

TDINV3000W050: 3kW Inverter Evaluation Board

Part 1: Getting started with the code

The design files and source code for the TDINV3000W050_0v1 3.0 kW inverter can be found at transphormusa.com/pv30kit and includes the complete project. The following instructions can be used to load and build the project on your computer for inspection or modification of the code. To take maximum advantage of the existing firmware, the TDINV3000W050 firmware is based on an open-source design from Texas Instruments (TI): The HV_SOLAR_DC_AC_v1.1 Kit and included in the firmware files. Full documentation of the TI kit and firmware is available from TI as part of the controlSUITE package.

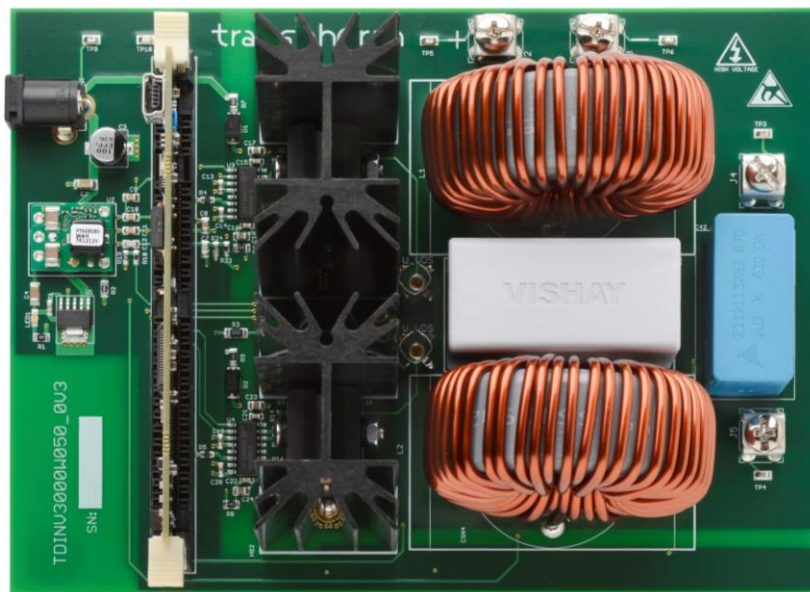
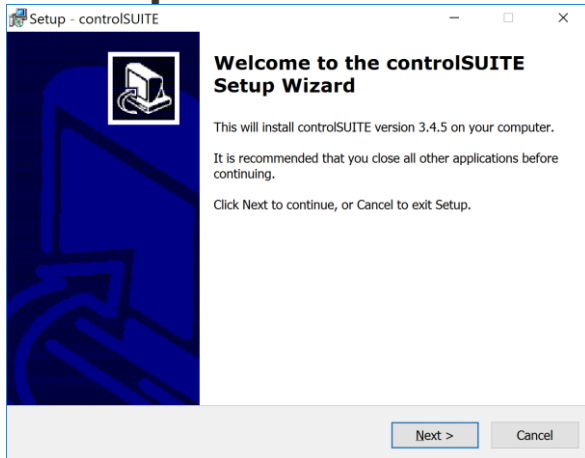


Figure 1. TDINV3000W050_0v1 3.0kW inverter evaluation board

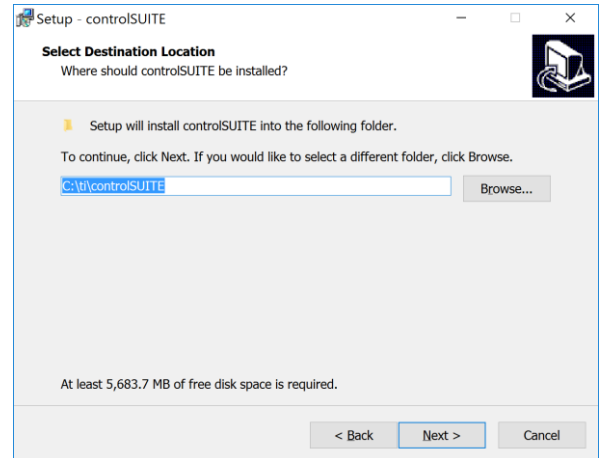
Installing Code Composer and controlSUITE

Code Composer Studio v7 is an integrated development environment based on the Eclipse open source framework and is available at processors.wiki.ti.com/index.php/Download_CCS.

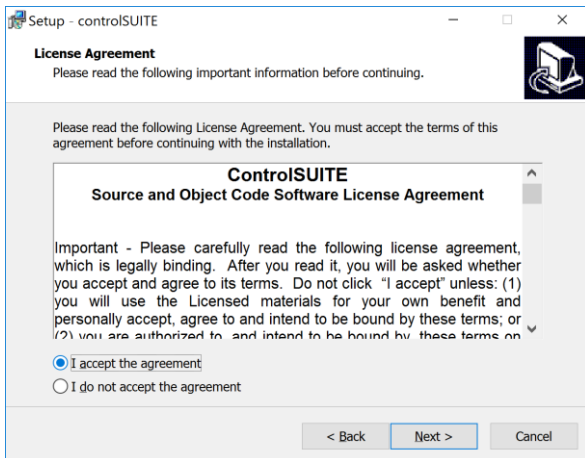
1. Install Texas Instruments Code Composer v7.x
2. The controlSUITE package is not required to work with the firmware but a great deal of documentation is provided with it; to download, go to ti.com/tool/controlsuite and run the controlSUITE installer, following these steps:



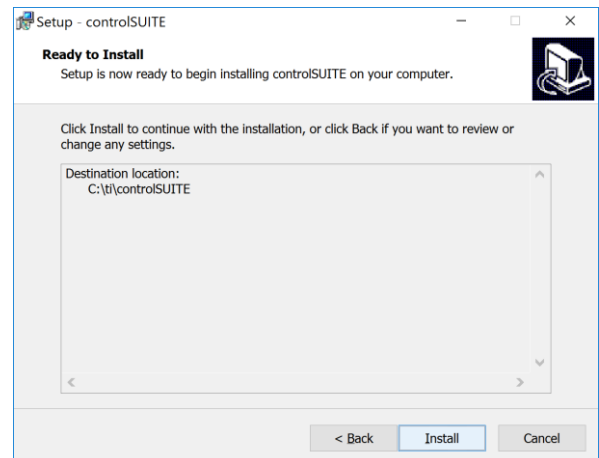
Step 1



Step 3



Step 2



Step 4

3. Once controlSUITE is installed, Code Composer can be set up to work with the TDINV3000W050 source code

Setting up Code Composer Studio to work with TDINV3000W050-KIT

4. Create a folder to use as a workspace, i.e., C:\transphorm\TDINV3000W050
5. Launch Code Composer Studio and select the desired workspace
6. Go to Project in the menu and select Import Existing CSS Eclipse Project
 - a. Navigate to the [firmware files](#), then locate TDINV3000W050_REV_01
 - b. Click Finish
7. Go to Project in the menu and select CLEAN which will perform a full build of the project

TDINV3000W050 Firmware Guide

- a. After finishing the build, the files will appear under Project Explorer. The TMS320F28035 will load as the target processor—to verify, click Target Configurations under the View menu item, and then select the SolarHv_DCAC.ccxml under Projects.

Load and run the code

8. Make sure jumper switch SW3 is in the down position on the control card to allow communication over the JTAG port. Once the code is written to flash, JP1 should be removed to allow booting from flash.
 - a. Connect a USB cable from the computer to the control card on the TDINV3000W050 inverter board (CN2)
 - b. Connect 12V_{CC} power adaptor to the inverter (J1)
9. After completion of the build, click the Debug button, located on the top-left side of screen. The IDE will now automatically connect to the target, load the output file into the device and change to the Debug perspective
10. Run the code by pressing the RUN button in the Debug tab

TDINV3000W050 Firmware Guide

Part 2: Working with the code

The Open-Loop build option is the only option currently proven and supported with this board. The alternate build options in the TI code (Closed Loop without PLL and Closed Loop with PLL) should be compatible with this board, but have not been verified.

The original code is named HV_SOLAR_DC_AC_v1.1, and can be imported from:

\controlSUITE\development_kits\HV_SOLAR_DC_AC_v1.1\SolarHv_DCAC_PiccoloB_Rev_02

Changing the switching frequency

The #define statements in SolarHv_DCAC-Settings.h are used to set the switching frequency. Three switching frequencies are provided, between 50kHz and 200kHz. There are four frequency-dependent parameters that need to be adjusted: the period, the decimation count, the sample time, and the amplitude control. Other frequencies can be generated following the same model. The decimation count is used to keep the loop update rate at 20kHz, regardless of switching frequency. The update rate can also be changed using the same parameters, but be careful to ensure that all computations can be completed in the time desired.

```
32 #define TRANSPHORM_SWITCHING_FREQUENCY_50KHZ 1 // 50kHz switching frequency
33 // #define TRANSPHORM_SWITCHING_FREQUENCY_100KHZ 1 // 100kHz switching frequency
34 // #define TRANSPHORM_SWITCHING_FREQUENCY_200KHZ 1 // 200kHz switching frequency
35
36 /* Parameters specific to switching frequency */
37 #if TRANSPHORM_SWITCHING_FREQUENCY_50KHZ
38 #define TRANSPHORM_PWM_PRD 1200
39 #define TRANSPHORM_LOOP_DECIMATION_COUNT 2.5 // Perform ADC conversions and run control loop every 50/2.5 = 20kHz
40 #define TRANSPHORM_SAMPLE_TIME (0.00002 * TRANSPHORM_LOOP_DECIMATION_COUNT) // 20e-6 * 2.5
41 #define TRANSPHORM_SIN_AMP 0.156
42 #elif TRANSPHORM_SWITCHING_FREQUENCY_100KHZ
43 #define TRANSPHORM_PWM_PRD 600
44 #define TRANSPHORM_LOOP_DECIMATION_COUNT 5 // Perform ADC conversions and run control loop every 100/5 = 20kHz
45 #define TRANSPHORM_SAMPLE_TIME (0.00001 * TRANSPHORM_LOOP_DECIMATION_COUNT) // 10e-6 * 5
46 #define TRANSPHORM_SIN_AMP 0.078
47 #elif TRANSPHORM_SWITCHING_FREQUENCY_200KHZ
48 #define TRANSPHORM_PWM_PRD 300
49 #define TRANSPHORM_LOOP_DECIMATION_COUNT 10 // Perform ADC conversions and run control loop every 200/10 = 20kHz
50 #define TRANSPHORM_SAMPLE_TIME (0.000005 * TRANSPHORM_LOOP_DECIMATION_COUNT) // 5e-6 * 10
51 #define TRANSPHORM_SIN_AMP 0.039
52 #endif
53
54 #define TRANSPHORM_LOOP_PRD (TRANSPHORM_PWM_PRD * TRANSPHORM_LOOP_DECIMATION_COUNT) // Decimate the control loop
55
```

Changing the output frequency

For open-loop operation, the output frequency is controlled by parameter dwPLL_Trace_Freq. in SolarHv_DCAC-main.c allow simple selection of 50 or 60Hz:

```
916
917 /* Uncomment one value below for dwPll_Trace_Freq depending on desired grid frequency */
918 dwPll_Trace_Freq = _IQ20(376.9911); //60Hz
919 //dwPll_Trace_Freq = _IQ20(314.1593); //50Hz
920
```

TDINV3000W050 Firmware Guide

Part 3: Protection Features

Under Voltage Lock Out

The TDINV3000W050 evaluation board supports an under-voltage lockout feature. Be sure the feature is enabled if used. The board will blink red (LED LD2) on the control card in the event the board is in under-voltage and the output will be held off. When the under-voltage threshold is crossed, the output will be on and LED LD2 will be steady. The hysteresis value applies to the under-voltage trip threshold.

```
10 #define UNDER_VOLTAGE_ENABLE    0
11 #define UNDER_VOLTAGE_TRIP      10
12 #define HYSTERESIS_SET           5
13 #define OVER_VOLTAGE_TRIP        420
```

Over Voltage Shut Down

The over-voltage feature is always enabled so be sure the overvoltage trip level is set high enough. In this example, the over-voltage threshold is 420V. In the event of an over-voltage event both red LEDs on the control card will blink in unison. The board must be powered down to recover from this type of event.

```
10 #define UNDER_VOLTAGE_ENABLE    0
11 #define UNDER_VOLTAGE_TRIP      10
12 #define HYSTERESIS_SET           5
13 #define OVER_VOLTAGE_TRIP        420
```

Current Limit Shut Down

The TDINV3000W050 evaluation board supports user-defined current limit thresholds in both the positive- and negative-going directions. The current limits are peak and in amps. In this example limits are set at 6A in both the positive- and negative- going directions. In most cases these trip points would be set to the same value. In an over-current event both red LEDs on the control card will blink in unison. The board must be powered down to recover from this type of event.

```
15 #define COMPARATOR_MAX           20
16 #define COMPARATOR_MIN           20
```