Firmware Guide

TDINV3500P100: 3.5kW Inverter Evaluation Board

Getting started with the code

The design files and source code for the TDINV3500P100_0v1 3.5kW inverter can be found at transphormusa.com/pv35kit 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 TDINV3500P100 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.

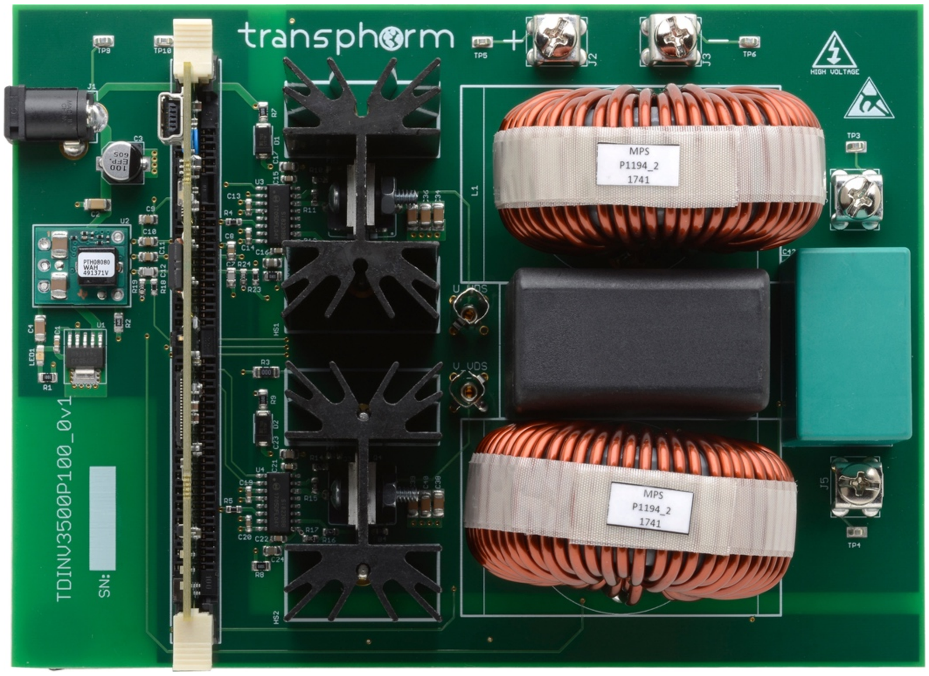
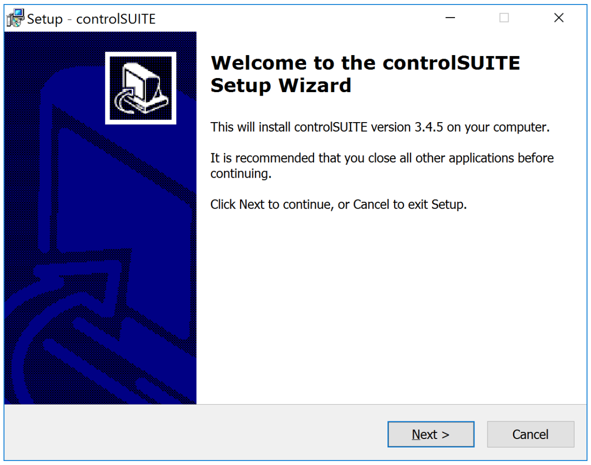


Figure 1. TDINV3500P100_0v1 3.5kW inverter evaluation board

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:



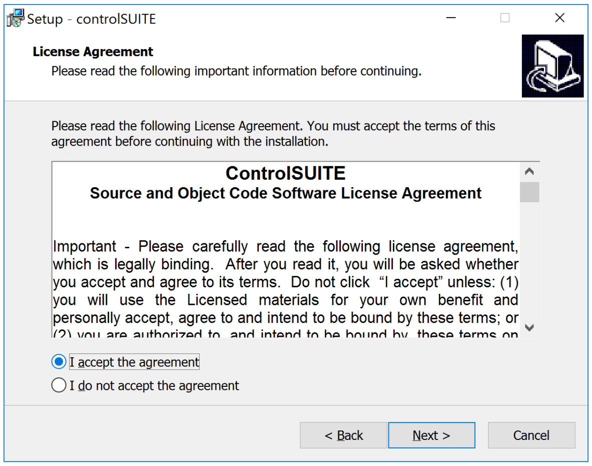
Welcome to the controlSUITE Setup Wizard

This will install controlSUITE version 3.4.5 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

Step 1



License Agreement

Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

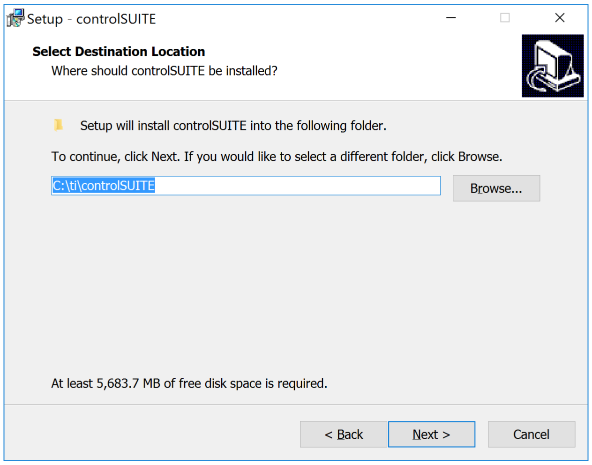
ControlSUITE
Source and Object Code Software License Agreement

Important - Please carefully read the following license agreement, which is legally binding. After you read it, you will be asked whether you accept and agree to its terms. Do not click "I accept" unless: (1) you will use the Licensed materials for your own benefit and personally accept, agree to and intend to be bound by these terms; or (2) you are authorized to and intend to be bound by these terms on

● I accept the agreement

○ I do not accept the agreement

Step 2



Select Destination Location

Where should controlSUITE be installed?

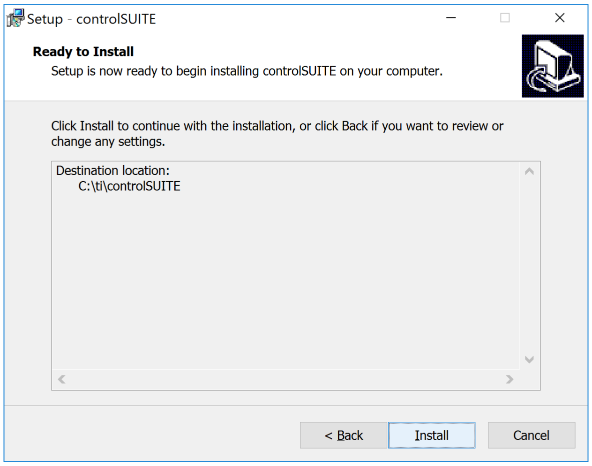
Setup will install controlSUITE into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\ti\controlSUITE

At least 5,683.7 MB of free disk space is required.

Step 3



Ready to Install

Setup is now ready to begin installing controlSUITE on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

Destination location:
C:\ti\controlSUITE

Step 4

3. Once controlSUITE is installed, Code Composer can be set up to work with the TDINV3500P100 source code

Setting up Code Composer Studio to work with TDINV3500P100-KIT

4. Create a folder to use as a workspace, i.e., C:\transphorm\TDINV3500P100
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 TDINV3500P100_REV_01
   b. Click Finish
7. Go to Project in the menu and select CLEAN which will perform a full build of the project
   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.
   b. Connect a USB cable from the computer to the control card on the TDINV3500P100 inverter board (CN2)
   c. 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

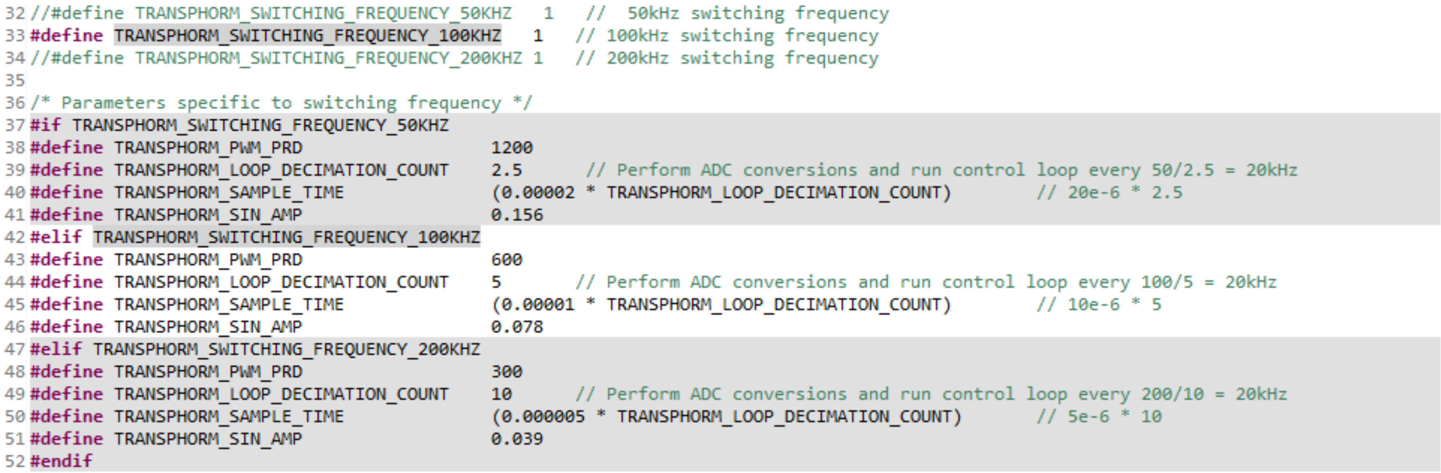
Working with the code

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

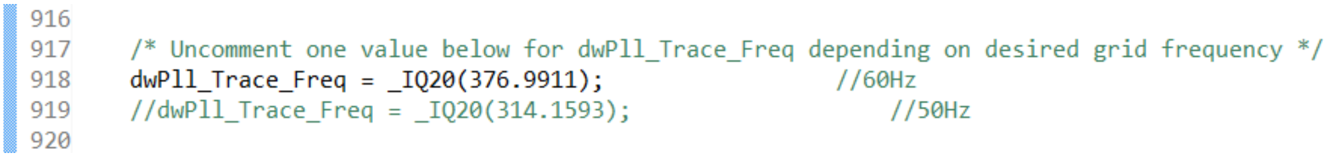
The #define statements in SolarHv_DCAC-Settings.h are used to set the switching frequency. Three switching frequencies are provided, between 50kHz and 200kHz. There are four frequency-dependent parameters that need to be adjusted: the period, the decimation count, the sample time, and the amplitude control. 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.



```
//#define TRANSPHORM_SWITCHING_FREQUENCY_50KHZ   1   //  50kHz switching frequency
#define TRANSPHORM_SWITCHING_FREQUENCY_100KHZ   1   // 100kHz switching frequency
//#define TRANSPHORM_SWITCHING_FREQUENCY_200KHZ 1   // 200kHz switching frequency

/* Parameters specific to switching frequency */
#if TRANSPHORM_SWITCHING_FREQUENCY_50KHZ
#define TRANSPHORM_PWM_PRD                  1200
#define TRANSPHORM_LOOP_DECIMATION_COUNT    2.5     // Perform ADC conversions and run control loop every 50/2.5 = 20kHz
#define TRANSPHORM_SAMPLE_TIME              (0.00002 * TRANSPHORM_LOOP_DECIMATION_COUNT)        // 20e-6 * 2.5
#define TRANSPHORM_SIN_AMP                  0.156
#elif TRANSPHORM_SWITCHING_FREQUENCY_100KHZ
#define TRANSPHORM_PWM_PRD                  600
#define TRANSPHORM_LOOP_DECIMATION_COUNT    5      // Perform ADC conversions and run control loop every 100/5 = 20kHz
#define TRANSPHORM_SAMPLE_TIME              (0.00001 * TRANSPHORM_LOOP_DECIMATION_COUNT)        // 10e-6 * 5
#define TRANSPHORM_SIN_AMP                  0.078
#elif TRANSPHORM_SWITCHING_FREQUENCY_200KHZ
#define TRANSPHORM_PWM_PRD                  300
#define TRANSPHORM_LOOP_DECIMATION_COUNT    10     // Perform ADC conversions and run control loop every 200/10 = 20kHz
#define TRANSPHORM_SAMPLE_TIME              (0.000005 * TRANSPHORM_LOOP_DECIMATION_COUNT)       // 5e-6 * 10
#define TRANSPHORM_SIN_AMP                  0.039
#endif
```

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 50Hz or 60Hz.

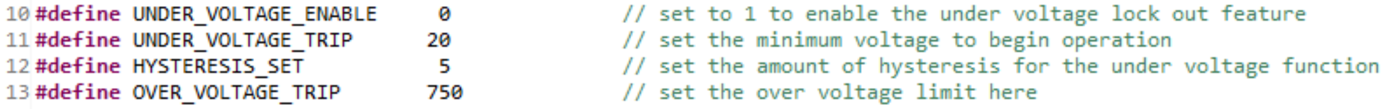


```

    /* Uncomment one value below for dwPll_Trace_Freq depending on desired grid frequency */
    dwPll_Trace_Freq = _IQ20(376.9911);                 //60Hz
    //dwPll_Trace_Freq = _IQ20(314.1593);                   //50Hz

```

Protection features

Under-voltage lockout

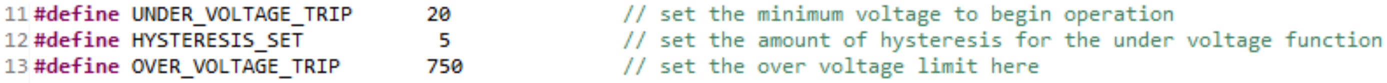
The TDINV3500P100 evaluation board supports an under-voltage lockout feature. Be sure the feature is enabled if used. The board will blink red (LED LD2) on the control card in the event the board is in under-voltage and the output will be held off. When the under-voltage threshold is crossed, the output will be on and LED LD2 will be steady. The hysteresis value applies to the under-voltage trip threshold.



```
#define UNDER_VOLTAGE_ENABLE     0              // set to 1 to enable the under voltage lock out feature
#define UNDER_VOLTAGE_TRIP      20              // set the minimum voltage to begin operation
#define HYSTERESIS_SET           5              // set the amount of hysteresis for the under voltage function
#define OVER_VOLTAGE_TRIP      750              // set the over voltage limit here
```

Over-voltage shutdown

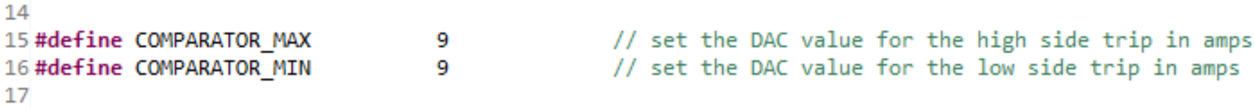
The over-voltage feature is always enabled so be sure the overvoltage trip level is set high enough. In this example, the over-voltage threshold is 750V. In the event of an over-voltage event both red LEDs on the control card will blink in unison. The board must be powered down to recover from this type of event.



```
#define UNDER_VOLTAGE_TRIP      20              // set the minimum voltage to begin operation
#define HYSTERESIS_SET           5              // set the amount of hysteresis for the under voltage function
#define OVER_VOLTAGE_TRIP      750              // set the over voltage limit here
```

Current limit shutdown

The TDINV3500P100 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 6.3A 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.



```

#define COMPARATOR_MAX           9              // set the DAC value for the high side trip in amps
#define COMPARATOR_MIN           9              // set the DAC value for the low side trip in amps

```