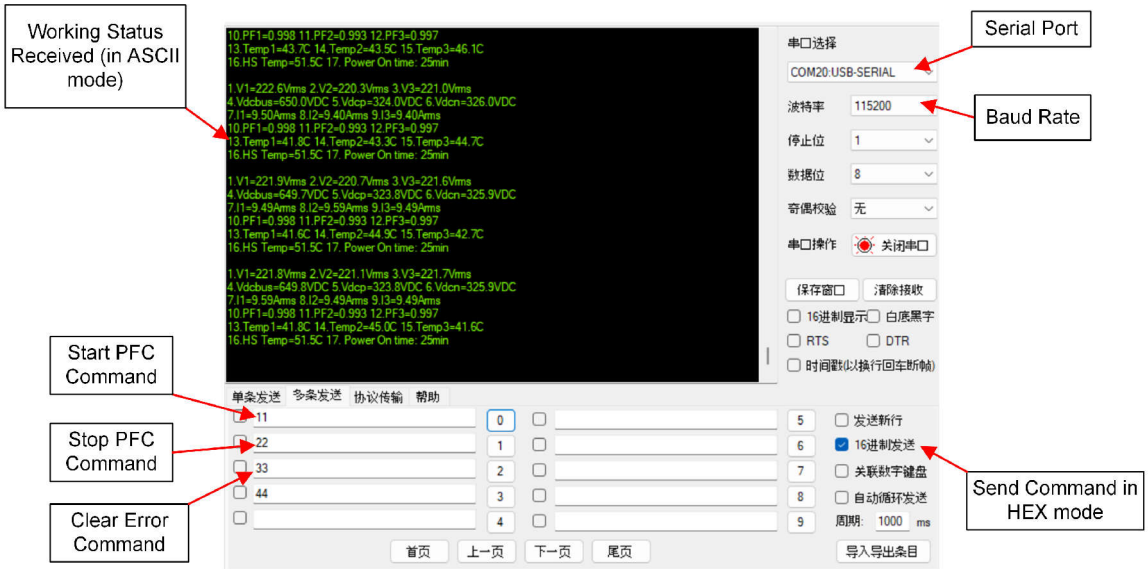


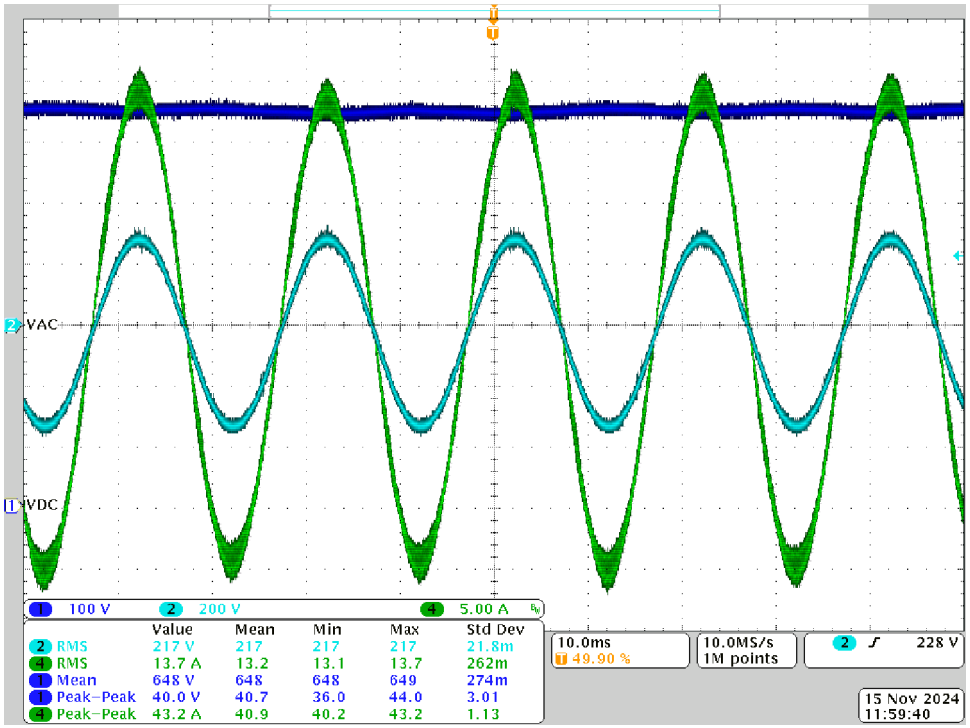
Figure 3-3. Overview of a GUI Software

3.2.1.3 Procedures of Test With GUI

Use the following steps to test the reference design with the GUI:

1. Connect the USB to UART adapter from the laptop to J9.

2. Connect the AC source(208AC to 400VAC$_{L-L}$) to J1, J2, and J3. J5 and J7 can be connected or can be left disconnected.
3. Connect the DC load to J4 and J8, set to no load.
4. Run the GUI software, choose the correct UART port and set the baud rate at 115200bps.
5. Power on the board, wait for the D25 LED to blink.
6. Wait for relays to close.
7. Check the GUI, the DC output voltage is about 530VDC at 380VAC input, since the PFC does not start.
8. Send the 0x11 command to start the PFC, until the DC output voltage is about 680VDC under no load.
9. Increase the load step by step, until the DC output voltage is about 650VDC (when the load is > 500W). Check the AC input current, DC output voltage, and temperature during loading. Figure 3-4 shows the waveforms under 380VAC, 650VDC, 9kW.



Note

- CH1 (Blue): DCBUS output voltage
- CH2 (Light blue): AC input phase A voltage
- CH4 (Green): AC Input phase A current

Figure 3-4. Waveforms 380VAC, 650VDC, 9kW

10. Unload the board step by step until there is no load.
11. Send the 0x22 command to stop the PFC.
12. To bring the system to a safe stop, bring the input AC voltage down to zero.

CAUTION

This reference design has electrolytic capacitors. The discharge rate of the capacitors is very slow without an external load. Always pay attention to the DCBUS voltage.

A load on the DCBUS can speed up discharge; otherwise, wait a long time for the DCBUS to get to zero.

3.2.2 Getting Started Firmware

The software of this design is available inside *DigitalPower Software Development Kit (SDK) for C2000™ Microcontrollers*.

Note

The firmware for the design is supported for the TMS320F2800137 real-time microcontroller.

3.2.2.1 Opening the Project Inside Code Composer Studio™

To start using the software:

1. Download and install Code Composer Studio™ IDE from the Code Composer Studio™ (CCS) Integrated Development Environment (IDE) tools folder. Version 20.0.0 or newer is recommended.
2. Download and install *DigitalPower Software Development Kit(SDK)for C2000 Microcontrollers*
3. Once the installation is complete, open CCS, and create a new workspace for importing the project by clicking *Project* → *Import CCS Projects*, and browse to `<install_location>\C2000Ware_DigitalPower_SDK_5_03_00_00\solutions\TIDA-010257\f280013x`, and then click on the project name and import the project.

3.2.2.2 Project Structure

Figure 3-5 shows the project explorer inside CCS once the project is imported.

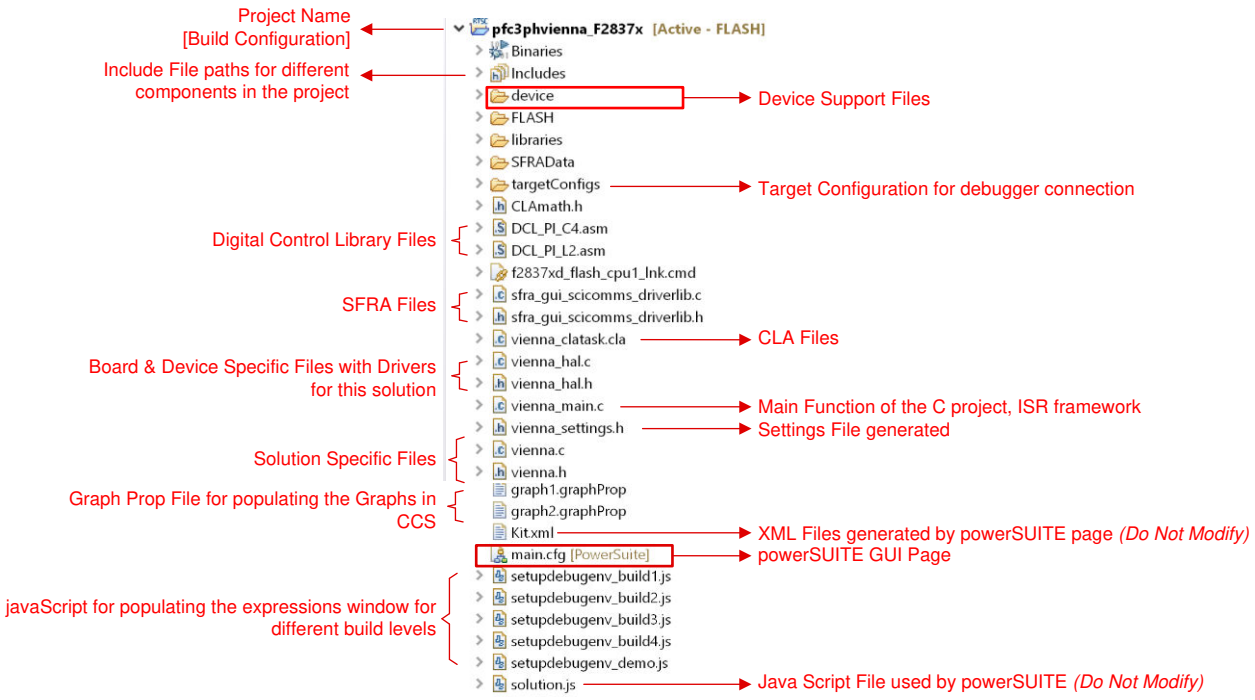


Figure 3-5. Project Explorer View of Design Project

The project consists of an interrupt service routine, which is called every PWM cycle, named `controlISR()`, where the control algorithm is executed. In addition to this, there are background tasks A0-A3 and B0-B3, which are called in a polling fashion and can be used to run slow tasks for which absolute timing accuracy is not required. A slower 10kHz routine `tenkHzISR()` is called for instrumentation and to run some slower tasks that require timing accuracy.

The software of this design guide is organized in four incremental builds (INCR_BUILD). The incremental build process simplifies the system bring up and design.

- INCR_BUILD 1: Open Loop Check
- INCR_BUILD 2: Closed Current Loop
- INCR_BUILD 3: Closed Voltage and Current Loop
- INCR_BUILD 4: Closed Voltage, Current, and Bus Cap Balance Loop

These build levels are detailed in Section 3.2.2.4.1 through Section 3.2.2.4.4.

3.2.2.3 Test Setup

Figure 3-6 shows the hardware blocks. Complete the following test setup steps before beginning each build level test:

1. Connect emulator to J9
2. Connect AC power supply to J1, J2, and J3, set to 0V output
3. Connect DC load to J4 and J8, set to 530Ω
4. Connect J9 to a laptop with a UART to USB adapter
5. Connect 15V, 1A DC adapter to J10

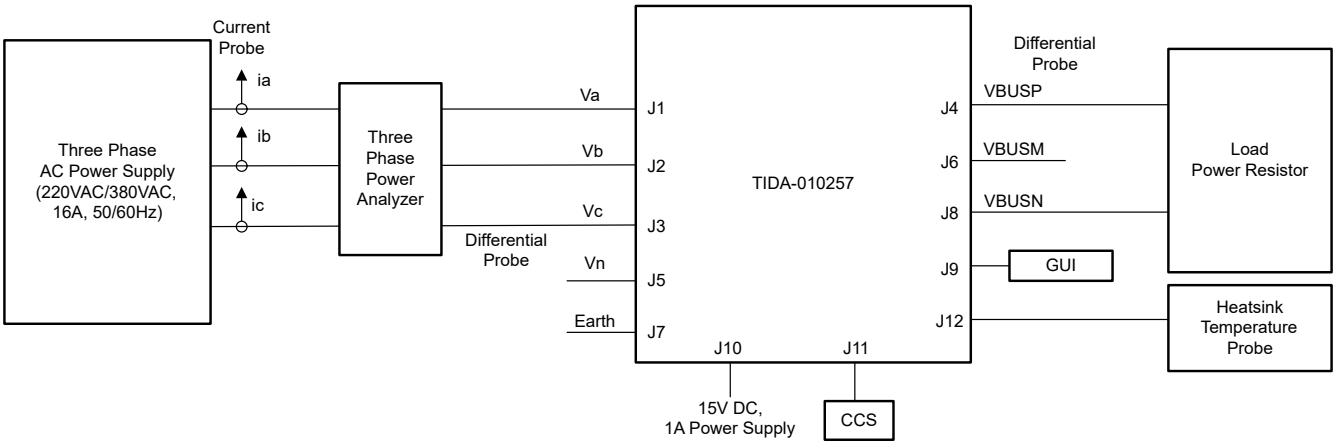


Figure 3-6. Hardware Blocks

3.2.2.4 Running Project

This section introduces firmware debugging for build levels 1–4. The variable name mentioned in the user guide is simplified compared with the real name in the code. For example, `VIENNA_guiVbus_Volts` is simplified as `guiVbus`.

3.2.2.4.1 INCR_BUILD 1: Open Loop

The board is run in an open-loop mode with a fixed-duty cycle. The duty cycle is controlled with the `dutyPU_DC` variable. This build verifies the sensing of feedback values from the power stage and also the operation of the PWM gate driver, which makes sure there are no hardware issues. Additionally, calibration of input and output voltage sensing can be performed in this build. The software structure for this build is shown in Figure 3-7. Blocks that run in the slower ISR are marked. Other blocks are run in the fast `controlISR`.

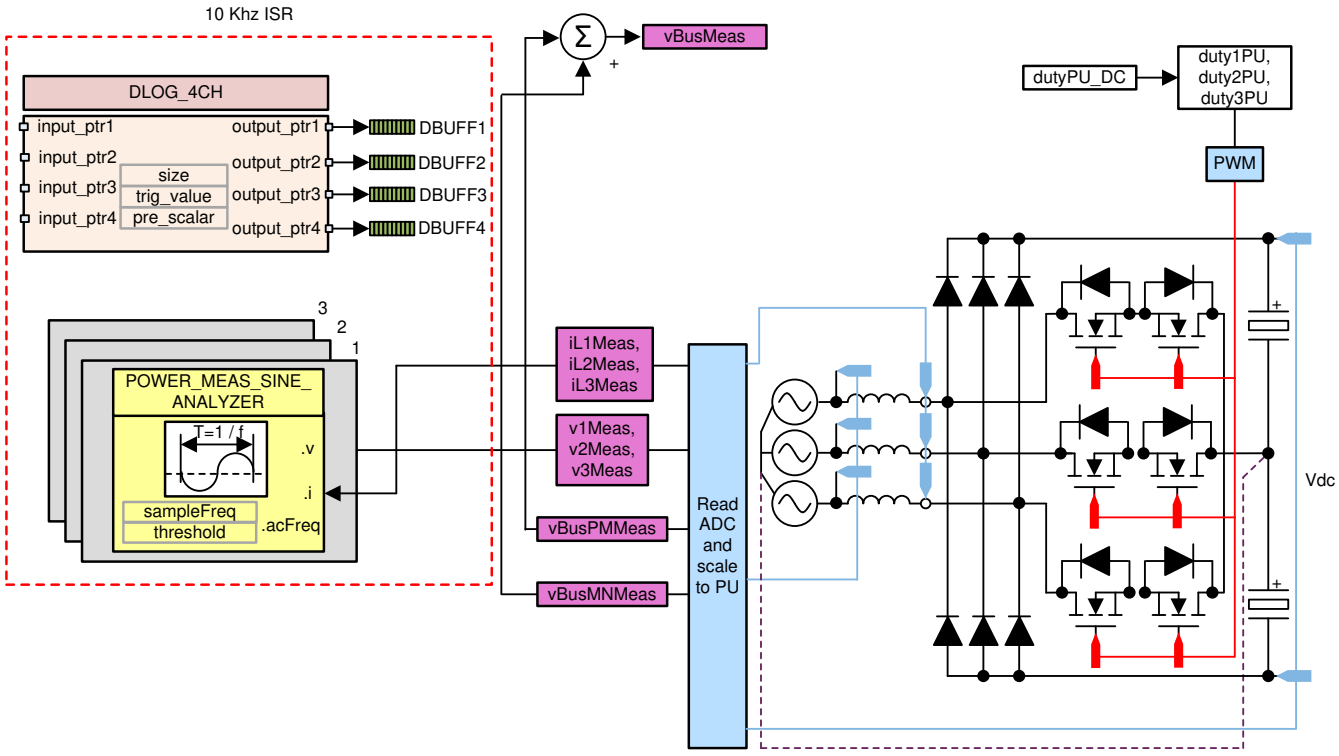


Figure 3-7. Build Level 1 Control Software Diagram: Open Loop Project

3.2.2.4.1.1 Setting, Building, and Loading the Project

1. Open the `vienna_settings.h` file, set VIENNA_INCR_BUILD to 1.
2. In the *Project Explorer* make sure the correct target configuration file is set as Active below `targetConfigs` (see Figure 3-5).
3. Right click on the project name and click *Rebuild Project*, the project builds successfully.
4. Click *Run* → *Debug*. This launches a debugging session.
5. The project then loads to the device, and CCS debug view becomes active. The code halts at the start of the main routine.

3.2.2.4.1.2 Setup Debug Environment Windows

1. To add the variables in the *Watch* or *Expressions* window, click *View* → *Scripting Console* to open the scripting console dialog box. On the upper right corner of this console, click the *Open* button, then browse to the `setupdebugenv_build1.js` script file located inside the project folder. This script file populates the *Watch* window with appropriate variables needed to debug the system. Click the *Continuous Refresh* button () on the *Watch* window to enable continuous update of values from the controller. Figure 3-8 shows the *Watch* window.

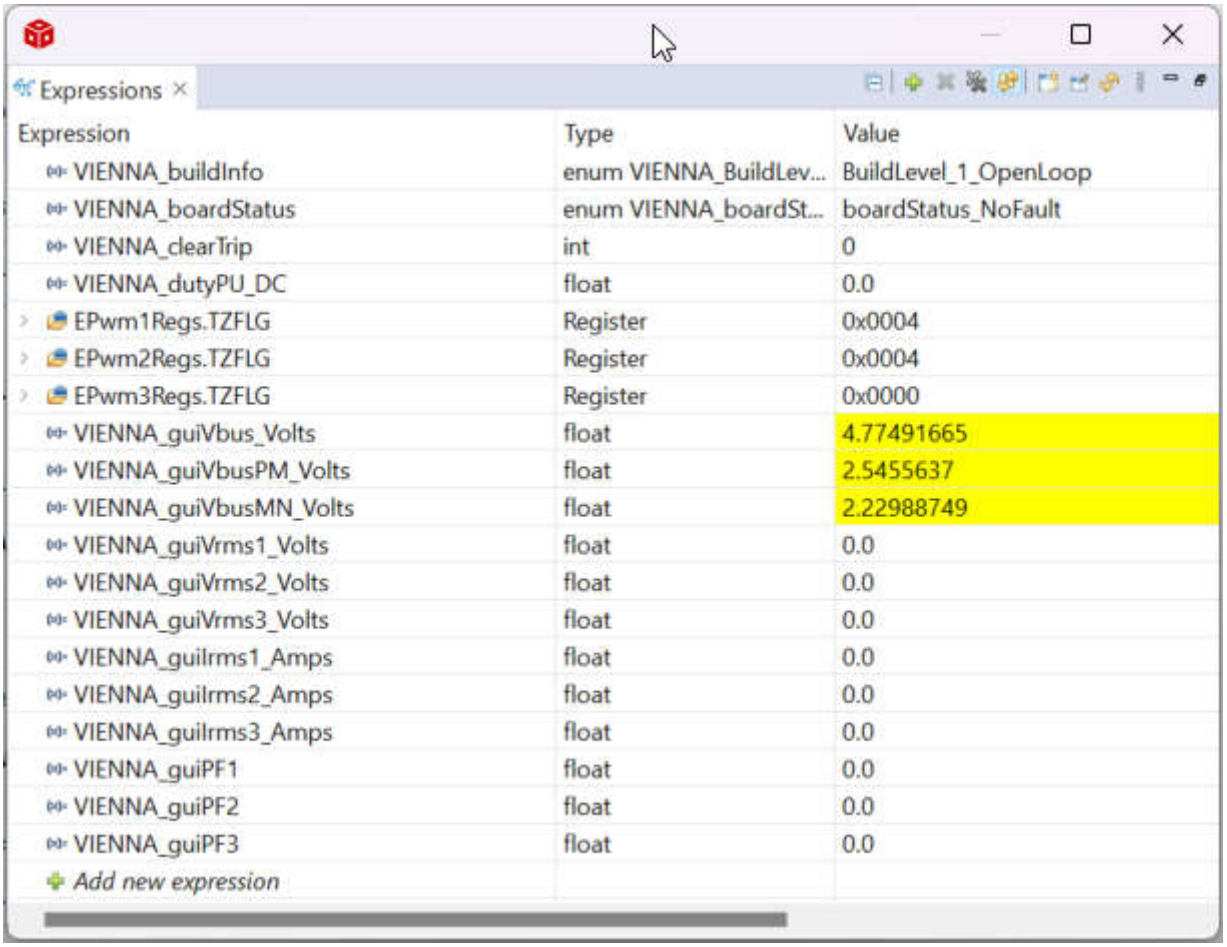


Figure 3-8. Build Level 1 Expressions View

2. The current and voltage measurements can be verified by viewing the data in the graph window. These values are logged in the slower 10kHz routine. Go to *Tools* → *Graph* → *DualTime*, and click *Import* and point to the `graph1.GraphProp` file inside the project folder. This file populates the *Graph Properties* window. Alternatively, enter the values as shown in the Figure 3-9. When the entries are verified, click the *OK* button.

   Two graphs appear in CCS. Click the *Continuous Refresh* button ( ) on these graphs. A second set of graphs can also be added by importing the `graph2.GraphProp` file.

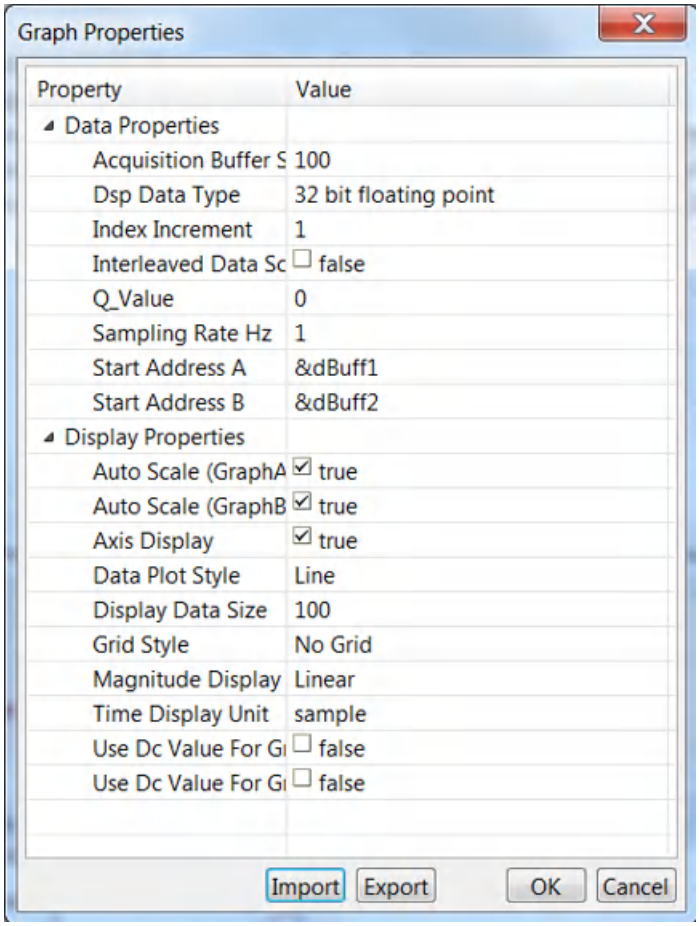
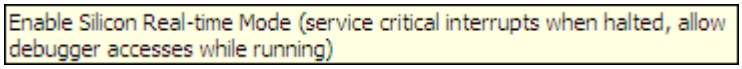


Figure 3-9. Graph Settings

3.2.2.4.1.3 Using Real-Time Emulation

Real-time emulation is a special emulation feature that allows windows within CCS to be updated while the MCU is running. This feature allows graphs and watch views to update but also allows the user to change values in *Watch* or memory windows and see the effect of these changes in the system without halting the processor.

1. Enable real-time mode by hovering the mouse on the buttons on the horizontal toolbar and clicking the button.

2. If a message box appears, select *YES* to enable debug events. This sets bit 1 (DGBM bit) of the status register 1 (ST1) to a *0*. The DGBM is the debug enable mask bit. When the DGBM bit is set to *0*, memory and register values can be passed to the host processor for updating the debugger windows.

3.2.2.4.1.4 Running Code (Build 1)

1. Run the project by clicking the  button.
2. In the watch view, periodically check if the `guiVbus` (`VIENNA_guiVbus_Volts` in the *Expression* window) variable is updating. If there is no change in the value, then make sure the real-time mode is enabled, and the hardware is set up correctly. Do not proceed further unless the update is verified.

Note

As no power is applied right now, this value is close to zero.

3. Slowly increase the input AC voltage from $0V_{RMS}$ to $80V_{RMS}$ L-N.
4. Verify the voltage sensing: Make sure `guiVbus`, `guiVbusPM`, and `guiVbusMN` display the correct values. For the $80V_{RMS}$ L-N, `guiVbus` is close to 190V, the graph function can show the waveform, as shown in Figure 3-10. The `guiVbusPM` and `guiVbusMN` variables are both close to 85V. The code runs a sine analyzer module, which computes the RMS value of the voltage and current. Figure 3-11 shows that the `guiVrms1`, `guiVrms2` and `guiVrms3` values are close to the input value, that is, $80V_{RMS}$. This verifies the voltage sensing of the board.

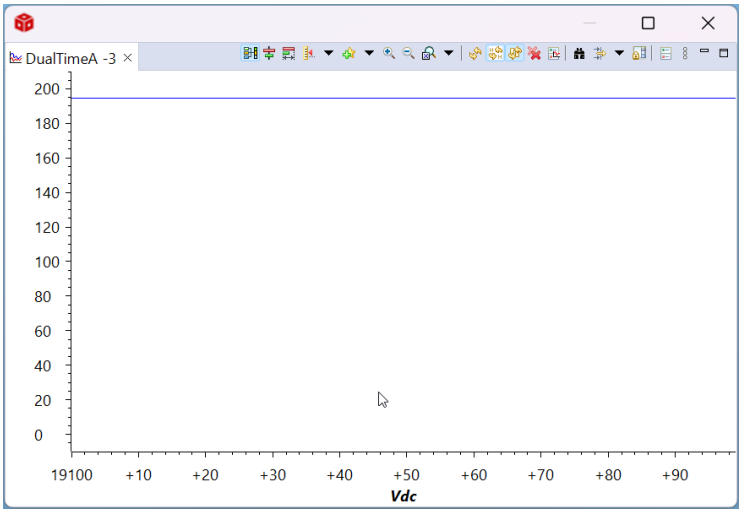


Figure 3-10. Build Level 1: Graph Showing Measured Output Voltages

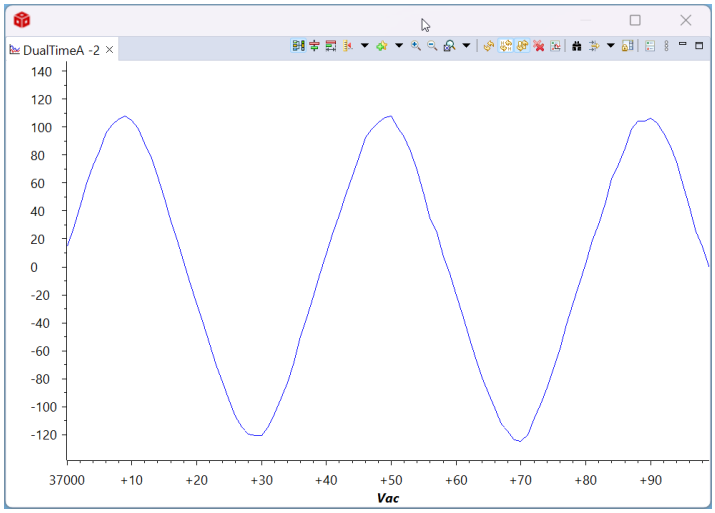


Figure 3-11. Build Level 1: Graph Showing Measured Input Voltages

5. Verifying the current sensing: Notice the `guiVrms1`, `guiVrms2`, and `guiVrms3` variables, for the given test condition these values are close to 0.5A. Additionally, the graphs must be seen to verify the current measurement. Figure 3-12 shows the currents on a graph.

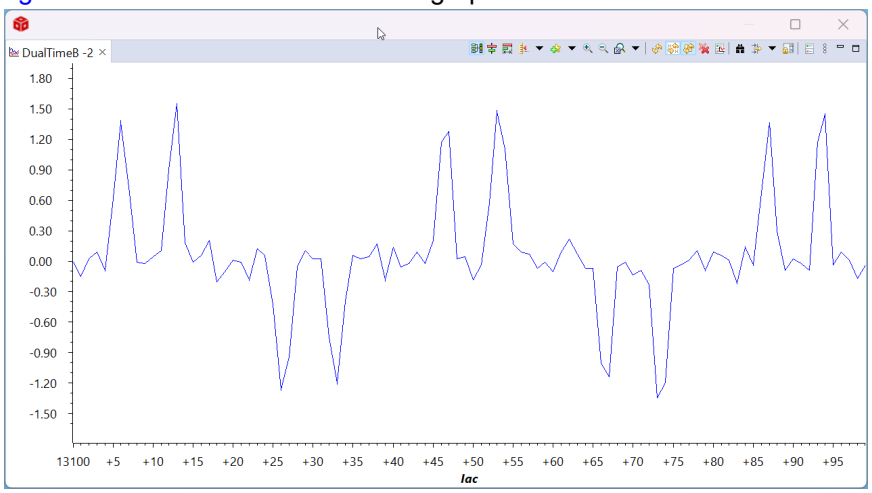
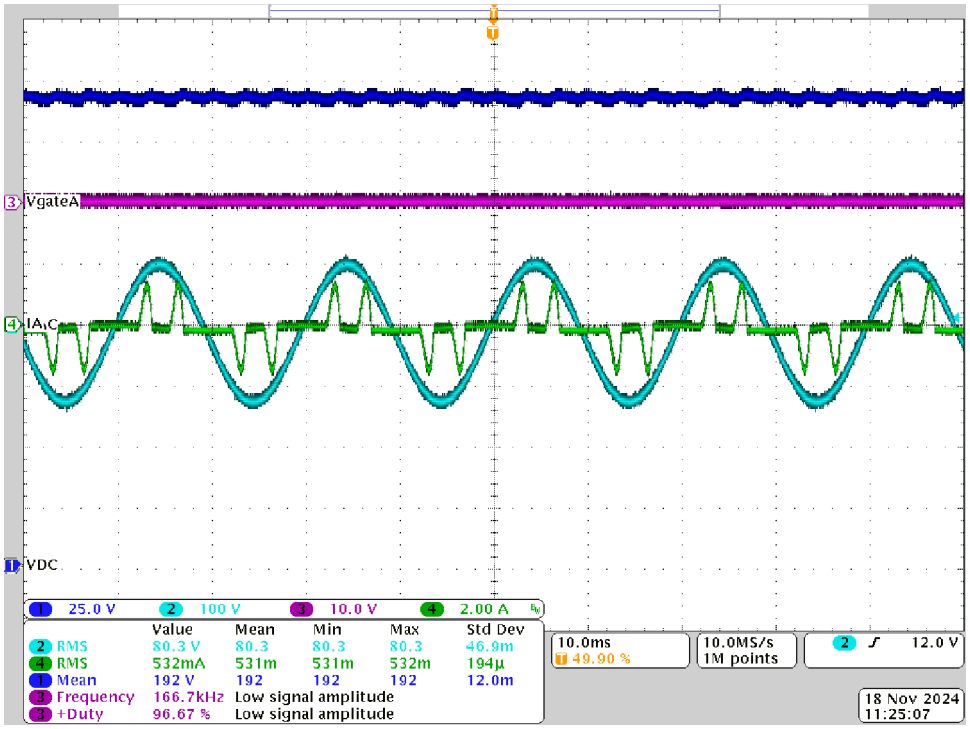


Figure 3-12. Build Level 1: Graph Showing Measured Currents

6. Figure 3-13 shows the scope capture of the input voltage and current.



- CH1 (Blue): DCBUS output voltage
- CH2 (Light blue): AC input phase A voltage
- CH3 (Pink): IGBT gate voltage
- CH4 (Green): AC Input phase A current

Figure 3-13. Build Level 1: Scope Capture Ia and Va (80V_{RMS} L-N) With PWM Tripped

7. Verify the PWM action, first reduce the input voltage to zero and wait for all the voltages to go down to zero.
8. Set the `dutyPU_DC` variable to 0.5 in the expressions view.
9. Clear the PWM trip by writing a *1* to `clearTrip`.
10. Slowly increase the input voltage and keep watching the input current. The duty cycle imparts a boost action. For example, when VAC is 80V_{RMS} without switching enabled, the `guiVbus` is about 190V; with switching, the `guiVbus` rises up to 380V. Thus, `guiVbusPM` and `guiVbusMN` are both higher than the input voltage maximum.

11. Below are the test conditions described in this build, the `guiVbus` variable rises to about 380V and `guiVbusPM` and `guiVbusMN` are close to 190V each, and the current is close to 1.1A_{RMS} when the input voltage reaches 80V_{RMS} L-N. The *Expressions* view appears as shown in step Figure 3-14. Make sure all the variables are accurate, that is, `guiVrms1`, `guiVrms2`, `guiVrms3`, `guiIrms1`, `guiIrms2`, `guiIrms3`, `guiPF1`, `guiPF2`, and `guiPF3`. If any variable is not accurate (as shown in Figure 3-14), this means there is a hardware issue with the sensing circuit.

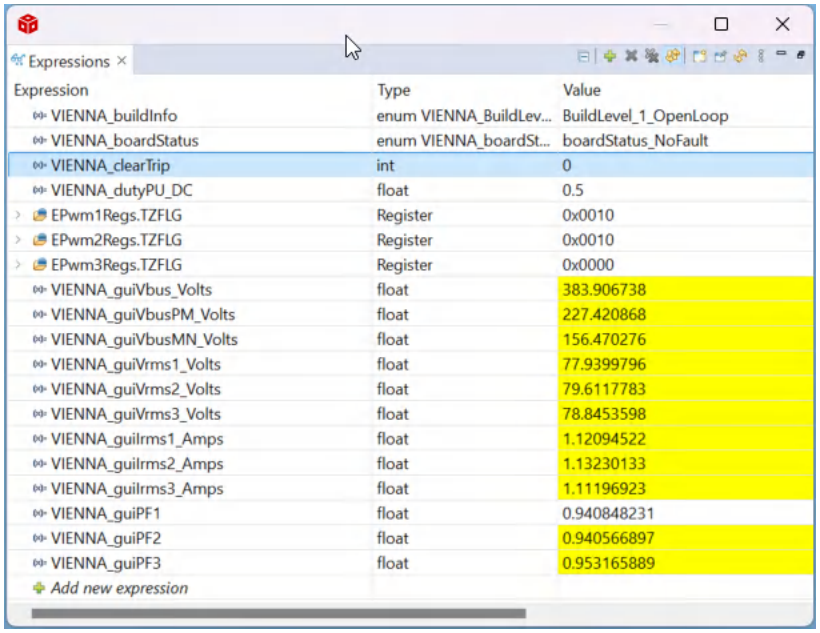
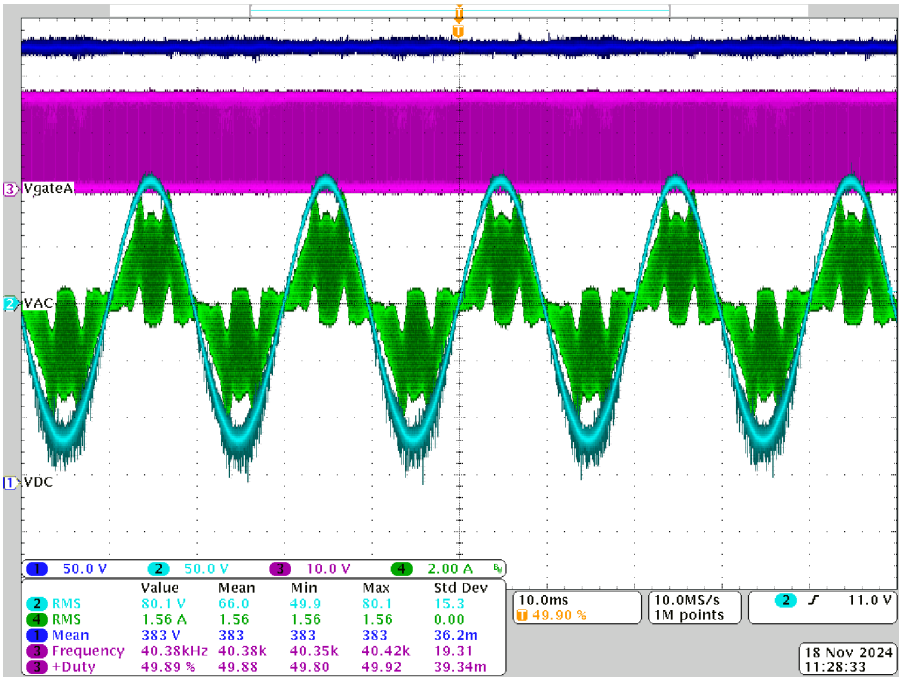


Figure 3-14. Build Level 1: Expressions View With Power Measurement

12. Figure 3-15 shows the scope capture.



- CH1 (Blue): DCBUS output voltage
- CH2 (Light blue): AC input phase A voltage
- CH3 (Pink): IGBT gate voltage
- CH4 (Green): AC Input phase A current

Figure 3-15. Build Level 1: Scope Capture Ia and Va (80V_{RMS} L-N) With Duty Cycle 0.5

13. This check verifies at a basic level the PWM driver and connection of the hardware.
14. Reduce the input voltage to zero, and watch for the bus voltages to reduce down to zero.
15. This completes the check for this build, the following items are verified on successful completion of this build:
    a. Sensing of voltages and currents and scaling to be correct
    b. Interrupt generation and execution of the build 1 code in the `controlISR` and `tenkHzISR()` variables
    c. PWM driver and switching

    If any issue is observed, carefully inspection the hardware to eliminate any build issues, and so forth.
16. The controller can now be halted, and the debug connection terminated.
17. Fully halting the MCU when in real-time mode is a two-step process. First halt the processor by using the *Halt* button on the toolbar () or by using *Target* → *Halt*. Then take the MCU out of real-time mode by clicking on the  icon. Finally, reset the MCU by clicking on the  button.
18. Close the CCS debug session by clicking on *Terminate Debug Session (Target > Terminate all)*.
    

3.2.2.4.2 INCR_BUILD 2: Closed Current Loop

In the BUILD 2 build, the inner current loop is closed, that is, the inductor current is controlled using a current compensator Gi. Both DC bus and output voltage feedforward are applied to the output of this current compensator to generate the duty cycle of the inverter, as shown in Equation 13. This action makes the plant for the current compensator simple, and a proportional (P) controller can be used to tune the loop of the inner current. The model for the current loop was derived in Section 2.2.1.2.

$$\text{duty1PU} = \frac{(\text{iL1Meas} - \text{iL1Ref}) \times \text{Gi\_GainKp} + \text{v1Meas}}{\text{v BusHalfMeas}} \tag{13}$$

Figure 3-16 illustrates the complete software diagram for this build.

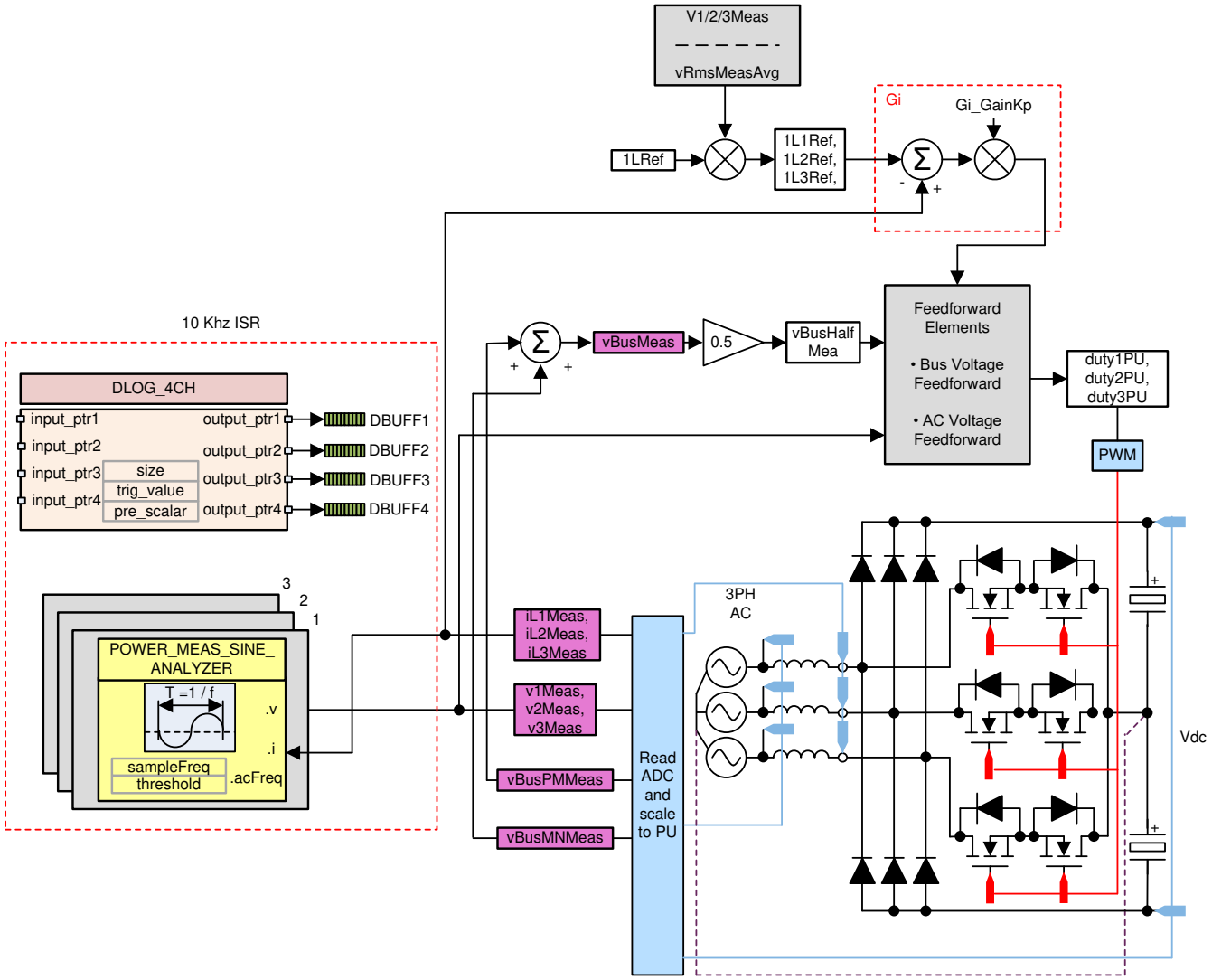
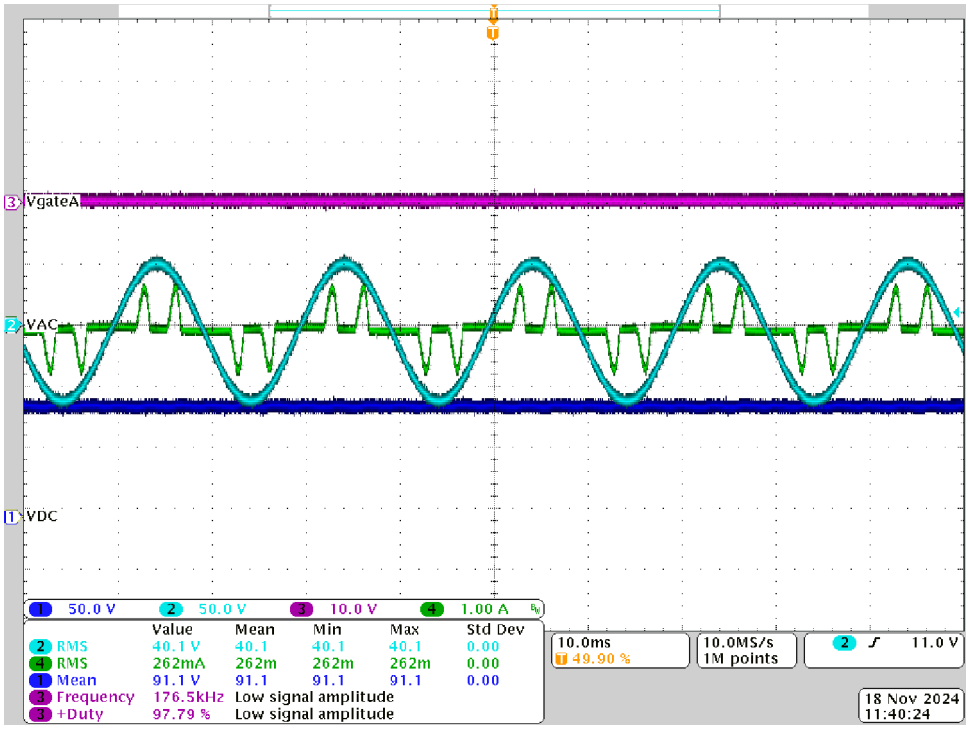


Figure 3-16. Build Level 2 Control Software Diagram: Closed Current Loop

3.2.2.4.2.1 Running Code (Build 2)

1. Run the project by clicking the  button.
2. Test first at a low voltage. Therefore, the input AC voltage is raised to only $40V_{RMS}$, 50Hz.
3. Figure 3-17 illustrates the input current and voltage waveform.

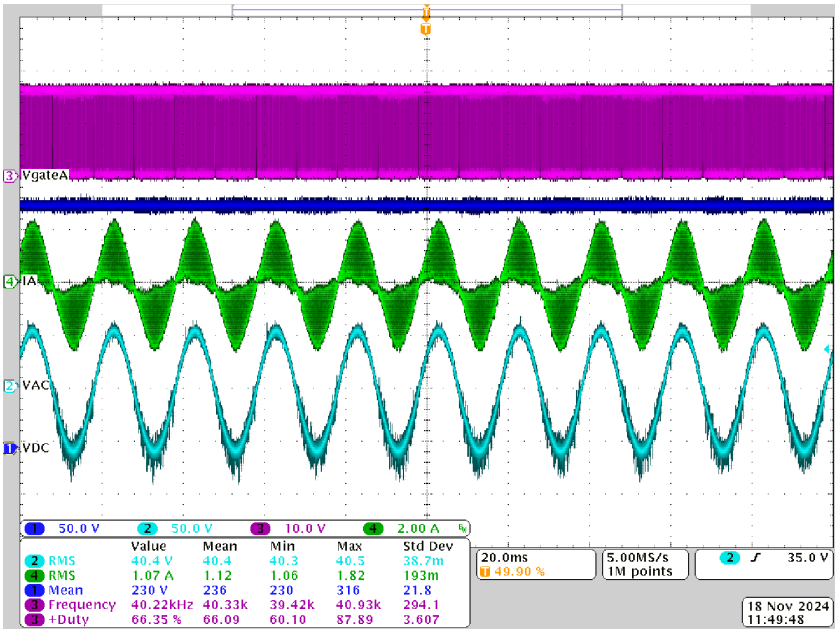


- CH1 (Blue): DCBUS output voltage
- CH2 (Light blue): AC input phase A voltage
- CH3 (Pink): IGBT gate voltage
- CH4 (Green): AC Input phase A current

Figure 3-17. Build Level 2: Scope Capture Ia and Va ($40V_{RMS}$ L-N) With PWM Tripped

4. A current reference is set by changing the `iLRef` variable in the *Expressions* view. This variable is set to 0.02.
5. Clear the trip by setting the `clearTrip` variable to *1*.

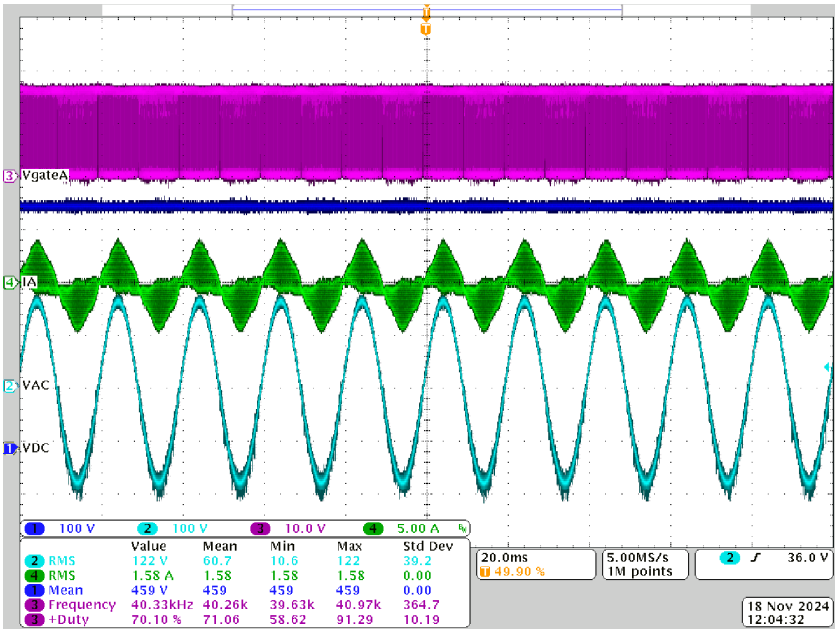
6. As soon as the trip is cleared, a sinusoidal current drawn from the input, which verifies correct operation of the current loop. Figure 3-18 shows the waveform.



- CH1 (Blue): DCBUS output voltage
- CH2 (Light blue): AC input phase A voltage
- CH3 (Pink): IGBT gate voltage
- CH4 (Green): AC Input phase A current

Figure 3-18. Build Level 2: Scope Capture Ia and Va (40V_{RMS} L-N) With `iLRef` = 0.02

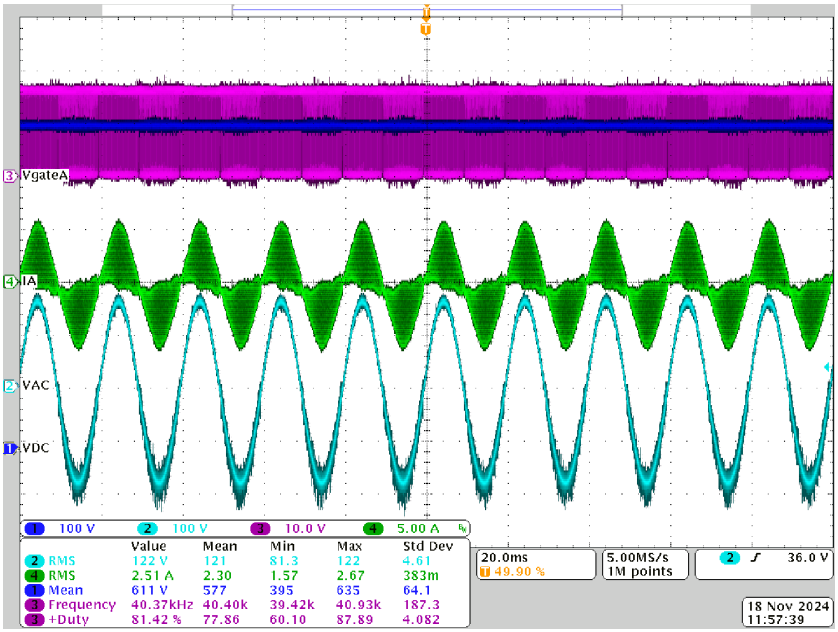
7. The `guivbus` variable is close to 230V, and the input AC current per phase is close to 1.07A.
8. Raise the input AC voltage slowly to 120V_{RMS}. The board maintains the input current to be constant as the input voltage rises. The output voltage is raised to 460V. Figure 3-19 shows what the waveforms look like.



- CH1 (Blue): DCBUS output voltage
- CH2 (Light blue): AC input phase A voltage
- CH3 (Pink): IGBT gate voltage
- CH4 (Green): AC Input phase A current

Figure 3-19. Build Level 2: Scope Capture Ia and Va (120V_{RMS} L-N) With `iLRef` = 0.02

9. Now raise the current reference `iLRef` to 0.05. Observe the bus voltage go to 610V and the input current to around 2.5A. Figure 3-20 shows the waveforms.



- CH1 (Blue): DCBUS output voltage
- CH2 (Light blue): AC input phase A voltage
- CH3 (Pink): IGBT gate voltage
- CH4 (Green): AC Input phase A current

Figure 3-20. Build Level 2: Scope Capture Ia and Va (120V_{RMS} L-N) With `iLRef` = 0.05

10. As only a proportional gain is used in the compensator, the current reference minus the feedback error is never zero. Notice the current drawn deviates slightly from the reference.
11. To bring the system to a safe stop, bring the input AC voltage down to zero, and observe that `guiVBus` comes down to zero as well.
12. Fully halting the MCU when in real-time mode is a two-step process. First halt the processor by using the *Halt* button on the toolbar ( ) or by using *Target* > *Halt*. Next take the MCU out of real-time mode by clicking on the button. Finally, reset the MCU ( ) .
13. Close the CCS debug session by clicking on *Terminate Debug Session (Target* > *Terminate all)*.

3.2.2.4.2.2 Building and Loading the Project and Setting Up Debug

1. Open the `vienna_settings.h` file, set VIENNA_INCR_BUILD to 2.
2. Right click on the project name, and click *Rebuild Project*. The project builds successfully. Click *Run* → *Debug*, which launches a debugging session. In the case of dual CPU devices, if a window appears indicating a CPU must be selected, then perform a debug. In this case, select CPU1. The project then loads on the device, and CCS debug view becomes active. The code halts at the start of the main routine.
3. To add the variables in the *Watch* or *Expressions* window, click *View* → *Scripting Console* to open the scripting console dialog box. On the upper right corner of this console, click on *Open* to browse to the `setupdebugenv_build2.js` script file, which is located inside the project folder. This file populates the *Watch* window with the appropriate variables needed to debug the system. Click the *Continuous Refresh* button () on the *Watch* window to enable continuous update of values from the controller. Figure 3-21 illustrates the *Watch* window.

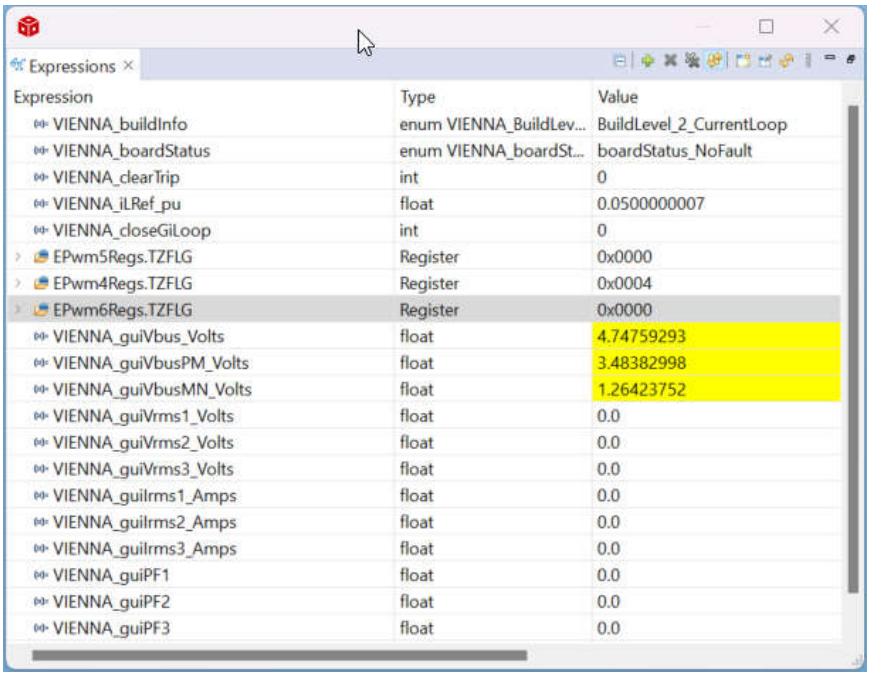


Figure 3-21. Build Level 2: Closed Current Loop Expressions View

4. Enable real-time mode by hovering the mouse on the buttons on the horizontal toolbar, and clicking the  button.

3.2.2.4.3 INCR_BUILD 3: Closed Voltage and Current Loop

In this build, the outer voltage loop is closed with the inner current loop closed (designed in BUILD 2). Figure 3-22 shows the software diagram for this build.

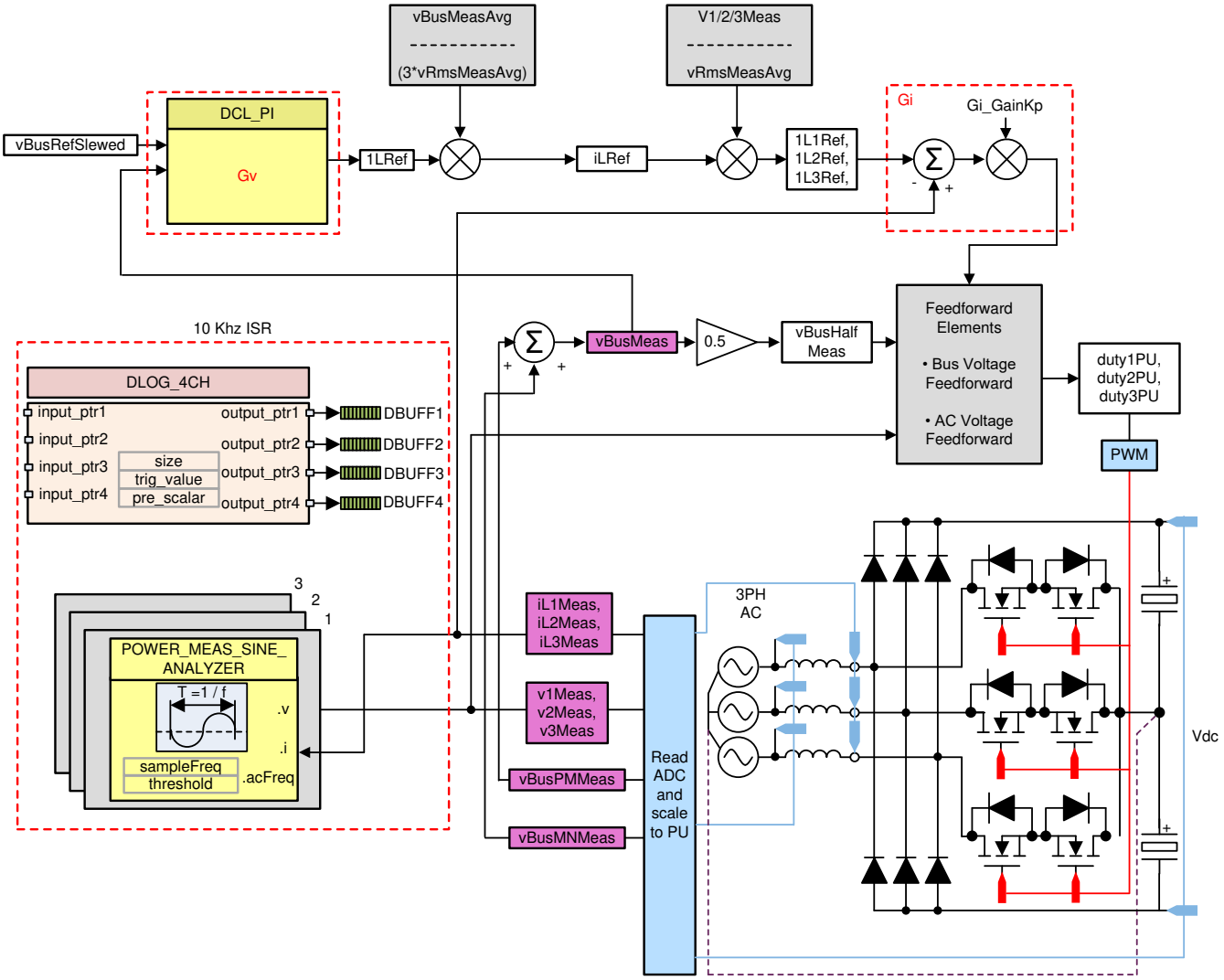


Figure 3-22. Build Level 3 Control Diagram: Output Voltage Control With Inner Current Loop

3.2.2.4.3.1 Building and Loading the Project and Setting Up Debug

1. Open the `vienna_settings.h` file, set VIENNA_INCR_BUILD to 3.
2. Right click on the project name, and click *Rebuild Project*. The project builds successfully. Click *Run* → *Debug*, which launches a debugging session. In the case of dual CPU devices, if a window appears indicating a CPU must be selected, then perform a debug. In this case, select CPU1. The project then loads on the device, and CCS debug view becomes active. The code halts at the start of the main routine.
3. To add the variables in the *Watch* or *Expressions* window, click *View* → *Scripting Console* to open the scripting console dialog box. On the upper right corner of this console, click on *Open* to browse to the *setupdebugenv_build3.js* script file located inside the project folder. This file populates the *Watch* window with appropriate variables needed to debug the system. Click on the *Continuous Refresh* button ( ) on the *Watch* window to enable continuous update of values from the controller.
   Figure 3-23 illustrates the *Watch* window.

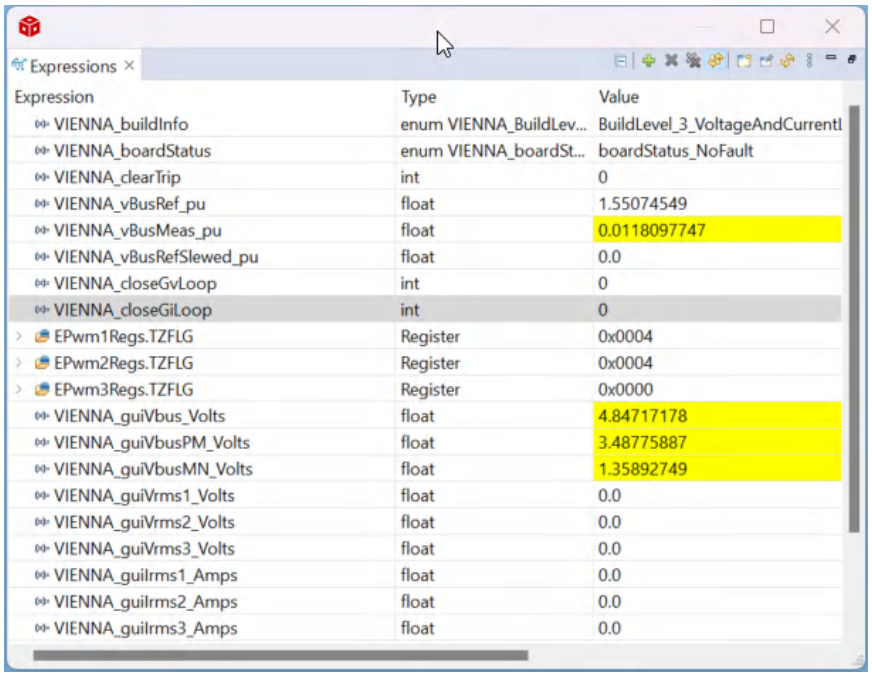


Figure 3-23. Build Level 3: Expressions View

4. Enable real-time mode by hovering the mouse on the buttons on the horizontal toolbar and clicking the button.

3.2.2.4.3.2 Running Code (Build 3)

1. Run the project by clicking the  button.
2. Raise the input AC voltage to $120V_{RMS}$ VL-N or $208V_{RMS}$ VL-L, 50Hz.
3. The DC voltage reference is set by the `vBusRef` variable. This value is set to as 1.0, which corresponds to 420V for this design.
4. Clear the trip by setting the `clearTrip` variable to *1*. The bus voltage then rises to be 420V.
5. Closed loop operation can be verified by comparing the `vBusRef` and `vBusMeas` in the expressions window as Figure 3-24 shows.

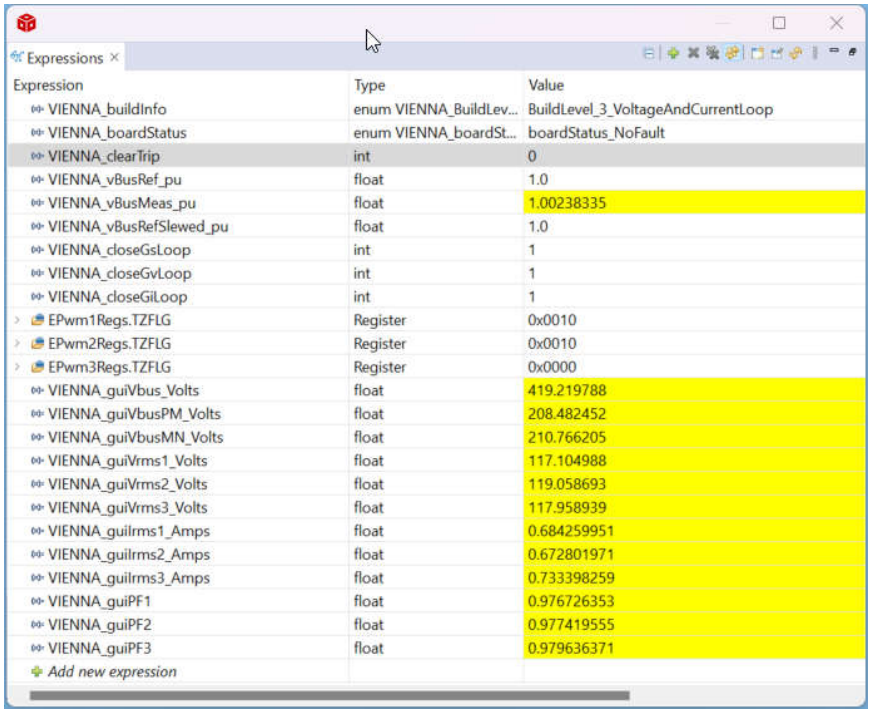
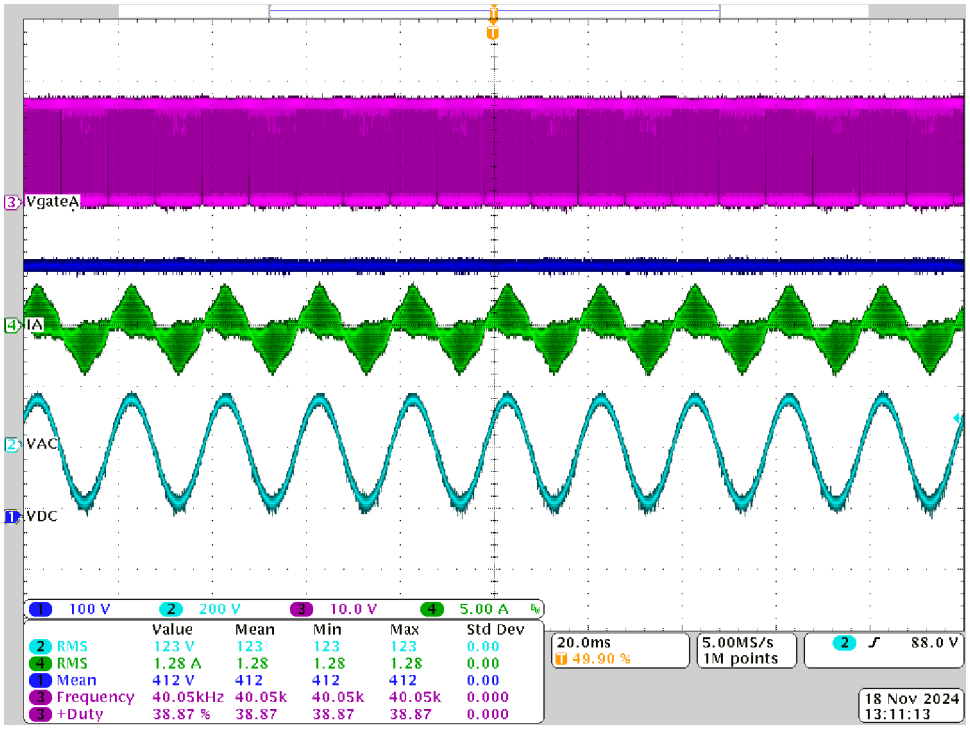


Figure 3-24. Build Level 3: Expressions Window With Vref = 1.0

6. Figure 3-25 shows the input voltage and current waveforms.



- CH1 (Blue): DCBUS output voltage
- CH2 (Light blue): AC input phase A voltage
- CH3 (Pink): IGBT gate voltage
- CH4 (Green): AC Input phase A current

Figure 3-25. Build Level 3: Scope Capture Ia and Va ($120V_{RMS}$ L-N) With Vref = 1.0

7. Now raise `vBusRef` to 1.5 step by step, the bus voltage is raised to 630V, the `vBusRef` and `vBusMeas` variables appear in the *Expressions* window as Figure 3-26 shows.

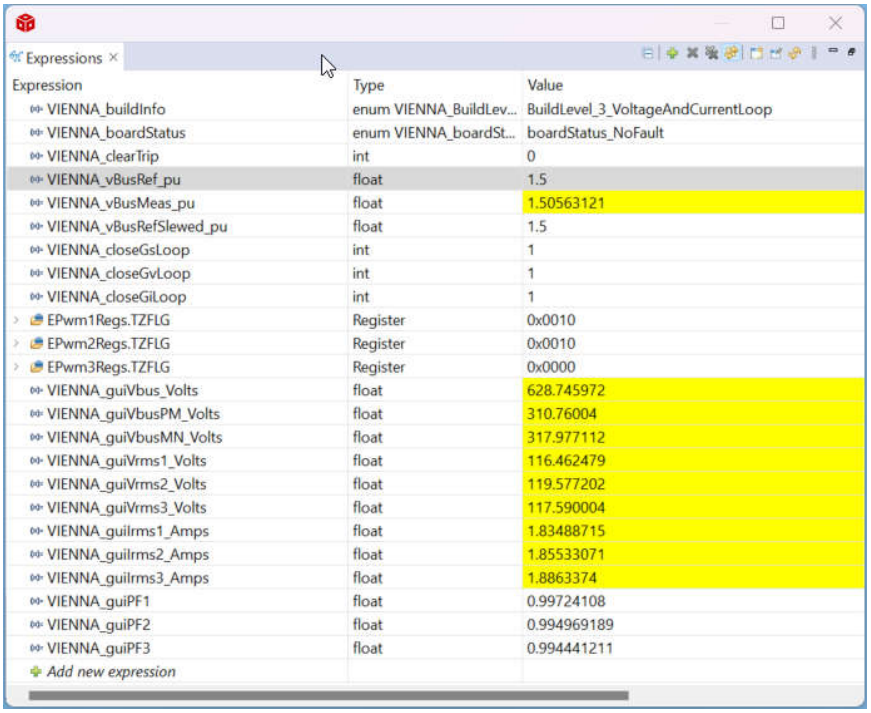
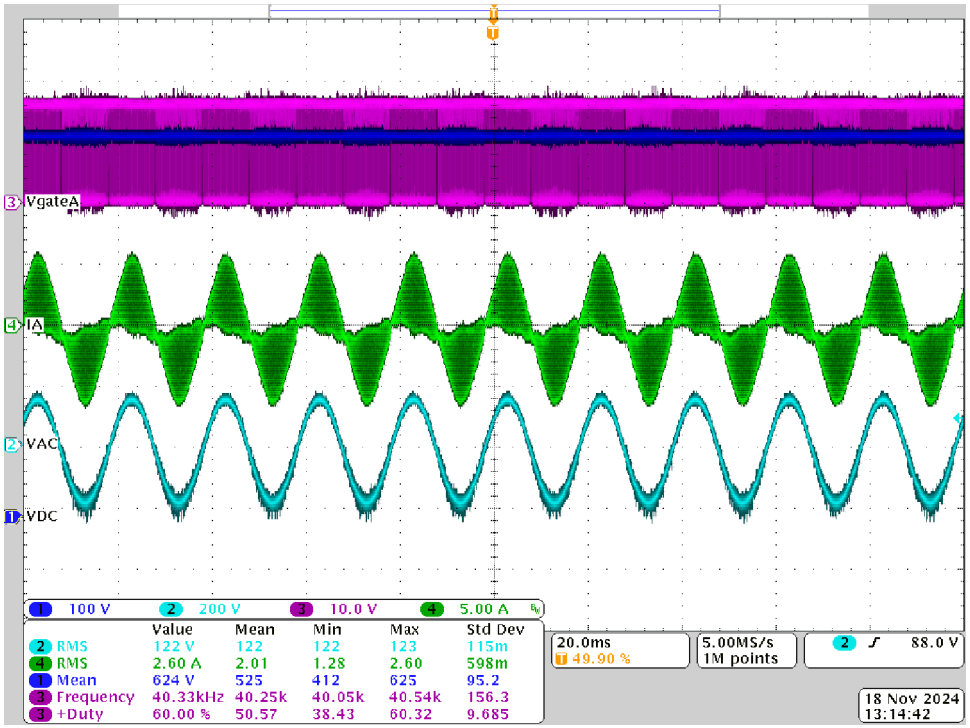


Figure 3-26. Build Level 3: Expressions Window With Vref = 1.5

Figure 3-25 shows the input voltage and current waveforms.



- CH1 (Blue): DCBUS output voltage
- CH2 (Light blue): AC input phase A voltage
- CH3 (Pink): IGBT gate voltage
- CH4 (Green): AC Input phase A current

Figure 3-27. Build Level 3: Scope Capture Ia and Va(120V_{RMS} L-N) With Vref=1.5

8. To bring the system to a safe stop, bring the input AC voltage down to zero, observe the `guiVBus` variable comes down to zero as well.
9. Fully halting the MCU when in real-time mode is a two-step process. First halt the processor by using the *Halt* button on the toolbar ( ) or by using *Target* > *Halt*. Next, take the MCU out of real-time mode by clicking on the board. Finally, reset the MCU ( ).
10. Close the CCS debug session by clicking on *Terminate Debug Session (Target* > *Terminate all)*.

3.2.2.4.4 INCR_BUILD 4: Closed Balance, Voltage, and Current Loop

In this build, the board is operated as a three-wire system, that is, the neutral of the power supply is not connected to the DC midpoint of the output. As Figure 3-28 illustrates, to maintain the DC bus balance, a balance loop with a simple proportional gain is added in the control structure. In this build a third harmonic injection is also carried out, which helps in stabilizing the DC bus balance point.

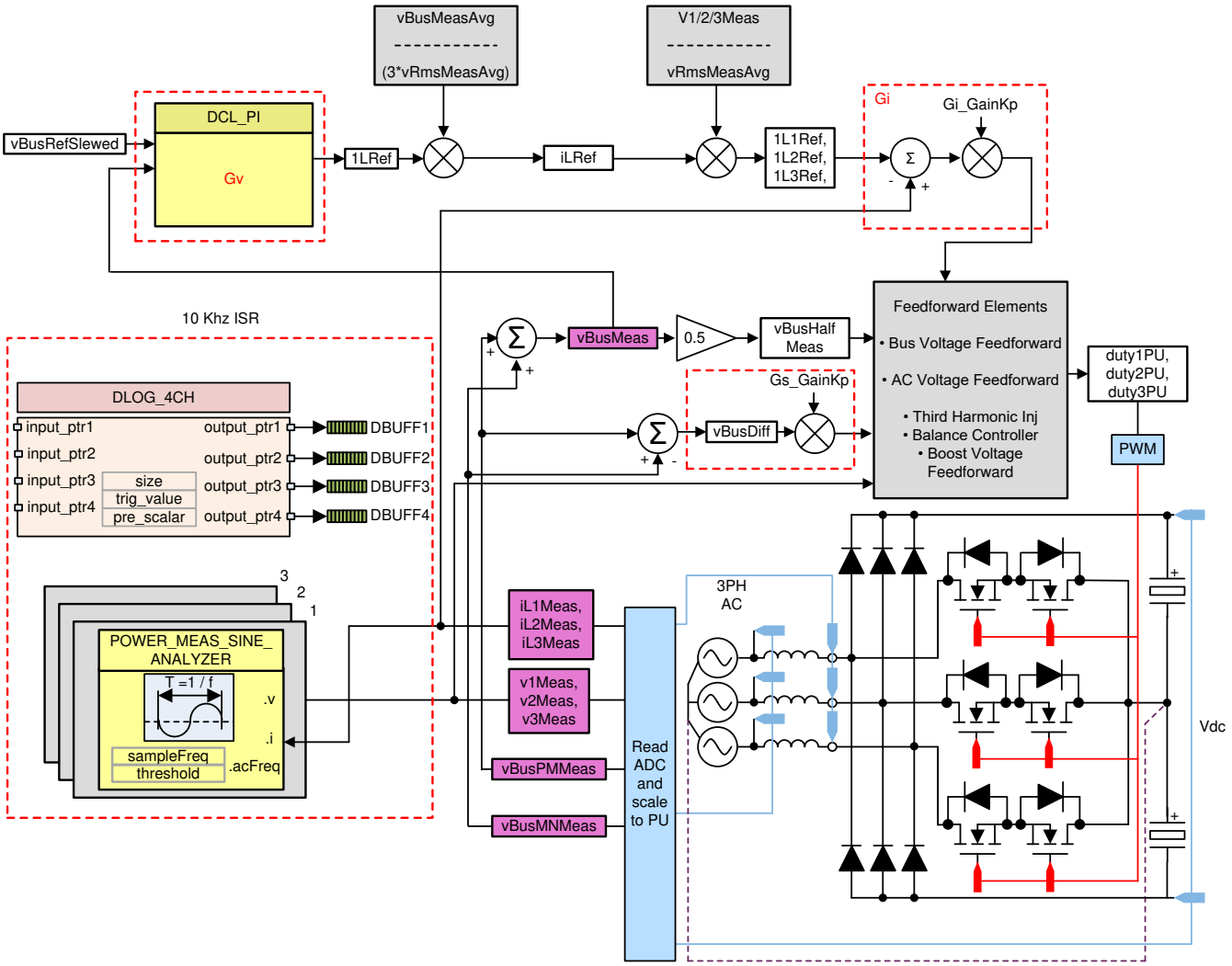


Figure 3-28. Build Level 4 Control Diagram: Output Voltage, Inductor Current, and Bus Cap Balance Loop

3.2.2.4.4.1 Building and Loading the Project and Setting Up Debug

1. Open the `vienna_settings.h` file, set VIENNA_INCR_BUILD to 4, and set VIENNA_VBUS_REF_SET_VOLTS to 650.
2. Right click the project name and click *Rebuild Project*. The project builds successfully. Click *Run* → *Debug*; this launches a debugging session. In the case of dual CPU devices, if a window appears indicating a CPU must be selected, then perform a debug. In this case, select CPU1. The project then loads on the device and CCS debug view becomes active. The code halts at the start of the main routine.
3. To add the variables in the *Watch* or *Expressions* window click *View* → *Scripting Console* to open the scripting console dialog box. On the upper-right corner of this console, click open to browse to the `setupdebugenv_build4.js` script file located inside the project folder. This file populates the *Watch* window with appropriate variables needed to debug the system. Click the *Continuous Refresh* button ( ) on the *Watch* window to enable continuous update of values from the controller. Figure 3-29 illustrates the *Watch* window.

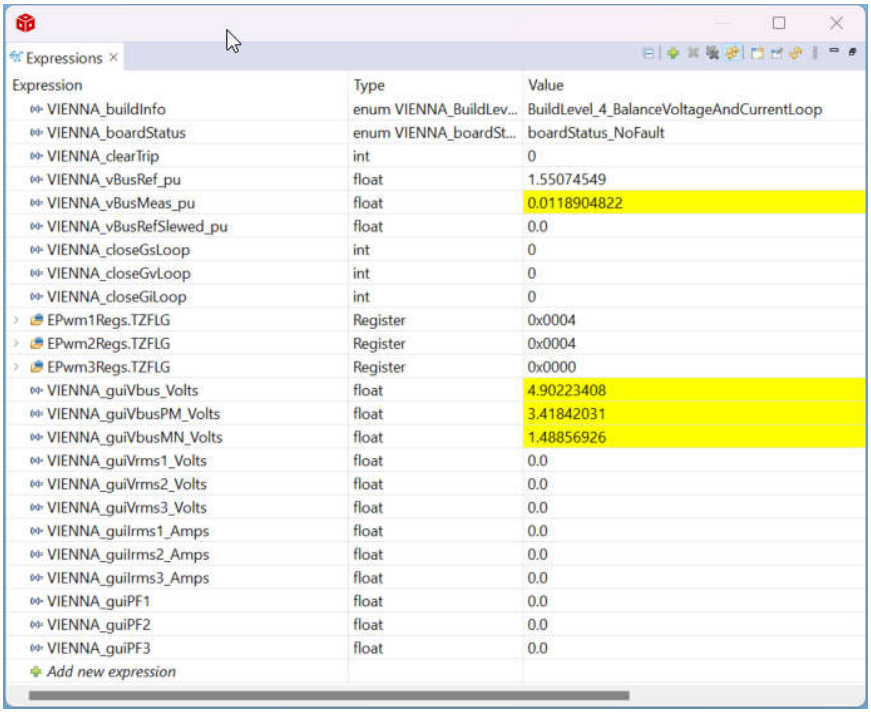
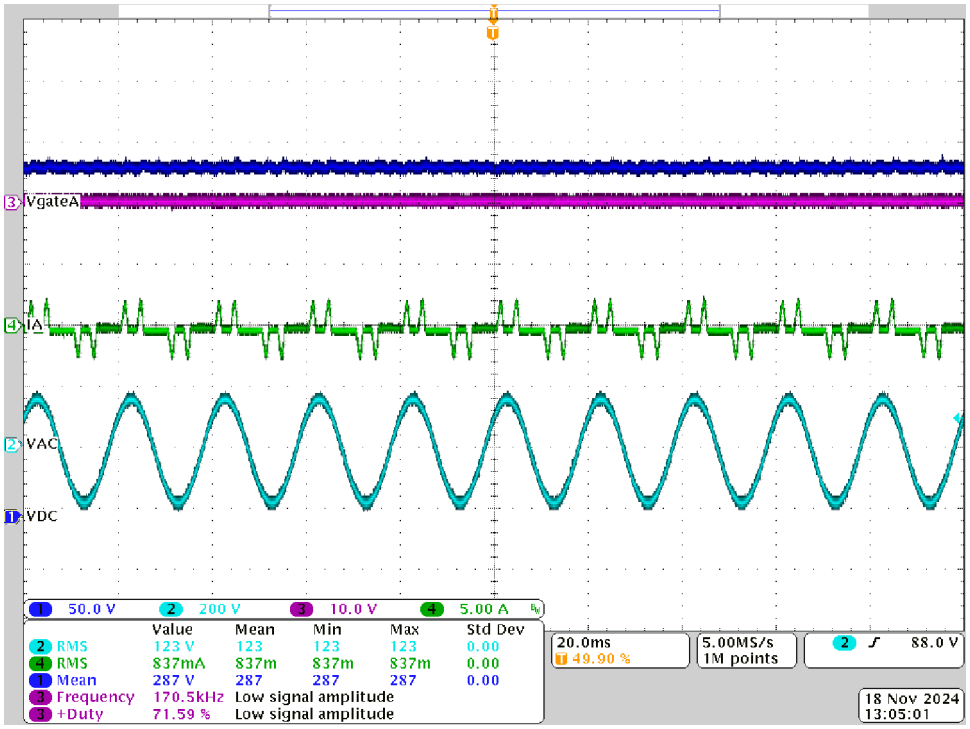


Figure 3-29. Build Level 4: Expressions View

4. Enable real-time mode by hovering the mouse on the buttons on the horizontal toolbar and clicking the button.

3.2.2.4.4.2 Running Code (Build 4)

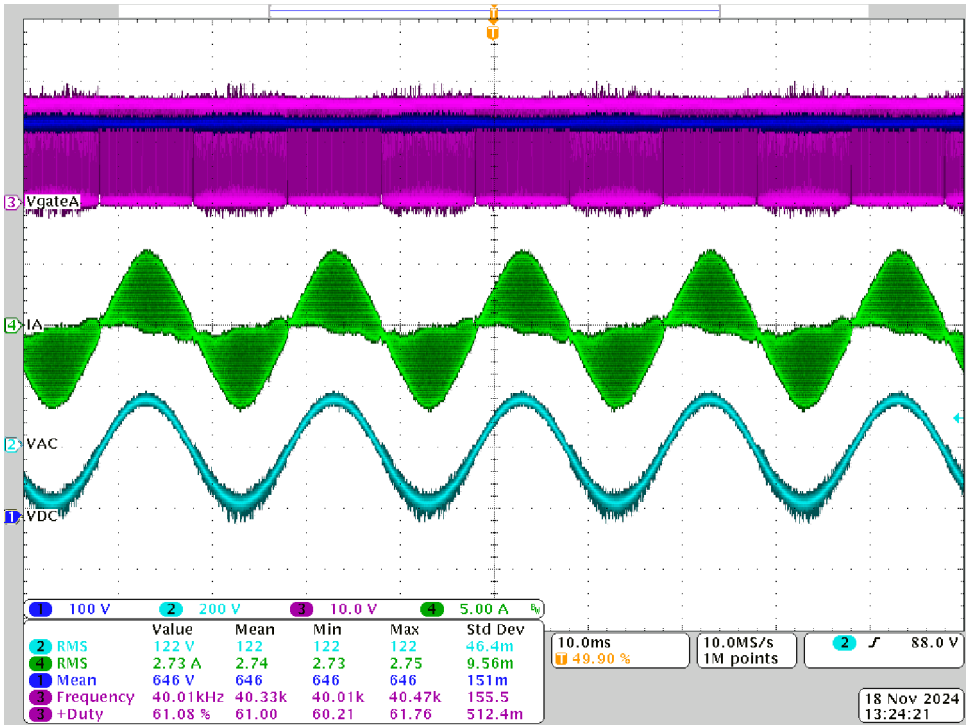
1. Run project by clicking the  button.
2. Raise the AC input to 120V_{RMS} V_{L-L} and 208V_{RMS} V_{L-L}, 50/60Hz. Figure 3-30 shows a rectified current that is going to be drawn from the input.



- CH1 (Blue): DCBUS output voltage
- CH2 (Light blue): AC input phase A voltage
- CH3 (Pink): IGBT gate voltage
- CH4 (Green): AC Input phase A current

Figure 3-30. Build Level 4: Scope Capture Ia and Va (120V_{RMS} L-N) With PWM Tripped

3. Bus voltage is set by the `vBusRef` variable, and is about 1.55V already, which corresponds to 650V for this design.
4. Start the PFC action by writing a *1* to the `clearTrip` variable.
5. The board now draws the sinusoidal current. Figure 3-31 shows the scope capture.



- CH1 (Blue): DCBUS output voltage
- CH2 (Light blue): AC input phase A voltage
- CH3 (Pink): IGBT gate voltage
- CH4 (Green): AC Input phase A current

Figure 3-31. Build Level 4: Scope Capture Ia and Va ($120V_{RMS}$ L-N) With Full PFC

6. Check the *Expressions* window shown in Figure 3-32. The DC bus voltages is also balanced, that is, the `guiVbusPM` and `guiVbusMN` variables are almost equal, which shows that the closed loop balance controller is working.

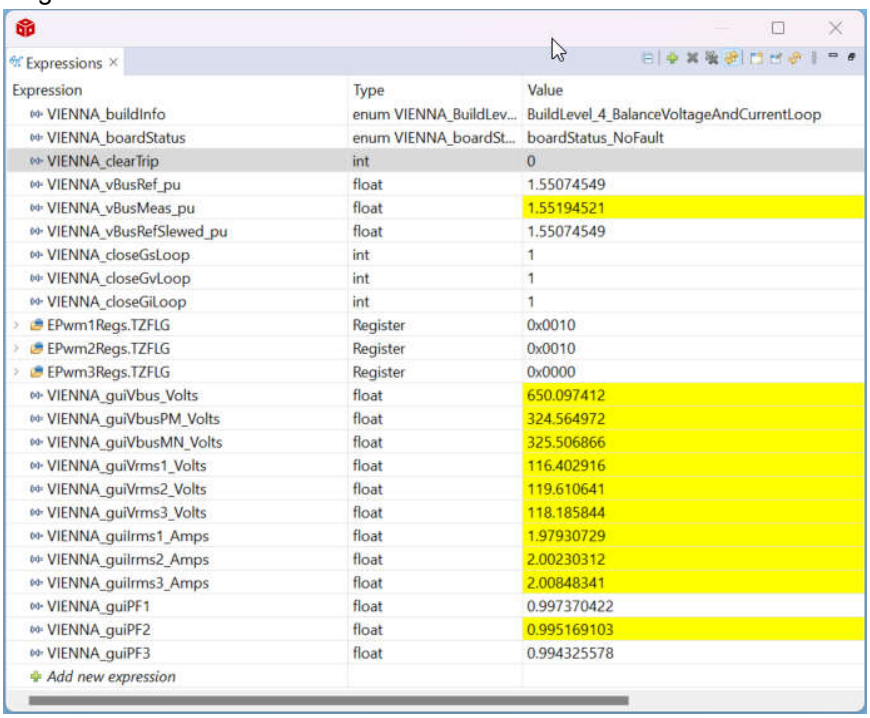


Figure 3-32. Build Level 4: Expressions Window With 120VAC and 650VDC

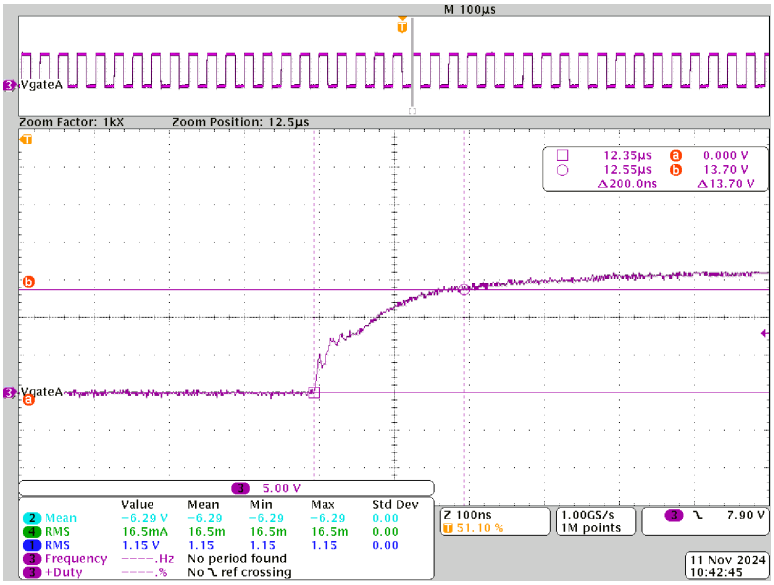
7. The balance loop open loop gain is controlled by the `Gs_GainKp` variable and can be adjusted in case the bandwidth is not enough. Though, for the balance loop, the bandwidth needs to be lower than the outer voltage loop and only 1Hz to 2Hz of bandwidth is sufficient.
8. To bring the system to a safe stop bring the input AC voltage down to zero, observe the `guiVBus` variable comes down to zero as well.
9. Fully halting the MCU when in real-time mode is a two-step process. First, halt the processor by using the *Halt* button on the toolbar () or by using *Target* > *Halt*. Then take the MCU out of real-time mode by clicking on the  button. Finally reset the MCU by selecting the  button.
10. Close the CCS debug session by clicking on *Terminate Debug Session (Target > Terminate all)*.  .

3.3 Test Results

Section 3.3.1 through Section 3.3.11 show the test results including current and voltage waveforms, and thermal performance.

3.3.1 IGBT Gate Rising and Falling Time

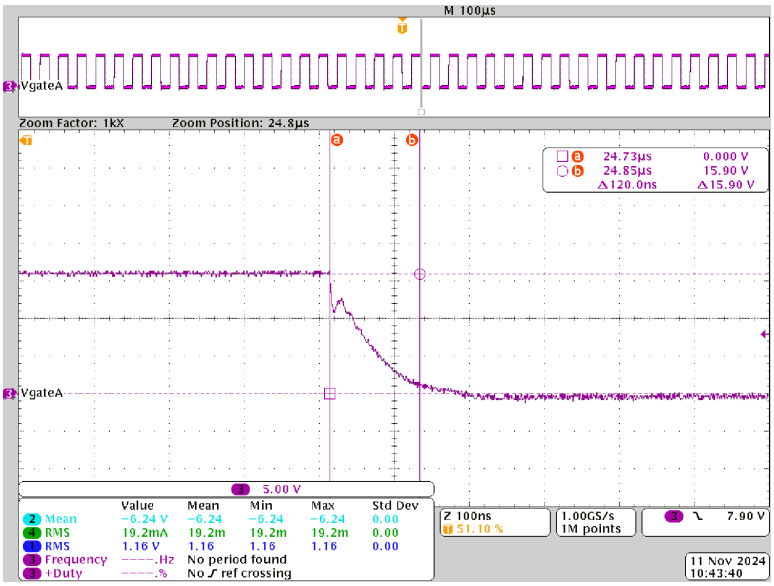
UCC5350MC is a 10A gate driver, this device drives the IGBT gate very fast, which can help to improve the IGBT power loss. Figure 3-33 shows 200ns rising time and 120ns falling time of IGBT gate voltage.



- CH3 (Pink): IGBT gate voltage

Figure 3-33. IGBT Gate Rising Edge

Figure 3-34 shows approximately 120ns of IGBT gate voltage falling time.

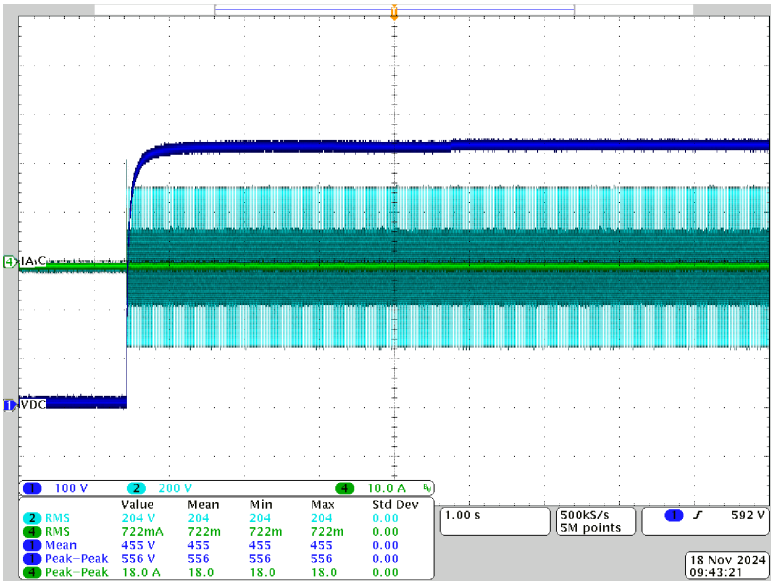


- CH3 (Pink): IGBT gate voltage

Figure 3-34. IGBT Gate Falling Edge

3.3.2 Power On Sequence

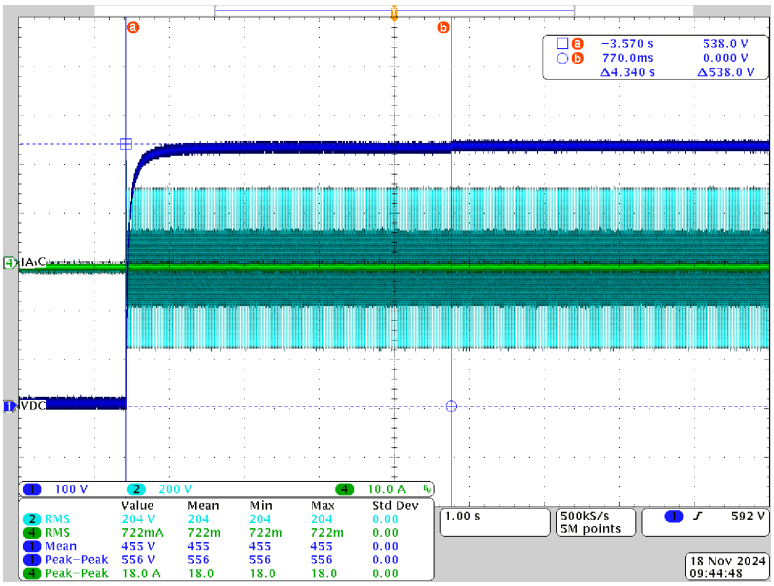
After power on, the charging current of the electrolytic capacitors (C15, C16, C24, and C25) is limited by PTC RT1. Once the DCBUS voltage is high enough, auxiliary power supply and isolated power supply start to work to provide low-voltage rails for the chips to work. The LED also starts to blink, and the relay is engaged after a delay. Make sure the DC load is less then 800Ω when the board is powered on, or else RT1 is triggered by a large inrush current. Figure 3-35 shows the waveform of the board powering on under 380VAC.



- CH1 (Blue): DCBUS output voltage
- CH2 (Light blue): AC input phase A voltage
- CH4 (Green): AC Input phase A current

Figure 3-35. Power On Sequence

Total power on time including DCBUS electrolytic capacitor charging time, auxiliary power supply start time, MCU reset, relay close delay. Figure 3-36 shows about 4 seconds power-on time under 380VAC input.

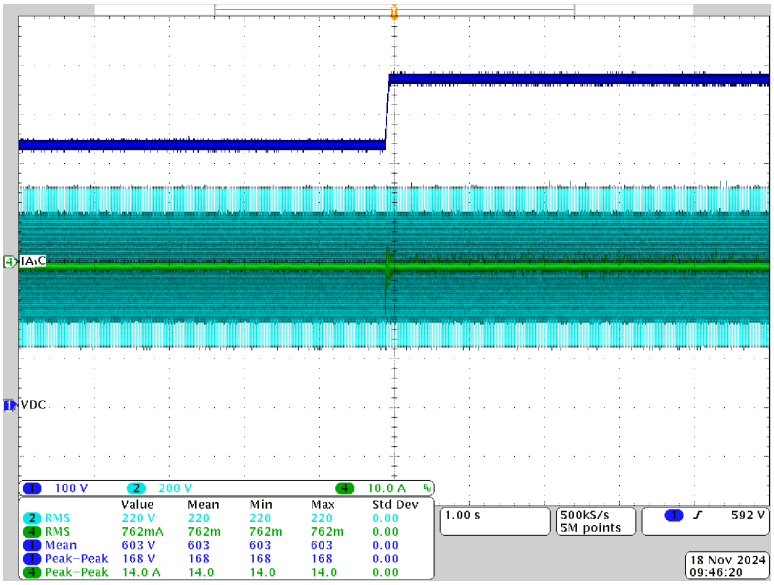


- CH1 (Blue): DCBUS output voltage
- CH2 (Light blue): AC input phase A voltage
- CH4 (Green): AC Input phase A current

Figure 3-36. Power On Time

3.3.3 PFC Started by GUI

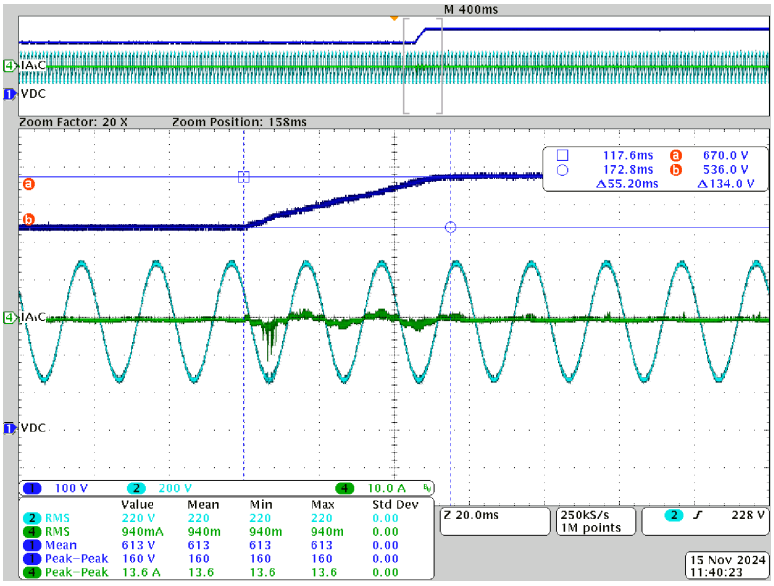
After power on, the DCBUS voltage is about 530VDC, once PFC starts, the DCBUS voltage is about 680VDC. Figure 3-37 shows the waveform when PFC is started with the GUI command 0x11.



- CH1 (Blue): DCBUS output voltage
- CH2 (Light blue): AC input phase A voltage
- CH4 (Green): AC Input phase A current

Figure 3-37. PFC Started by GUI

DCBUS rise from about 530V to about 680VDC under no load when PFC starts. Figure 3-38 shows the rising time is approximately 55ms.

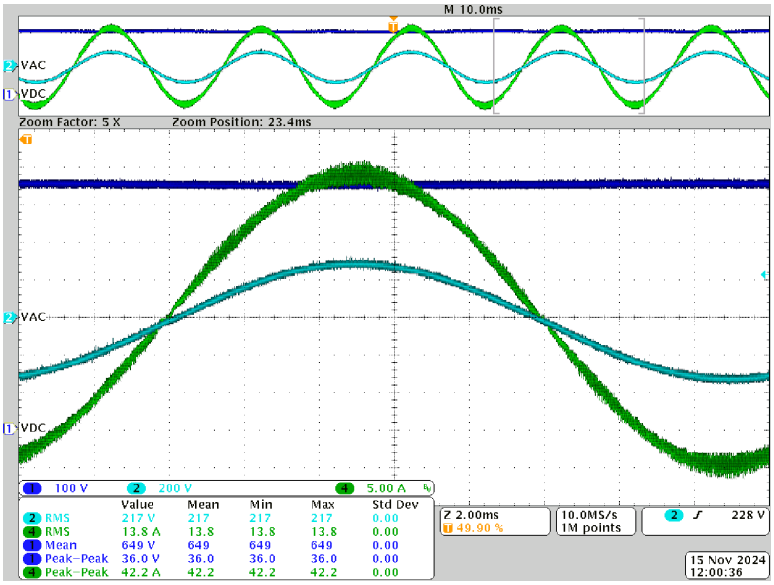


- CH1 (Blue): DCBUS output voltage
- CH2 (Light blue): AC input phase A voltage
- CH4 (Green): AC Input phase A current

Figure 3-38. DCBUS Rising Time When PFC Starts

3.3.4 Zero Crossing Under 380VAC, 9kW

Figure 3-39 shows the zero crossing time matches each other for the AC input voltage and current.

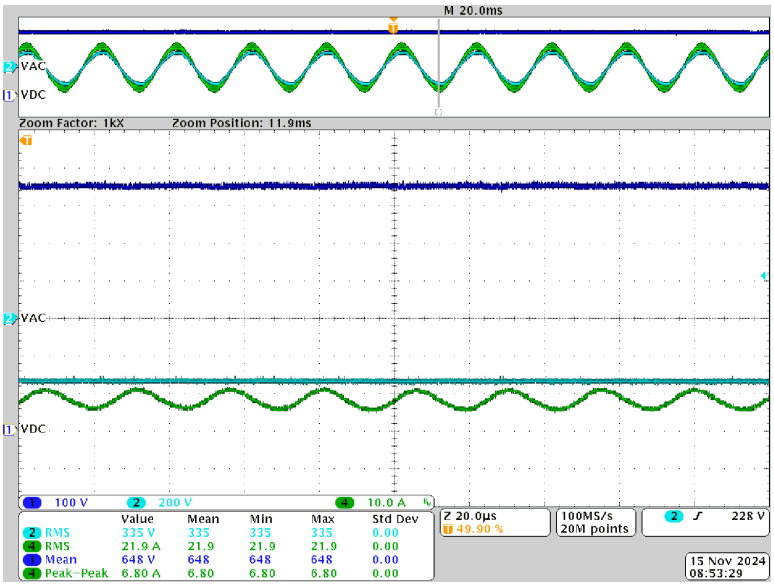


- CH1 (Blue): DCBUS output voltage
- CH2 (Light blue): AC input phase A voltage
- CH4 (Green): AC Input phase A current

Figure 3-39. Zero Crossing Time Under 380VAC, 9kW

3.3.5 Current Ripple Under 380VAC,10kW

Figure 3-40 shows the 6.8A boost inductor current ripple under 380VAC, 10kW, which is about ±15.5% of the peak AC input current 21.9A.

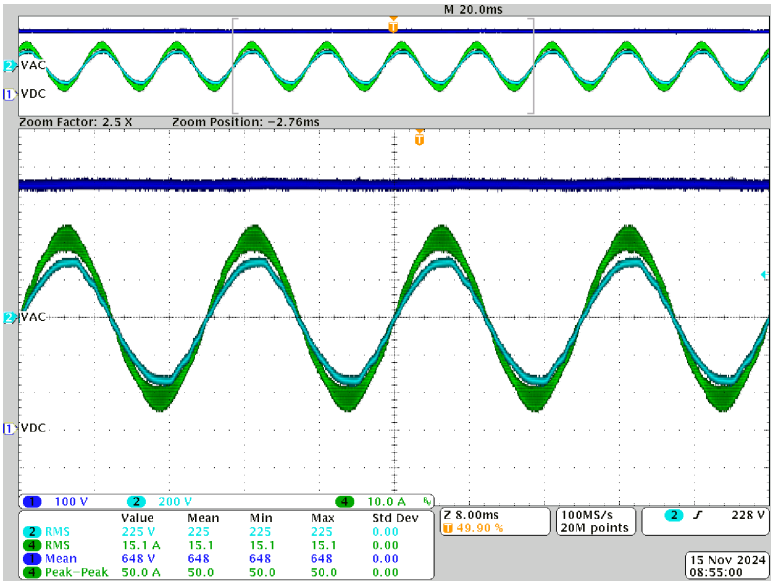


- CH1 (Blue): DCBUS output voltage
- CH2 (Light blue): AC input phase A voltage
- CH4 (Green): AC Input phase A current

Figure 3-40. Current Ripple Under 380VAC, 10kW

3.3.6 10kW Load Test With Grid Power

Figure 3-41 shows load test waveforms under 10kW with grid power.

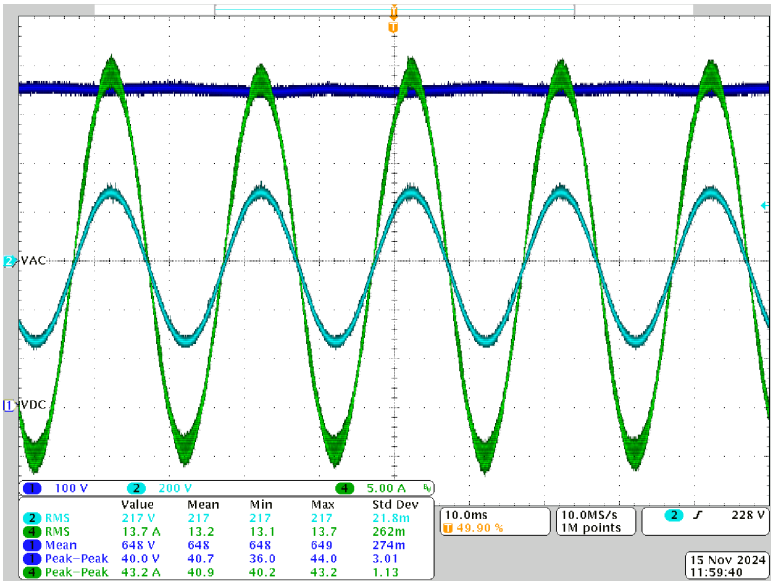


- CH1 (Blue): DCBUS output voltage
- CH2 (Light blue): AC input phase A voltage
- CH4 (Green): AC Input phase A current

Figure 3-41. 10kW Load Test With Grid Power

3.3.7 9kW Load Test With AC Power Source

Figure 3-42 shows the 9kW load test with an AC power source.



- CH1 (Blue): DCBUS output voltage
- CH2 (Light blue): AC input phase A voltage
- CH4 (Green): AC Input phase A current

Figure 3-42. 9kW Load Test With AC Power Source

3.3.8 Power Analyzer Results

Figure 3-43 shows the power analyzer results under 10kW with grid power.

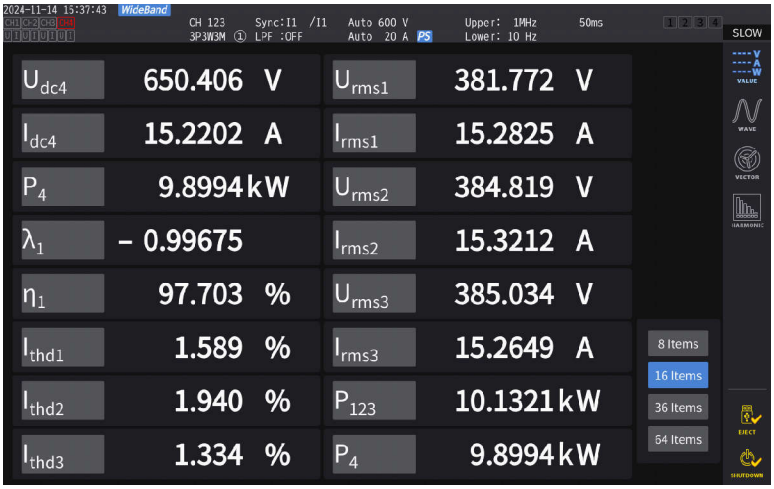


Figure 3-43. Power Analyzer Result Under 10kW Grid Power

Figure 3-44 shows power analyzer results under 9kW with an AC power source.

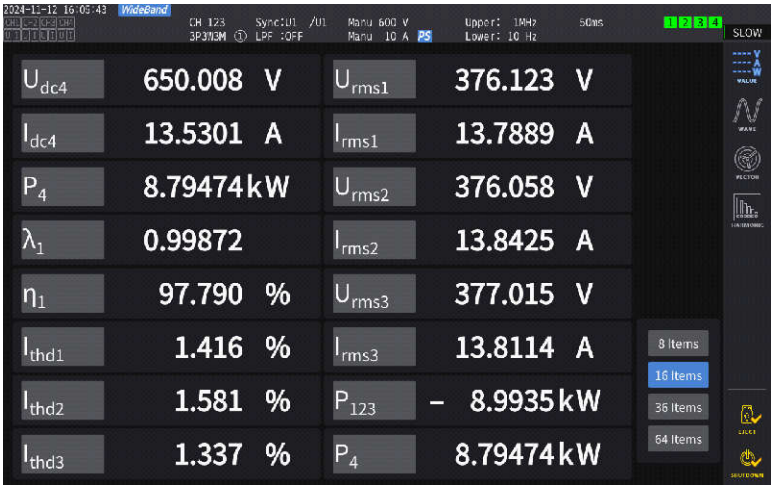


Figure 3-44. Power Analyzer Result Under 9kW AC Power Sources

3.3.9 Thermal Performance

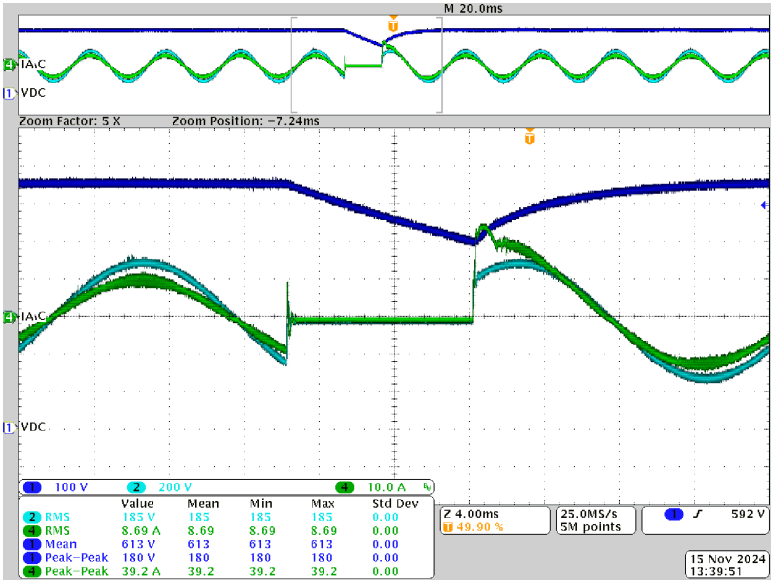
Figure 3-45 shows the thermal performance under 10kW load and 25°C ambient temperature in a open area without a cooling fan. The highest temperature rising is about 55°C of the MOSFET Q8. For auxiliary power supplies, the boost inductor temperature rising is about 23°C, the IGBT temperature rising is approximately 40°C.

The temperature of power devices and the heat sink temperature can also be read from the GUI software shown in Figure 3-3.

Figure 3-45. Thermal Performance

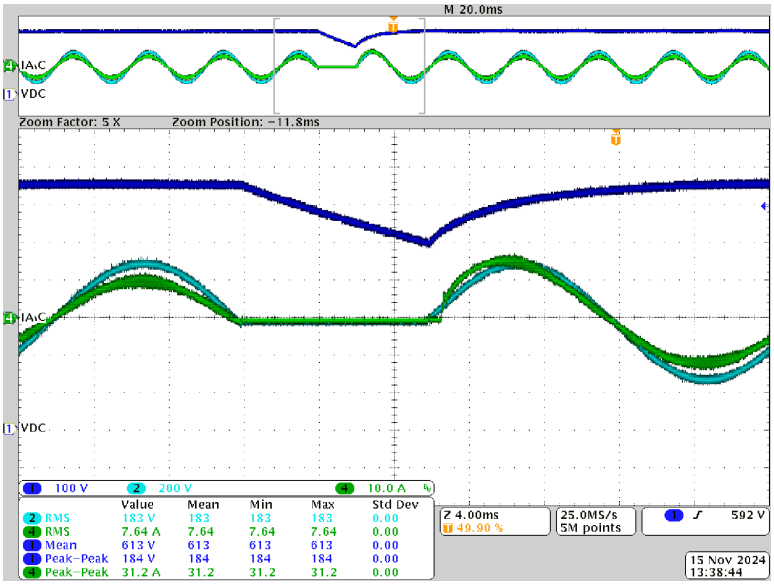
3.3.10 Voltage Short Interrupt Test

Figure 3-46 , Figure 3-47, and Figure 3-48 show voltage short interrupt tests under 5kW at 90, 0, and 45 phase angles. The current on boost inductors is monitored by DSP internal CMPSS, if this current is too big, IGBT is turned off on each PWM cycle. This is the cycle-by-cycle protection. However, DSP cannot protect current on diode bridges, since there is still uncontrolled rectified current with diode bridges when IGBTs are off.



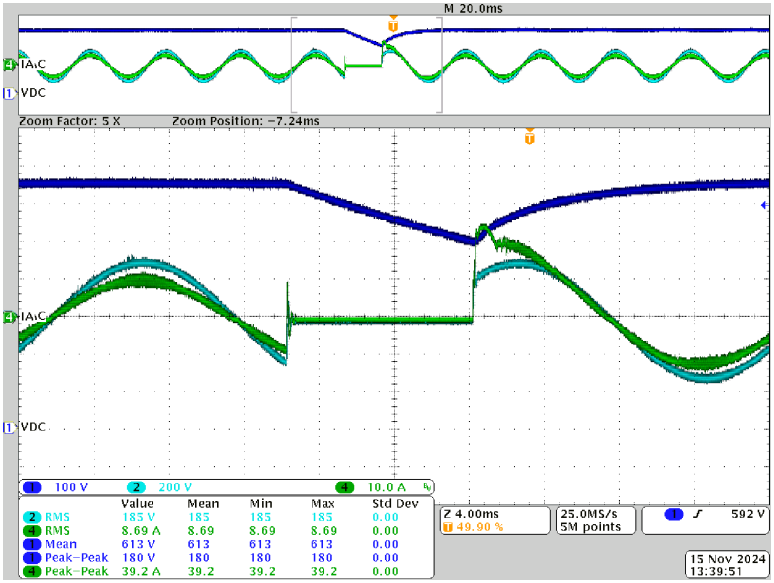
- CH1 (Blue): DCBUS output voltage
- CH2 (Light blue): AC input phase A voltage
- CH4 (Green): AC Input phase A current

Figure 3-46. Voltage Short Interrupt Test at 90 Phase Angle



- CH1 (Blue): DCBUS output voltage
- CH2 (Light blue): AC input phase A voltage
- CH4 (Green): AC Input phase A current

Figure 3-47. Voltage Short Interrupt Test at 0 Phase Angle



- CH1 (Blue): DCBUS output voltage
- CH2 (Light blue): AC input phase A voltage
- CH4 (Green): AC Input phase A current

Figure 3-48. Voltage Short Interrupt Test at 45 Phase Angle

3.3.11 Efficiency, iTHD, and Power Factor Results

Figure 3-49 shows average efficiency test results at each power level under 325VAC, 380VAC, and 400VAC. Average peak efficiency is 98% across 3kW–4.5kW under 380VAC and 400VAC.

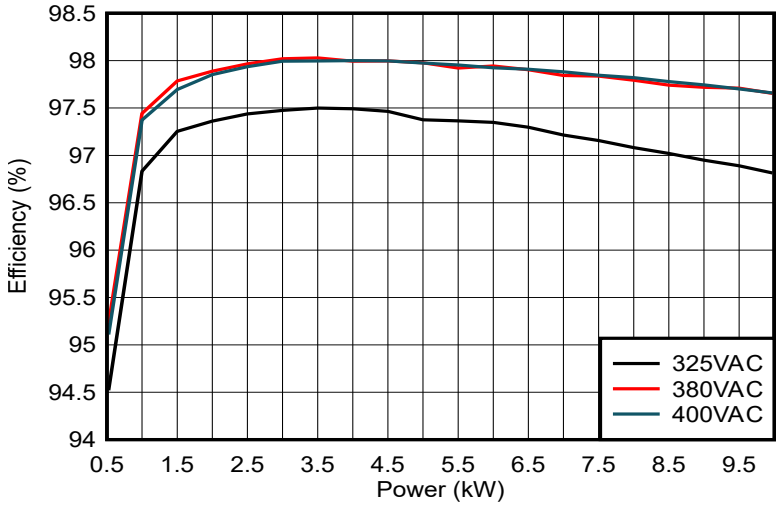


Figure 3-49. Efficiency Results

Figure 3-50 shows average iTHD test results at each power level under 325VAC, 380VAC, and 400VAC. Average iTHD is less than 5% at > 2.5kW under 380VAC. iTHD is less than 1.5% at 10kW under 380VAC.

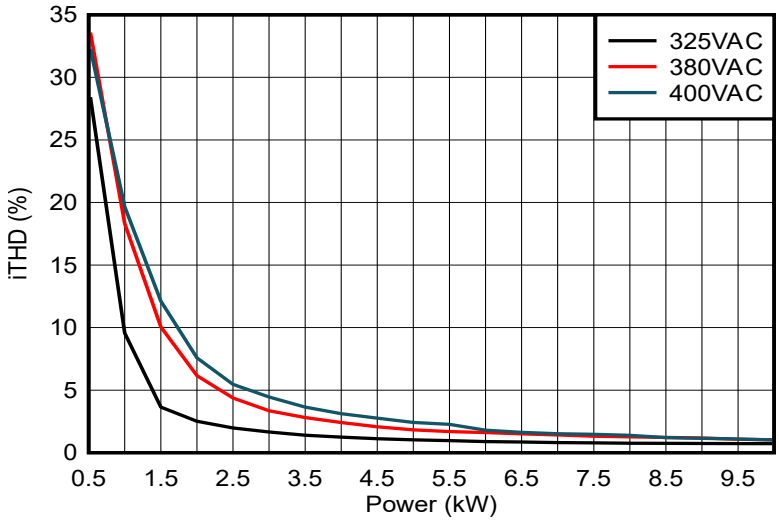


Figure 3-50. iTHD Test Results

Figure 3-51 shows average power factor test results at each power level under 325VAC, 380VAC, and 400VAC.

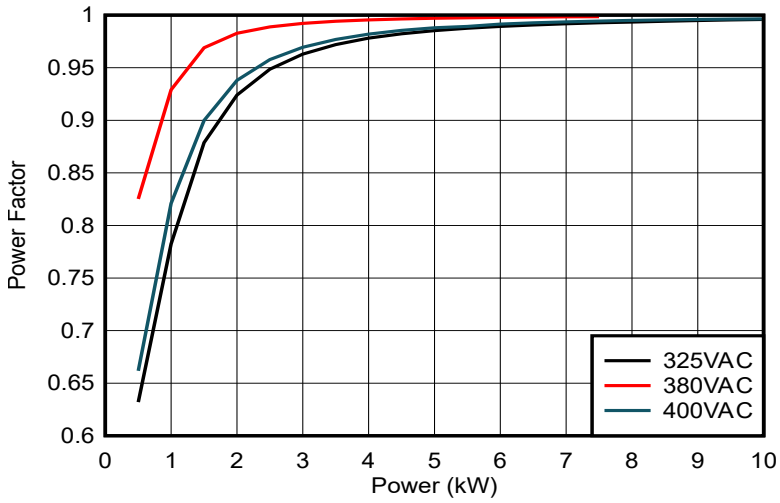


Figure 3-51. Power Factor Test Results

4 Design and Documentation Support

4.1 Design Files

4.1.1 Schematics

To download the schematics, see the design files at TIDA-010257.

4.1.2 Bill of Material (BOM)

To download the BOM, see the design files at TIDA-010257.

4.2 Tools and Software

Tools

LAUNCHXL-F2800137	TMS320F2800137 LaunchPad™ development kit for C2000™ real-time MCU

Software

C2000Ware DigitalPower SDK	DigitalPower software development kit (SDK) for C2000™ MCUs

4.3 Documentation Support

1. Texas Instruments, *TMS320F280013x Microcontrollers Data Sheet*
2. Texas Instruments, *TMS320F280013x Real-Time Microcontrollers Technical Reference Manual*
3. Texas Instruments, *C2000™ Software Frequency Response Analyzer (SFRA) Library and Compensation Designer User's Guide*

4.4 Support Resources

TI E2E™ support forums are an engineer's go-to source for fast, verified answers and design help — straight from the experts. Search existing answers or ask your own question to get the quick design help you need.

Linked content is provided "AS IS" by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's Terms of Use.

4.5 Trademarks

C2000™, TI E2E™, and Code Composer Studio™ are trademarks of Texas Instruments.
All trademarks are the property of their respective owners.

5 About the Author

HELY ZHANG is a System Application Engineer at Texas Instruments where he is responsible for developing home appliance related power delivery and motor inverters. Hely earned his master's degree from Anhui University of Science and Technology with Power electronics in 2002, and worked in SolarEdge and General Electric before joining TI.

JIM CHEN is a system engineer at Texas Instruments where he is responsible for specifying and developing system designs for Appliances. Jim earned his master's degree from Nanjing University of Aeronautics and Astronautics with Electrical Engineering in 2022.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), 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 will 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 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.

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

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265