Software User Guide: TIDA-011008

3.6-kW, GaN-based single-phase string inverter reference design

Copyright © 2024 Texas Instruments Incorporated

The board needs an external 12 V power supply.

1 Hardware, Software, Testing Requirements, and Test Results

1.1 Required Hardware and Software

The hardware design for the TIDA-011008 has been discussed in detail in the associated hardware design guide. Please refer to that documentation for details on the H/W design. This section details the hardware setup required for the experiments and testing of different software labs, Labs 1 through 8 as outlined in this user guide.

1.1.1 Hardware

The hardware in this design is set up and operated in several pieces:

- One TIDA-011008 DC/DC board
- **Isolated** 4-pin or 6-pin JTAG debugger
- Laptop or other computer

The test equipment required to power and evaluate the design is as follows:

- Isolated 12-V, 2-A bench style supply for primary board power
- 1 DC supplies (>500 V, >15 A)
- 1 DC load (>520 V, > 10 A)
- 1 Bidirectional AC Grid Emulator (> 230 V_{RMS}, 16 A_{RMS})

NOTE: The MCU GND is referred to the negative rail of the power. Hence it is mandatory and extremely important to *use an isolated 12 V supply and an isolated debugger*!

1.1.2 Getting Started with Firmware

This section details the CCS software project for this TIDA-011008 and testing of different software labs.

1.1.2.1 Opening the CCS Project

The software of this design is available from MSS. To open the project:

1. Install CCS (version 20 or above)
2. Install C2000Ware DigitalPower SDK from the tool page
3. Install C2000Ware
4. Open CCS, and create a new workspace
5. Import TIDA-011008 project into CCS

1.1.2.2 Interrupts and Lab Structure

The project consists of one ISR (INT_ADC_A_1_ISR), which is reserved for the control loops and the PWM update of the boost converters and the DC/AC converter. INT_ADC_A_1_ISR is triggered by the EOC of ADC_A which is triggered by DCAC1_BASE (EPWM1) when time-base counter equal to zero. 4 events will trigger the interrupt.

The software for this reference design is organized in various labs. Table 1-1 lists the labs for the DC/DC and DC/AC portions running on ISR. All the labs are run using the C28x Main CP.

Copyright © 2024 Texas Instruments Incorporated

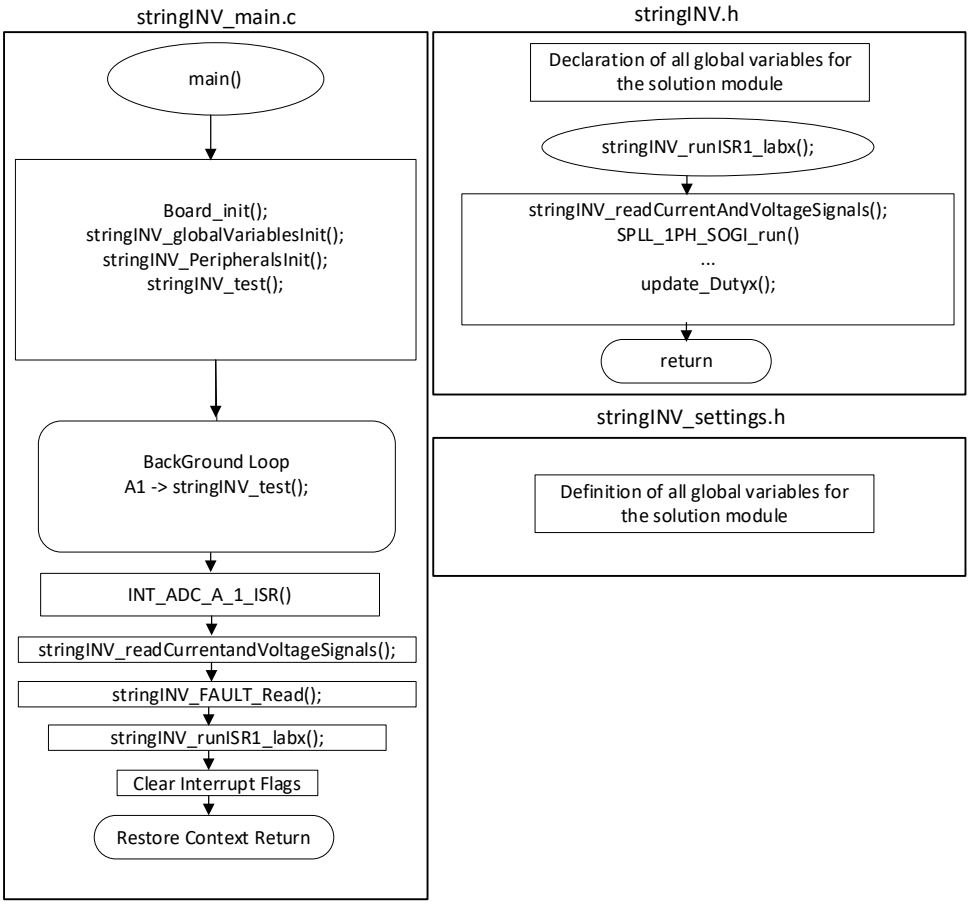


Figure 1-0 Software Flow Diagram

Table 1-1 Overview of Labs to Test Reference Design

LAB NUMBER	DESCRIPTION	COMMENTS	TEST ENVIRONMENT
1	Run PWM and ADC check	Check PWM and sensing	
2	Run Boost Converters in open-loop	Check functionality of the Boost converters	Power Stage Hardware + DC sources + DC Load connected to the DC bus rail
3	Run Boost Converters in close-loop with MPPT	Check control loops + MPPT	Power Stage Hardware + DC sources + DC Load connected to the DC bus rail
4	Run DC/AC Converter in open-loop (HERIC, H-Bridge Unipolar and Bipolar)	Check functionality of the DC-AC Converter	Power Stage Hardware + AC Load + DC source
5	Run DC/AC Converter in close-loop	Check control loops and functionality of the DC-AC Converter	Power Stage Hardware + AC Load + DC source
6	Run DC/AC Converter in close-loop with PLL and AC simulator (grid-tied)	Check control loops and functionality of the DC-AC Converter	Power Stage Hardware + AC grid emulator + DC source
7	Run DC/AC Converter in close-loop with PLL and AC simulator (grid-tied) with Voltage control	Check control loops and functionality of the DC-AC Converter	Power Stage Hardware + AC grid emulator + DC source
8	Run DC/AC Converter in close-loop with PLL and AC simulator (grid-tied) with Voltage control along with Boost Converters with MPPT	Check control loops and functionality of the DC-AC Converter and Boost Converters + MPPT	Power Stage Hardware + AC grid emulator + DC sources

Copyright © 2024 Texas Instruments Incorporated

1.1.2.3 Protection Scheme: overvoltage, overcurrent and overtemperature protections

There are two overcurrent protections (OCP) set up in the entire system using the comparator sub-system (CMPSS) internal to the MCU as described in Table 1-1. The overcurrent protection has been set up by using the instruction defined in the stringINV.syscfg.

The value of OCP can be directly changed by changing the following variables in stringINV_settings.h file. This sets the corresponding values of high comparator DAC value and low comparator DAC value accordingly.

- stringINV_ACDC_iAC_TRIP_AMPS
- stringINV_ACDC_CS_Boost_TRIP_AMPS

2 Testing and Results

2.1 Testing Lab 1 (ADC and PWM check)

The PWM signals of all the DC/DC converter and DC/AC can be seen on the on the pins of the board. Do not apply power on PV or AC port of the board when testing Lab1, apply only 12V aux power.

Please note that the DC/AC stage in this lab runs in Bipolar mode and if other modes need to be checked, corresponding changes need to be made in the software. 2 for Bipolar, 3 for Unipolar

- `#define stringINV_DCAC_Modulation_Scheme 2`

➢ IMPORTANT, PLEASE CHECK THE VERSION OF PCB BOARD

Please check the PCB revision on top left of the board:

- if it's E1, then `#define stringINV_PCB_Version 1`
- if it's E2, then `#define stringINV_PCB_Version 2`

- To set the project to Lab 1, change the Lab Number in the < stringINV_settings.h> file.
  `#define SINV_LAB 1`
- Right-click on the project name and click *Rebuild Project*. The project builds successfully.
- To load the project, first make sure in the Project Explorer the correct target configuration file is set as Active under targetConfigs (*.ccxml file). Then, click *Run → Debug* to launch a debugging session. The project will then load on the device and the CCS debug view will become active. The code will halt at the start of the main routine. Click Run → Resume to run the program.
- To populate this window with the correct variables, click Debug → WATCH. Then right click on the Expressions tab on CCS and select Remove All. This will clear the Expressions window. Now right click again on the cleared Expressions window and select Import. Then browse to the project folder and select Lab1.json. This will populate the watch window with the appropriate variables needed to debug the system. Enable Continuous Refresh button on the watch window to enable continuous update of values from the control.
- Enable the PWM of the converters by setting stringINV_clearPWMTrip to 1.
- To check PWM of the Boost converters, set the duty cycle in the variables stringINV_Boost1_Duty_Ref.
- stringINV_clearPWMTrip sets the PWM for the converter. The code enables operation of the bipolar mode but the functionality of the converter can be examined already.
- Check the corresponding pins to see the PWM signals.

Copyright © 2026 Texas Instruments Incorporated

2.2 Testing Labs 2 through 8

- It is important to mention that in all the labs the voltages can be higher than 60 V. 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 precaution and appropriate safeguards to avoid injuring yourself or damaging property.
- Also note that the electrolytic capacitors do not discharge automatically. It is task of the user to discharge them.
- In all the labs, the 12 V power supply can be provided externally by connecting an isolated 12 V power supply.
- If the PV emulator is not available, you can comment Boot_MPPT() function and uncomment Boost_Currentloop() function in Lab3 and Lab8. In Boost_Currentloop() function, the boost input current is controlled by `stringINV_Boost_CS_Ref`.

```
//stringInverter_runISR1_lab_Boost_MPPT();
stringInverter_runISR1_lab_Boost_Currentloop();
```

- For the DC/AC part, please pay attention to the following points:
  - Set the modulation scheme by changing the following in the < stringINV_settings.h> file.

    #define stringINV_DCAC_Modulation_Scheme xx

    - 2 is for H-Bridge in bipolar and 3 is for H-Bridge in double frequency unipolar

2.2.1.Testing Lab 2 (Run Boost converters in open-loop)

In this lab, the DC/DC boosts are run in open-loop. One independent DC power sources on inputs and DC load connected to the DC bus rail are needed. In E1 version board, please remove one DC link capacitor as the connector for DC Load for testing the boost stage.

Figure 2-1 shows how the DC load and the DC sources need to be connected.

Copyright © 2026 Texas Instruments Incorporated

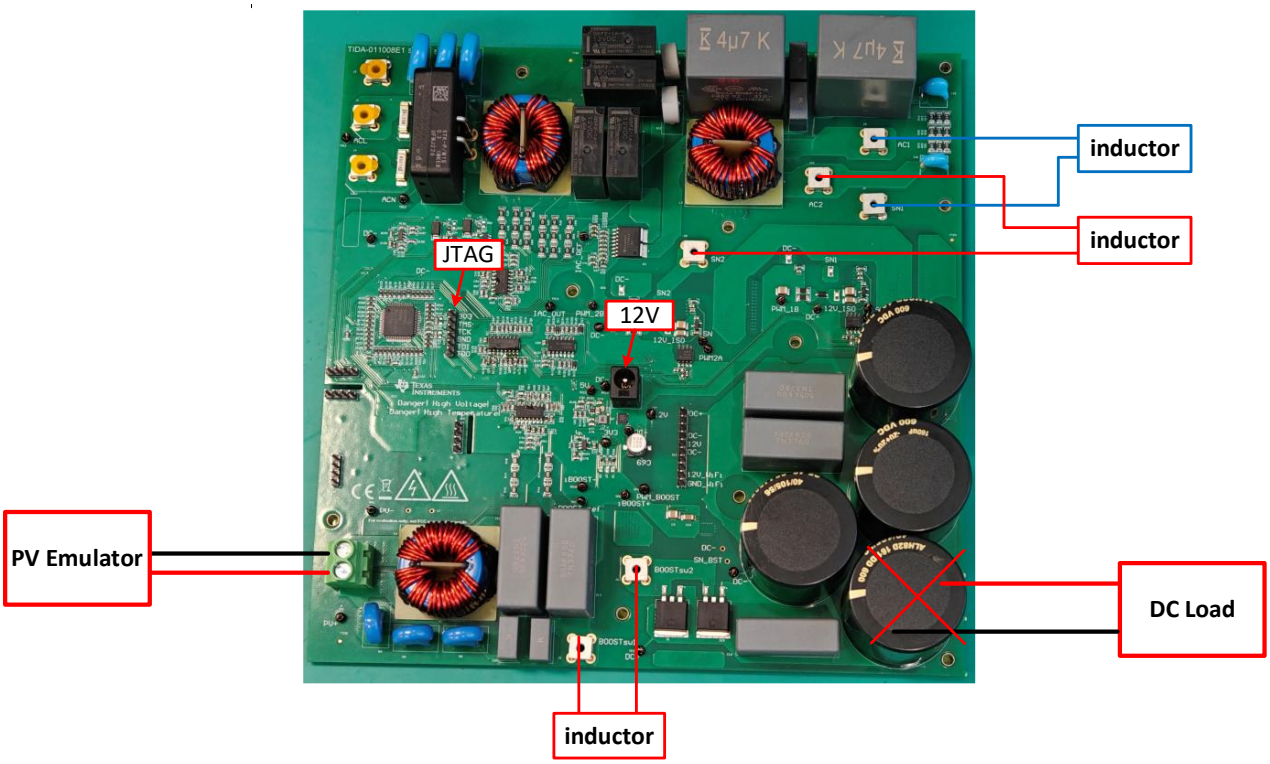


Figure 2-1 Testing the board when setting up Lab2, Lab3

The following steps need to be taken in order to implement Lab 2 and run the system:

- Set the project to Lab 2 by changing the Lab Number in the < stringINV_settings.h> file.
  ```
  #define SINV_LAB 2
  ```
- Build and load the code, use the Lab2.json file to populate the watch variables in the CCS window.
- Turn-on the DC load and set up the voltage at 50 V with current limit up to 15 A (voltage control mode, CV).
- Turn-on the DC power supply to reach with appropriate voltage limit and limit the current to 2 A. Later on, after working normally, the current limitation can set to 15A.
- Enable the PWM of the Boost converters by setting stringINV_clearPWMTripBoost to 1. The Boost start with an open circuit and that should be visible on the DC load.
- Choose the duty-cycle corresponding to the Boost converter you are interested in running. Set the values of stringINV_Boost1_Duty_Ref accordingly. You can notice the slowly ramped values of the Boost duty cycles. The voltages on the DC power supply will automatically be set according to the duty cycle requested.
- Ramp up the DC load to reach 400 V in steps of 50 V also DC supply to reach the required input voltage. Boost converter has been tested up to 3.6 kW (with a panel input voltage of 250 V).
- To stop the converter, set stringINV_stop = 1.
- Turn-off the DC source first and then the DC load.

Copyright © 2026 Texas Instruments Incorporated

2.2.2.Testing Lab 3 (Run Boost Converters in close-loop with MPPT)

In this lab, the DC/DC boosts are run along with MPPT algorithm. DC power sources on input and DC load connected to the DC bus rail are needed. The MPPT algorithm used is incremental conductance.

- If the PV emulator is not available, you can comment Boot_MPPT() function and uncomment Boost_Currentloop() function in Lab3 and Lab8. In Boost_Currentloop() function, the boost input current is controlled by `stringINV_Boost_CS_Ref`.

```
//stringInverter_runISR1_lab_Boost_MPPT();
stringInverter_runISR1_lab_Boost_Currentloop();
```

Figure 4-3 shows how the DC load and the DC sources need to be connected.

- Set the project to Lab 3 by changing the Lab Number in the < stringINV_settings.h> file.

```
#define SINV_LAB 3
```

- Build and load the code, use the Lab3.json file to populate the watch variables in the CCS window.
- Turn-on the DC load and set up the voltage at 50 V with current limit up to 30 A (voltage control mode, CV).
- Turn-on the DC power supply to reach with appropriate voltage limit and limit the current to 2 A. Later on, after working normally, the current limitation can set to 15A.
- Enable the PWM of the Boost converters by setting stringINV_clearPWMTripBoost to 1.
- The MPPT algorithm starts working, and you will notice the DC power supply go in current control mode with up to 15 A pushing through and voltage moves in accordance to achieve the maximum power tracking.
- Slowly start increasing the voltages of DC source and load, to reach the required voltages. Boost converter has been tested up to 3.6 kW (with a panel input voltage of 250 V).
- To stop the converter, set stringINV_stop = 1.
- Turn-off the DC source first and then the DC load.

Copyright © 2026 Texas Instruments Incorporated

2.2.3.Testing Lab 4 (Run DC/AC Converter in open-loop across a resistive load)

In this lab the DC/AC is operated in open-loop. A DC power supply is connected to the input. On the AC side, the output needs to be connected across resistors ranging down to 14.7 Ohm, the resistor need to handle 3.6kW, so a resistor with 5kW capability is good, a fan is needed to cool down the resistor.

Figure 2-2 shows how the AC load and the DC source need to be connected.

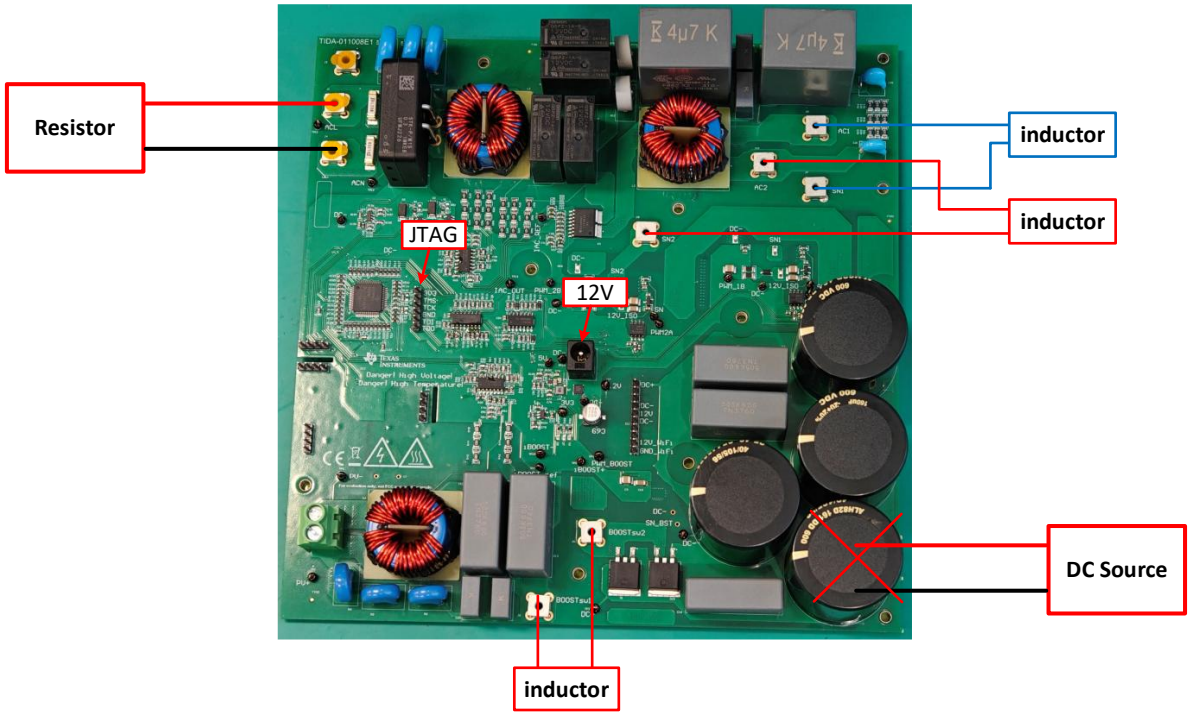


Figure 2-2 Testing the board when setting up Lab4, Lab5.

The following steps need to be taken in order to implement Lab 4 and run the system:

- Set the project to Lab 4 by changing the Lab Number in the < stringINV_settings.h> file.
  ```
  #define SINV_LAB 4
  ```
- Set the required modulation scheme by changing the following in the < stringINV_settings.h> file.

  #define stringINV_DCAC_Modulation_Scheme xx

  2 is for H-Bridge in Bipolar and 3 is for H-Bridge in Unipolar
- Build and load the code, use the Lab4.json file to populate the watch variables in the CCS window.
- The AC relay can be closed by setting relay_safe_enable and relay_precharge_enable to 1.
- Enable the PWM of the DC/AC by setting up stringINV_clearPWMTripDCAC to 1.
- Turn the DC supply on and set it to 50 V with corresponding current limitation. Note if the grid voltage and current waveforms look as expected.
- Ramp-up the input DC power supply up to 400 V in steps of 50 V to achieve 3.6 kW (in case of 14.7 Ohms across load).
- To shut the system down, ramp the DC source down in steps of 50 V to 50 V.
- To stop the converter, set stringINV_stop = 1. This will stop the converter at zero crossing and will also turn the AC relay off.
- Turn-off the DC source.

Copyright © 2026 Texas Instruments Incorporated

2.2.4.Testing Lab 5 (Run DC/AC Converter in close-loop)

In this lab, the DC/AC converter is tested in close-loop. DC link voltage is provided by the DC power supply. On the the AC side, the output needs to be connected across resistors.

Figure 2-2 shows how the AC Load and the DC source needs to be connected.

- Set the project to Lab 5 by changing the Lab Number in the < stringINV_settings.h> file.
  ```
  #define SINV_LAB 5
  ```
- Set the required modulation scheme by changing the following in the < stringINV_settings.h> file.

  #define stringINV_DCAC_Modulation_Scheme xx

  2 is for H-Bridge in Bipolar and 3 is for H-Bridge in Unipolar
- Build and load the code, use the Lab5.json file to populate the watch variables in the CCS window.
- The AC relay can be closed by setting relay_safe_enable and relay_precharge_enable to 1.
- Enable the PWM of the DC/AC by setting up stringINV_clearPWMTripDCAC to 1.
- Turn on the DC supply with the input voltage set to 50 V with corresponding current limitation.
- Note that the initial value for stringINV_ACDC_iAC_real_Ref is 0.5A, also note if the grid voltage and current waveforms look as expected.
- Ramp-up the input DC power supply up to 400 V in steps of 50 V.
- Change the current value from the DC/AC converter by changing the variable stringINV_ACDC_iAC_real_Ref. The current ramps up slowly to the requested value through a ramp generation function stringINV_RAMP_IDCACREF(). The current reference can be changed till the system achieves 3.6 kW.
- To shut the system down, ramp the DC source down in steps of 50 V to 50 V.
- To stop the converter, set stringINV_stop = 1. This will stop the converter at zero crossing and will also turn the AC relay off.
- Then turn-off the DC source.

Copyright © 2026 Texas Instruments Incorporated

2.2.5.Testing Lab 6 (Run DC/AC Converter in close-loop with PLL and AC Simulator)

In this lab, the DC/AC converter is tested in close-loop and PLL aids in synchronizing to the grid frequency. DC link voltage is provided to the DC Bus by the power supply. On the AC side, the output needs to be connected to a grid emulator. In case the grid emulator cannot sink power, connect appropriate resistors across it.

Figure 2-3 shows how the AC grid emulator and the DC source needs to be connected.

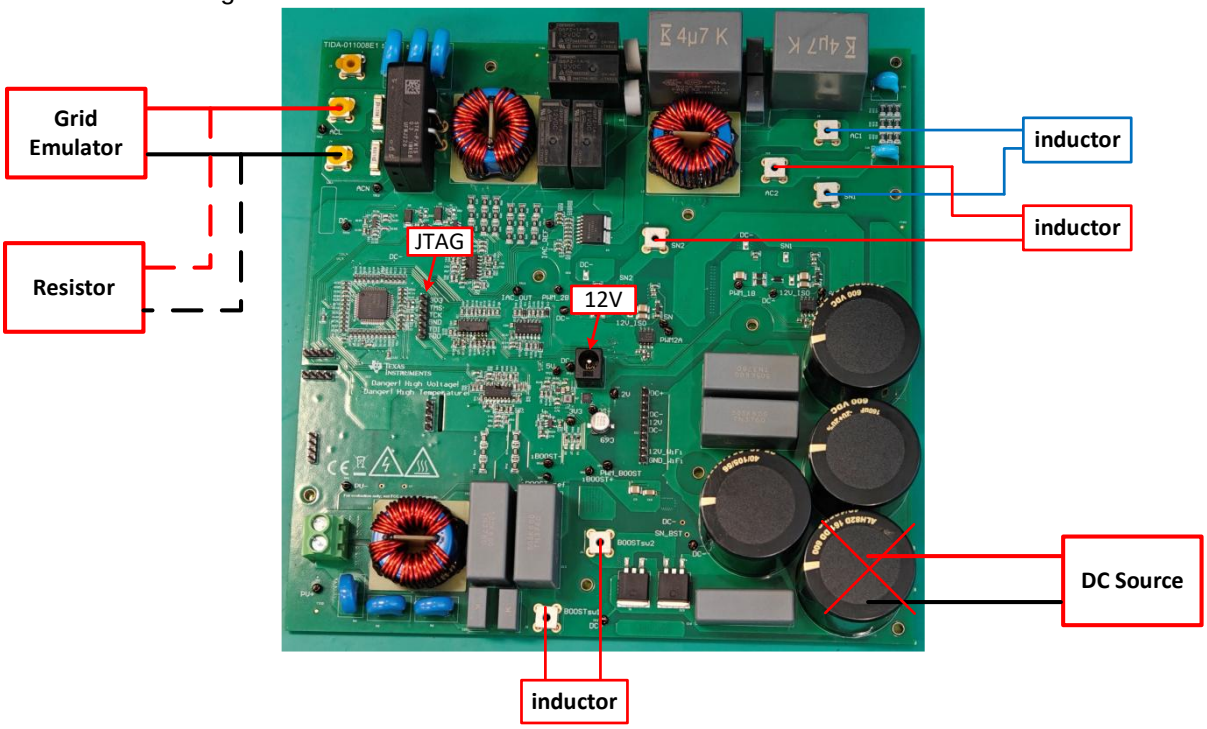


Figure 2-3 Testing the board when setting up Lab6, Lab7.

- Set the project to Lab 6 by changing the Lab Number in the < stringINV_settings.h> file.
  ```
  #define SINV_LAB 6
  ```
- Set the required modulation scheme by changing the following in the < stringINV_settings.h> file.

  #define stringINV_DCAC_Modulation_Scheme xx

  2 is for H-Bridge in Bipolar and 3 is for H-Bridge in Unipolar
- Build and load the code, use the Lab6.json file to populate the watch variables in the CCS window.
- Turn on the DC supply with the input voltage set to 50 V with corresponding current limitation.
- Turn on the AC grid emulator at a voltage setting of 30 V and at a frequency of 50 Hz, notice the grid voltage waveform.
- The AC relay can be closed by setting relay_safe_enable and relay_precharge_enable to 1.
- Enable the PWM of the DC/AC by setting up stringINV_clearPWMTripDCAC to 1.
- Note if the grid voltage and current waveforms look as expected.
- Ramp-up the input DC power supply up to 400 V in steps of 50 V and correspondingly the AC grid emulator in steps of around 30 V to reach 230 V.
- Ask for more current from the DC/AC converter by changing the variable stringINV_ACDC_iAC_real_Ref. The current ramps up slowly to the requested value through a ramp generation function stringINV_RAMP_IDCACREF(). The current reference can be changed till the system achieves 3.6 kW.
- To shut the system down, ramp the DC source down in steps of 50 V, and AC grid emulator in steps of 30 V.
- To stop the converter, set stringINV_stop = 1. This will stop the converter at zero crossing and will also turn the

Copyright © 2026 Texas Instruments Incorporated

AC relay off.

- Then turn-off the DC source first, then the grid simulator.

Copyright © 2026 Texas Instruments Incorporated

2.2.6.Testing Lab 7 (Run DC/AC Converter in close-loop with PLL, AC Simulator and Voltage control)

In this lab, the DC/AC converter is tested in close-loop and PLL aids in synchronizing to the grid frequency. DC link voltage will be regulated to a constant value by a voltage control loop. On the AC side, the output needs to be connected to a grid emulator. In case the grid emulator cannot sink power, connect appropriate resistors across it as shown in Figure 2-3.

Figure 2-3 shows how the AC grid emulator and the DC source needs to be connected.

- Set the project to Lab 7 by changing the Lab Number in the < stringINV_settings.h> file.
  ```
  #define SINV_LAB 7
  ```
- Set the required modulation scheme by changing the following in the < stringINV_settings.h> file.

  #define stringINV_DCAC_Modulation_Scheme xx

  2 is for H-Bridge in Bipolar and 3 is for H-Bridge in Unipolar
- Build and load the code, use the Lab7.json file to populate the watch variables in the CCS window.
- Set the DC Bus reference voltage reference by changing the value of the variable stringINV_DCDC_VDCext_Ref. This can also be set in the stringINV.c file. Set this voltage to 400 V. The ramped value can be seen in the variable stringINV_DCDC_vDC_Ref_Slewed.
- Turn on DC source at a voltage, say 430 V (keeping some margin) higher than the controlled voltage value set, and a current limit of say 0.5 A.
- Turn on the AC grid emulator at a voltage setting of 230 V and at a frequency of 50 Hz, notice the grid voltage waveform.
- The AC relay can be closed by setting relay_safe_enable and relay_precharge_enable to 1.
- Enable the PWM of the DC/AC by setting up stringINV_clearPWMTripDCAC to 1. You will see voltage being regulated.
- Slowly increase the current from the DC source to push more power into the converter. This can be changed till the system achieves 3.6 kW.
- To stop the converter, set stringINV_stop = 1. This will stop the converter at zero crossing and will also turn the AC relay off.
- Turn-off the DC source first, then the AC grid emulator.

Copyright © 2026 Texas Instruments Incorporated

2.2.7.Testing Lab 8 (Run DC/AC Converter in close-loop with PLL and AC simulator (grid-tied) with Voltage control along with Boost Converters)

In this lab, the DC/AC converter is tested in close-loop and PLL aids in synchronizing to the grid frequency. DC link voltage will be regulated to a constant value by a voltage control loop. On the AC side, the output needs to be connected to a grid emulator. Additionally, the DC/DC boosts are run along with MPPT algorithm. DC power sources on input is needed.

- If the PV emulator is not available, you can comment Boot_MPPT() function and uncomment Boost_Currentloop() function in Lab3 and Lab8. In Boost_Currentloop() function, the boost input current is controlled by `stringINV_Boost_CS_Ref`.

```
//stringInverter_runISR1_lab_Boost_MPPT();
stringInverter_runISR1_lab_Boost_Currentloop();
```

Figure 2-4 shows how the AC grid emulator and the DC source needs to be connected. In case the grid emulator cannot sink power, connect appropriate resistors across it.

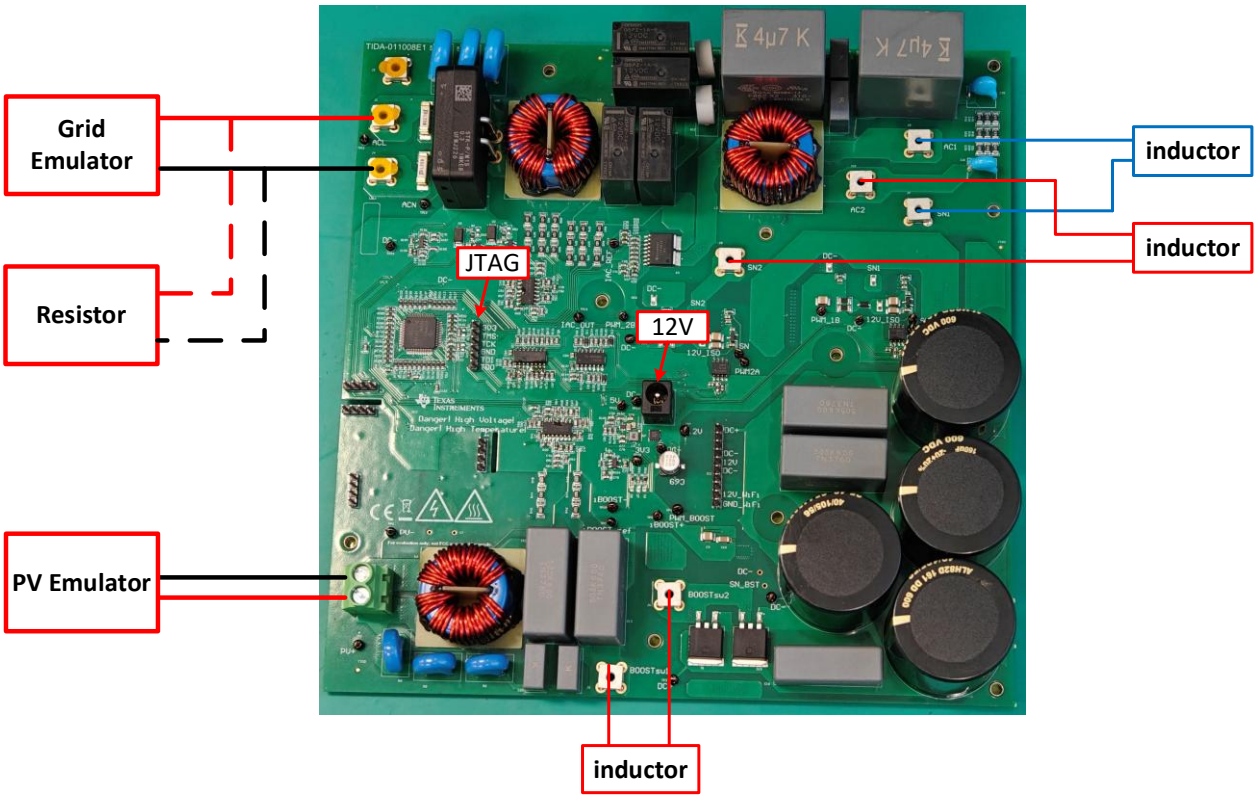


Figure 2-4 Testing the board when setting up Lab8

- Set the project to Lab 8 by changing the Lab Number in the < stringINV_settings.h> file.

```
#define SINV_LAB 8
```

- Set the required modulation scheme by changing the following in the < stringINV_settings.h> file.

  #define stringINV_DCAC_Modulation_Scheme xx

  2 is for H-Bridge in Bipolar and 3 is for H-Bridge in Unipolar

- Build and load the code, use the Lab8.json file to populate the watch variables in the CCS window.
- Set the DC Bus reference voltage reference by changing the value of the variable stringINV_DCDC_VDCext_Ref. This can also be set in the stringINV.c file. Set this voltage to 400 V. The ramped value can be seen in the variable stringINV_DCDC_vDC_Ref_Slewed.
- The AC relay can be closed by setting relay_safe_enable to 1.

Copyright © 2026 Texas Instruments Incorporated

- Turn on the AC grid emulator at a voltage setting of 230 V and at a frequency of 50 Hz, notice the grid voltage waveform.
- After the precharge of DC link capacitor is finished, set relay_precharge_enable to 1.
- Enable the PWM of the DC/AC by setting up stringINV_clearPWMTripDCAC to 1.
- Turn-on the DC power supply to reach with appropriate voltage limit and limit the current to 2 A.
- Enable the PWM of the Boost converters by setting stringINV_clearPWMTripBoost to 1.
- The MPPT algorithm starts working, and you will notice the DC power supply go in current control mode. If works stable, ramp up the current limitation to 15 A pushing through and voltage moves in accordance to achieve the maximum power tracking.
- The DC Bus voltage would be set according to the reference value.

  Please note that the maximum power which can be supplied into the grid is 3.6 kW. The power supplied by the boost converters must not overpass this value.
- To stop the converter, set stringINV_stop = 1. This will stop the converter at zero crossing and will also turn the AC relay off.
- Turn-off the DC source first, and then the AC grid emulator.

Trademarks

TMS320C2000™, C2000™ are trademarks of Texas Instruments.
All trademarks are the property of their respective owners.

Copyright © 2026 Texas Instruments Incorporated

3 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Revision A. June 2026

Copyright © 2026 Texas Instruments Incorporated

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 (https: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 © 2026, Texas Instruments Incorporated

Copyright © 2026 Texas Instruments Incorporated