Digitally Controlled HV Solar MPPT DC-DC Converter

User's Guide

Literature Number: TIDU404
September 2014

Contents

List of Figures

List of Tables

User's Guide
TIDU404–September 2014

Digitally Controlled HV Solar MPPT DC-DC Converter

This guide details how to implement a digitally controlled DC-DC converter that is used as a front-end converter for solar inverter (DC-AC) application. This converter implements an isolated DC-DC stage with maximum power point tracking (MPPT) algorithm to use the full capacity of a 500-W solar panel. The panel maintains its input voltage at the reference set point generated by the MPPT algorithm and delivers power to a downstream DC-AC inverter when connected across its output. The DC-AC inverter transfers the power from the DC-DC stage to an emulated grid connected across its own output. A C2000™ Piccolo™-B control card and a 500-W isolated DC-DC stage EVM are used to implement the complete DC-DC system.

This EVM comes with a Piccolo-B control card and not the Piccolo-A card. However, a Piccolo-A control card can also be used to implement full control of the EVM.

1 Introduction

Photovoltaic (PV) systems based on solar energy offer an environmentally friendly source of electricity. A key feature of such PV systems is the efficiency of conversion at which the power converter stage can extract the energy from the PV arrays and deliver to the load. The MPPT of the PV output for all sunshine conditions reduces the cost of installation and maximizes the power output from the PV panel. Therefore, a DC-DC converter employing some MPPT algorithm is generally used as a front-end converter to efficiently extract the PV output power and convert the PV output voltage to a high voltage DC-BUS. Depending on the system requirement, the DC-DC converter can use either an isolated power stage or a nonisolated stage. The high voltage bus from the DC-DC converter is then fed to power the DC-AC inverter that eventually supplies the load and connects to the grid.

This C2000 MPPT DC-DC EVM uses an isolated DC-DC stage as is shown in Figure 1. The EVM consists of two DC-DC stages: a 2-ph interleaved boost converter, and an isolated half-bridge LLC resonant converter.

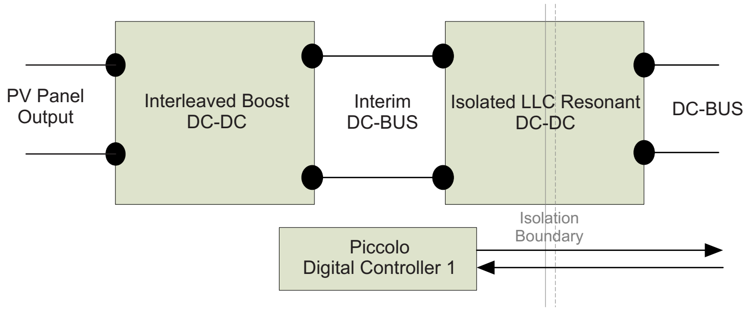


Figure 1. Isolated DC-DC Converter Block Diagram

The DC-DC converter draws DC current from the PV panel so the panel operates at its maximum power transfer point. This action requires maintaining the panel output (for example, the DC-DC converter input at a level determined by the MPPT algorithm), which happens in the 2-ph interleaved boost converter stage. The isolated LLC resonant converter simply isolates the high frequency for the DC-DC stage.

A C2000 Piccolo microcontroller with its on-chip PWM, ADC, and analog comparator modules is able to digitally control the MPPT DC-DC system completely.

C2000, Piccolo, Code Composer Studio, TMS320C2000 are trademarks of Texas Instruments.
All other trademarks are the property of their respective owners.

1.1 DC-DC Stage Implementation

Figure 2 illustrates a C2000-based MPPT DC-DC converter control system. The PV panel output voltage (Vpv) is applied to the 2-ph interleaved boost stage.

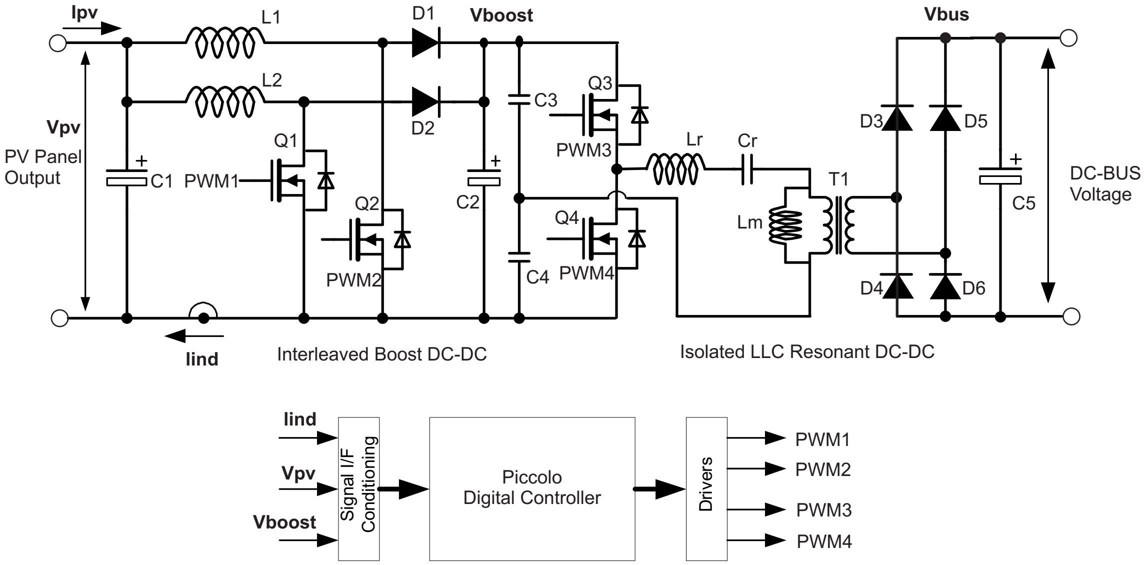


Figure 2. MPPT DC-DC Converter Control using C2000 Microcontroller

Inductor L1, MOSFET Q1, and diode D1 together form one of the boost stages while, L2, Q2, and D2 form the other. The capacitor C2 at the boost converter output acts as an energy reservoir and provides boost voltage to the resonant LLC stage.

The H-bridge LLC resonant stage consists of MOSFETs Q3 to Q4, input capacitors C3 to C4, resonant inductor Lr, resonant capacitor Cr, transformer T1, output rectifiers D3 to D6, and output capacitor C5. This stage has a voltage ratio of 1 and provides the isolation between the primary and secondary side.

Figure 2 indicates all the interface signals needed for full control of this DC-DC converter using a C2000 microcontroller (MCU). The MCU controls the hardware using three feedback signals and four PWM outputs. The signals that are sensed and fed back to the MCU include the Vpv, the boost output voltage (Vboost), and the total boost inductor currents (Iind). These sensed signals implement the voltage and current control loops for the DC-DC boost stage. The interleaved boost DC-DC topology is chosen to boost the variable DC output to a fixed DC-BUS voltage. The main reason for using this topology is the wide input voltage variation. The PWM signals for the power switches Q1 and Q2 is phase-shifted by 180 degrees. This phase-shift helps reduce the ripple in the PV panel current.

The LLC stage runs at open loop with its PWM frequency set to be the same as the resonant frequency. The Piccolo controller shares the common ground with the primary side of the LLC stage, and the controller does not receive isolated feedback from the LLC secondary output terminals. Thus, the LLC is run under open loop and is necessary to maintain a voltage conversion factor of 1. The LLC achieves this (1) by making the LLC PWM frequency the same as the resonant frequency, and (2) by maintaining a minimum load of about 10 W across the LLC output.

Figure 3 shows the DC-DC interleaved boost converter control loops. These loops use current mode control. However, the goal is to control the PV panel output (Vpv), which is the input to the DC-DC stage and allows the PV panel (array) to operate at its maximum power point at all times. Regulate input current by adjusting the duty cycles of the power switches Q1 and Q2. Regulate input voltage by adjusting the input current. A MPPT algorithm described in Section 1.2 is responsible for determining the set point (Vpv_ref) for the PV panel voltage. Notice that the input voltage control loop works quite differently compared to conventional feedback used in output voltage control. Under this control scheme, when the PV panel voltage tends to go higher than the reference panel voltage set by the MPPT algorithm, the control loop increases the panel current command (reference current for inner current loop Iind_ref) and thereby controls the panel voltage at its reference level (Vpv_ref). When the panel voltage tends to go lower than the reference, the control loop reduces the panel current command in order to reestablish the panel voltage to its reference level.

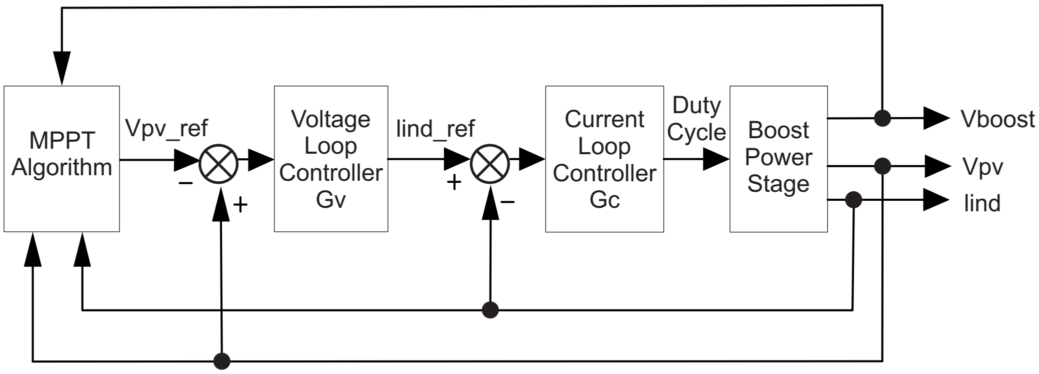


Figure 3. MPPT DC-DC Converter Control Loops

The panel voltage Vpv, sensed through one of the ADC channels, is compared against the reference voltage Vpv_ref set by the MPPT algorithm. The resulting error signal (Ev) is then input the voltage loop controller (Gv), which regulates the panel voltage at the reference level. The voltage controller Gv has the form of a two-pole two-zero (2P2Z) compensator. The output of Gv is the reference current command for the inner inductor current loop. The average value of the inductor current is the panel current Ipv. Therefore, by controlling the average value of the inductor current, the current controller Gc essentially controls the panel current.

This reference current command Iind_ref for the current control loop is compared against the feedback inductor current Iind sensed through another ADC channel. The resulting current error signal is then input the current loop controller Gc which generates the boost converter PWM duty ratio command for the boost switches Q1 and Q2.

In addition to implementing the voltage and current loop controllers, the C2000 MCU also monitors the boost output voltage for over voltage protection. The ADC channel that monitors the boost voltage has an internal analog comparator with user programmable threshold. This threshold for the comparator is set by use of an internal 10-bit DAC. Whenever the DC-BUS voltage reaches an upper limit corresponding to the user programmable comparator threshold, the comparator initiates a pulse-by-pulse duty limit for the boost PWM signals. This threshold limits the boost inductor current and, as a result, the boost bus voltage to its desired upper limit.

The C2000 MCU also generates two PWM outputs to drive the isolated LLC stage. This stage runs in an open loop fashion with unity voltage conversion ratio (voltage gain), which means the boost voltage and the LLC output voltage Vbus is almost equal. However, this stage requires a minimum load of about 10 W across Vbus (16 kΩ at 400 V). With no load connected across Vbus and the boost output voltage set to 400 V, the LLC stage gain might be higher than 1, resulting in high voltage across LLC output (Vbus). The user must prevent this condition by always maintaining a minimum load resistor of about 16 kΩ across the LLC output (Vbus).

All the time critical functions related to the DC-DC control loops are implemented in a fast sampling loop enabled by the C2000 microcontroller high speed CPU, interrupts, on chip 12-bit ADC module and high frequency PWM modules. A detailed description of the software algorithm is provided in Section 2.

1.2 *DC-DC Electrical Specifications*

The following lists the key highlights of the C2000 MPPT DC-DC EVM:

- Panel Voltage: 200 (Minimum) to 300 V (Maximum)
- 400-V DC Output
- 500-W Output Power
- Full load efficiency greater than 94%

2 Software Overview

2.1 Software Control Flow

The Code Composer Studio™ (CCS) project for the C2000 MPPT DC-DC mostly makes use of the “Cbackground/ ASM-ISR” framework. The main fast interrupt service routine (ISR; 50 kHz) runs in assembly environment. However, a slower ISR (10 kHz) also runs from a C environment. This slow ISR is made interruptible by the fast ISR. Also, a third ISR runs from the C environment at a much slower frequency to implement the local interconnect network (LIN) based communication with the DC-AC inverter stage. The frequency of the LIN interrupt is set by the inverter at 50 Hz.

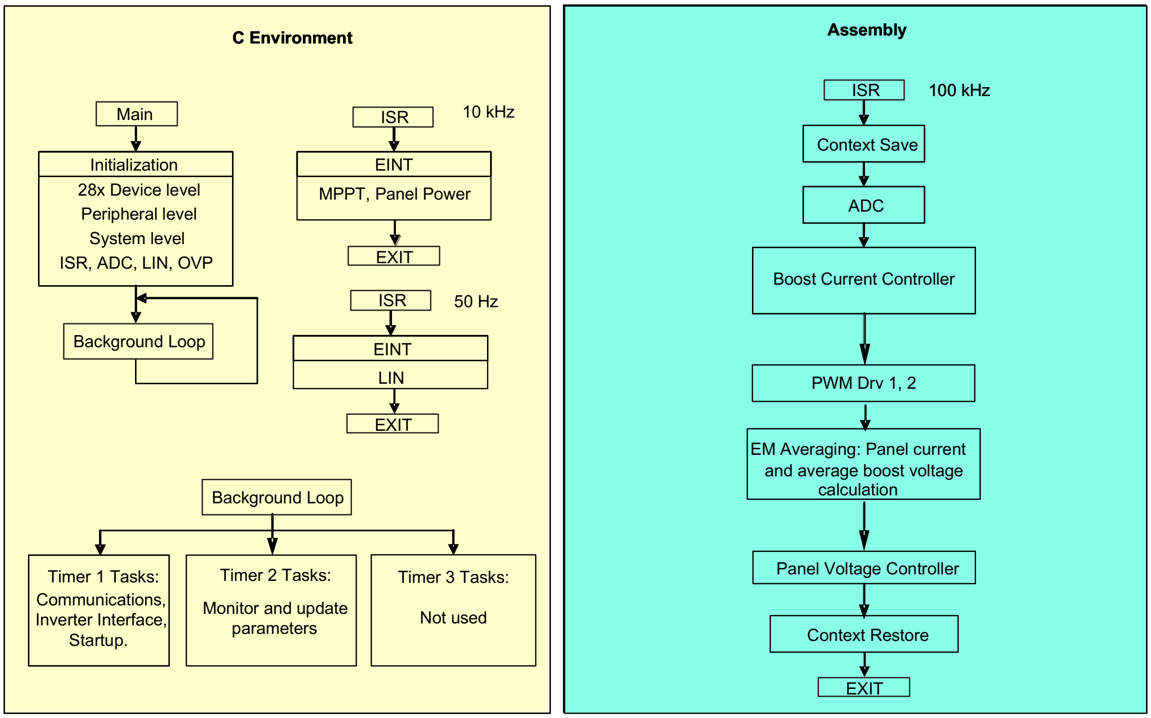


Figure 4. MPPT DC-DC Software Flow Diagram

The project uses C-code as the main supporting program for the application and is responsible for all system management tasks, decision making, intelligence, and host interaction. The assembly code is strictly limited to the fast ISR, which runs all the critical control code. Typically, this critical control code includes reading ADC values, control calculations, and PWM updates. The slower ISRs in the C environment implement the MPPT algorithm to calculate the reference PV panel voltage and establish LIN communication with the inverter stage. Figure 4 depicts the general software flow for this project.

The key framework C files used in this project are:

HV_Solar_DC-DC-Main.c— This file initializes, runs, and manages the application.

HV_Solar_DC-DC -DevInit_F2803x.c— This file is responsible for a one-time initialization and configuration of the F280xx device, and includes functions such as setting up the clocks, PLL, GPIO, and so on. This file comes from the control card (2803x) used in the MPPT DC-DC EVM, and one of these files will be in the CCS project.

The fast ISR consists of a single file:

HV_Solar_DC-DC-DPL-ISR.asm— This file contains the all-time critical “control type” code. This file has an initialization section (one-time execute) and a run-time section thati executes at half the rate (50 kHz) as the PWM time-base (100 kHz) used to trigger it.

The slow ISR that runs at 10 kHz consists of two files. The user selects one of the two files to implement the MPPT algorithm:

Mppt_incc.h— This file contains code for calculating the panel voltage for MPPT using the incremental conductance method. This file has an initialization section (one-time execute) and a run-time section that executes at a 10-kHz rate.

Mppt_pno.h— This file contains code for calculating the panel voltage for MPPT using the perturb and observe method. This file has an initialization section (one-time execute) and a run-time section that executes at a 10-kHz rate.

The second slow ISR that runs at 50 Hz consists of one file:

SolarHv_DCDC_Lin.C— This file contains code for establishing LIN communication with the inverter stage.

The power library functions (or modules) are “called” from the fast ISR framework.

These power library modules may have both a C and an assembly component. In this project, five library modules are used. The C and corresponding assembly module names are listed in Table 1:

Table 1. Library Modules

C CONFIGURE FUNCTION	ASM INITIALIZATION MACRO	ASM RUN-TIME MACRO
PWM_1ch_UpDwnCnt_Cnf.c	PWMDRV_1ch_UpDwnCnt_INIT n	PWMDRV_1ch_UpDwnCnt n
ADC_SOC_Cnf.c	ADCDRV_1ch_INIT m,n,p,q	ADCDRV_1ch m,n,p,q
PWM_CompPairDB_Cnf.C		
	MATH_EMAVG_INIT n	MATH_EMAVG n
	CNTL_2P2Z_INIT n	CNTL_2P2Z n

The assembly modules can also be represented graphically as shown in Figure 5.

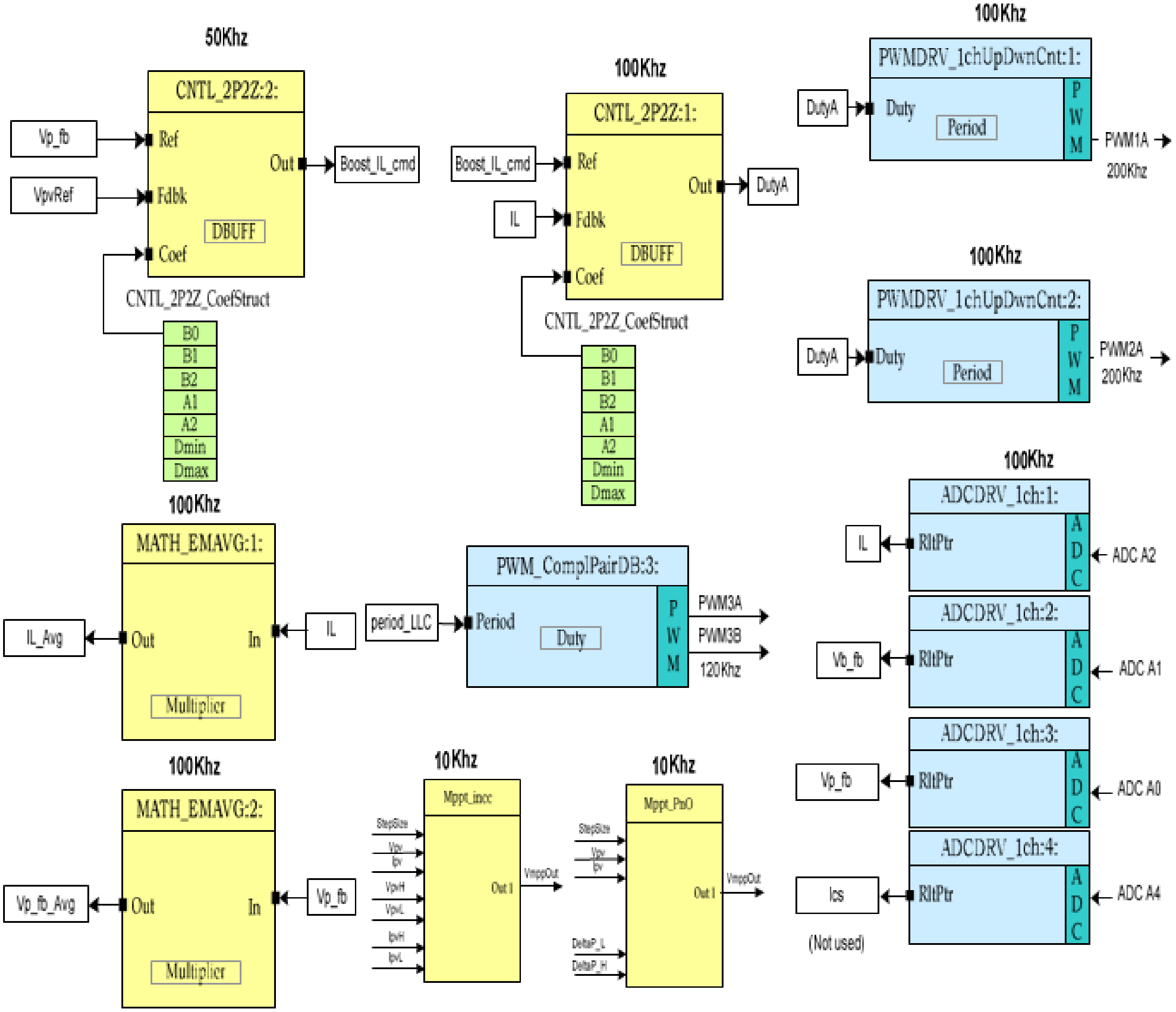


Figure 5. Software Blocks

Note the color coding used for the modules in Figure 5. The blocks in dark blue represent the on-chip hardware modules in the C2000 controller. The blocks in blue are the software drivers associated with these modules. The blocks in yellow are part of the computation carried out on various signals. The controllers used for voltage and current loops have the form of a 2P2Z compensator. However, these controllers can be of other forms such as PI, PID, 3-pole 3-zero or any other controller suitable for the application.

The modular library structure makes it convenient to visualize and understand the complete system software flow as shown in Figure 6. The structure also allows for easy use and additions or deletions of various functionalities. This fact is demonstrated in this project by implementing an incremental build approach. This approach is discussed in more detail in the next section.

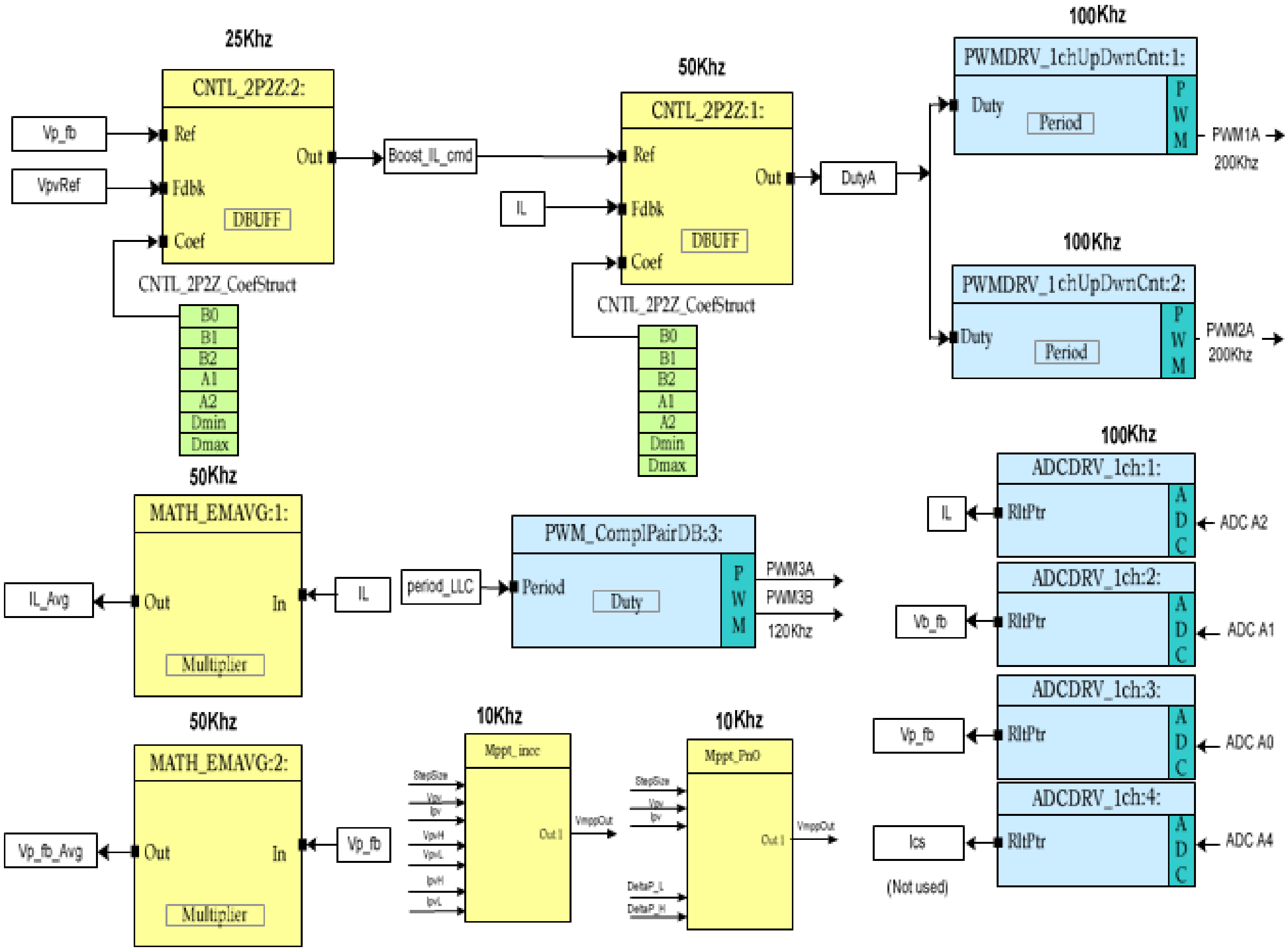


Figure 6. Software Control Flow

As mentioned in Section 1.1, the MPPT DC-DC system is controlled by two feedback loops. The outer voltage loop maintains the panel voltage at the level calculated by the MPPT algorithm while a faster inner current loop controls the average boost inductor current. Figure 6 also gives the rate the software modules are executed. For example, the current controller is executed at a rate of 50 kHz (half of the PWM switching frequency) while the voltage controller is executed at a rate of 25 kHz.

2.2 Incremental Builds

This project is divided into three incremental builds. This approach provides the user with a step-by-step method to get familiar with the software and understand how it interacts with the MPPT DC-DC hardware. This approach also simplifies the task of debugging and testing the boards.

The build options are shown in Table 2. To select a particular build option, set INCR_BUILD, found in the HV_Solar_DC-DC-Settings.h file, to the corresponding build selection in Table 2. Once the build option is selected, compile the complete project by selecting the Rebuild-All Compiler option. Section 3 provides more details on how to run each of the build options.

Table 2. Incremental Build Options for MPPT DC-DC

INCREMENTAL BUILD OPTIONS	DETAILS
INCR_BUILD = 1	Open loop check for boost, LLC action, and ADC feedback (Check sensing circuitry)
INCR_BUILD = 2	Open voltage loop and closed current loop control of boost
INCR_BUILD = 3	Closed voltage and current loop control of boost with MPPT

3 Procedure for Running the Incremental Builds

All software files related to this C2x controlled MPPT DC-DC system (for example, the main source files, ISR assembly files and the project file for C framework) are located in the directory …\controlSUITE\development_kits\MPPT DC-DC_v1.0\MPPT DC-DC. The projects included with this software are targeted for CCSv4.

CAUTION

This board produces high voltages and should only be handled by experienced power supply professionals in a lab environment. To safely evaluate this board, use a PV panel emulator with an appropriate power rating to power the unit. Before power is applied to the board, attach a voltmeter and an appropriate resistive or electronic load to the output. These additions will discharge the bus capacitor quickly when the PV power is turned off. The board has no output overcurrent protection implemented on the board, so take appropriate measures for preventing any output short circuits.

Follow the steps in Section 3.1, Section 3.2, and Section 3.3 to build and run the examples included in the DC-DC software.

3.1 Build 1: Open Loop Boost With ADC Measurements

3.1.1 Objective

The objectives of this build are:

- To evaluate MPPT DC-DC PWM and ADC software driver modules
- To verify MOSFET gate driver circuit, voltage, and current sensing circuit
- To become familiar with the operation of CCS. Under this build, the system runs in open-loop mode, so the measured ADC values are used for circuit verification and instrumentation purposes only.

The following section explain the steps required for building and running a CCS project.

3.1.2 Overview

The software in Build 1 has been configured so the user can quickly evaluate the PWM driver module by viewing the related waveforms on a scope and observing the effect of duty cycle change on DC-DC output voltage. From the CCS watch window, either adjust the PWM duty cycle or evaluate the ADC driver module by viewing the ADC sampled data.

The PWM and ADC driver macro instantiations are executed inside the _DPL_ISR. Figure 7 shows the software blocks used in this build. The two PWM signals for the two BOOST switches are obtained from ePWM module 1 and 2. ePWM1A drives one of the BOOST switches while ePWM2A drives the other. The two PWM signals for the two LLC stage switches are obtained from ePWM module 3. ePWM3A drives the upper (high side) LLC switch while ePWM3B drives the low side switch.

The quantities that are sensed and fed back to the MCU include the panel voltage (Vp_fb), the combined BOOST inductor current (IL), and the boost DC bus voltage (Vb_fb). These quantities are read using the ADC driver module and are indicated in Figure 7. The fourth channel for the ADC driver macro is not used in this application. The ADC driver module converts the 12-bit ADC result to a 32-bit Q24 value.

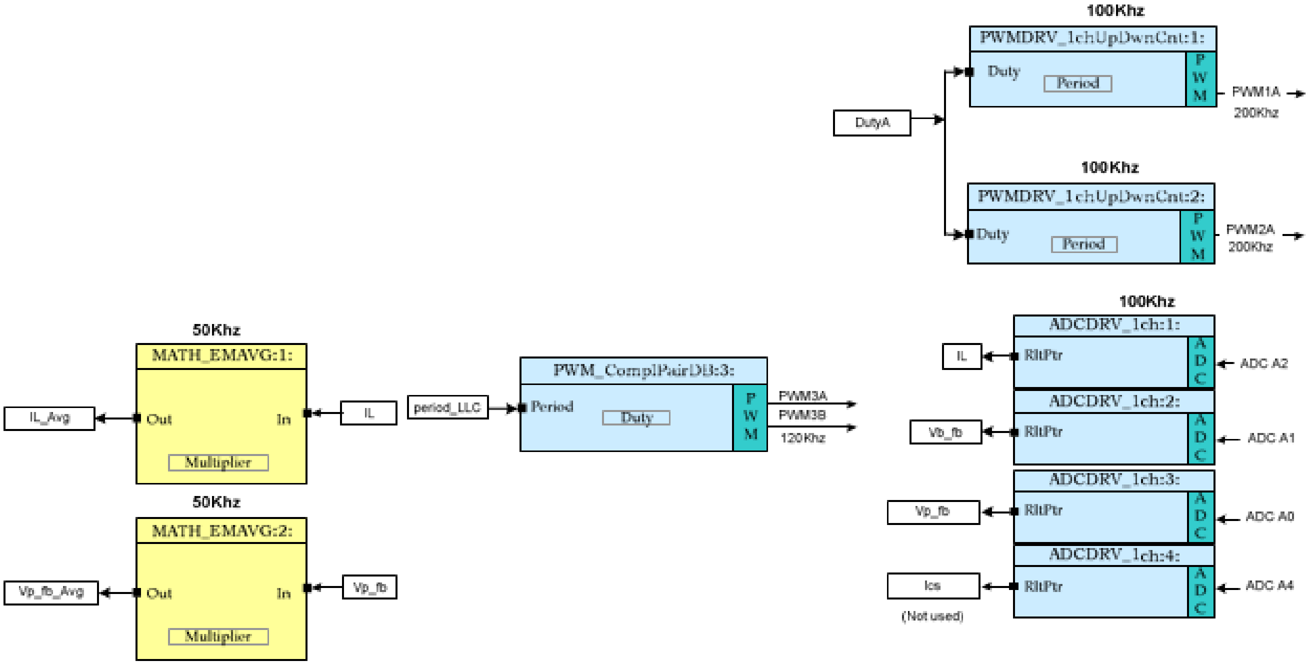


Figure 7. Build 1 Software Blocks

The boost PWM signals are generated at a frequency of 100 kHz, that is a period of 10 µs. With the controller operating at 60 MHz, one count of the time base counter of ePWM1 corresponds to 16.6667 ns. This count implies a PWM period of 10 µs is equivalent to 600 counts of the time base counter (TBCNT1, TBCNT2). The ePWM1 and ePWM2 modules are configured to operate in up-down count mode as shown in Figure 8. This count means a time base period value of 300 (period register value) will give a total PWM period value of 600 counts (or 10 µs).

The LLC PWM signals are generated at a frequency of 120 kHz, that is a period of 8.33 µs. With the controller operating at 60 MHz, one count of the time base counter of ePWM3 corresponds to 16.6667 ns. This count implies a PWM period of 8.33 µs is equivalent to 500 counts of the time base counter (TBCNT3). The ePWM3 module is configured to operate in up-down count mode. This count means a time base period value of 250 (period register value) will give a total PWM period value of 500 counts (or 8.33 µs).

The BOOST inductor current is sampled at the midpoint of the PWM1A ON pulse because the sampled value represents the average inductor current under continuous conduction mode (CCM) condition. Under DCM condition, this sampled current value represents a fraction of the average inductor current.

The other two voltage signal conversions are also initiated at this time. This is indicated in Figure 8. The flexibility of ADC and PWM modules on C2000 devices allow for precise and flexible ADC start of conversions. In this case, ePWM1 is used as a time base to generate a start of conversion (SOC) trigger when the TBCNT1 reaches zero. A dummy ADC conversion is performed at this point to ensure the integrity of the ADC results.

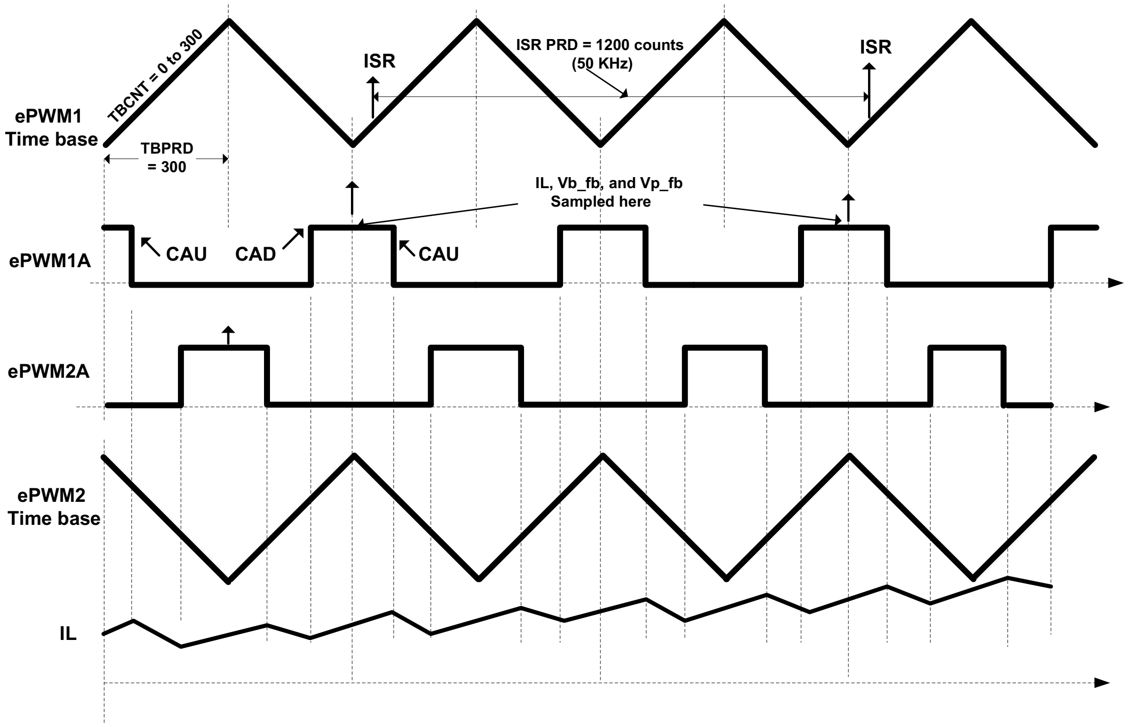


Figure 8. PWM Generation and ADC Sampling

On a CAU event (TBCNT1 = CMPA and counting up), ePWM1A output is Reset, while on a CAD event (TBCNT1 = CMPA and counting down), ePWM1A output is Set. ePWM2A is also configured in a similar manner with a 180 phase shift with respect to ePWM1A.

The CMPA value is derived from the input BOOSTDuty (Q24 variable) command.

The ADC module is configured to use SOCA of ePWM1 so SOCA is triggered at the TBCNT1 = ZERO event. All conversions are completed using this SOCA trigger. These three ADC results are read in the ISR by executing the ADC driver module from the 50-kHz ISR labeled as _DPL_ISR.

This ISR in assembly (_DPL_ISR) is triggered by EPWM1 on a CMPB match event on the up count. CMPB is set to 80 so that the ISR is triggered only after the ADC conversions are complete. This trigger is where the PWMDRV_1ch_UpDwnCnt macros are executed and the PWM compare shadow registers updated. These macros and registers are loaded into the active register at the next TBCNT = ZERO event.

NOTE: The ISR trigger frequency is half that of the PWM switching frequency as shown in Figure 8.

3.1.3 Protection

A mechanism for an overvoltage protection (OVP) is implemented in the software for this MPPT DC-DC EVM. This OVP applies only for the boost output and not for the LLC stage output. Since the Piccolo controller is on the primary side of the isolation, the controller has no knowledge of the isolated LLC output. Therefore, connect a minimum load of 10 W across the LLC output to make the open loop LLC output follow its input voltage, that is the output voltage from the boost stage. The minimum load across the LLC output helps maintain the LLC stage voltage conversion factor of 1. This way the max LLC output will also be limited by the max boost output, which in turn is protected by the OVP mechanism.

Compare the sensed boost stage DC-BUS output voltage from the ADC input against the OVP threshold set. The default OVP threshold set point is 404 V. This threshold parameter is programmed inside the file HV_SOLAR_DC_DCMain.h. This programming is done by the following initialization of the on-chip DAC reference voltage.

Comp3Regs.DACVAL.bit.DACVAL = 808;

Since the 10-bit DAC full scale value represents a max voltage of 511 V, the initial value of 808 represents the OVP threshold of 404 V.

In case of an overvoltage condition, the PWM outputs are duty limited pulse by pulse using the trip zone (TZ) registers. The flexibility of the trip mechanism on C2000 devices provides the possibilities for taking different actions on different trip events.

3.1.4 Procedure

3.1.4.1 Start CCS and Open a Project

Use the following steps to execute this build:

1. Connect the USB connector to the Piccolo controller board for emulation. Power up the 12-V bias supply at JP1. By default, the Piccolo control card jumpers (see the Piccolo control card documentation for your device-specific data sheet) are configured such that the device boot from FLASH. Change these jumper settings to allow code execution from RAM under CCS control.
2. Start CCS. In CCS, a project contains all the files and build options needed to generate 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*, and under *Select Root Directory*, navigate to and select ..\controlSUITE\development_kits\MPPT DC-DC_v1.0\MPPT DC-DC directory. Make sure that under the *Projects* tab MPPT DC-DC is checked. Click *Finish*. This project will invoke all the necessary tools (compiler, assembler, and linker) for building the project.
3. In the project window on the left, click the plus sign (+) to the left of Project. Your project window will look like the following in Figure 9:

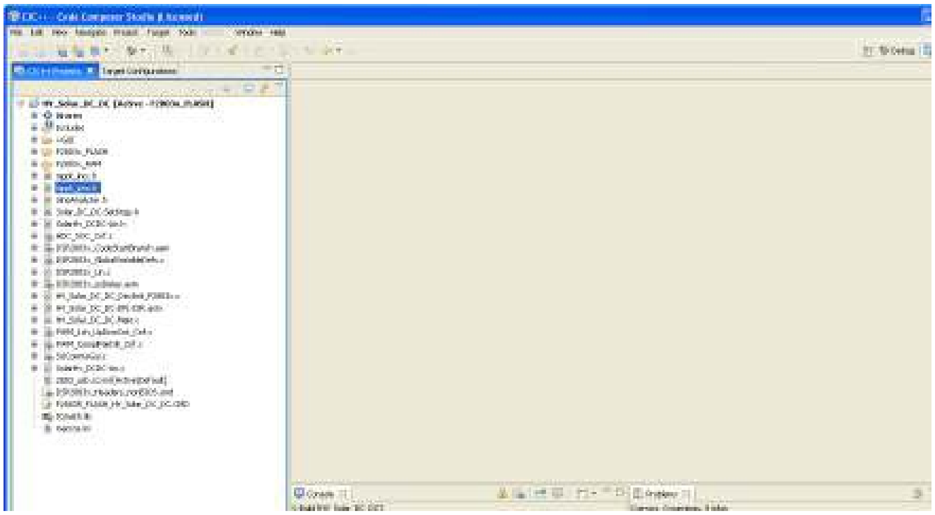


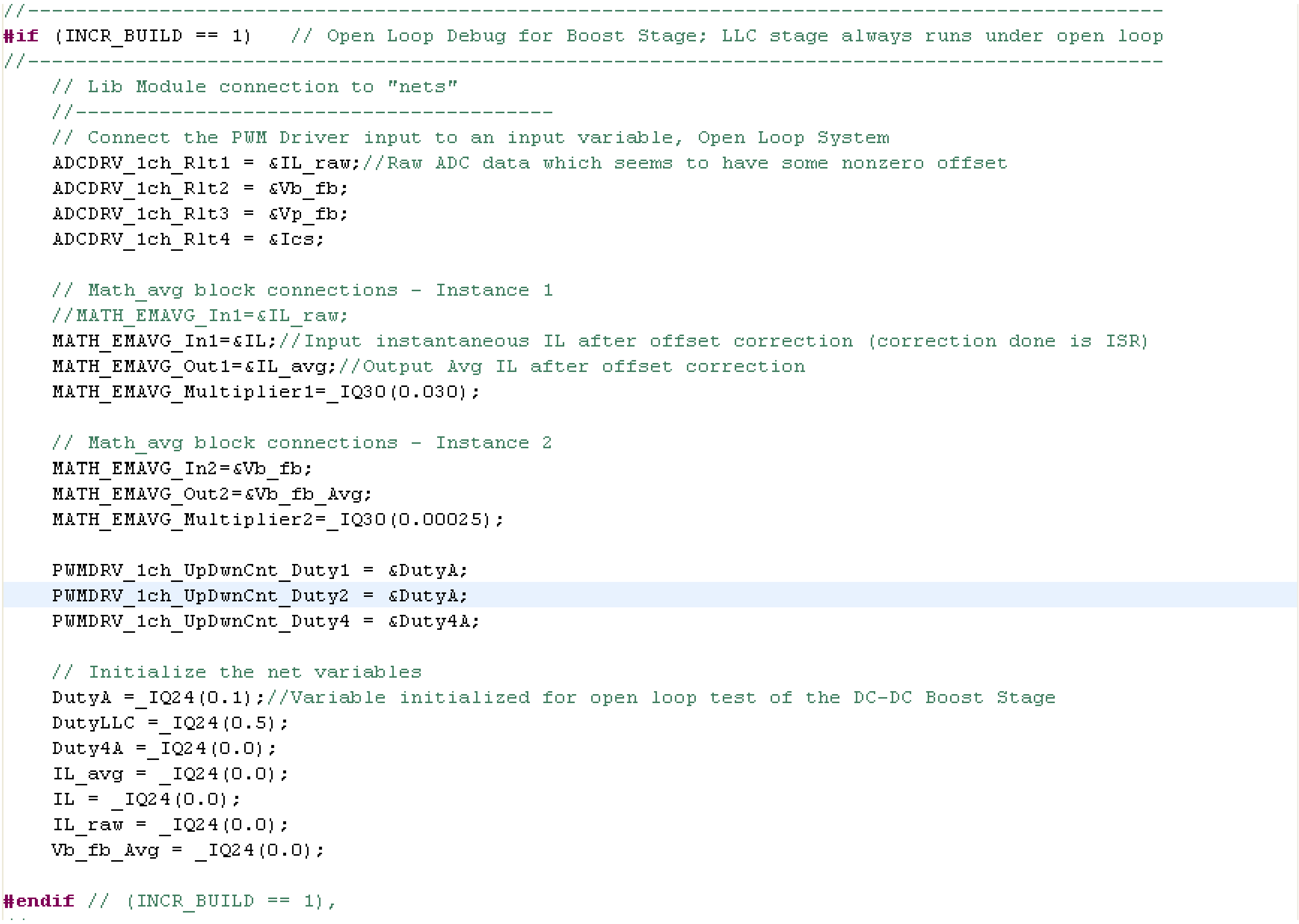
Figure 9. CCS Project Window

3.1.4.2 Device Initialization, Main, and ISR Files

NOTE: Do not make any changes to the source files—only inspect.

1. Open and inspect SOLAR_DC_DC-DevInit_F2803x.c by double clicking on the filename in the project window. Note that system clock, peripheral clock Preliminary 15 TMS320C2000™ Systems Applications Collateral prescale, and peripheral clock enables have been setup. Next, notice that the shared GPIO pins have been configured.
2. Open and inspect SOLAR_DC_DC -Main.c. Notice the call made to the *DeviceInit()* function and other variable initialization. Also, notice the code for different incremental build options, the ISR intialization and the background for(;;) loop.

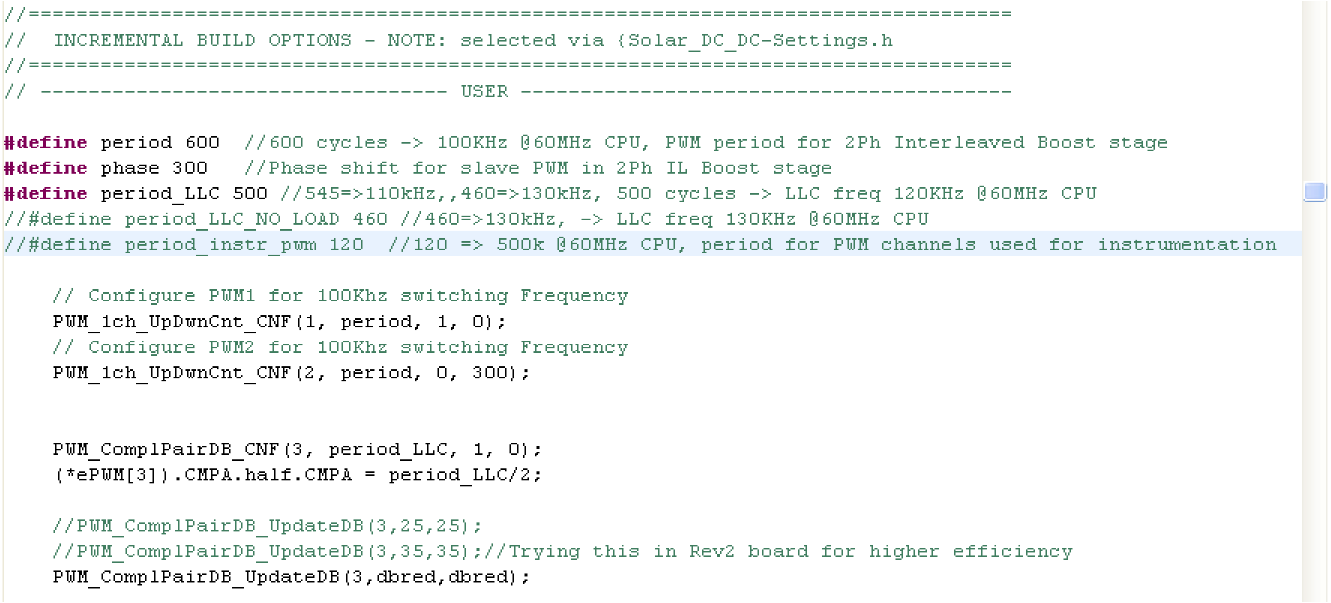
3. Locate and inspect the following code in the main file under initialization code specific for Build 1. This file is where the PWMDRV_1ch_UpDwnCnt and ADCDRV_1CH blocks connect in the control flow.



```
//------------------------------------------------------------------------------------------------
#if (INCR_BUILD == 1)   // Open Loop Debug for Boost Stage; LLC stage always runs under open loop
//------------------------------------------------------------------------------------------------
    // Lib Module connection to "nets"
    //----------------------------------------
    // Connect the PWM Driver input to an input variable, Open Loop System
    ADCDRV_1ch_Rlt1 = &IL_raw;//Raw ADC data which seems to have some nonzero offset
    ADCDRV_1ch_Rlt2 = &Vb_fb;
    ADCDRV_1ch_Rlt3 = &Vp_fb;
    ADCDRV_1ch_Rlt4 = &Ics;

    // Math_avg block connections - Instance 1
    //MATH_EMAVG_In1=&IL_raw;
    MATH_EMAVG_In1=&IL;//Input instantaneous IL after offset correction (correction done is ISR)
    MATH_EMAVG_Out1=&IL_avg;//Output Avg IL after offset correction
    MATH_EMAVG_Multiplier1=_IQ30(0.030);

    // Math_avg block connections - Instance 2
    MATH_EMAVG_In2=&Vb_fb;
    MATH_EMAVG_Out2=&Vb_fb_Avg;
    MATH_EMAVG_Multiplier2=_IQ30(0.00025);

    PWMDRV_1ch_UpDwnCnt_Duty1 = &DutyA;
    PWMDRV_1ch_UpDwnCnt_Duty2 = &DutyA;
    PWMDRV_1ch_UpDwnCnt_Duty4 = &Duty4A;

    // Initialize the net variables
    DutyA =_IQ24(0.1);//Variable initialized for open loop test of the DC-DC Boost Stage
    DutyLLC =_IQ24(0.5);
    Duty4A =_IQ24(0.0);
    IL_avg = _IQ24(0.0);
    IL = _IQ24(0.0);
    IL_raw = _IQ24(0.0);
    Vb_fb_Avg = _IQ24(0.0);

#endif // (INCR_BUILD == 1),
```

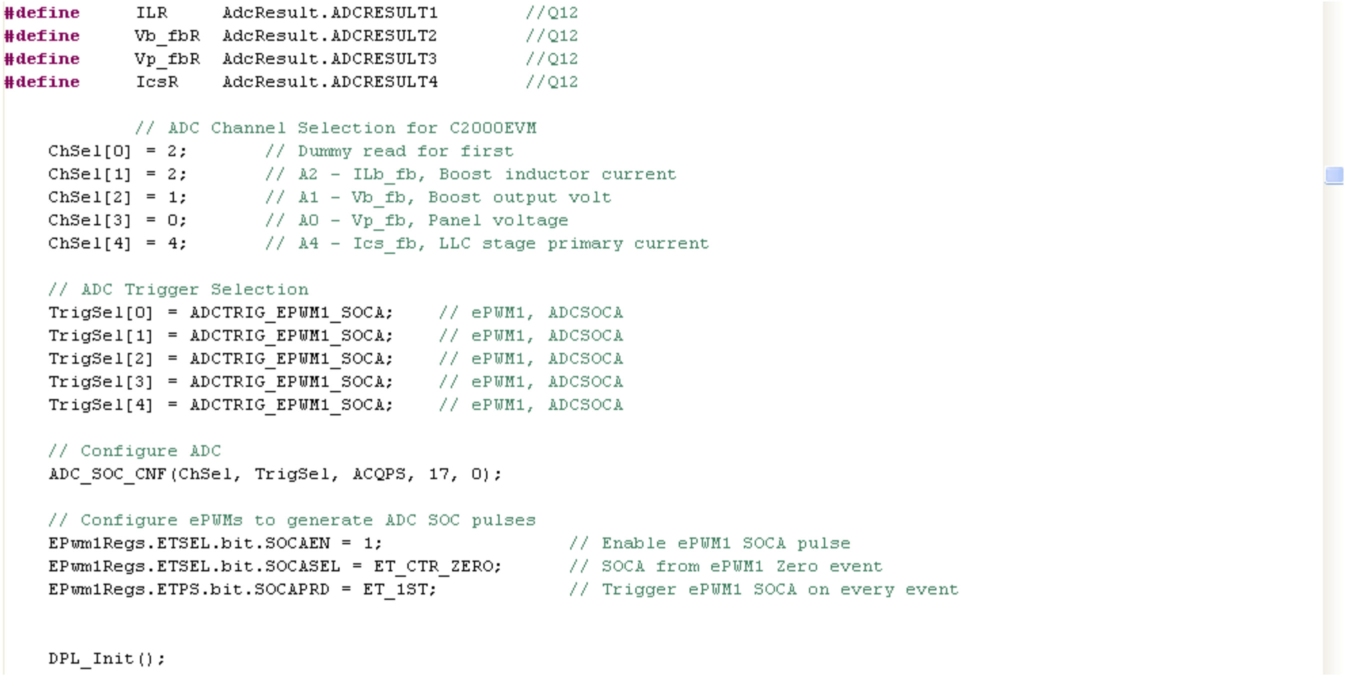
Figure 10. Code

4. Locate and inspect the code in the main file under initialization code, shown in Figure 11. This file is where the PWMDRV_1ch_UpDwnCnt block is configured and initialized. This action is common for all incremental builds. This PWM driver module inputs the total PWM period value of 600 and internally calculates the period register value of 300.
   Also locate and inspect the code in the main file under initialization code shown in Figure 12. This file is where the ADCDRV_1CH block is configured and initialized. This action is also common for all incremental builds.



```
//===================================================================================
//  INCREMENTAL BUILD OPTIONS - NOTE: selected via {Solar_DC_DC-Settings.h
//===================================================================================
// ---------------------------------- USER ----------------------------------------

#define period 600  //600 cycles -> 100KHz @60MHz CPU, PWM period for 2Ph Interleaved Boost stage
#define phase 300   //Phase shift for slave PWM in 2Ph IL Boost stage
#define period_LLC 500 //545=>110kHz,,460=>130kHz, 500 cycles -> LLC freq 120KHz @60MHz CPU
//#define period_LLC_NO_LOAD 460 //460=>130kHz, -> LLC freq 130KHz @60MHz CPU
//#define period_instr_pwm 120  //120 => 500k @60MHz CPU, period for PWM channels used for instrumentation

    // Configure PWM1 for 100Khz switching Frequency
    PWM_1ch_UpDwnCnt_CNF(1, period, 1, 0);
    // Configure PWM2 for 100Khz switching Frequency
    PWM_1ch_UpDwnCnt_CNF(2, period, 0, 300);


    PWM_ComplPairDB_CNF(3, period_LLC, 1, 0);
    (*ePWM[3]).CMPA.half.CMPA = period_LLC/2;

    //PWM_ComplPairDB_UpdateDB(3,25,25);
    //PWM_ComplPairDB_UpdateDB(3,35,35);//Trying this in Rev2 board for higher efficiency
    PWM_ComplPairDB_UpdateDB(3,dbred,dbred);
```

Figure 11. Code 2



```
#define      ILR      AdcResult.ADCRESULT1        //Q12
#define      Vb_fbR   AdcResult.ADCRESULT2        //Q12
#define      Vp_fbR   AdcResult.ADCRESULT3        //Q12
#define      IcsR     AdcResult.ADCRESULT4        //Q12

          // ADC Channel Selection for C2000EVM
    ChSel[0] = 2;       // Dummy read for first
    ChSel[1] = 2;       // A2 - ILb_fb, Boost inductor current
    ChSel[2] = 1;       // A1 - Vb_fb, Boost output volt
    ChSel[3] = 0;       // A0 - Vp_fb, Panel voltage
    ChSel[4] = 4;       // A4 - Ics_fb, LLC stage primary current

    // ADC Trigger Selection
    TrigSel[0] = ADCTRIG_EPWM1_SOCA;    // ePWM1, ADCSOCA
    TrigSel[1] = ADCTRIG_EPWM1_SOCA;    // ePWM1, ADCSOCA
    TrigSel[2] = ADCTRIG_EPWM1_SOCA;    // ePWM1, ADCSOCA
    TrigSel[3] = ADCTRIG_EPWM1_SOCA;    // ePWM1, ADCSOCA
    TrigSel[4] = ADCTRIG_EPWM1_SOCA;    // ePWM1, ADCSOCA

    // Configure ADC
    ADC_SOC_CNF(ChSel, TrigSel, ACQPS, 17, 0);

    // Configure ePWMs to generate ADC SOC pulses
    EPwm1Regs.ETSEL.bit.SOCAEN = 1;              // Enable ePWM1 SOCA pulse
    EPwm1Regs.ETSEL.bit.SOCASEL = ET_CTR_ZERO;   // SOCA from ePWM1 Zero event
    EPwm1Regs.ETPS.bit.SOCAPRD = ET_1ST;         // Trigger ePWM1 SOCA on every event


    DPL_Init();
```

Figure 12. Code 3

5. Open and inspect SOLAR_DC_DC-DPL-ISR.asm. The _DPL_Init and _DPL_ISR sections under Build 1 are where the PWM and ADC driver macro instantiation is done for initialization and runtime, respectively.

3.1.4.3 Build and Load the Project

1. Select the incremental build option as 1 in the SOLAR_DC_DC-Settings.h file.

NOTE: Whenever you change the incremental build option in SOLAR_DC_DCSettings. h, always select *Rebuild All*.

2. Click the *Project Rebuild All* button and watch the tools run in the build window.
3. Click *Target Debug Active Project*. CCS will ask you to open a new target configuration file if one has not already been selected. If a valid target configuration file has been created for this connection, you may jump to Step 1 in Section 3.1.4.5. In the *New Target Configuration Window*, type in the name of the .ccxml file for the target you will be working with (example, xds100-F28035.ccxml). Check *Use shared location* and click *Finish*.
4. In the .ccxml file that opens up, select *Connection* as “Texas Instruments XDS100v2 USB Emulator”, and under the device, scroll down and select *TMS320F28035*. Click *Save*.
5. Click *Target Debug Active Project*. Select the project configuration as F2803x_FLASH. The program will be loaded into the Flash. You should now be at the start of Main().

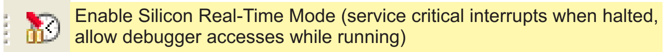
3.1.4.4 Debug Environment Windows

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. If a watch view did not open when the debug environment was launched, open a new watch view and add various parameters to it by following these steps:

1. Click *View Watch* on the menu bar.
2. Click the *Watch (1)* tab at the top watch view. Add any variables to the watch view. In the empty box in the *Name* column, type the symbol name of the variable you want to watch and press *Enter* on keyboard. Be sure to modify the format as needed.

3.1.4.5 Using Real-Time Emulation

Real-time emulation is a special emulation feature that allows the windows within CCS to be updated at a rate up to 10 Hz while the MCU is running. This not only allows graphs and watch views to update, but also allows the user to change values in watch or memory windows, and see the effect of these changes in the system.

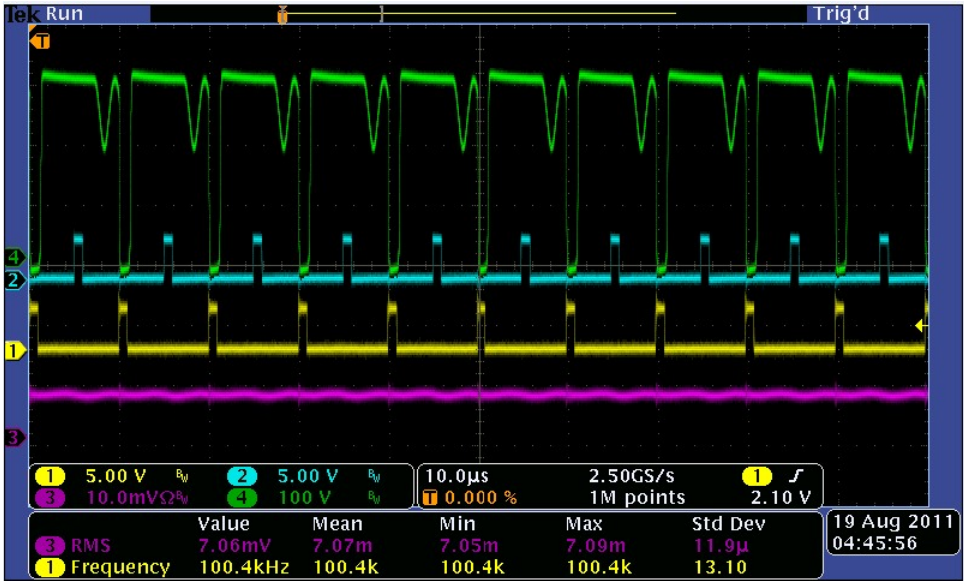
1. Enable real-time mode by hovering your mouse on the buttons on the horizontal toolbar and clicking the  Enable Silicon Real-Time Mode (service critical interrupts when halted, allow debugger accesses while running) button.
2. A message box may appear. If so, select YES to enable debug events. This will set bit 1 (DGBM bit) of status register 1 (ST1) to a “0”. The DGBM is the debug enable mask bit. When the DGBM bit is set to “0”, memory and register values can be passed to the host processor for updating the debugger windows.
3. Click on the Continuous Refresh button for the watch view.

3.1.4.6 Run the Code

1. Run the code by using the *F8* key, using the *Run* button on the toolbar, or using *Target Run* on the menu bar.
2. In the watch view, add the variable DutyA and set it to 0.1 (=1677721 in Q24). This variable sets the duty cycle for the boost converter.
3. Apply a resistive load to the DC-DC EVM output terminal (10 to 100 W).
4. Use a high-voltage, isolated DC supply to power the DC-DC EVM. Measure and verify the boost DC-BUS voltage corresponding to applied input voltage and the duty ratio.
5. Use DutyA to slowly change the duty from the watch window. The boost converter output voltage should change accordingly, and this, in turn, will change the EVM output.

NOTE: Observe the output voltage carefully. The output voltage should not be allowed to exceed the maximum voltage rating of the board.

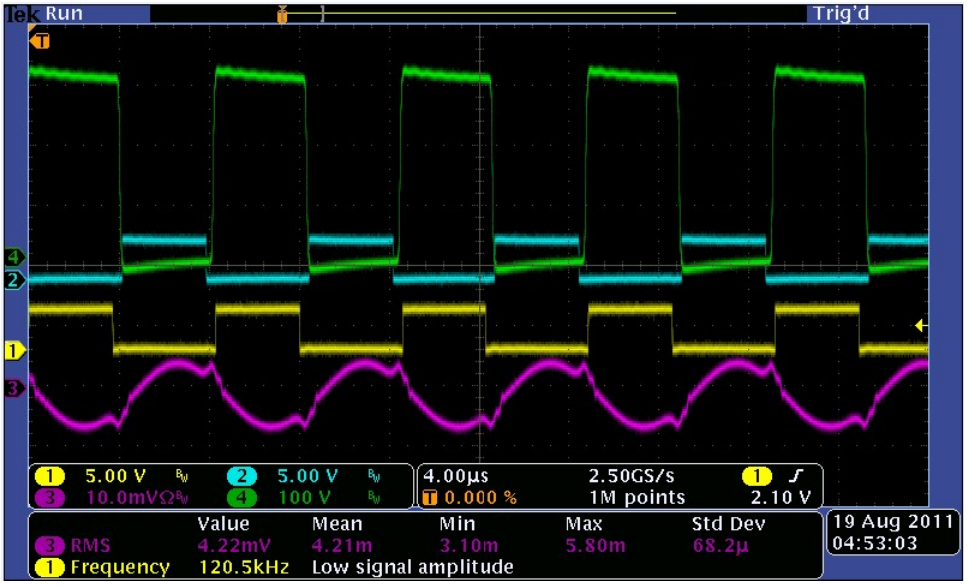
6. Add the other variables such as Vb_fb and Vp_fb, and verify the different ADC results in the watch view.
7. Figure 13 shows two PWM outputs (Ch1 and Ch2), the DC source input current (Ch3), and the boost MOSFET drain to source voltage (Ch4) when the output DC-BUS load is 1 KΩ, the input DC voltage is 250 V, and the set duty ratio is about 10%. The PWM frequency is measured as 100 kHz.



	Value	Mean	Min	Max	Std Dev
3 RMS	7.06mV	7.07m	7.05m	7.09m	11.9µ
1 Frequency	100.4kHz	100.4k	100.4k	100.4k	13.10

Figure 13. Graph

8. Figure 14 shows two PWM outputs (Ch1 and Ch2) for the LLC stage. In this figure, Ch4 represents LLC stage primary switch node voltage, and Ch3 is the LLC primary current when the boost input is 250 V, the boost output (LLC input) is 300 V, the LLC output is 307 V, the LLC output load is 1 KΩ, and the boost duty ratio is set to about 10%.



	Value	Mean	Min	Max	Std Dev
3 RMS	4.22mV	4.21m	3.10m	5.80m	68.2µ
1 Frequency	120.5kHz	Low signal amplitude			

Figure 14. Graph 2

9. Try different duty cycle values and observe the corresponding ADC results. Increase the duty cycle value in small steps. Always observe the output voltage carefully, which should not be allowed to exceed the capabilities of the board. Different waveforms, like the PWM gate drive signals, input voltage and current, and output voltage may also be probed and verified using an oscilloscope. Take the appropriate safety measures while probing these high voltage signals.
10. Fully halting the MCU when in real-time mode is a two-step process. With the DC input turned off wait until the DC bus capacitor is fully discharged. First, halt the processor by using the Halt button on the toolbar, or by using Target Halt. Then take the MCU out of real-time mode. Finally reset the MCU.
11. Leave CCS running for the next exercise or close the program.

3.2 Build 2: MPPT DC-DC With Closed Current Loop

3.2.1 Objective

The objective of this build is to verify the operation of the MPPT DC-DC under closed current loop and open voltage loop mode. Since the voltage loop is open, there is no MPPT operation under this build.

3.2.2 Overview

Figure 15 shows the software blocks used in this build, which features one additional software block compared to the Build 1 diagram (Figure 7). This block is shown in Figure 15 as CNTL_2P2Z:1, which represents a 2p2z controller and is used for the current control loop. Depending on the control loop requirements, other control blocks such as a PI or a 3p3z controller can also be used.

As shown in Figure 15, the current loop control block is executed at a 50-KHz rate. CNTL_2P2Z is a second-order compensator realized from an IIR filter structure. This function is independent of any peripherals and therefore does not require a CNF function call.

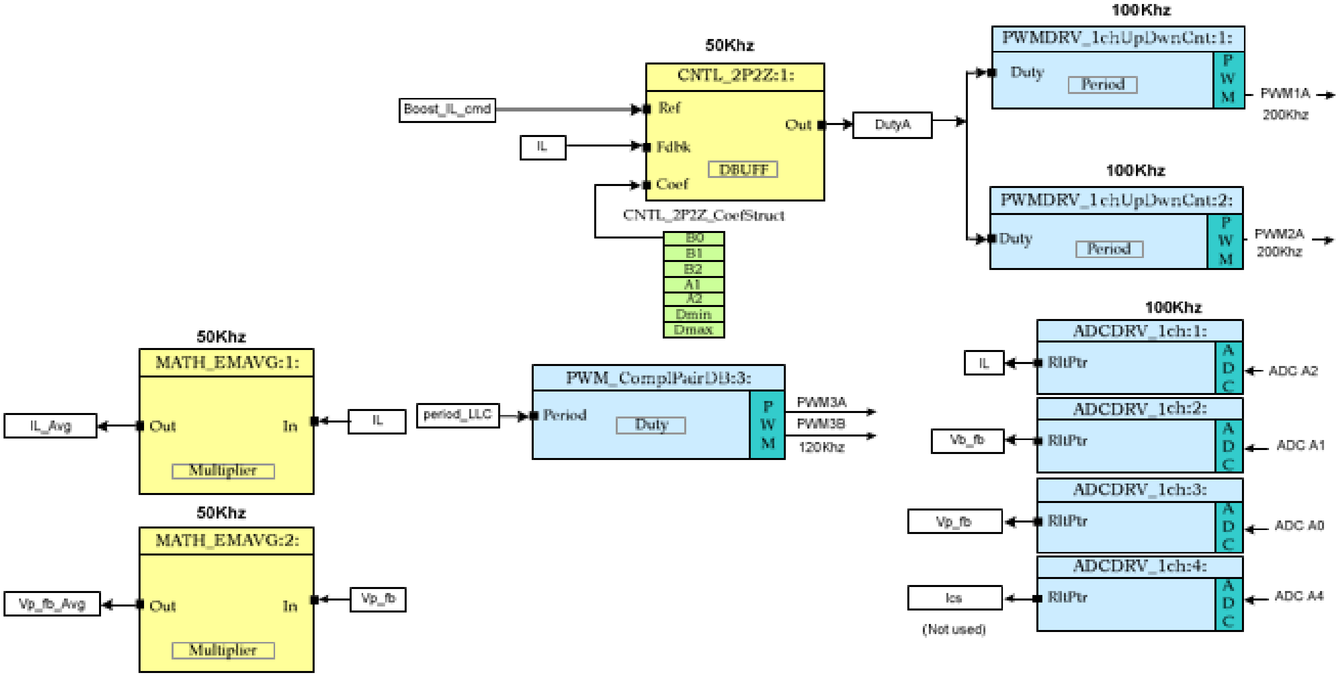


Figure 15. Build 2 Software Blocks

This 2p2z controller requires five control coefficients. These coefficients and the clamped output of the controller are stored as the elements of a structure named CNTL_2P2Z_CoefStruct1. The CNTL_2P2Z block can be instantiated multiple times if the system needs multiple loops. Each instance can have separate set of coefficients. The CNTL_2P2Z instance for the current loop uses the coefficients stored as the elements of structure CNTL_2P2Z_CoefStruct1. This way a second instantiation of CNTL_2P2Z with a different structure, CNTL_2P2Z_CoefStruct2, can be used for voltage loop control, as shown in Section 3.3 with Build 3.

The controller coefficients can be changed directly by modifying the values for B0, B1, B2, A1, and A2 inside the structure CNTL_2P2Z_CoefStruct1. Alternately, the 2p2z controller can be expressed in PID form and the coefficients can be changed by changing the PID coefficients. The equations relating the five controller coefficients to the three PID gains are given below. For the current loop these P, I and D coefficients are named as: Pgain_I, Igain_I and Dgain_I respectively. For the voltage loop, used in Build 3, these coefficients are named as: Pgain_V, Igain_V and Dgain_V, respectively. These coefficients are used in Q26 format.

The compensator block (CNTL_2P2Z) has a reference input and a feedback input. The feedback input labeled as, Fdbk, comes from the ADC. The reference input labeled as *Ref* normally comes from the voltage loop controller output. This build has no voltage loop controller, so the variable Boost_IL_cmd is used to change the reference current under user control. The z-domain transfer function for CNTL_2P2Z is shown in Equation 1:

$$\frac{U(z)}{E(z)}=\frac{b_0+b_1z^{-1}+b_2z^{-2}}{1+a_1z^{-1}+a_2z^{-2}} \tag{1}$$

The recursive form of the PID controller is given by the difference equation:

$$u(k)=u(k-1)+b_0e(k)+b_1e(k-1)+b_2e(k-2)$$

where

- $b_0=K_p+K_i+K_d$
- $b_1=-K_p+K_i-2K_d$
- $b_2=K_d$ (2)

And the z-domain transfer function of this PID is:

$$\frac{U(z)}{E(z)}=\frac{b_0+b_1z^{-1}+b_2z^{-2}}{1-z^{-1}} \tag{3}$$

Comparing this equation with the general form, PID is a special case of CNTL_2P2Z control where a1 is –1 and a2 is 0.

The exponential moving average (MATH_EMAVG) blocks shown in Figure 15 calculate the average of the boost inductor current and the panel voltage. The average inductor current represents the panel current and is used to implement the MPPT algorithm.

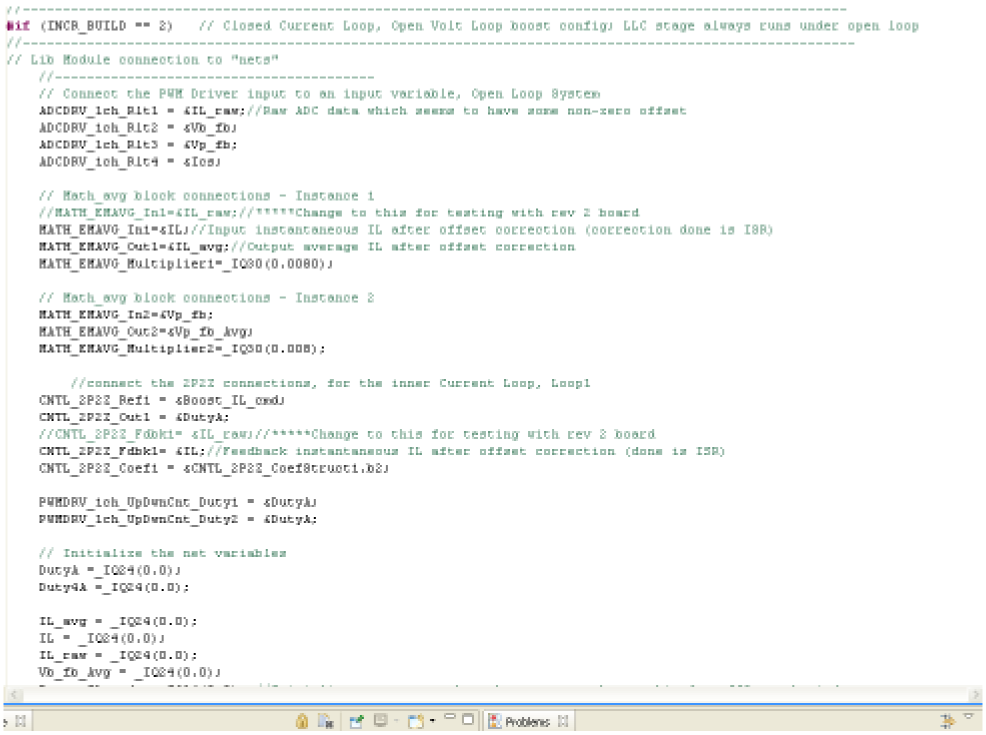
3.2.3 Procedure

Follow the steps below to execute this build.

3.2.3.1 Build and Load Project

Locate and inspect the following code in the main file under initialization code specific for Build 2. This file is where all the software blocks related to Build 2 connect in the control flow.

Follow Step 1 in Section 3.1.4.1 through Step 4 in Section 3.1.4.2 exactly as in Build 1 , except that in Step 3 of the latter section, select the Build 2 option instead of Build 1. Then complete the following steps.



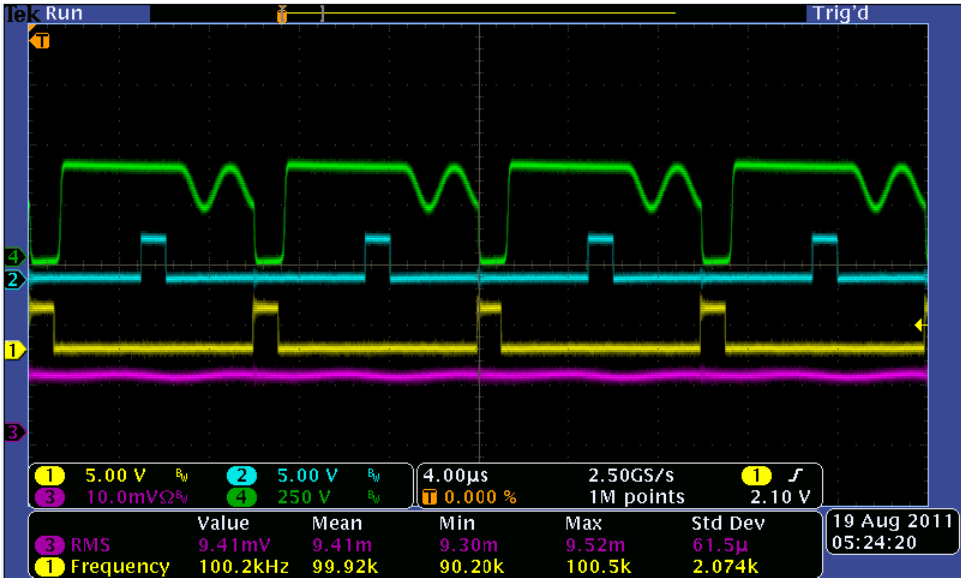
```
//-----------------------------------------------------------------------------------------
#if (INCR_BUILD == 2)   // Closed Current Loop, Open Volt Loop boost config; LLC stage always runs under open loop
//-----------------------------------------------------------------------------------------
// Lib Module connection to "nets"
    //----------------------------------------
    // Connect the PWM Driver input to an input variable, Open Loop System
    ADCDRV_1ch_Rlt1 = &IL_raw;//Raw ADC data which seems to have some non-zero offset
    ADCDRV_1ch_Rlt2 = &Vb_fb;
    ADCDRV_1ch_Rlt3 = &Vp_fb;
    ADCDRV_1ch_Rlt4 = &Ios;

    // Math_avg block connections - Instance 1
    //MATH_EMAVG_In1=&IL_raw;//*****Change to this for testing with rev 2 board
    MATH_EMAVG_In1=&IL;//Input instantaneous IL after offset correction (correction done is ISR)
    MATH_EMAVG_Out1=&IL_avg;//Output average IL after offset correction
    MATH_EMAVG_Multiplier1=_IQ30(0.0080);

    // Math_avg block connections - Instance 2
    MATH_EMAVG_In2=&Vp_fb;
    MATH_EMAVG_Out2=&Vp_fb_Avg;
    MATH_EMAVG_Multiplier2=_IQ30(0.008);

        //connect the 2P2Z connections, for the inner Current Loop, Loop1
    CNTL_2P2Z_Ref1 = &Boost_IL_cmd;
    CNTL_2P2Z_Out1 = &DutyA;
    //CNTL_2P2Z_Fdbk1= &IL_raw;//*****Change to this for testing with rev 2 board
    CNTL_2P2Z_Fdbk1= &IL;//Feedback instantaneous IL after offset correction (done is ISR)
    CNTL_2P2Z_Coef1 = &CNTL_2P2Z_CoefStruct1.b2;

    PWMDRV_1ch_UpDwnCnt_Duty1 = &DutyA;
    PWMDRV_1ch_UpDwnCnt_Duty2 = &DutyA;

    // Initialize the net variables
    DutyA =_IQ24(0.0);
    Duty4A =_IQ24(0.0);

    IL_avg = _IQ24(0.0);
    IL = _IQ24(0.0);
    IL_raw = _IQ24(0.0);
    Vb_fb_Avg = _IQ24(0.0);
```

Figure 16. Code Build 2-1

1. Open and inspect SOLAR_DC_DC-DPL-ISR.asm. The _DPL_Init and _DPL_ISR sections under Build 2 are where all the macro instantiations under Build 2 are done for initialization and runtime, respectively.
2. Select the *Incremental build option* as 2 in the SOLAR_DC_DC-Settings.h file. Then follow Step 2 in Section 3.1.4.3 through Step 1 in Section 3.1.4.6 in Build 1 to run the code. When all these steps are completed, you should now be at the start of *Main()*.

NOTE: Whenever you change the incremental build option in SOLAR_DC_DCSettings. h always select *Rebuild All*.

3. In the watch view, add the variable Boost_IL_cmd and set it to 0.05 (=838861 in Q24). This variable sets the magnitude of the reference current command for the current control loop.
4. Connect an appropriate resistive load across the DC-DC output. For example, use a 1-kΩ resistor of a 400-W rating. This resistor will provide a load of 160 W at 400-V bus voltage.
5. Slowly apply DC power to the board from an isolated DC source. Monitor the DCDC EVM output voltage as the input voltage is raised slowly to 300 V. In increments of 0.01, slowly adjust the value for Boost_IL_cmd to set the output voltage to about 385 V. Use an oscilloscope with voltage and current probes to observe the input voltage, input current, boost MOSFET voltage, LLC primary current and PWM outputs. With a 300-V boost input and 1-kΩ resistive load when the boost output voltage is set to 373 V, you should see the LLC output voltage (EVM output) of 385 V. The following scope plot is captured under this condition. Here Ch1 and Ch2 show the boost PWM outputs. Ch3 is the boost input current, and Ch4 is the voltage across the boost MOSFET.



1 5.00 V 2 5.00 V
3 10.0mVΩ 4 250 V

4.00µs 2.50GS/s 1 2.10 V
0.000 % 1M points

	Value	Mean	Min	Max	Std Dev
3 RMS	9.41mV	9.41m	9.30m	9.52m	61.5µ
1 Frequency	100.2kHz	99.92k	90.20k	100.5k	2.074k

19 Aug 2011
05:24:20

Figure 17. Graph Build 2-1

6. In increments of 0.01, increase Boost_IL_cmd slightly and observe the bus voltage settle to a higher value. Increasing Boost_IL_cmd increases the magnitude of the current reference signal, and the bus voltage will rise. Therefore, apply caution and set the OVP threshold to a value less than 400 V.
7. Follow Step 10 and Step 11 in Section 3.1.4.6 to turn off the power and reset the MCU.

3.3 Build 3: MPPT DC-DC With Closed Voltage and Current Loop

3.3.1 Objective

The objective of this build is to verify the operation of the complete MPPT DC-DC project from the CCS environment.

3.3.2 Overview

Figure 18 shows the software blocks used in this build. Compared to Build 2 in Figure 15, this build uses an additional 2p2z control block labeled as CNTL_2P2Z:2. This control block is the second instantiation of the 2p2z control block to implement the MPPT DC-DC voltage loop control. This voltage loop controller is executed at a 25-kHz rate, which is half the rate for current loop. The output from this control block drives the input node Boost_IL_cmd of the current controller block.

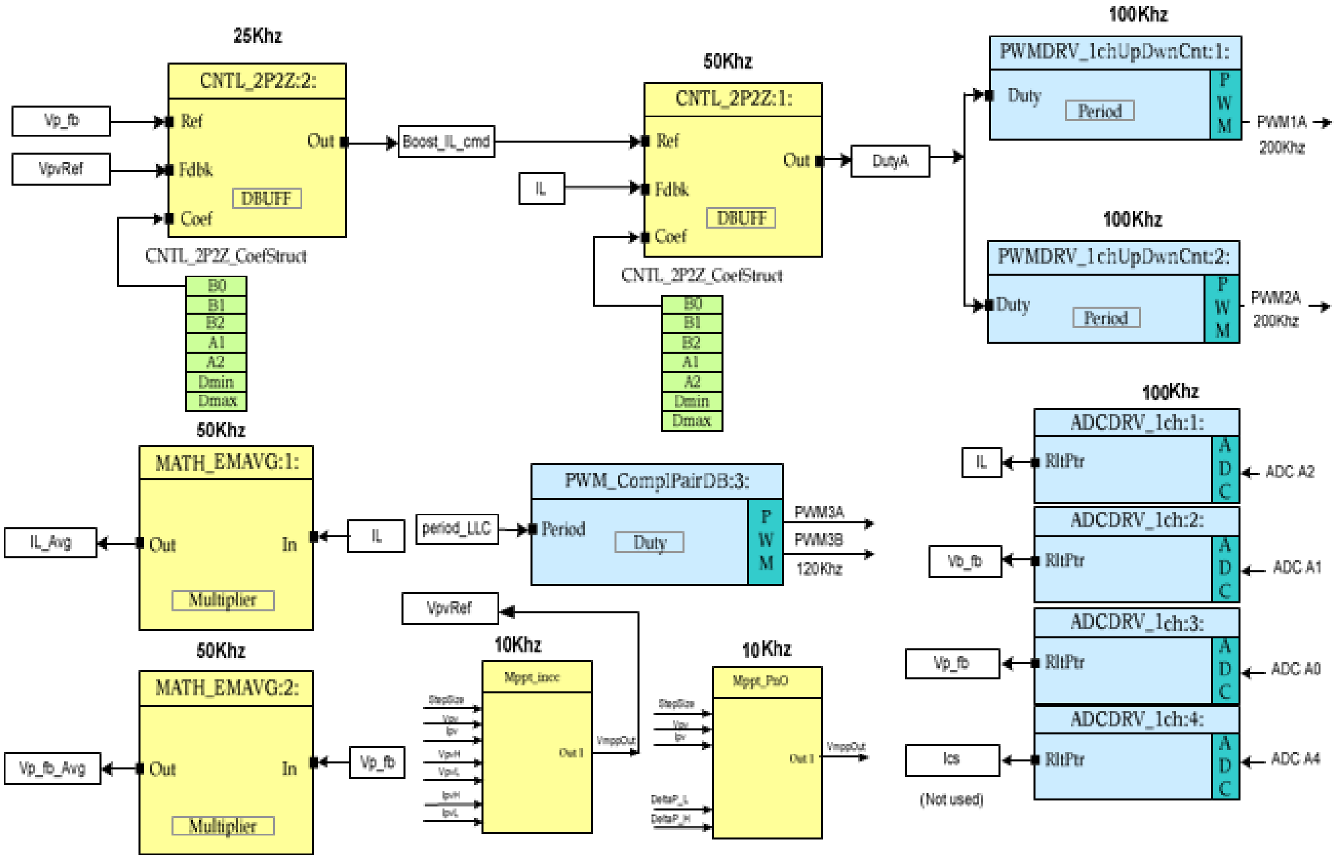


Figure 18. Build 3 Software Blocks

Similar to current loop controller, CNTL_2P2Z:2 also requires five control coefficients. These coefficients and the clamped output of the controller are stored as the elements of a second structure named CNTL_2P2Z_CoefStruct2. The coefficients for this controller can be changed directly by modifying the values for B0, B1, B2, A1, and A2 inside the structure CNTL_2P2Z_CoefStruct2, or by changing the equivalent PID gains as discussed in Section 3.2.2.

Figure 18 also shows two additional blocks implementing two different MPPT algorithms that are used in this EVM. The default code setting uses incremental conductance algorithm for the MPPT, so the output from this MPPT block is connected to the feedback terminal (Fdbk) of the voltage loop controller. These MPPT blocks are run from a 10-kHz ISR.

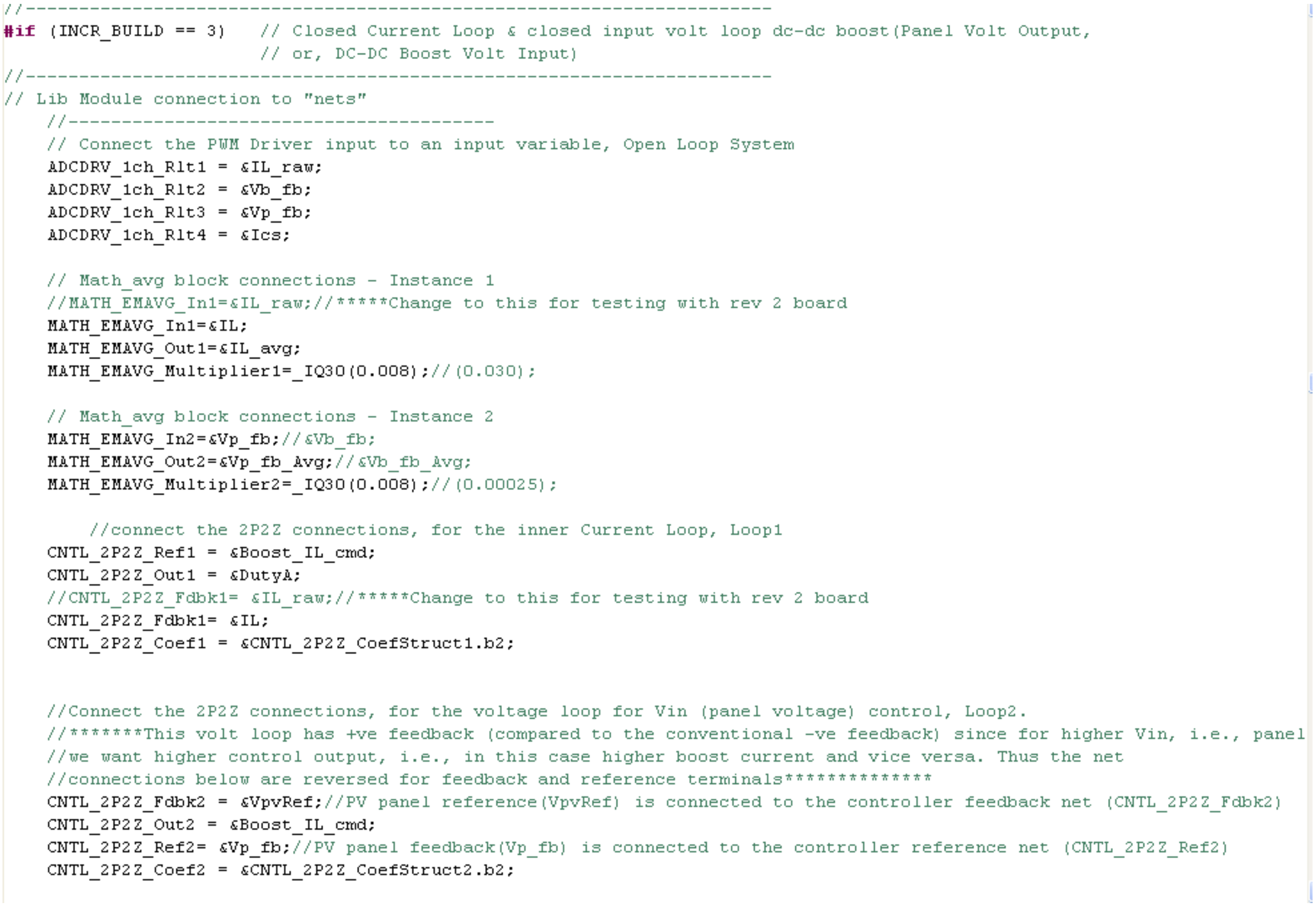
3.3.3 Procedure

Follow the steps below to execute this build.

3.3.3.1 Build and Load Project

Locate and inspect the following code in the main file under initialization code specific for Build 3. This file is where all the software blocks related to Build 3 connect in the control flow.

Follow Step 1 in Section 3.1.4.1 through Step 4 in Section 3.1.4.2 exactly as in Build 1 , except that in Step 3 of the latter section, select the Build 3 option instead of Build 1. Then complete the following steps.



```
//----------------------------------------------------------------------
#if (INCR_BUILD == 3)   // Closed Current Loop & closed input volt loop dc-dc boost(Panel Volt Output,
                        // or, DC-DC Boost Volt Input)
//----------------------------------------------------------------------
// Lib Module connection to "nets"
    //----------------------------------------
    // Connect the PWM Driver input to an input variable, Open Loop System
    ADCDRV_1ch_Rlt1 = &IL_raw;
    ADCDRV_1ch_Rlt2 = &Vb_fb;
    ADCDRV_1ch_Rlt3 = &Vp_fb;
    ADCDRV_1ch_Rlt4 = &Ics;

    // Math_avg block connections - Instance 1
    //MATH_EMAVG_In1=&IL_raw;//*****Change to this for testing with rev 2 board
    MATH_EMAVG_In1=&IL;
    MATH_EMAVG_Out1=&IL_avg;
    MATH_EMAVG_Multiplier1=_IQ30(0.008);//(0.030);

    // Math_avg block connections - Instance 2
    MATH_EMAVG_In2=&Vp_fb;//&Vb_fb;
    MATH_EMAVG_Out2=&Vp_fb_Avg;//&Vb_fb_Avg;
    MATH_EMAVG_Multiplier2=_IQ30(0.008);//(0.00025);

        //connect the 2P2Z connections, for the inner Current Loop, Loop1
    CNTL_2P2Z_Ref1 = &Boost_IL_cmd;
    CNTL_2P2Z_Out1 = &DutyA;
    //CNTL_2P2Z_Fdbk1= &IL_raw;//*****Change to this for testing with rev 2 board
    CNTL_2P2Z_Fdbk1= &IL;
    CNTL_2P2Z_Coef1 = &CNTL_2P2Z_CoefStruct1.b2;


    //Connect the 2P2Z connections, for the voltage loop for Vin (panel voltage) control, Loop2.
    //*******This volt loop has +ve feedback (compared to the conventional -ve feedback) since for higher Vin, i.e., panel
    //we want higher control output, i.e., in this case higher boost current and vice versa. Thus the net
    //connections below are reversed for feedback and reference terminals**************
    CNTL_2P2Z_Fdbk2 = &VpvRef;//PV panel reference(VpvRef) is connected to the controller feedback net (CNTL_2P2Z_Fdbk2)
    CNTL_2P2Z_Out2 = &Boost_IL_cmd;
    CNTL_2P2Z_Ref2= &Vp_fb;//PV panel feedback(Vp_fb) is connected to the controller reference net (CNTL_2P2Z_Ref2)
    CNTL_2P2Z_Coef2 = &CNTL_2P2Z_CoefStruct2.b2;
```

Figure 19. Code Build 3-1

1. Open and inspect SOLAR_DC_DC-DPL-ISR.asm. The _DPL_Init and _DPL_ISR sections under Build 3 are where all the macro instantiations under Build 3 are done for initialization and runtime, respectively.
2. Select the *Incremental build option* as 3 in the SOLAR_DC_DC-Settings.h file. Then follow Step 2 in Section 3.1.4.3 through Step 1 in Section 3.1.4.6 in Build 1 to run the code. When all these steps are completed, you should now be at the start of *Main()*.

NOTE: Whenever you change the incremental build option in SOLAR_DC_DC-Settings.h, always select *Rebuild All*.

3. In the watch view, add four variables inverter_connected, Start_DC_DC, Vp_fb,and Vb_fb. Set the Q-format for the last two variables (Vp_fb and Vb_fb) to Q24. These two variables represent the boost input and output voltages, respectively. These variables will slowly increase as the DC-DC starts up when PV panel emulator power is applied and the MPPT is turned on. To start MPPT from CCS first (under this build), modify the code as explained in Step 5 and reload the program to Flash memory. Then, from the CCS watch window, set the variables Start_DC_DC to 1 and inverter_connected to 0.

4. Configure a solar panel emulator (200 to 300 V, 500-W maximum) to provide input power to the EVM. Configure the panel emulator to emulate the following solar panel characteristics, and connect it to the EVM input. Do not turn on panel power at this time.
   Connect an appropriate resistive load to the EVM output terminals (Vo-R and GND terminals). As in the example above, if the panel emulator is configured to supply 165 W of power at MPPT point, then select a load resistor value of 970 Ω so that the EVM output voltage is limited to about 400V (P = 400*400/970, approximately 165 W). A smaller resistor will also work as long as the output voltage does not fall below 350 V, which means that the smallest resistor that can be chosen for this load set up (165 W) is about 742 Ω (R = 350*350/165 = 742 Ω). A resistor value larger than 970 Ω will cause output voltage higher than 400 V for this load set up. This output overvoltage condition must be prevented by choosing the maximum resistor value of 970 Ω for this load set up of 165 W. Use a resistor with a power rating >200 W for this load setting.

Table 3. Example Panel Emulator Parameters

VALUE	PURPOSE	MEASUREMENT
Voc	Open circuit panel voltage	260 V
Vmpp	Panel voltage for MPPT	220 V
Impp	Panel current for MPPT	0.75 A
Isc	Short circuit panel current	1 A

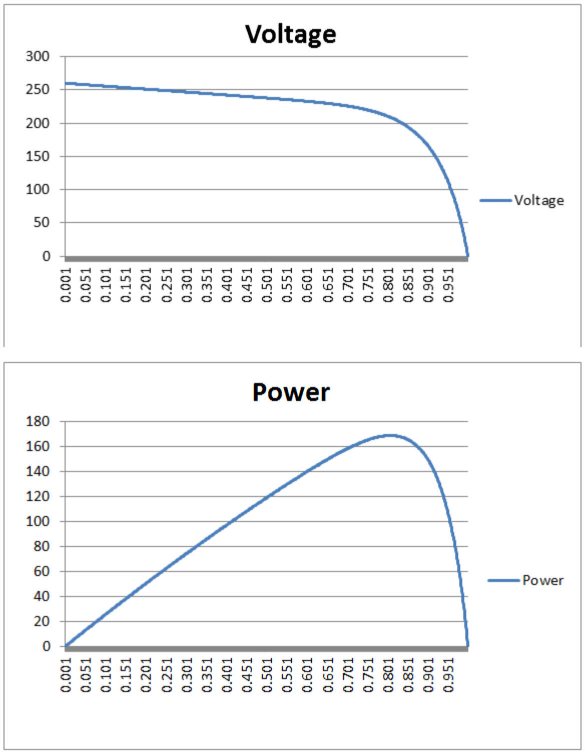
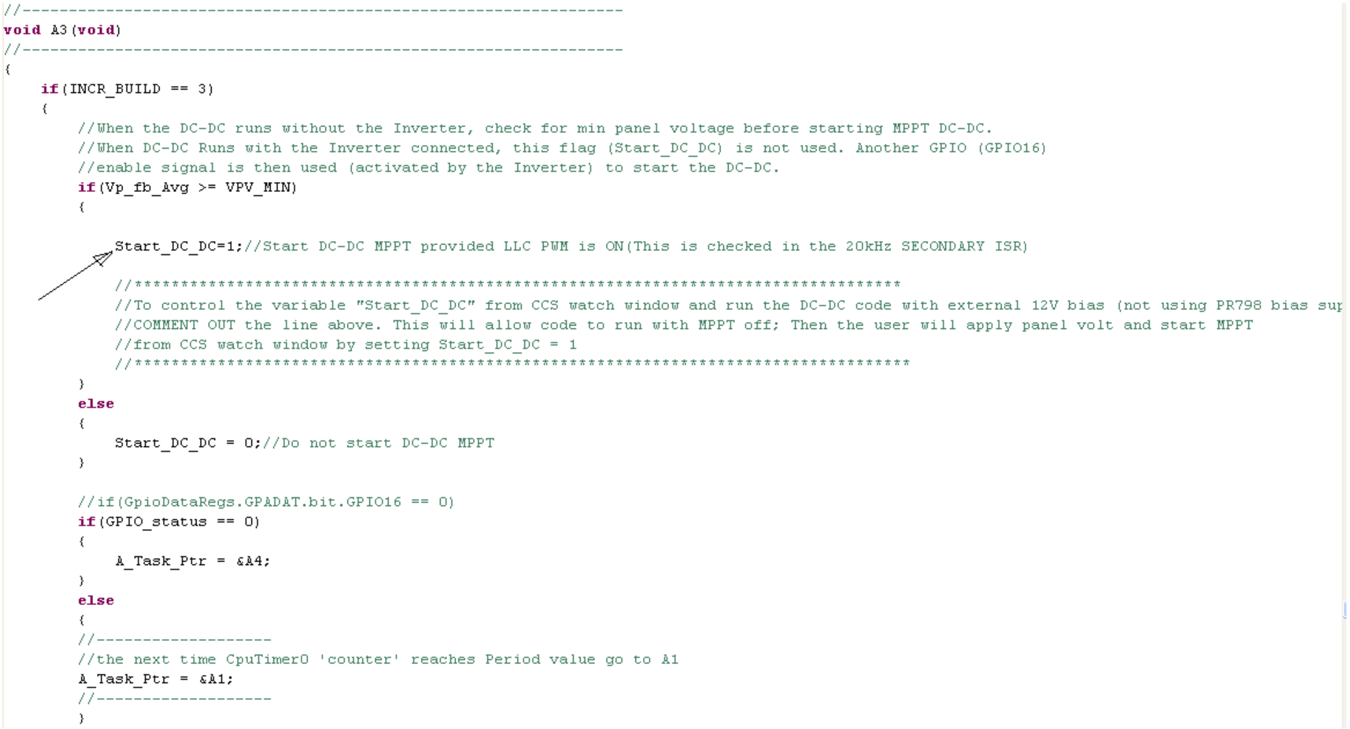


Figure 20. V/P Graph

5. Starting the MPPT algorithm (in Build 3) from the CCS watch window, with the PV panel emulator power applied to the EVM input, will require changing one line of code as explained below. Without this change, the code automatically starts the MPPT algorithm (1) when the minimum panel input voltage is applied, and (2) from the GUI the user set the variable inverter_connected to 0. When the EVM is shipped, the default code loaded in the flash memory allows this GUI-based power up in standalone mode. Therefore, this code modification is needed only if you want to start the MPPT from the CCS watch window.
   Open the CCS project file HV_Solar_DC_DC-Main.c and locate the code where the variable Start_DC_DC is set to 1 as shown in Figure 21.

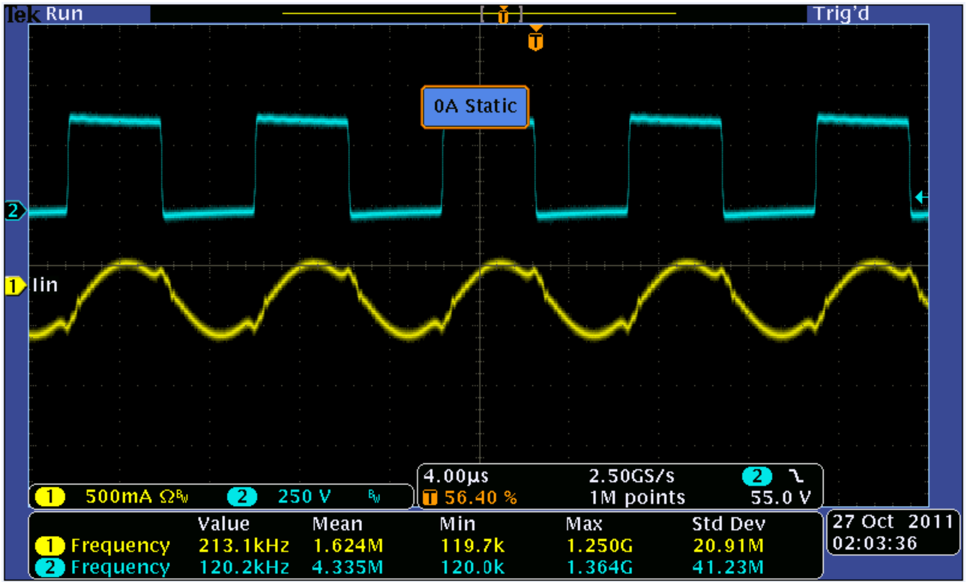
Comment out this one line of code, save the file, recompile, and reload the project in the Flash memory. Follow Step 1 through Step 2 (as described before) and run the code. Enable the real-time mode and then click on the *Continuous Refresh* button for the watch view. The variables Start_DC_DC will be set to 0 and inverter_connected will be set to 1.



```
//-----------------------------------------------------------------
void A3(void)
//-----------------------------------------------------------------
{
    if(INCR_BUILD == 3)
    {
        //When the DC-DC runs without the Inverter, check for min panel voltage before starting MPPT DC-DC.
        //When DC-DC Runs with the Inverter connected, this flag (Start_DC_DC) is not used. Another GPIO (GPIO16)
        //enable signal is then used (activated by the Inverter) to start the DC-DC.
        if(Vp_fb_Avg >= VPV_MIN)
        {

            Start_DC_DC=1;//Start DC-DC MPPT provided LLC PWM is ON(This is checked in the 20kHz SECONDARY ISR)

            //************************************************************************************
            //To control the variable "Start_DC_DC" from CCS watch window and run the DC-DC code with external 12V bias (not using PR798 bias sup
            //COMMENT OUT the line above. This will allow code to run with MPPT off; Then the user will apply panel volt and start MPPT
            //from CCS watch window by setting Start_DC_DC = 1
            //*************************************************************************************
        }
        else
        {
            Start_DC_DC = 0;//Do not start DC-DC MPPT
        }

        //if(GpioDataRegs.GPADAT.bit.GPIO16 == 0)
        if(GPIO_status == 0)
        {
            A_Task_Ptr = &A4;
        }
        else
        {
        //-------------------
        //the next time CpuTimer0 'counter' reaches Period value go to A1
        A_Task_Ptr = &A1;
        //-------------------
        }
```

Figure 21. Code Build 3-2

6. Use a voltmeter to monitor the DC-BUS voltage across the EVM output. Now turn on the PV panel emulator with the setup described in Step 4. At this point, the MPPT will still remain off, so the EVM input should be around 260 V and the output voltage, with a 1-kΩ load resistor, will also be at the same level. From the CCS watch window, set the variables inverter_connected to 0 and Start_DC_DC to 1. This change will start the MPPT algorithm and the EVM output will rise to around 400 V. With the MPPT turned on and the input voltage loop controller connected, the panel output voltage (that is, the boost input voltage) will be around 220 V. The EVM will deliver 165 W of panel power at the MPPT reference voltage of 220 V.
   Use an oscilloscope to capture the switch node voltage and transformer primary current from the LLC stage under this operating condition. This oscilloscope is shown in Figure 22 where Ch2 represents the LLC primary switch node voltage, and Ch1 represents the LLC primary current.



	Value	Mean	Min	Max	Std Dev
1 Frequency	213.1kHz	1.624M	119.7k	1.250G	20.91M
2 Frequency	120.2kHz	4.335M	120.0k	1.364G	41.23M

Figure 22. Graph Build 3-1

7. Now use the scope probes to capture the boost stage MOSFET drain to source voltage and the PV emulator current under this operating condition as shown in Figure 23 where Ch2 represents the boost MOSFET drain to source voltage and Ch4 represents the panel current.

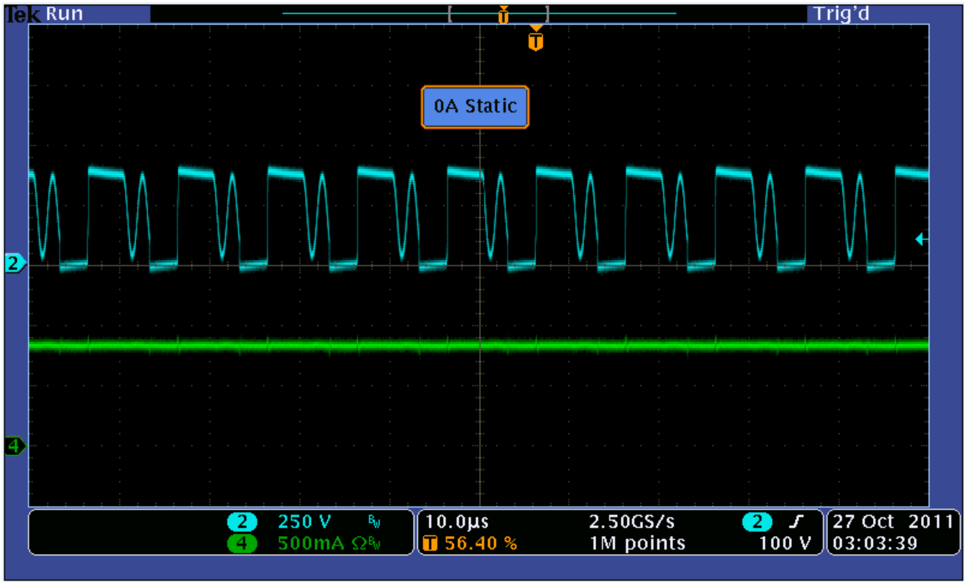


Figure 23. Graph Build 3-2

8. Observe the variables on the watch window. The variable Vp_fb should show a value of about 0.4297 (=220/512) when the Q format is set to Q24. The maximum panel voltage set by the sense resistors is about 512 V that corresponds to maximum ADC input of 3.3 V. Therefore, the normalized or per unit value will be about 0.4297 when the actual panel voltage is 220-V DC. The variable Vb_fb should show a value of about 0.7813 (=400/512) when the Q format is set to Q24. The maximum boost output voltage set by the sense resistors is about 512 V that corresponds to maximum ADC input of 3.3 V. Therefore, the normalized or per unit value will be about 0.7813 when the actual boost output voltage is 400-V DC.

9. Follow Step 10 and Step 11 in Section 3.1.4.6 to turn off the power and reset the MCU. Undo the change in code performed in , then recompile and reload the code into the Flash memory for standalone operation of the EVM. Set the Piccolo control card jumpers (see the Piccolo control card documentation for your device-specific data sheet) appropriately so the device can boot from flash.

4 References

For more information please refer to the following guides:

1. MPPT DC-DC-GUI-QSG – A quick-start guide for quick demo of the MPPT DC-DC EVM using a GUI interface. ..\controlSUITE\development_kits\HV_SOLAR_DC_DC\~Docs\QSG_HV_SOLAR_DC_DC_GUI_Rev1.0.pdf
2. MPPT DC-DC_Rel-1.0-HWdevPkg – A folder containing various files related to the Piccolo-B controller card schematics and the MPPT DC-DC schematic. ..\controlSUITE\development_kits\HV_SOLAR_DC-DC\HV_SOLAR_DC-DC_HWDevPkg
3. F28xxx User's Guides http://www.ti.com/lsds/ti/microcontrollers_16-bit_32-bit/c2000_performance/real-time_control/f2802x_f2803x_f2806x/overview.page

5 Design Files

5.1 Schematics

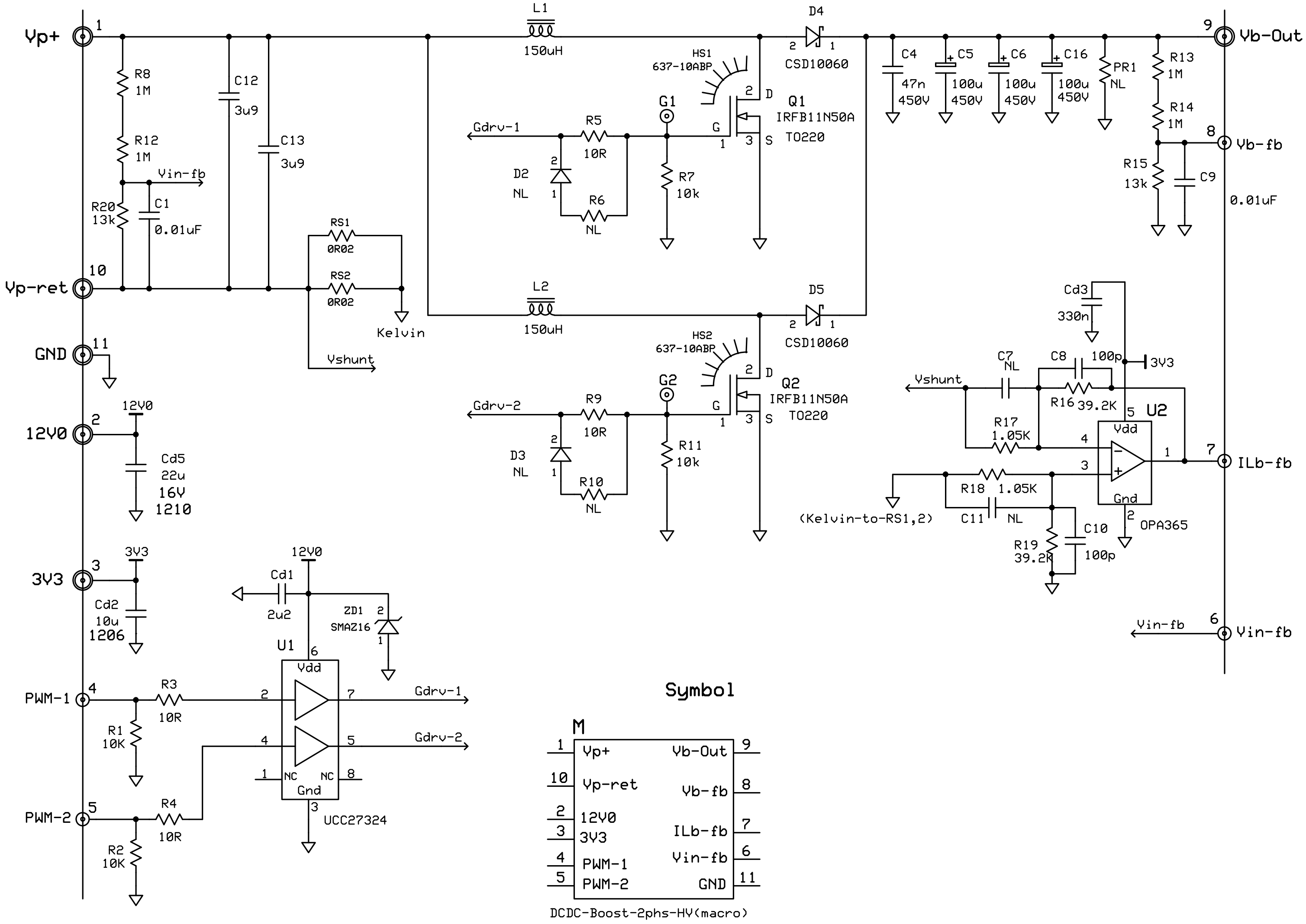


Figure 24. DC-DC Boost

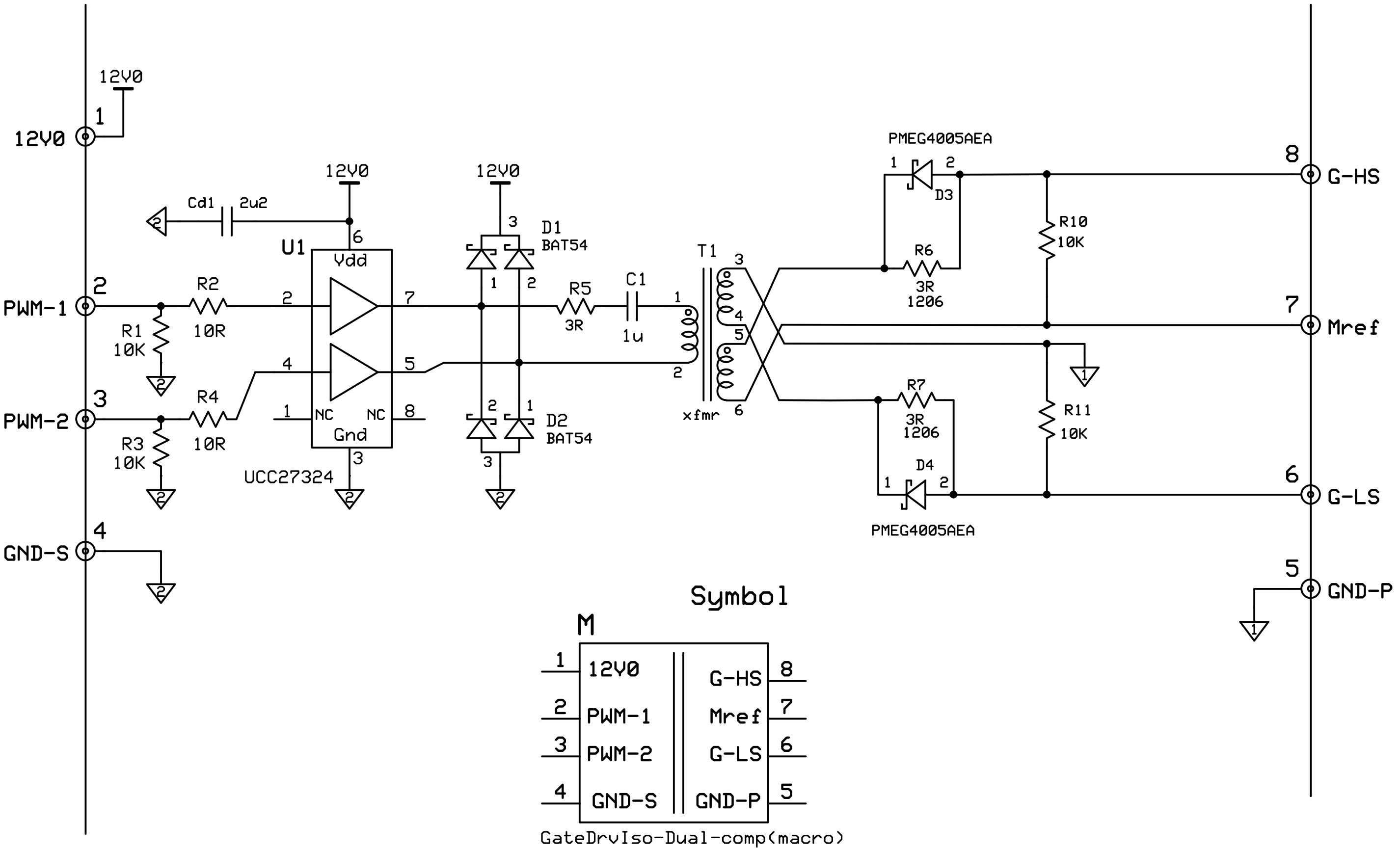


Figure 25. Gate Driver

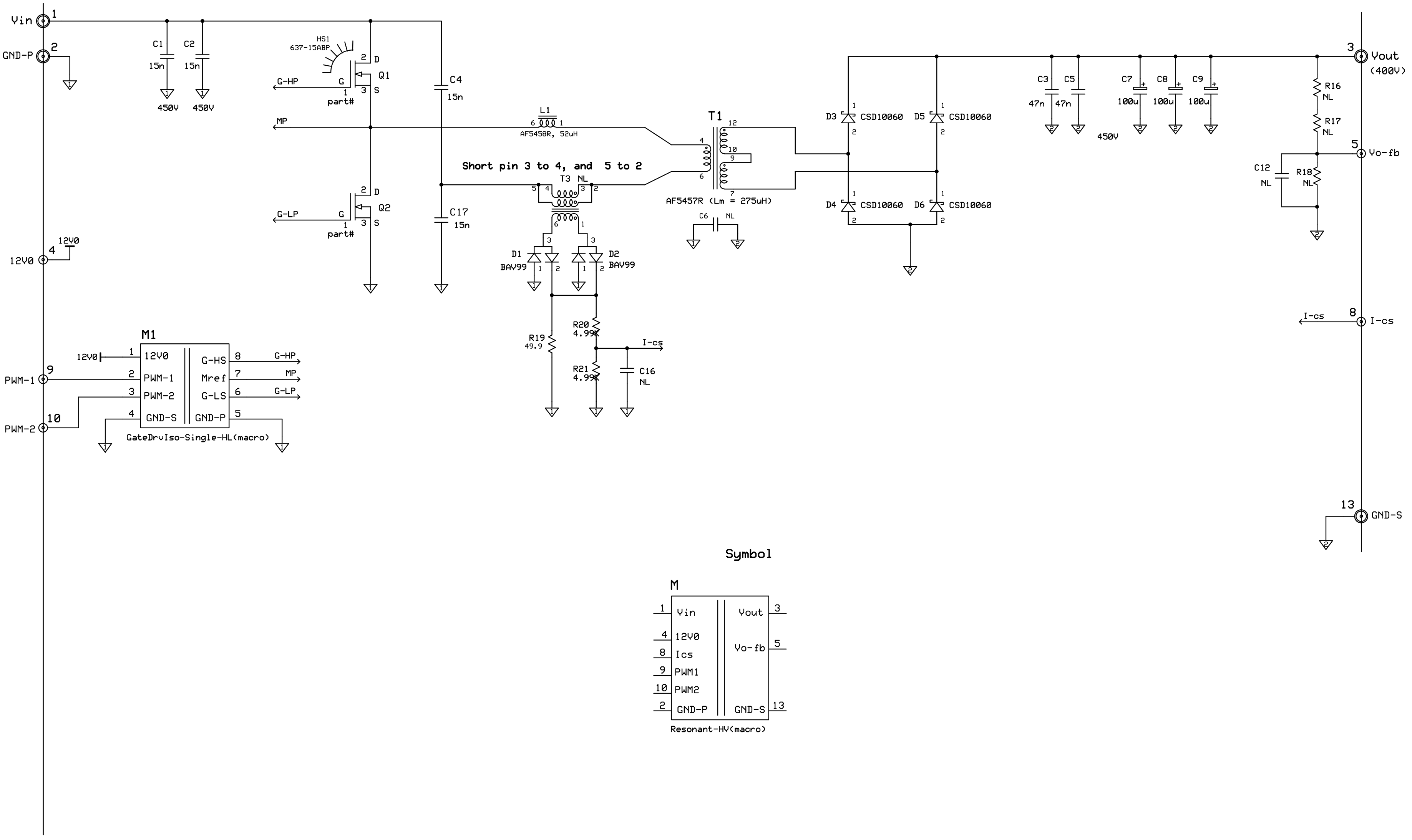


Figure 26. Resonant HV

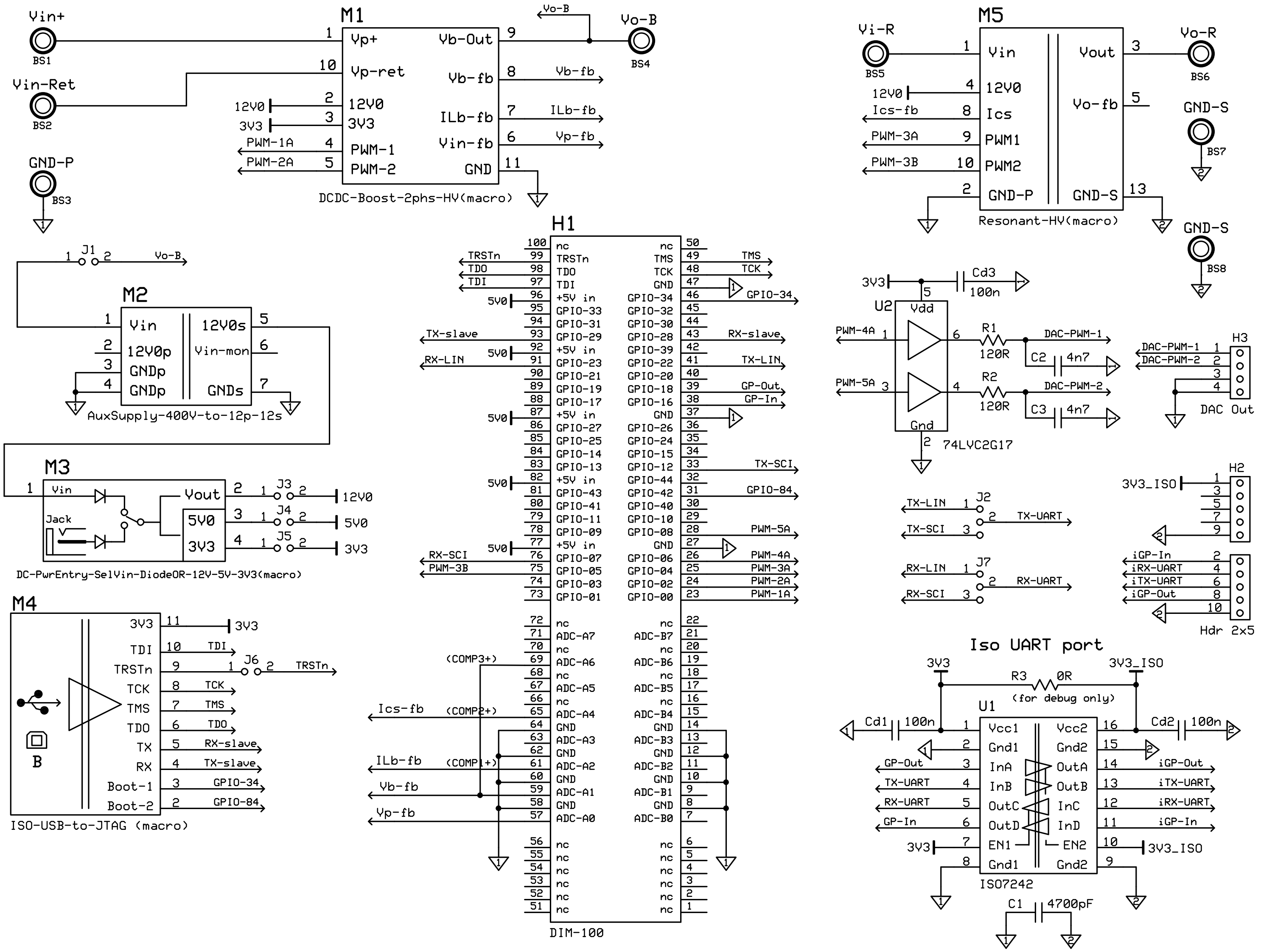


Figure 27. Solar HV

5.2 Bill of Materials (BOM)

Table 4. Bill of Materials (BOM)

MODULE	DESCRIPTION	MANEX PART NO	MFGR	MFGR PART NO	VENDOR	VENDOR PART NO	LOCATION	QTY/ BOARD
Solar HV Prim Main Board[R4]	100 pin DIMM socket - Molex 0876301001	160-0000913	Molex	87630-1001		538-87630-1001	H1	1
Solar HV Prim Main Board[R4]	AuxSupply-400 Vin-to-12 V	PR798				Custom Made	M2	1
Solar HV Prim Main Board[R4]	Ceramic, 1000 pF, 0805	121-0002344	Panasonic-ECG	ECJ-2VB1H102K -Obsoleted		PCC102BNTR-ND	C2, C3	2
Solar HV Prim Main Board[R4]	Banana Connector and Screws	260-0005067	Pomona Electronics	3267		501-1115-ND	BS1~ BS8	8
Solar HV Prim Main Board[R4]	Standard 0.1" SIL headers, cut to fit, total needed 2x5, Right Angle				DIGI	A34346-40-ND	H2	1
Solar HV Prim Main Board[R4]	Standard 0.1" SIL headers, cut to fit, total needed 2x5, DELETE, Changed to Right Angle header	260-0009772					H2	0
Solar HV Prim Main Board[R4]	Standard 0.1" SIL headers, cut to fit, total needed 1x4	260-0008027					H3	1
Solar HV Prim Main Board[R4]	Standard 0.1" SIL headers, cut to fit, total needed 1x3	260-0008027					J2, J7	2
Solar HV Prim Main Board[R4]	Standard 0.1" SIL headers, cut to fit, total needed 1x2	260-0008027					J1, J3,J4,J5,J6	5
Solar HV Prim Main Board[R4]	CAP CERM 4700PF 3000 V X7R 20% RA				DIGI	478-4042-ND	C1	0
Solar HV Prim Main Board[R4]	CAP CER .1 µF 50 V 10% X7R 0805	DIGIKEY 490-1666-1-ND					Cd1, Cd2, Cd3	3
Solar HV Prim Main Board[R4]	10K	DIGIKEY RHM10.0KCCT-ND					R1, R2	2
Solar HV Prim Main Board[R4]	NL	NL					R3	0
Solar HV Prim Main Board[R4]	74LVC2G17, DELETE	150-0018026	Texas Instruments	SN74LVC2G17DCKR		296-13013-2-ND	U2	0
Solar HV Prim Main Board[R4]	74LVC2G17, corrected the DIGI p/n (see DIGI p/n column)		Texas Instruments	SN74LVC2G17DCKR		296-13012-2-ND	U2	1

Table 4. Bill of Materials (BOM) (continued)

MODULE	DESCRIPTION	MANEX PART NO	MFGR	MFGR PART NO	VENDOR	VENDOR PART NO	LOCATION	QTY/ BOARD
Solar HV Prim Main Board[R4]	ISO7242	150-0018027	Texas Instruments	ISO7242ADWR		ISO7242ADWR-ND	U1	1
USB-JTAG-ISO[R1]	Cer. Cap - 1 µ, 6.3 V, X5R, 0603	121-0000154	Panasonic-ECG	ECJ-1VB1A105K -Obsoleted		PCC2174CT-ND	C1, C4, C5	3
USB-JTAG-ISO[R1]	Cer. Cap - 2 µ2, 6.3 V, X5R, 0603	121-0000169	Murata Electronics	GRM188R60J225ME01D		490-1551-1-ND	C2	1
USB-JTAG-ISO[R1]	Cer. Cap - 100n, 16 V, X7R, 0603	121-0000110	Murata Electronics	GRM188R71C104KA01D		GRM188R71C104KA01D	C3, C6-C16	12
USB-JTAG-ISO[R1]	Cer. Cap - 0.1 µf, 100 V, 20% , Axial	221-0005072	AVX	SA111E104MAR		478-3154-1-ND	C17	1
USB-JTAG-ISO[R1]	Diode Array, Quad, SOT363	125-0000911	Diodes	BAW567DW-7-F		BAW567DW-FDICT-ND	DA1	1
USB-JTAG-ISO[R1]	Standard 0.1" SIL headers, cut to fit, total needed 1x8	260-0008027					J1, J3, J4, J5	4
USB-JTAG-ISO[R1]	Standard 0.1" DIL header, cut to fit, remove pin 6, total needed 2x7	260-0009772					J2	1
USB-JTAG-ISO[R1]	Connector - USB Type B	160-0000912	Mill-Max	897-43-004-90-000000		ED90064-ND	JP1	1
USB-JTAG-ISO[R1]	Inductor, 22 µH, 0805	131-0004603	Murata	LQM21FN220N00L		490-4030-1	L1, L2	2
USB-JTAG-ISO[R1]	LED, green, 0805	144-0000911	Stanley	PG1112H-TR		404-1021-1	LD1	1
USB-JTAG-ISO[R1]	Res. - 470R, 5%, 0.1W, 0603	101-0000460	Rohm	MCR03EZPJ471		RHM470GCT-ND	R1	1
USB-JTAG-ISO[R1]	Res. - 27R, 5%, 0.1W, 0603	101-0000359	Rohm	MCR03EZPJ270		RHM27GCT-ND	R2, R3	2
USB-JTAG-ISO[R1]	Res. - 2K2, 5%, 0.1W, 0603	101-0000307	Rohm	MCR03EZPJ222		RHM2.2KGCT-ND	R4, R6, R9, R10, R14	5
USB-JTAG-ISO[R1]	Res. - 1M, 1%, 0.1W, 0603	101-0000209	Rohm	MCR03EZPFX1004		RHM1.00MHCT-ND	R5	1
USB-JTAG-ISO[R1]	Res. - 10K, 5%, 0.1W, 0603	101-0000247	Rohm	MCR03EZPJ103		RHM10KGCT-ND	R7	1
USB-JTAG-ISO[R1]	Res. - 1K, 5%, 0.1W, 0603	101-0000297	Rohm	MCR03EZPJ102		RHM1.0KGCT-ND	R8	1
USB-JTAG-ISO[R1]	Res. - 680R, 5%, 0.1W, 0603	101-0000520	Rohm	MCR03EZPJ681		RHM680GCT-ND	R11	1
USB-JTAG-ISO[R1]	Res. - 47K, 5%, 0.1W, 0603	101-0000466	Rohm	MCR03EZPJ473		RHM47KGCT-ND	R13	1

Table 4. Bill of Materials (BOM) (continued)

MODULE	DESCRIPTION	MANEX PART NO	MFGR	MFGR PART NO	VENDOR	VENDOR PART NO	LOCATION	QTY/ BOARD
USB-JTAG-ISO[R1]	FTDI UART/FIFO Dual, 48-LQFP	150-0017881	FTDI	FT2232D-REEL		768-1010-1-ND	U1	1
USB-JTAG-ISO[R1]	EEPROM, SOT23-6	150-0000913	Microchip	93LC46BT-I/OT		93LC46BT-I/OTCT-ND	U2	1
USB-JTAG-ISO[R1]	DFF - LVC2G74	150-0000407	Texas Instruments	SN74LVC2G74DCTR		296-13273-1-ND	U3	1
USB-JTAG-ISO[R1]	LDO - 3.3 V - TPS73033	150-0000469	Texas Instruments	TPS73033DBVR		296-17580-1-ND	U4	1
USB-JTAG-ISO[R1]	Digital Isolator, Quad 4/0, 25 Mbps	150-0004608	Texas Instruments	ISO7240CDW		595-ISO7240CDW	U5	1
USB-JTAG-ISO[R1]	Digital Isolator, Quad 2/2, 25 Mbsp	150-0004611	Texas Instruments	ISO7242CDW		595-ISO7242CDW	U6	1
USB-JTAG-ISO[R1]	Resonator - 6 MHz	000-0000911	Murata	CSTCR6M00G53Z-R0		490-1218-1-ND	X1	1
Solar HV DC-DC Boost[R2]	Film Cap - 47n, 630 VDC METAL POLY CAP	221-0007814	Panasonic-ECG	ECQ-E6473KZ -Obsoleted		E6473-ND	C4	1
Solar HV DC-DC Boost[R2]	Elec. Cap - 100 µf, 450 V, Alum, radial	221-0010341	United Chemi-Con	EKXG451ELL101MM40S		565-1481-ND	C5,C6,C16	3
Solar HV DC-DC Boost[R2]	CAP CER 0.01 µF 50 V 5% 0805	121-0000237	Murata	GRM216R71H103KA01D		490-1664-1-ND	C1, C9	2
Solar HV DC-DC Boost[R2]	Cer. Cap - 2u2, 25 V, X7R, 0805	121-0010984	Panasonic-ECG	ECJ-2FF1E225Z -obsoleted		PCC2231CT-ND	Cd1	1
Solar HV DC-DC Boost[R2]	Cer Cap, 22 µ, 1210,	121-0007264	Murata	GRM32ER71C226KE18L		490-4524-1-ND	Cd5	1
Solar HV DC-DC Boost[R2]	MLC Cap - 3.9 µF, 500 VDC, CAP, (AVX p/n SK087C395KAR)	221-0018031	AVX	SK087C395KAR		478-5685-ND	C12,C13	2
Solar HV DC-DC Boost[R2]	RES 10.0K Ω 1/8W 1% 0805 SMD	101-0000627	Rohm	MCR10EZHF1002		RHM10.0KCCT-ND	R1,R2,R7,R11	4
Solar HV DC-DC Boost[R2]	RES 10.0 Ω 1/8W 1% 0805 SMD	101-0000608	Rohm	MCR10EZHF10R0		RHM10.0CCT-ND	R3,R4,R5,R9	4
Solar HV DC-DC Boost[R2]	NL	NL	Rohm	MCR10EZHF2491		RHM2.49KCCT-ND	R6,R10	0
Solar HV DC-DC Boost[R2]	Res. - 1.0M, 0.1%, 0.1W, 0805	101-0000572	Panasonic-ECG	ERA-6AEB105V		P1MDACT-ND	R8,R12	2
Solar HV DC-DC Boost[R2]	Res. - 1.0M, 0.1%, 0.1W, 0805	101-0000572	Panasonic-ECG	ERA-6AEB105V		P1MDACT-ND	R13,R14	2
Solar HV DC-DC Boost[R2]	RES 13.0K Ω 1/8W 1% 0805 SMD,	101-0000661	Rohm	MCR10EZHF1302		RHM13.0KCCT-ND	R15	1

Table 4. Bill of Materials (BOM) (continued)

MODULE	DESCRIPTION	MANEX PART NO	MFGR	MFGR PART NO	VENDOR	VENDOR PART NO	LOCATION	QTY/ BOARD
Solar HV DC-DC Boost[R2]	RES 13.0K Ω 1/8W 1% 0805 SMD,	101-0000661	Rohm	MCR10EZHF1302		RHM13.0KCCT-ND	R20	1
Solar HV DC-DC Boost[R2]	RESISTOR .020 Ω 1W 1% 2512	101-0009399	Panasonic-ECG	ERJ-M1WSF20MU		P20MCT-ND	RS1, RS2	1
Solar HV DC-DC Boost[R2]	DIODE ZENER 1W 16 V SMA	125-0007804	Micro Commercial	SMAZ16-TP		SMAZ16-TPMSCT-ND	ZD1	1
Solar HV DC-DC Boost[R2]	Dual Power Mosfet Driver HS, 4A, 8-SOIC, UCC27324QDRQ1	150-0007815	Texas Instruments	UCC27324QDRQ1		296-23486-1-ND	U1	1
Solar HV DC-DC Boost[R2]	IPP60R250CP, MOSFET N-CH 650 V 12A TO-220	240-0007816	Infineon	IPP60R250CP		IPP60R250CPIN-ND	Q1,Q2	2
Solar HV DC-DC Boost[R2]	CSD10060G, DIODE SCHOTTKY 600 V 10A TO263-2	125-0007819	Cree	CSD10060G -obsoleted		CSD10060G-ND	D4,D5	2
Solar HV DC-DC Boost[R2]	DO NOT POPULATE	NL					D2, D3	0
Solar HV DC-DC Boost[R2]	Heat Sink, 2.0", Wakefield, (DELETE)	676-0007857	Wakefield Thermal Sol	637-20ABP		345-1030-ND	HS1,HS2	0
Solar HV DC-DC Boost[R2]	Heat Sink, 1.5", Wakefield,		Wakefield Thermal Sol	637-15ABPE		345-1112-ND	HS1,HS2	2
Solar HV DC-DC Boost[R2]	Nylon Washers for Heatsink Stad offs , 0.120x0.375 Nylon	675-0009334	Keystone	3118		3118K		4
Solar HV DC-DC Boost[R2]	Heat Pads Insulators	251-0011236	Bergquist	SP400-0.007-00-51		BER205-ND		2
Solar HV DC-DC Boost[R2]	Shoulder washer for Imperial 440 Screws				Mouser	749-SW-032-135	Ref Q1, Q2	
Solar HV DC-DC Boost[R2]	Imperial 440 Screws						Ref Q1, Q2	
Solar HV DC-DC Boost[R2]	NL	NL					PR1	
Solar HV DC-DC Boost[R2]	Cooper Inductors , CTX16-18833-R, 100 Khz, 500 µH, 3A Peak , 400W Boost Inductor	131-0018033				CTX16-18833-R	L1,L2	2
Solar HV DC-DC Boost[R2]	Amplifier, OPA365, SOT-23-5		Texas Instruments	OPA365AIDBVT		296-20645-2-ND	U2	1
Solar HV DC-DC Boost[R2]	Amplifier, OPA354, SOT-23-5, DELETE	150-0005126	Texas Instruments	OPA354AIDBVT		296-13033-1-ND 595-OPA354AIDBVT	U2	0

Table 4. Bill of Materials (BOM) (continued)

MODULE	DESCRIPTION	MANEX PART NO	MFGR	MFGR PART NO	VENDOR	VENDOR PART NO	LOCATION	QTY/ BOARD
Solar HV DC-DC Boost[R2]	RES 4.99K Ω 1/8W 1% 0805 SMD, DELETE	NL	Rohm	MCR10EZHF4991		RHM4.99KCCT-ND	R16, R19	0
Solar HV DC-DC Boost[R2]	RES 39.2K Ω 1/8W 0.1% 0805 SMD		Rohm				R16, R19	2
Solar HV DC-DC Boost[R2]	Res. - 100R, 0.1%, 0.1W, 0805, DELETE	NL	Rohm	MCR10EZHF1000		RHM100CCT-ND	R17, R18	0
Solar HV DC-DC Boost[R2]	Res. - 1.05K, 0.1%, 0.1W, 0805						R17, R18	
Solar HV DC-DC Boost[R2]	DO NOT POPULATE	NL					C7, C11	0
Solar HV DC-DC Boost[R2]	Cer. Cap - 10p, 50 V, X7R, 0805, DELETE	121-0000279	Yageo	CC0805JRNP09BN100		311-1099-1-ND	C8, C10	0
Solar HV DC-DC Boost[R2]	Cer. Cap - 100p, 25 V, X7R, 0805						C8, C10	
Solar HV DC-DC Boost[R2]	Cer. Cap- 330n, 16 V, X5R, 0805	121-0011235	TDK	C2012X7R1C334K/1.25		445-1356-1-ND	Cd3	1
DC-PwrEntry1Sw [R1]	Cer. Cap - 22 µ, 25 V, X5R, 1206	121-0005129	Taiyo Yuden	EMK316BJ226ML-T		587-1433-1-ND	C1	1
DC-PwrEntry1Sw [R1]	Cer. Cap - 10 µ, 25 V, X5R, 1206	121-0002909	Panasonic-ECG	ECJ-3YB1E106M -Obsoleted		PCC2326CT-ND	C2	1
DC-PwrEntry1Sw [R1]	Cer. Cap - 4 µ7, 25 V, X5R, 0805	121-0000389	Murata Electronics	GRM21BR61E475KA12L		490-3335-1-ND	C3	2
DC-PwrEntry1Sw [R1]	Cer. Cap - 0.1 µf, 25 V, X5R, 0805	121-0000249	Panasonic-ECG	ECJ-2VB1E104K -Obsoleted		PCC1828CT-ND	C4	1
DC-PwrEntry1Sw [R1]	Elec. Cap - 330 µf, 63 V, Alum, radial	221-0009366	Panasonic-ECG	ECA-1EHG331		P5542-ND	C5,C6,C7	3
DC-PwrEntry1Sw [R1]	Res. - 330R, 1%, 0.1W, 0805	101-0000881	Rohm	MCR10EZHF3300		RHM330CTR-ND	R1	1
DC-PwrEntry1Sw [R1]	Res. - 470R, 5%, 0.1W, 0805	101-0000964	Rohm	MCR10EZPJ471		RHM470ARCT-ND	R2	1
DC-PwrEntry1Sw [R1]	LED, green, 0805	144-0000911	Stanley	PG1112H-TR		404-1021-1	LD1	1
DC-PwrEntry1Sw [R1]	Toggle Switch - Miniature, SPDT	249-0000911	Mountain Switch	108-2AS1T1203-EVX		108-2AS1T1203-EVX	SW1	1
DC-PwrEntry1Sw [R1]	Power Module - PTH08080	425-0006612	Texas Instruments	PTH08080WAH		296-20432-ND	M1	1
DC-PwrEntry1Sw [R1]	LDO - 3.3 V - TPS79533	150-0002191	Texas Instruments	TPS79533DCQR		296-13810-1-ND	U1	1

Table 4. Bill of Materials (BOM) (continued)

MODULE	DESCRIPTION	MANEX PART NO	MFGR	MFGR PART NO	VENDOR	VENDOR PART NO	LOCATION	QTY/ BOARD
DC-PwrEntry1Sw [R1]	Standard 0.1" SIL headers, cut to fit, total needed 1x2	260-0008027						1
DC-PwrEntry1Sw [R1]	Power jack 2.1 x 5.5 mm	260-0005141	CUI	PJ-002AH		CP-002AH-ND	JP1	1
Solar Resonant-HV-SSC [R2]	CAP .015 µF 630 V MET PLYPRO RAD	221-0014804	Nichicon	QXP2J153KRPT		493-3654-ND	C4,C17	2
Solar Resonant-HV-SSC [R2]	CAP .015 µF 630V MET PLYPRO RAD	221-0014804	Nichicon	QXP2J153KRPT		493-3654-ND	C1,C2	2
Solar Resonant-HV-SSC [R2]	NL	NL					C6	1
Solar Resonant-HV-SSC [R2]	Film Cap - 47n, 630VDC METAL POLY CAP	221-0007814	Panasonic-ECG	ECQ-E6473KZ -obsoleted		E6473-ND	C3,C5	1
Solar Resonant-HV-SSC [R2]	Elec. Cap - 100 µf, 450 V, Alum, radial	221-0010341	United Chemi-Con	EKXG451ELL101MM40S		565-1481-ND	C7, C8, C9	3
Solar Resonant-HV-SSC [R2]	NL	NL					C12	1
Solar Resonant-HV-SSC [R2]	NL	NL					C16	1
Solar Resonant-HV-SSC [R2]	DIODE SW DUAL 75 V 350MW SOT23-3	125-0012375	Diodes	BAV99-7-F		BAV99-FDICT-ND	D1,D2	2
Solar Resonant-HV-SSC [R2]	HEATSINK TO-220 VERT MT BLK 1.5" (DELETE)	676-0005123	Wakefield Thermal Sol	637-15ABP		345-1029-ND	HS1	0
Solar Resonant-HV-SSC [R2]	HEATSINK TO-220 VERT MT BLK 1.5" (DELETE)	676-0005123	Wakefield Thermal Sol	637-15ABP		345-1029-ND	HS2	0
Solar Resonant-HV-SSC [R2]	HEATSINK TO-220 VERT MT BLK 1.5"		Wakefield Thermal Sol	637-15ABPE		345-1112-ND	HS1, HS2	2
Solar Resonant-HV-SSC [R2]	Standard 0.1" SIL headers, cut to fit, 1x2	260-0008027					GND-P	1
Solar Resonant-HV-SSC [R2]	Standard 0.1" SIL headers, cut to fit, 1x2	260-0008027					GND-S	1
Solar Resonant-HV-SSC [R2]	Heat Pads Insulators	251-0011236	Bergquist	SP400-0.007-00-51		BER205-ND		2
Solar Resonant-HV-SSC [R2]	Shoulder washer for Imperial 440 Screws				Mouser	749-SW-032-135	Ref Q1, Q2	
Solar Resonant-HV-SSC [R2]	Imperial 440 Screws						Ref Q1, Q2	
Solar Resonant-HV-SSC [R2]	LLC Resonant Inductor, 55 µH, 120kHz	AF5458R	VITEC			Custom from Vitec	L1	1

Table 4. Bill of Materials (BOM) (continued)

MODULE	DESCRIPTION	MANEX PART NO	MFGR	MFGR PART NO	VENDOR	VENDOR PART NO	LOCATION	QTY/ BOARD
Solar Resonant-HV-SSC [R2]	MOSFET N-CH 600 V 20A, 0.19 Ω TO-220AB, SPP20N60S%	150-0018030	Infineon	SPP20N60C3		SPP20N60C3IN-ND	Q1,Q2	2
Solar Resonant-HV-SSC [R2]	CSD10060G, DIODE SCHOTTKY 600 V 10A TO263-2	125-0007819	Cree	CSD10060G -Obsoleted		CSD10060G-ND	D3, D4, D5, D6	4
Solar Resonant-HV-SSC [R2]	NL		Rohm	MCR10EZPF4991		RHM4.99KCRCT-ND	R16	1
Solar Resonant-HV-SSC [R2]	NL		Rohm	MCR10EZHJ000		RHM0.0ACT-ND	R17	1
Solar Resonant-HV-SSC [R2]	NL		Rohm	MCR10EZHF4750		RHM475CCT-ND	R18	1
Solar Resonant-HV-SSC [R2]	RES 49.9 Ω 1/8W 1% 0805 SMD	101-0000975	Rohm	MCR10EZPF49R9		RHM49.9CRCT-ND	R19	1
Solar Resonant-HV-SSC [R2]	RES 4.99K Ω 1/8W 1% 0805 SMD	101-0000937	Rohm	MCR10EZPF4991		RHM4.99KCRCT-ND	R20	1
Solar Resonant-HV-SSC [R2]	RES 4.99K Ω 1/8W 1% 0805 SMD	101-0000937	Rohm	MCR10EZPF4991		RHM4.99KCRCT-ND	R21	1
Solar Resonant-HV-SSC [R2]	LLC Transformer, 1:1:1, 120 kHz, OCL=275 µH	AF5457R	VITEC			Custom from Vitec	T1	1
Solar Resonant-HV-SSC [R2]	WCM-601-6 (DO NOT POPULATE) (Place SHORTs, pin 4 to 3, and pin 5 to 2)	254-0016340					T3	1
GateDrvIso-Single-HL [R3]	IC MOSFET DRVR DUAL HS 4A 8-SOIC	150-0005125	Texas Instruments	UCC27324D		296-12531-5-ND UCC27324	U1	1
GateDrvIso-Single-HL [R3]	WCM-701-6	254-0016338				West Coast Magnetics	T1	1
GateDrvIso-Single-HL [R3]	RES 3.00 Ω 1/4W 1% 0805(R5) & 1206 (R6,R7) SMD (DELETE R8, R9), (USE 1206 for R6, R7)	101-0014811	Rohm	MCR10EZHFL3R00		RHM3.00CCT-ND	R5,R6,R7,R8, R9	5
GateDrvIso-Single-HL [R3]	RES 10.0 Ω 1/8W 1% 0805 SMD	101-0000608	Rohm	MCR10EZPF10R0		RHM10.0CRCT-ND	R2,R4	2
GateDrvIso-Single-HL [R3]	RES 10.0K Ω 1/8W 1% 0805 SMD	101-0000627	Rohm	MCR10EZHF1002		RHM10.0KCCT-ND	R1,R3,R10,R11	4
GateDrvIso-Single-HL [R3]	MOSFET P-CH 30 V 1.5A SSOT3, DELETE	140-0014809	Fairchild	FDN358P		FDN358PCT-ND	Q1,Q2	0
GateDrvIso-Single-HL [R3]	SCHOTTKY RECT 40 V 0.5A SOD323, DELETE	125-0018029	NXP	PMEG4005AEAT/R		771-PMEG4005AEAT/R	D3,D4	0

Table 4. Bill of Materials (BOM) (continued)

MODULE	DESCRIPTION	MANEX PART NO	MFGR	MFGR PART NO	VENDOR	VENDOR PART NO	LOCATION	QTY/ BOARD
GateDrvIso-Single-HL [R3]	SCHOTTKY RECT 40 V 1.0A SOD323F, DELETE		NXP	PMEG4010CEH			D3,D4	
GateDrvIso-Single-HL [R3]	SCHOTTKY RECT 40 V 1.0A SOD323		NXP	PMEG4010BEA		568-6523-1-ND	D3,D4	2
GateDrvIso-Single-HL [R3]	DIODE SCHOTTKY 30 V 200MW SOT23-3	125-0000039	Diodes	BAT54A-7-F		BAT54A-FDICT-ND	D2	1
GateDrvIso-Single-HL [R3]	DIODE SCHOTTKY DUAL 30 V SOT23-3	125-0014807	Diodes	BAT54C-7-F		BAT54C-FDICT-ND	D1	1
GateDrvIso-Single-HL [R3]	CAP CER 2.2 µF 16 V X7R 10% 1206	121-0000491	TDK	C3216X7R1C225K/1.60		445-1384-1-ND	Cd1	1
GateDrvIso-Single-HL [R3]	CAP CER 1.0 µF 25 V X7R 10% 0805	121-0000317	TDK	C2012X7R1E105K		445-1354-1-ND	C1	1

5.3 PCB Layouts

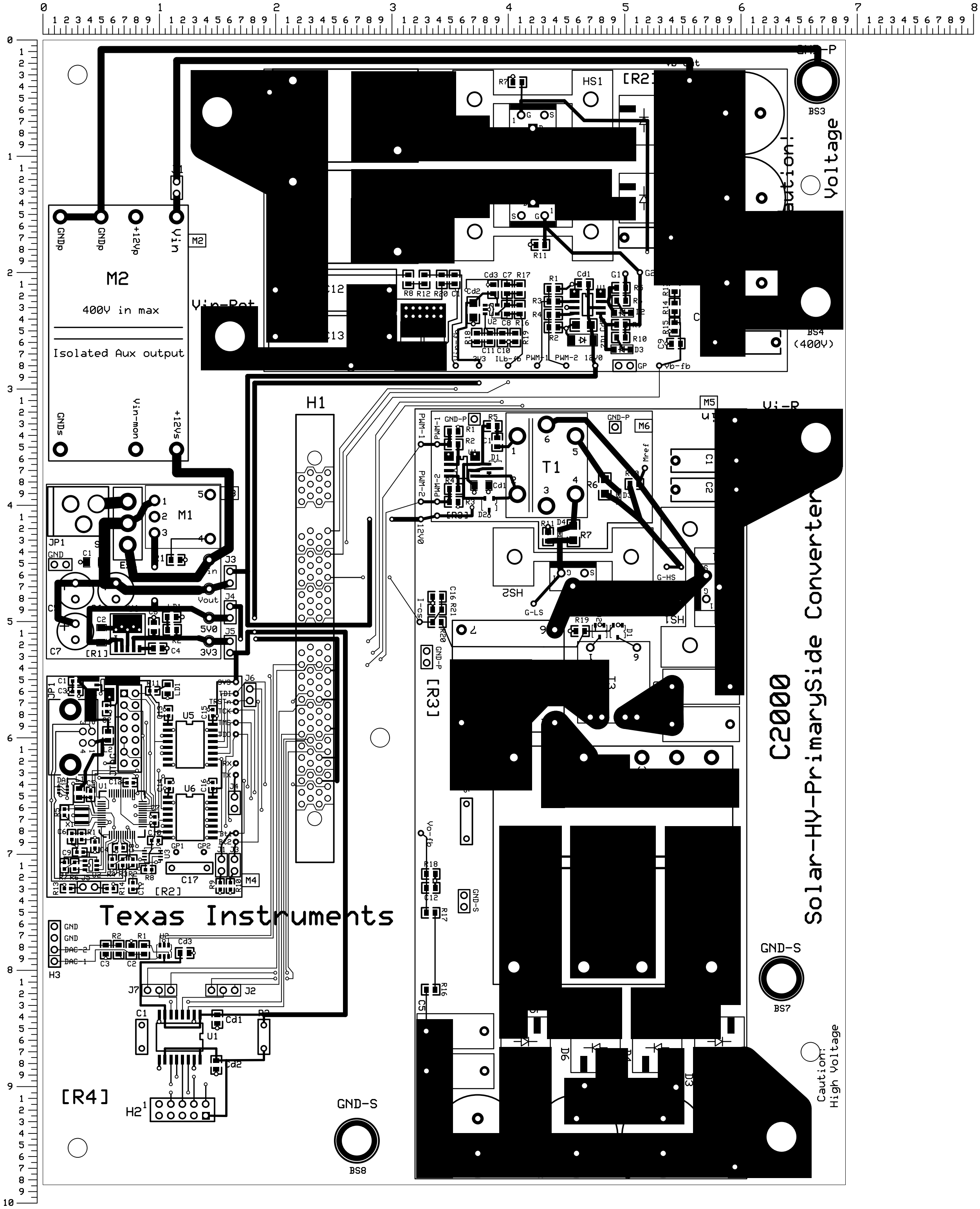


Figure 28. Top Layer

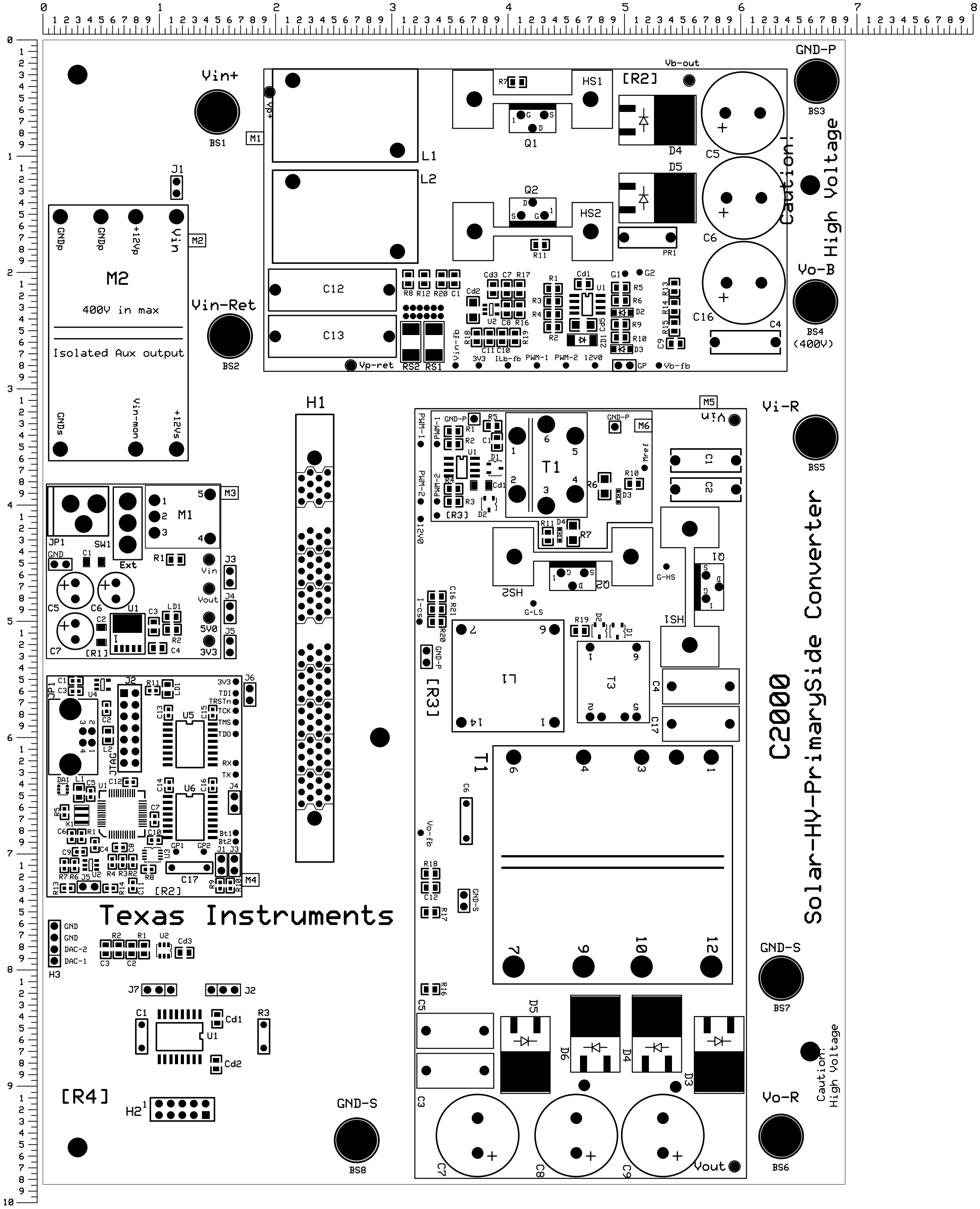


Figure 29. Silk Screen Layer

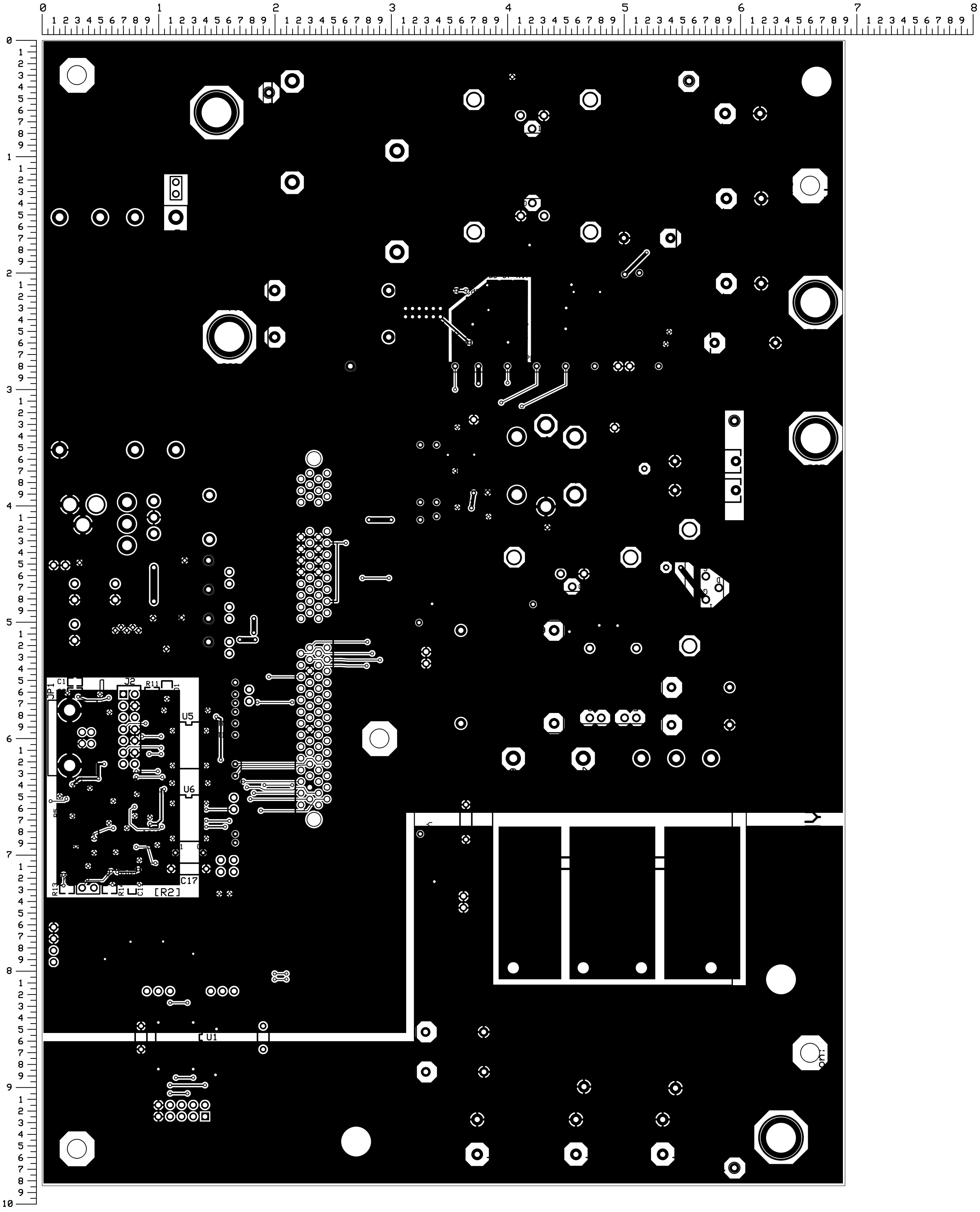


Figure 30. Bottom Layer

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as "components") are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI's terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers' products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers' products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI's goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or "enhanced plastic" are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

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