**Design Guide: TIDA-00961**

**High-Efficiency, 1.6-kW, High-Density, GaN-Based, 1-MHz CrM Totem-Pole PFC Converter Reference Design—Software Guide**

**Description**

This reference design is a critical-conduction-mode (CrM), totem-pole power factor correction (PFC) design that uses TI's 600-V GaN power stage LMG3410 and Piccolo™ F280049 controller. This high-density (165 × 84 × 40 mm), two-stage, interleaved, 1.6-kW design is ideal for many space-constrained applications such as servers, telecom, and industrial power supplies. Interleaving the power stages reduces input and output ripple currents. For hardware design details and detailed test results, see the TIDA-00961 design guide.

**Features**

- Super High-Peak Efficiency > 98.7% at Full Load and 230-V AC Input
- Provides Ready Platform of GaN-Based CrM Totem-Pole PFC to Address Various Power Supplies up to 2 kW
- GaN Power Stage Switching at Frequencies Approaching 1.2 MHz Results in Very Compact Power Stage (165 × 84 × 40 mm)
- Wide Operating Input Range: 85-V to 245-V AC Input
- Full Digital Control Using TI's Piccolo F280049 Controller
- TI's LMG3410 GaN Power Stage With Integrated Driver and Protection Ensures Circuit Reliability and Eases Design
- External Cooling Not Required; up to 55°C Ambient Operation for Loads ≤ 800 W
- C2000™ powerSUITE™ Support for Easy Adaptation of Design for User Requirement
- Software Frequency Response Analyzer (SFRA) for Quick Measurement of Open Loop Gain
- Protects for Output Overcurrent, Overvoltage, and Undervoltage Conditions

**Applications**

- Merchant network and server PSU
- Merchant telecom rectifiers
- Industrial AC-DC power supplies
1 System Description

The main power supplies used in telecom, server, and industrial PSU systems convert AC line power to an isolated constant DC voltage output suitable for the loads they power—typically, 12 V for server PSUs, 48 V for telecom rectifiers, and 24 V for industrial PSUs. These main power supply systems range typically from 1 kW to 5 kW. All systems need a front-end power factor correction (PFC) circuit to shape the input current of the power supply so as to meet the power factor and current THD norms such as IEC 61000-2-3.

This reference design is an attempt to meet the above challenges of telecom, server, and industrial power supplies. It makes use of the fast switching capability of the driver integrated GaN FET from TI, LMG3410 and the advanced features of TI’s microcontroller TMS320F280049 to realize a very fast switching (up to 1.2 MHz) ZVS transition mode totem pole (TTPL) PFC. Two channel interleaving is implemented to demonstrate the design’s capability to cater to higher power applications by adding parallel interleaved power stages. High efficiency of up to 99% is demonstrated, though it is possible to improve it even further by optimizing the design of the magnetics. The design is able to meet the PF and current THD requirements applicable to the systems addressed. The main features of this reference design are the high switching frequency (made possible due to the LMG3410 GaN power switches), ZVS operation (made easier by the advanced features of TMS320F280049), bridgeless TTPL operation and implementation of interleaving the transition mode operation.

The accompanying software allows programming the controller and experimenting with different control parameters to tune the control loop for good system performance. This design supports the use of C2000 powerSUITE tools like the compensation designer and the software frequency response analyzer (SFRA). The software project allows users to evaluate the complete system with the help of these supported tools. This software is part of DigitalPower SDK for C2000™ microcontrollers (MCU). The DigitalPower SDK is a cohesive set of software infrastructure, tools, and documentation designed to minimize C2000 MCU based digital power system development time.

This guide provides software design details along with experimental results wherever necessary. This guide also describes a structured step-by-step method to evaluate this solution by starting with a simple open-loop excitation and then working towards a complete well-tuned closed-loop system. For hardware design details and detailed test results, see High-Efficiency, 1.6-kW High-Density GaN-Based 1-MHz CrM Totem-Pole PFC Converter Reference Design.
1.1 **Key System Specifications**

Table 1 describes the power specifications of the TTPL PFC reference design.

**Table 1. Key System Specifications**

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TEST CONDITIONS</th>
<th>MIN</th>
<th>NOM</th>
<th>MAX</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>INPUT CONDITIONS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input voltage ($V_{INAC}$)</td>
<td>85</td>
<td>230</td>
<td>245</td>
<td>V AC</td>
<td></td>
</tr>
<tr>
<td>Input frequency ($f_{LINE}$)</td>
<td>47</td>
<td>50/60</td>
<td>63</td>
<td>Hz</td>
<td></td>
</tr>
<tr>
<td>No load power ($P_{NL}$)</td>
<td>$V_{NAC} = 230$ V, $I_{OUT} = 0$ A</td>
<td>1</td>
<td>W</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>OUTPUT CONDITIONS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output voltage</td>
<td>390</td>
<td></td>
<td>V DC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output voltage</td>
<td>4.1</td>
<td></td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Line regulation</td>
<td>0.5%</td>
<td></td>
<td>%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Load regulation</td>
<td>0.5%</td>
<td></td>
<td>%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output ripple</td>
<td>Peak-to-peak</td>
<td>25</td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output power (High line)</td>
<td>$V_{NAC} = 230$ V</td>
<td></td>
<td>1600</td>
<td>W</td>
<td></td>
</tr>
<tr>
<td>Output power (Low line)</td>
<td>$V_{NAC} = 120$ V</td>
<td></td>
<td>1250</td>
<td>W</td>
<td></td>
</tr>
<tr>
<td><strong>SYSTEM CHARACTERISTICS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Efficiency ($\eta$)</td>
<td>$V_{IN} = V_{NOM}$ and full load</td>
<td>98.75</td>
<td>%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PWM switching frequency</td>
<td>200</td>
<td>1200</td>
<td>kHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operating ambient</td>
<td>Open frame, 200LFM</td>
<td>−10</td>
<td>25</td>
<td>55</td>
<td>°C</td>
</tr>
<tr>
<td>Power line harmonics</td>
<td>As per IEC 61000-3-2 Class A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Board size</td>
<td>Length × width × height</td>
<td>165 × 84 × 40</td>
<td>mm$^3$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**WARNING**

TI intends this reference design to be operated in a lab environment only and does not consider it 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 may 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 the 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 may 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 over-voltage and over-current 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 design or components connected to the design.*

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

Some components may 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 may be present.
2 System Overview

2.1 Block Diagram

Figure 1 shows the high-level block diagram of the design. The key TI parts used in this design are the LMG3410 GaN power stage and TMS320F280049 Piccolo MCU. Daughter cards are used for each of the half-bridge (HB) switching power stages and the control card. An onboard auxiliary power supply for housekeeping and fan is implemented using the UCC28740. There is a silicon MOSFET-based synchronous rectifier driven by the UCC27712 HB driver to reduce power loss in the low-frequency HB at the input. The input current is sensed by a Hall effect sensor, the output of which is processed by a variable gain amplifier stage built using OPA237 to get better accuracy at low current levels.

Figure 1. Block Diagram of 1.6-kW PFC Regulator
### 2.2 Resources Guide

Table 2 shows the key signal connections between the TMS320F280049C controlCARD and the TIDA-00961 base board.

#### Table 2. Key Signal Connections

<table>
<thead>
<tr>
<th>SIGNAL NAME</th>
<th>FUNCTION (100-PIN F28004x DEVICE ON 120-PIN controlCARD)</th>
<th>HSEC PIN NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Si_GATE_H</td>
<td>EPWM-1A: High-side drive signal for silicon MOSFET synchronous rectifier</td>
<td>49</td>
</tr>
<tr>
<td>Si_GATE_L</td>
<td>EPWM-1B: Low-side drive signal for silicon MOSFET synchronous rectifier</td>
<td>51</td>
</tr>
<tr>
<td>HS_GATE1</td>
<td>EPWM-2A: High-side drive signal for phase 1 HB</td>
<td>53</td>
</tr>
<tr>
<td>LS_GATE1</td>
<td>EPWM-2B: Low-side drive signal for phase 1 HB</td>
<td>55</td>
</tr>
<tr>
<td>HS_GATE2</td>
<td>EPWM-3A: High-side drive signal for phase 2 HB</td>
<td>50</td>
</tr>
<tr>
<td>LS_GATE2</td>
<td>EPWM-3B: Low-side drive signal for phase 2 HB</td>
<td>52</td>
</tr>
<tr>
<td>ACL_SENSE_O</td>
<td>ADC2 Ch B3: AC input voltage line</td>
<td>20</td>
</tr>
<tr>
<td>ACN_SENSE_O</td>
<td>ADC1 Ch A3: AC input voltage neutral</td>
<td>17</td>
</tr>
<tr>
<td>AC_I_SENSE</td>
<td>ADC2 Ch B2 with CMPSS3: AC return current sensing and overcurrent protection</td>
<td>18</td>
</tr>
<tr>
<td>AC_CUR_VREF</td>
<td>ADC2 Ch B0: Hall effect current sensor reference</td>
<td>12</td>
</tr>
<tr>
<td>HV_SENSE</td>
<td>ADC3 Ch C14: AC return current sensing and overcurrent protection</td>
<td>36</td>
</tr>
<tr>
<td>ZVS1_2</td>
<td>CMPSS4: ZVS adjustment phase 1</td>
<td>24</td>
</tr>
<tr>
<td>ZVS2_2</td>
<td>CMPSS7: ZVS adjustment phase 2 (not used)</td>
<td>14</td>
</tr>
<tr>
<td>AC_CUR_FAULT</td>
<td>GPIO23: Input current fault signal from the Hall effect current sensor</td>
<td>64</td>
</tr>
<tr>
<td>GaN_Fault1</td>
<td>GPIO37: GaN fault signal for phase 1</td>
<td>58</td>
</tr>
<tr>
<td>GaN_Fault2</td>
<td>GPIO35: GaN fault signal for phase 2</td>
<td>60</td>
</tr>
<tr>
<td>RELAY_CTRL</td>
<td>GPIO12: Controls in-rush relay</td>
<td>57</td>
</tr>
<tr>
<td>FAN_CTRL</td>
<td>GPIO15: Controls on-board cooling fan</td>
<td>63</td>
</tr>
<tr>
<td>G_AC_I</td>
<td>GPIO25: Controls current sensing circuit gain</td>
<td>77</td>
</tr>
</tbody>
</table>
## 2.3 Software Overview

### 2.3.1 Software Flow

The software project makes use of the C-background and C-ISR framework. The project uses C-background code as the main supporting program for the application, which is responsible for all system management tasks, decision making, intelligence, and host interaction. The C-ISR code consists of three components, which are executed inside time critical interrupt service routines (ISRs) and runs all the critical control code. This code includes ADC reading, control calculations, and PWM updates. The Slow Control ISR is executed at a fixed rate of 10 kHz using CPU timer2. The Fast Control ISR is triggered by PWM1 and is executed at a rate of 50 kHz. The PWM ISR executes at a variable rate determined by the PWM switching frequency and is triggered every third PWM cycle.

Figure 2 shows the general software flow for this project.

![Figure 2. Software Flow](image)

<table>
<thead>
<tr>
<th>Timer 0 Tasks:</th>
<th>Timer 1 Tasks:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trip level adjust</td>
<td>Instrumentation</td>
</tr>
<tr>
<td>Fault management</td>
<td>SFRA Background</td>
</tr>
<tr>
<td>Start-up</td>
<td>In-rush relay control</td>
</tr>
<tr>
<td>Shutdown</td>
<td>Serial comms</td>
</tr>
</tbody>
</table>

### 2.3.1.1 Background Loop

A task state-machine has been implemented as part of the background code. Tasks are arranged in groups (A1, A2, A3…, B1, B2, B3…). Each group is executed according to two CPU timers, which are configured with periods of 2.5 ms and 100 ms, respectively. Within each group (for example, B), each task is run in a round-robin manner. For example, group B executes every 100 ms, and there are three tasks in group B. Therefore, B1, B2, and B3 execute once every 300 ms.

### 2.3.1.2 Slow Control ISR

This ISR is responsible for running the voltage loop. Voltage loop uses a filtered version of the bus voltage feedback. A slew limit algorithm and software overvoltage protection algorithm are also implemented inside this ISR. Part of the phase shedding algorithm is executed here and is only used for disabling the second phase once the system is operating under the phase shedding threshold. Once the controller makes the decision to disable the second phase, the phase is disabled at the next negative to positive zero crossing transition of the input voltage in the state-machine run inside the Fast Control ISR.

<table>
<thead>
<tr>
<th>Slow Control ISR</th>
<th>Fast Control ISR</th>
<th>PWM ISR</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 kHz ISRs</td>
<td>50 kHz ISRs</td>
<td></td>
</tr>
<tr>
<td>SFRA_INJECT()</td>
<td>SFRA_INJECT()</td>
<td>SFRA_INJECT()</td>
</tr>
<tr>
<td>ADC_VBUS()</td>
<td>ADC_AC()</td>
<td></td>
</tr>
<tr>
<td>Vloop</td>
<td>4x oversampling</td>
<td></td>
</tr>
<tr>
<td>EMAVG()</td>
<td>loop</td>
<td></td>
</tr>
<tr>
<td>Datalog (optional)</td>
<td>Period, Dead-time Calculations</td>
<td></td>
</tr>
<tr>
<td>Instrumentation (RMS calculations)</td>
<td>Phase Re-enable - Fast</td>
<td></td>
</tr>
<tr>
<td>Phase Shedding - Slow</td>
<td>AC Cycle/ Crossover state machine , SW PLL</td>
<td></td>
</tr>
<tr>
<td>OVP</td>
<td>PWM register updates</td>
<td></td>
</tr>
<tr>
<td>SFRA_COLLECT()</td>
<td>SFRA_COLLECT()</td>
<td>EXIT</td>
</tr>
<tr>
<td>EXIT</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2.3.1.3 Fast Control ISR

This ISR is responsible for executing the AC cycle state machine, which tracks the positive or negative halves of the input signal and provides clean transitions from one half-cycle to the next. When a new half-cycle begins, the switching transition involves using soft-start for the active switch, turning on the correct silicon MOSFET rectifier switch after a delay, and finally enabling the synchronous rectifier GaN switch after the current starts flowing.

A software phase locked loop that uses the same basic structure as the SOGI-FLL module discussed in Interleaved CCM Totem Pole Bridgeless PFC Reference Design Using C2000™ MCU provides the input sinusoidal signal angle information to the state machine. The frequency adaptive feature implemented through a frequency locked loop (FLL) allows operation over the full input frequency range (47 Hz to 63 Hz). The software PLL uses 4× oversampled input voltage waveform. Similarly, the current loop, which is executed inside this ISR, uses 4× oversampled feedback data. Along with the voltage loop output, the current loop output provides the on-time for the active switch. The off-time is calculated by applying volt-sec balance to the boost inductor. The dead times are also calculated inside this ISR.

Part of the phase shedding algorithm is also executed in this ISR. It re-enables the second phase once the load is above the phase re-enable threshold. Once the controller starts to re-enable the second phase, the second phase is re-enabled as soon as possible to avoid stress on the functioning enabled phase.

2.3.1.4 PWM ISR

This solution makes use of the global and one-shot reload feature available on C2000 devices with type-4 PWM modules. This is a critical feature that allows seamless transitions between PWM switching frequencies for all PWM outputs of the two phases. This feature is used inside the PWM ISR. The PWM ISR is also responsible for maintaining correct interleaving (180° phase shift) between the two phases.

When instantaneous input voltage is greater than half of the output voltage, some negative current is needed to achieve or approach ZVS as it may not be achieved using dead-time and off-time calculations alone. The PWM ISR executes a ZVS adjustment routine to achieve or approach ZVS under these conditions. This algorithm adjusts the off-time of the active switch and, thereby, adjusts the period of the PWMs.

2.3.1.5 powerSUITE™ Tools

Note the references to SFRA library functions in Figure 2. Texas Instruments’ SFRA library is designed to enable frequency response analysis on digitally controlled power converters using software alone. This feature enables performing frequency response analysis of the power converter with relative ease as no external connections or equipment is required. The optimized library can be used in high-frequency power conversion applications to identify the plant and the open-loop characteristics of a closed-loop power converter, which can be used to get stability information such as bandwidth, gain margin, and phase margin to evaluate the control loop performance. For more information, refer to the SFRA library documentation.

In addition to SFRA, this kit supports the use of another powerSUITE tool called the Compensation Designer. The Compensation Designer tool allows the design of different styles of compensators to achieve the desired closed loop performance, which can be done using the measured power stage or plant data from the SFRA Tool. The coefficients that must be programmed on the device are generated by the Compensation Designer and can be copied into the code directly. These tools help users evaluate the complete system, adapt it for their end application, and tune it for improved performance.
2.3.2 Incremental Builds

This project is divided into three incremental builds, which makes learning and getting familiar with the board and software easier. This approach is also good for debugging and testing boards. Table 3 shows the incremental build options. To select a particular build option, select the corresponding `INCR_BUILD` option in `main.syscfg`. Once the build option is selected, compile the complete project by selecting rebuild-all compiler option. Section 3 provides more details to run each of the build options.

<table>
<thead>
<tr>
<th>BUILD OPTION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>INCR_BUILD = 1</td>
<td>Open-loop check (check PWM drive circuit and sensing circuit)</td>
</tr>
<tr>
<td>INCR_BUILD = 2</td>
<td>Closed current loop check</td>
</tr>
<tr>
<td>INCR_BUILD = 3</td>
<td>Closed voltage loop check (full PFC)</td>
</tr>
</tbody>
</table>

The SFRA and compensation designer tools can be used when the system is operated in builds 2 or 3. Make sure that the `SFRA_TYPE` in `main.syscfg` or `pfc2philrmttptl_settings.h` is set correctly to measure the desired current and voltage loop response.
3 Procedures for Running the Incremental Builds

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

3.1 Test Condition

For input, the power supply source \(V_{\text{IN}}\) must range from 85-V to 245-V AC. Set the input current limit of input AC source to 15 A for full power tests, but start with a lower current limit during initial board bring-up.

For output, use an electronic variable load or a variable resistive load, which must be rated for \(\geq 400\) V and must vary the load current from 0 to 5 A.

3.2 Test Equipment Required for Board Validation

- Isolated AC source
- Single-phase power analyzer
- Digital oscilloscope
- Multimeters
- Electronic or resistive load

3.3 controlCARD Modifications and Settings

Certain modifications and settings on the device control card are required to control the base-board, communicate over JTAG and use the isolated UART port. The following are the required for revision E2 of the F280049C controlCARD. Users can also refer to the information sheet located inside C2000Ware at <install_path>\c2000ware\boards\controlcards\TMDSCNCD280049C or alternatively get it from the Piccolo F280049C controlCARD Information Guide.

1. Set S1:A on the controlCARD on both ends to the "ON (up)" position to enable JTAG connection to the device and UART connection for SFRA GUI. If this switch is "OFF (down)", one cannot use the isolated JTAG built in on the controlCARD nor can SFRA GUI communicate to the device.

2. Remove the capacitor connected between the isolated grounds on the controlCARD, C26:A, for the best performance of this reference design.

3. Set position 4 on S7 to the OFF (down) position.

4. Populate R40 with a 47-\(\Omega\) resistor, and populate C43 with a 47-nF capacitor.

5. Populate R53, R64, and R41 each with 1-k\(\Omega\) resistors.

6. Populate C59, C54 and C50 each with 10-nF capacitors.

3.4 Test Setup

1. Connect the GaN daughter cards and C2000 controlCARD to the design board.

2. Connect input terminals (Pin 1 and Pin 3 of connector J101) of the reference board to the AC (or DC as described in the step-by-step procedure in Section 3.5) power source. Do not turn on the power source until indicated to do so in the following procedure.

3. Connect output terminals (Pin 1 and Pin 2 of connector J2) to the electronic load, maintaining correct polarity. Pin 2 is the VDC output and Pin 1 is the GND terminal.

4. Connect a current limited (1 A) 12-V DC bench power supply between TP304 (+) and TP104 (–). Do not turn on the power source until indicated to do so in the following procedure.

5. For the initial bring-up of the board, set and maintain a minimum load of about 800 mA (500 \(\Omega\)) at a 390-V DC output.

6. Connect voltmeter, oscilloscope probes, and other measurement equipment to probe or analyze various signals and parameters as desired. Only use appropriately rated equipment and follow proper isolation and safety practices.
7. Connect a USB B to A cable between the PC and the C2000 controlCARD. Do not turn on any of the power supplies at this time.

8. Use an external fan for cooling with the air flow directed towards the two inductors and the two GaN daughter cards under all test conditions. A Sunon SP100A fan is used for these tests.

---

CAUTION
There are high voltages present on the board. Only experienced power supply professionals should handle this board in a lab environment. To safely evaluate this board, use an appropriate isolated and current limited power source. Before power is applied to the board, an appropriate resistive or electronic load must be connected at the output. Do not handle the unit when power is applied to it. Only use appropriately rated equipment and follow proper isolation and safety practices.

---

### 3.5 Test Procedure

Before installing software for this solution, download and install C2000WARE-DIGITALPOWER-SDK software package from the link provided by TI. Install this software in its default folder. The software project would then reside inside C2000Ware Digital Power SDK folder at `<install_location>`\`solutions\`\ida_00961`. Follow these steps to build and run this code with different incremental builds.

#### 3.5.1 Build 1: Open Loop Check With ADC Feedback

The objective of this build is to:
- Evaluate the open loop operation of the system
- Verify the PWM and ADC driver modules
- Verify the FET driver circuit and sensing circuit on the board
- Become familiar with the operation of CCS

Because this system is running open-loop, the ADC measured values are only used for instrumentation purposes in this build. Only DC input is used in this build.

##### 3.5.1.1 Start CCS and Open a Project

To quickly execute this build:

1. Connect an isolated DC power supply capable of providing at least 100 V at up to 1 A to the input terminals (Pin 1 and Pin 3 of connector J101) of the reference board. Positive or negative polarity can be used for this connection. The variable `neg_cycle` in the watch view automatically sets to 1 for negative input and resets to 0 for positive input once the code is running. Set the power supply to output 100 V with a current limit of 1 A. Do not turn the power supply on at this time.

2. Connect a 400-V resistive load of about 500 Ω or 333 Ω at the output.

3. Open CCSv7 (or newer). Maximize the program to fill the screen. Close the welcome screen if it opens up.

4. A project contains all the files and build options needed to develop an executable output file (.out), which can be run on the MCU hardware. On the menu bar click `Project → Import Existing CCS/CCE Eclipse Project`. Under `Select root directory`, navigate to the C2000Ware Digital Power SDK folder and select `<install_location>`\`solutions\`\ida_00961`. Click `Finish`.

   This project invokes all the necessary tools (compiler, assembler, linker) to build the project.

5. In the project window on the left, click the plus sign (+) to the left of Project. Figure 3 shows an example project window.
3.5.1.2 Device Initialization, Main, and ISR Files

NOTE: DO NOT make any changes to the source files—ONLY INSPECT.

1. Open the main.syscfg file by double-clicking on the file name in the Project window.
2. Under Project Options, select Open Loop as the INCR_BUILD option as shown in Changed Figure 4: Build 1: main_syscfg. Save the main.syscfg file. Rebuild the project if there are any changes in main.syscfg.
3. Open and inspect `pfc2philtrmttpl_main.c`. Notice the call made to the `setupDevice()` function and other initialization functions like `setupPwms()`, `setupADC()`, and so on. Also notice code for the ISRs and the background for(;;) loop.

4. Open and inspect `pfc2philtrmttpl_main.h`. Notice the code for different incremental build options (specifically the build to compile now) inside `pfcControlCode()`.

### 3.5.1.3 Build and Load the Project

1. Open the `pfc2philtrmttpl_settings.h` file. Notice that the incremental build (`INCR_BUILD`) option is set to 1, `DEMO_MODE` is set to 0, and `DC_CHECK` is set to 1.

2. If another build option was built previously, right click on that project name and click on `Clean Project`. Click `Project → Build All` button, and watch the tools run in the build window.

3. Turn on the 12-V DC bench power supply. Click on the `Debug` button or click `Run → Debug`. The build 1 code should compile and load.

4. Notice the CCS Debug icon in the upper right-hand corner, indicating that the user is now in the Debug Perspective view. The program should be stopped at the start of `main()`.

### 3.5.1.4 Debug Environment Windows

It is standard debug practice to watch local and global variables while debugging code. There are various methods for doing this in CCS, such as memory views and watch views. Additionally, CCS has the ability to make time (and frequency) domain plots. This ability allows the user to view waveforms using graph windows.

1. Populate the expressions window entries by clicking on `View → Scripting console` on the menu bar and then opening the `setupdebugenv_allbuilds.js` file from the project directory using the scripting console `Open File` command. Figure 5 shows an example of the expressions window. Note that some of the variables have not been initialized at this point in the main code and may contain some useless values.
usEnablePwms, when set, enables all PWM outputs to the power stage. When this parameter is reset, all PWM outputs are disabled. A set value for start_flag indicates the system is operational with PWM switching enabled. neg_cycle is set to 1 for the negative DC input or the negative half-cycle of the AC input and reset to 0 for the positive input or positive half-cycle. pwrsrce_ocpThreshold and pwrsrce_ovpThreshold variable set the peak input overcurrent and output overvoltage protection thresholds in amperes and volts, respectively. closeGiLoop and closeGvLoop indicate whether the current loop and voltage loops are closed and are used in builds 2 and 3. ac_cur_sensed is the raw instantaneous current feedback value, and ac_cur_ref provides the reference current value set for the current loop. These are used in build 2. Similarly, vBus_sensedFiltered_notch2 is the raw filtered output voltage feedback value, and vBusRef provides the reference voltage value set for the voltage loop. These are used in build 3.

The variable coeff_change allows changing current and voltage loop coefficients from the watch view by first changing the coefficient values (new_gi_Kp, new_gi_Ki, new_gv_Kp, and new_gv_Ki) and then setting coeff_change to 1. guiVrmsEMAvg and guiIrmsEMAvg provide RMS values of the input voltage and current, while guiVbus provides the output voltage value. new_ton_calc is the on-time of the active switch, which is directly controlled in build 1. phase_shedding_EN, when set, allows phase shedding algorithm to be active. phase_OFF indicates that the second phase has been turned off as part of the phase shedding operation. ovp_Fault and ocp_Fault indicate overvoltage and overcurrent faults, respectively.

### 3.5.1.5 Using Real-Time Emulation

Real-time emulation is a special emulation feature that allows the windows within CCS to be updated at up to a 10-Hz rate while the MCU is running. This emulation not only allows graphs and watch views to update, but also allows the user to change values in watch or memory windows and have those changes affect the MCU behavior. This is very useful when tuning control law parameters on the fly, for example.

1. Enable real-time mode by hovering the mouse on the buttons on the horizontal toolbar and clicking Enable Silicon Real-time Mode (service critical interrupts when halted, allow debugger accesses while running).
2. A message box may appear. If so, select YES to enable debug events. This will set bit 1 (DGBM bit) of status register 1 (ST1) to 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. When a large number of windows are open, as bandwidth over the emulation link is limited, updating
too many windows and variables in continuous refresh can cause the refresh frequency to bog down.

Right click on the button in the Expressions window and select *Continuous Refresh Interval*… One can slow down the refresh rate for the expressions window variables by changing the continuous refresh interval (milliseconds) value. A rate of 1000 ms is usually enough for these exercises.

4. Click on the *Continuous Refresh* button ( ) for the expressions view.

### 3.5.1.6 Run the Code

1. Run the code by using the <F8> key, or by using the *Run* button on the toolbar.
2. Turn on the 100-V power supply.
3. *guiVrmsEMAvg*, *guiIrmsEMAvg*, and *guiVbus* should correctly reflect input and output parameters. If this is the case, proceed to the next step. If this is not the case, an option is to debug the sensing circuit for the parameters that do not update correctly.
4. In the watch view, the variable *usEnablePwms* should be set to 0, while *new_ton_calc* should be set to 20. Set *usEnablePwms* to 1.
5. With a 333-Ω load, the output voltage should go up to about 122 V.
6. Increase *new_ton_calc* to 25. The output voltage should increase to about 138 V. It is now safe to experiment with different *new_ton_calc* values. Make sure that this value is changed slowly, making sure the output voltage does not go above 400 V.
7. You may also experiment with different load values and different input voltage values. Please make sure that the board is never operated out of specifications.
8. One may also probe the switch node voltages for the two phases using appropriately rated voltage probes and oscilloscope. Figure 6 provides the switch node voltage waveforms for both phases (*Vsw1* and *Vsw2*) with a 100-V input, a load of 333 Ω, and a *new_ton_calc* value of 30. Note that zero voltage switching (ZVS) is approached or achieved.

![Figure 6. Build 1: Switch Node Voltages](image)

Figure 7 shows a corresponding watch view for this test.
9. Set the `pwrste_ovpThreshold` below the current output voltage. The PWMs should shut down and the `ovp_Fault` should be set in the watch window. When this happens, first turn off the DC power source. Take the MCU out of real-time mode and then reset the device.

10. Follow all of the above steps again except the one for changing `pwrste_ovpThreshold`. This time set the `pwrste_ocpThreshold` below the input current level that is being drawn. The PWMs should again shut down and the `ocp_Fault` should be set in the watch window. This validates build 1 of the software along with all the sensing circuit and protection mechanisms.
11. Fully halting the MCU when in real-time mode is a two-step process. First turn off the 100-V supply.
   Wait a few seconds. Halt the processor by using the Suspend button ( ), or by using Target →
   Halt. Then click the button again to take the MCU out of real-time mode. Reset the MCU.
12. Either leave CCS running for the next exercise or close CCS.

3.5.2 Build 2: Closed Current Loop Check

The objective of this build is to evaluate the closed current loop operation of the system. Although a DC or
AC input can be used with this build, it is recommended to use an AC input because the following steps
are applicable to an AC input. If this is the first time this board is powered, start with build 1 in
Section 3.5.1.

3.5.2.1 Start CCS and Open a Project

To quickly execute this build:
1. Connect a programmable, isolated AC power supply capable of providing universal AC input up to 1.6
   kW to the input terminals (Pin 1 and Pin 3 of connector J101) of the reference board. Set the power
   supply current limit to 8 A and the output frequency to 60 Hz. Do not turn the power supply on at this
time.
2. Connect a 400-V resistive load of about 333 Ω at the output.
3. Follow Steps 3 to 5 of Section 3.5.1.

3.5.2.2 Build and Load the Project

1. Open main.syscfg file by double-clicking on the file name in the project window. Under Project Options,
   select Closed Current Loop as the INCR_BUILD option. Save the main.syscfg file. Rebuild the whole
   project if there are any changes in main.syscfg.
2. Open the pfc2pfilrmttpl_settings.h file. Notice that the incremental build (INCR_BUILD) option is set to
   2, DEMO_MODE is set to 0, and DC_CHECK is set to 0.
3. Make sure DC_CHECK is set to 0 above.
4. If another build option was built previously, right click on the project name and click on Clean Project.
   Click Project → Build All button and watch the tools run in the build window.
5. Turn on the 12-V DC bench power supply. Click on the Debug button or click Run → Debug. The build
   2 code should compile and load.
6. Notice the CCS Debug icon in the upper right-hand corner, which indicates the
   Debug Perspective
   view. The program should be stopped at the start of main().
7. Follow Step 1 in Section 3.5.1.4 to Step 4 in Section 3.5.1.5.

3.5.2.3 Run the Code

1. Run the code by using the <F8> key, or by using the Run button on the toolbar.
2. With the AC source set to output 0 V at 60 Hz, turn on the AC power supply.
3. Slowly increase the input voltage from 0-V to 120-V AC.
4. guiVrmsEMAvg, guilmrsEMAvg, and guiVbus should correctly reflect input and output parameters.
5. In the watch view, the variable usEnablePwms should be set to 0, while ac_cur_ref should be set to
   0.03. Set usEnablePwms to 1.
6. With a 333-Ω load, the output voltage should boost up to about 194 V.
7. Increase ac_cur_ref to 0.04. The output voltage should increase. Keep increasing ac_cur_ref in
   increments of 0.01 until ac_cur_ref is increased to a value of 0.1. The ac_cur_sensed value may
   change rapidly and is not the same as the ac_cur_ref value. This change is because ac_cur_sensed is
   the instantaneous input current value while ac_cur_ref is the amplitude reference for the current
   command.
8. With a 120-V AC input, a load of about 333 Ω, and an ac_cur_ref of 0.1, the output voltage should be
around 308 V. Figure 8 shows the input voltage and current waveforms under this condition.

Figure 8. Build 2: Input Waveforms

Figure 9 shows a corresponding watch view for the above test.

Figure 9. Build 2: Expressions Window at Run Time

NOTE: The following three steps are optional.

9. Open the SFRA GUI from the main.syscfg powerSUITE page. Make sure Current is selected for SFRA current loop analysis. If not, change the settings in main.syscfg and rebuild the whole project.

10. Connect the GUI to the controller:
   a. Open the SFRA GUI.
   b. Select Floating Point.
   c. Click on Setup.
d. Select the USB COM port being used for this connection.
e. Uncheck Boot on Connect.
f. Click on Connect.

11. Click on Start Sweep to start a frequency sweep, and notice the frequency injection in the current waveforms on the oscilloscope. The SFRA GUI plots and displays the SFRA response of the current loop at the end of the sweep. The captured SFRA data is also available in the compensation designer window.

12. One can experiment with different load values and different input voltage values. Never operate the board out of the specifications.

13. One can also probe the switch node voltages for the two phases using appropriately rated voltage probes and oscilloscopes. This validates closed current loop operation.

14. Fully halt the MCU when in real-time mode:
   a. First turn off the 100-V supply, and wait a few seconds.
   b. Halt the processor by using the Suspend button ( ), or by using Target → Halt. Then click the button again to take the MCU out of real-time mode. Reset the MCU.

15. Either leave CCS running for the next exercise or close CCS.

3.5.3 Build 3: Closed Voltage Loop Check (Full PFC)

The objective of this build is to evaluate the complete PFC system. Use AC input with this build. If this is the first time the board is powered, start with build 1 in Section 3.5.1.

3.5.3.1 Start CCS and Open a Project

To quickly execute this build:

1. Connect a programmable isolated AC power supply capable of providing universal AC input up to 1.6 kW to the input terminals (Pin 1 and Pin 3 of connector J101) of the reference board. Set the power supply current limit to 15 A and the output frequency to 50 or 60 Hz. Do not turn on the power supply at this time.

2. Connect a 400-V resistive load of about 333 Ω at the output.

3. Follow Steps 3 to 5 of Section 3.5.1.

3.5.3.2 Build and Load the Project

1. Open the main.syscfg file by double-clicking on the file name in the project window. Under Project Options, select Closed Voltage & Current Loop as the INCR_BUILD option. Save the main.syscfg file. Rebuild the project if there are any changes in main.syscfg.

2. Open the pfc2phltrmttpl_settings.h file. Notice that the incremental build (INCR_BUILD) option is set to 3, DEMO_MODE is set to 1, and DC_CHECK is set to 0.

3. Make sure DC_CHECK is set to 0 above.

4. If another build option was built previously, right click on the project name and click on Clean Project. Click Project → Build All button and watch the tools run in the build window.

5. Turn on the 12-V DC bench power supply. Click on the Debug button or click Run → Debug. The build 3 code should compile and load.

6. Notice the CCS Debug icon in the upper right-hand corner, which indicates the Debug Perspective view. The program should be stopped at the start of main().

7. Follow Step 1 in Section 3.5.1.4 to Step 4 in Section 3.5.1.5.

3.5.3.3 Run the Code

1. Run the code by using the <F8> key, or by using the Run button on the toolbar.

2. With the AC source set to output 0 V at 60 Hz, turn on the AC power supply.
3. Increase the input voltage from 0-V to 120-V AC.
4. guiVrmsEMAvg, guiIrmsEMAvg, and guiVbus should correctly reflect input and output parameters.
5. After a few seconds, the converter should start working and the output should voltage ramp up to about 390 V DC, which is the default value set in the Output Vbus (V) text box in main.cfg.

**NOTE:** Make sure the Output Vbus (V) value in main.cfg is always within board limits. This value must not be set below 390 V.

6. Figure 10 shows the input voltage and current waveforms with a 120-V AC input and a load of about 333 Ω.

**Figure 10. Build 3: Input Waveforms**

![Figure 10](image)

Figure 11 shows a corresponding watch view for the above test.

**Figure 11. Build 3: Expressions Window at Run Time**

![Figure 11](image)
NOTE: The following three steps are optional.

7. Open the SFRA GUI from the main.syscfg powerSUITE page.

8. Connect the GUI to the controller:
   a. Open the SFRA GUI.
   b. Select *Floating Point*.
   c. Click on *Setup*.
   d. Select the USB COM port being used for this connection.
   e. Uncheck *Boot on Connect*.
   f. Click on *Connect*.

9. Click on *Start Sweep* to start a frequency sweep, and notice the frequency injection in the current waveforms on the oscilloscope, as shown in Figure 13.

**Figure 12. Build 3: Frequency Injection During SFRA Frequency Sweep**
The SFRA GUI plots and displays the SFRA response of the current loop at the end of the sweep, as shown in Figure 13.

**Figure 13. Build 3: SFRA Current Loop Response**
The default loop parameters can be tuned to improve the frequency response using the compensation designer tool. The current loop response in Figure 14 was obtained with a single-phase operation and some aggressive loop parameters.

Figure 14. SFRA Current Loop Response With Better Loop Tuning

Note that the loop parameters resulting in Figure 14 have not been used for this version of the software as they require more tuning and testing across all operating points.

10. One can experiment with different load values and different input voltage values. Never operate the board out of the specifications. Also, it is not recommended to use load transients higher than about 300 W and line transients greater than 20 V at this time.

11. One can also probe the switch node voltages for the two phases using appropriately rated voltage probes and oscilloscope. The following waveforms provide the node voltage waveforms along with input voltage and current under full load condition at a 120-V AC input and 230-V AC input.
Figure 15. Positive Half-Cycle 1 (Full Load With 120-V AC Input)

Figure 16. Positive Half-Cycle 2 (Full Load With 120-V AC Input)
Figure 17. Positive Half-Cycle 3 (Full Load With 120-V AC Input)

Figure 18. Negative Half-Cycle 1 (Full Load With 120-V AC Input)
Figure 19. Negative Half-Cycle 2 (Full Load With 120-V AC Input)

Figure 20. Negative Half-Cycle 3 (Full Load With 120-V AC Input)
12. To fully halt the MCU when in real-time mode:
   a. First turn off the 100-V supply, and wait a few seconds.
   b. Halt the processor by using the \textit{Suspend} button ( ), or by using \textit{Target} $\rightarrow$ \textit{Halt}. Then click button again to take the MCU out of real-time mode. Reset the MCU.

13. Close CCS.
4 Software Files
To download the software files, see C2000WARE-DIGITALPOWER-SDK.

5 Related Documentation
1. Texas Instruments, LMG3410 600-V 12-A Integrated GaN Power Stage Data Sheet
2. Texas Instruments, TMS320F28004x Piccolo™ Microcontrollers Data Sheet
3. Texas Instruments, UCC27712 620-V, 1.8-A, 2.8-A High-Side Low-Side Gate Driver with Interlock Data Sheet
5. Texas Instruments, Topic 1: Safety Considerations in Power Supply Design Seminar
6. Texas Instruments, PFC THD Reduction and Efficiency Improvement by ZVS or Valley Switching Application Report

5.1 Trademarks
Piccolo, E2E, C2000, powerSUITE are trademarks of Texas Instruments.
All other trademarks are the property of their respective owners.

6 About the Authors
HRISHIKESH NENE is a systems engineer with C2000 microcontroller solutions and Member Group Technical Staff (MGTS) at Texas Instruments. He is a senior member of IEEE (SMIEEE). He received a bachelor of engineering (B.E.) degree in electronics and telecommunications from the University of Pune (India) in 2003 and a master’s degree in electrical engineering (M.S.E.E.) from the University of Houston in 2006. At TI he has worked extensively on design and development of digitally controlled power electronics systems and on defining the industry leading C2000 control peripherals.

SALIL CHELLAPPAN is a systems manager and Member Group Technical Staff (MGTS) at Texas Instruments, where he is responsible for developing customized power solutions as part of the power delivery, industrial segment. Salil brings to this role his extensive experience in power electronics, power conversion, EMI/EMC, power and signal integrity, and analog circuits design spanning many high-profile organizations. Salil holds a bachelor of technology degree from the University of Kerala.
### Revision History

#### Changes from B Revision (June 2019) to C Revision

<table>
<thead>
<tr>
<th>Changes</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Changed main.cfg to main.syscfg throughout document</td>
<td>9</td>
</tr>
<tr>
<td>Added Rebuild the project if there are any changes in main.syscfg</td>
<td>12</td>
</tr>
<tr>
<td>Changed Figure 4: Build 1: main_syscfg</td>
<td>13</td>
</tr>
<tr>
<td>Added Rebuild the whole project if there are any changes in main.syscfg</td>
<td>17</td>
</tr>
<tr>
<td>Added If not, change the settings in main.syscfg and rebuild the whole project</td>
<td>18</td>
</tr>
<tr>
<td>Added Rebuild the project if there are any changes in main.syscfg</td>
<td>19</td>
</tr>
</tbody>
</table>

#### Changes from A Revision (August 2018) to B Revision

<table>
<thead>
<tr>
<th>Changes</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Changed figure 2 Fast Control ISR section from 10 kHz to 50 kHz</td>
<td>7</td>
</tr>
</tbody>
</table>

#### Changes from Original (March 2018) to A Revision

<table>
<thead>
<tr>
<th>Changes</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Changed globally: all “…280049M” references to “…280049C”, where applicable</td>
<td>1</td>
</tr>
<tr>
<td>Added below Resources, changed “TMDXDOCK280049M” to “TMDSCNCD280049C” Design Folder part number and .url link to point to “F280049C controlCARD Evaluation Module” versus the docking station</td>
<td>1</td>
</tr>
<tr>
<td>Changed Key System Specifications table to separately list high line and low line for MAX Output power values</td>
<td>3</td>
</tr>
</tbody>
</table>
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, 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 (www.ti.com/legal/termsofsale.html) 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.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2020, Texas Instruments Incorporated