High Voltage Solar Inverter DC-AC Kit

User’s Guide
Version 1.0

Abstract

The inverters especially the solar inverter gains more and more attention in the recent years. Solar inverter has the solar energy input, then feed the solar energy to the grid. So the grid-tie technology and some of the protection is the key points when design a solar inverter system.

This document describes the implementation of the inverter kit that used as a DC-AC part of the High Voltage Solar Kit. The kit has a nominal input of 400VDC, and its output is 600W which can be fed to the grid. In this document, the basic knowledge of the inverter and the introduction of the kit will be present first. The hardware Introduction, the firmware design as well as the closed loop controllers design will be described. Besides, the build steps will be introduced as the third part. Finally, the test result and the waveform will be shown.

Key words: Solar Inverter, PLL, 2P2Z, Grid-Tie, DC-AC, Full-bridge

1 Introduction

The inverter has been widely used in many fields, such as the motor control, the UPS, and the solar inverter systems. The main function of the inverter is to convert the DC power to AC power by using the power electronics like the IGBT, and MOSFET. Traditionally, many inverter systems will be implemented by the analog components. As the development of the digital processors, more and more low cost and high performance micro-controllers had got into the market. At the same time, more and more inverter systems trends to use the micro-controllers to implement the digital controller which can not only simplify the system structure but also improve the output performance of the inverters.

Among different inverter systems, there are 2 different types. The first type is the voltage output type, which output the AC voltage as a voltage source. For example, the inverter in the UPS system is a typical voltage type inverter. The other type is the current type, which output the AC current in a specified power factor. The motor control inverter and the solar inverter are the current type inverters. This document will mainly discuss the current type inverter for the solar system.

At present, many different topologies for inverter have come out. In our demo, we use the full-bridge topology including 4 IGBT as a reference design, which is easy to get started and transplant to the real product.

1.1 The basic principles

The full bridge current type inverter topology is shown in the Figure 1.1.

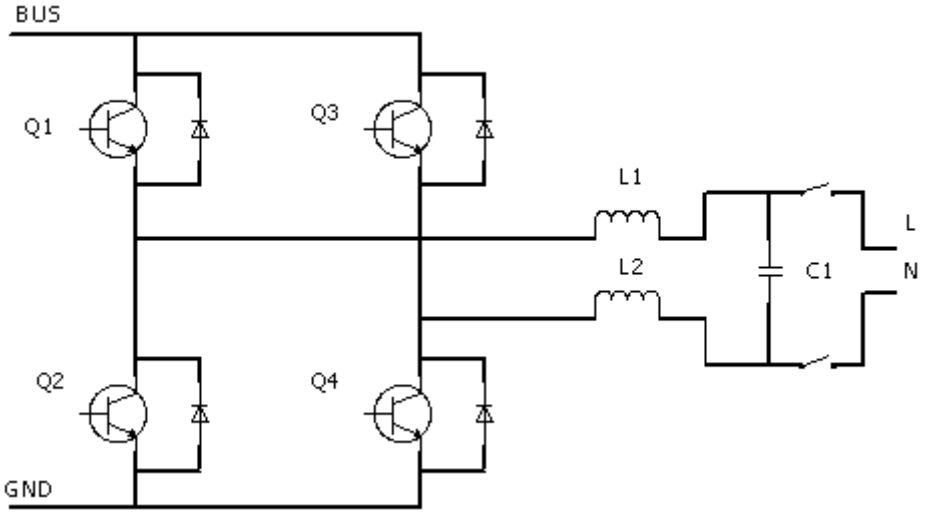


Figure1.1 Full Bridge Current Type Inverter

There are many SPWM(Sinusoid Pulse Width Modulation) control strategies for the full bridge topology to have a AC output. Among these strategies, they can be divided to the following 2 categories:

- Single polar modulation
- Dual polar modulation

1.1.1 Single Polar Modulation Theory

The single polar means the voltage in AC side of the inverter has only positive or only negative voltage. An example of the single polar modulation is shown in the Figure 1.2.

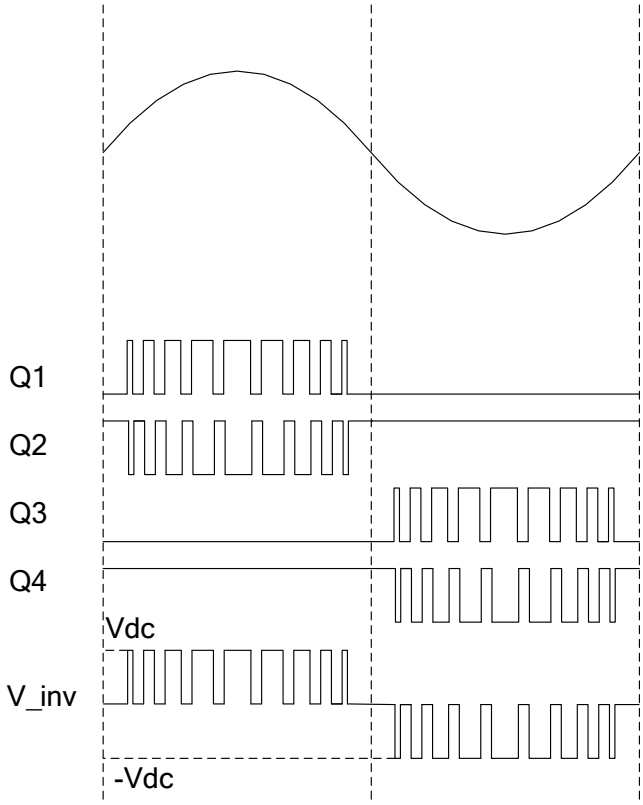


Fig 1.2 Single polar modulation theory

In Figure1.2, we can see that when in the positive cycle of the sine wave, the output voltage of the inverter is changing from the Vdc to 0, while the negative cycle is the –Vdc to 0. So in the positive cycle, if the duty of Q1 is d , then we can get the relation between the output voltage V_o and the DC bus voltage V_{bus} :

$$V_o = dV_{bus} \qquad (1\text{-}1)$$

1.1.2 The controller loop

For the current type inverter, the output current will be controlled. Besides, in most of the solar inverter systems, there is a DC-DC part in front of the DC-AC part, which is used to boost up the panel voltage and execute the MPPT. The DC-DC will not control the DC bus voltage but controls the input panel voltage and works in the power output mode. So it is the responsibility for the DC-AC part (inverter) to control the DC bus voltage.

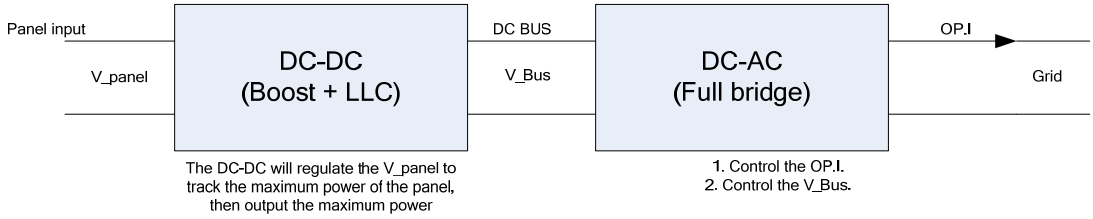


Fig 1.3 The typical solar inverter structure

The DC BUS works as a link between the DC-DC and DC-AC part, when the DC BUS voltage rise up, the DC-AC will increase its output current to keep the DC BUS in a specified value, then the output power of the system will be increased. When the DC BUS tends to fall down, the DC-AC will decrease its output current to prevent the DC BUS from falling down, by which the output power will be decreased.

So the typical controller structure for the inverter part is below:

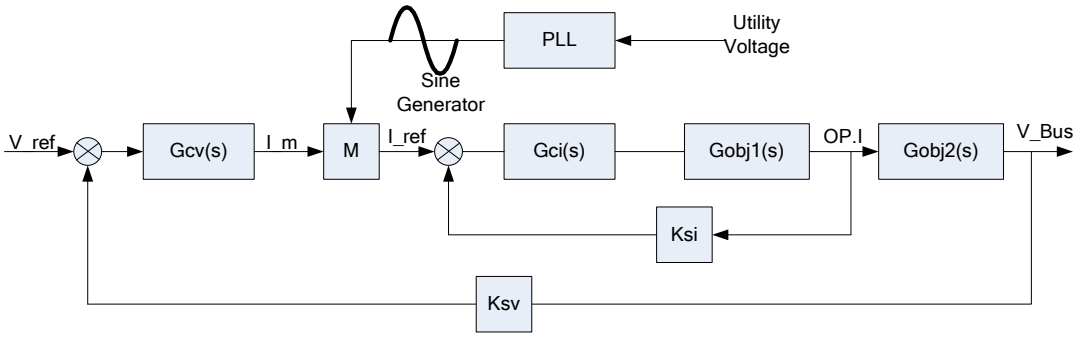


Fig 1.4 The controller loop of the inverter part in solar system

The double loop control system is used in Fig 1.4. The internal loop is the output current loop, it will trace the I_ref which is the product of the I_m and Sine. The external loop is the DC BUS voltage loop, it will keep the bus voltage to V_ref. Besides, there is a PLL to ensure the synchronization of the grid voltage and the output current.

Notes: When there is no DC-DC part and the DC-source in CV mode is connected, the external loop must be disabled.

2 Design Introduction

2.1 Hardware

2.1.1 The Key Component

The key components of the kit are shown in Fig 2.1. In this kit, 4 pieces of 600V IGBT are used. The IGBT drivers are designed to the module type. Besides, 2 pieces of 2.5mH inductor are used. 2 pieces of relay are used to control the grid-tie connection. Finally, the Hall current sensor is used to sense the inductor current.

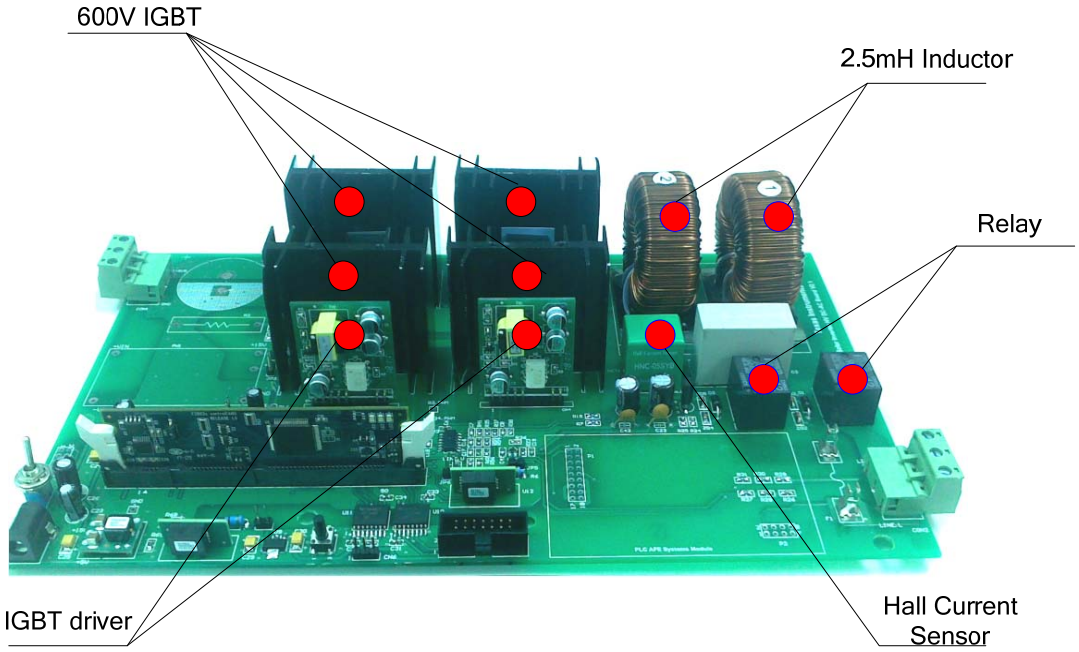


Fig 2.1 The Key Components on the board

The PCB placement is below:

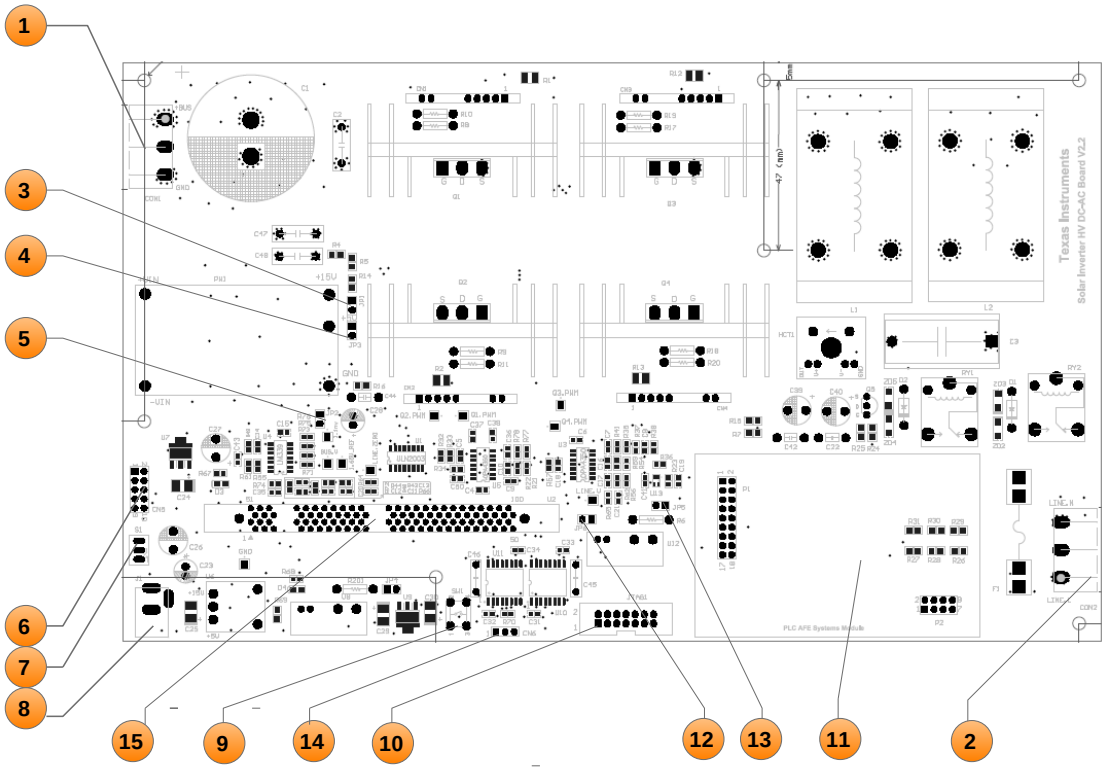


Fig 2.2 The PCB placement

Item No	Points Name	Comment
1	CON1	The DC bus connector for the DC-DC input.
2	CON2	The Utility connector L and N.
3	JP1	Onboard +15V Jumper
4	JP3	Onboard +5V Jumper
5	JP2	IGBT Driver +15V Jumper
6	CN5	DC-DC board signal interface
7	S1	External +15V Adapter Switch
8	J1	External +15V input Jack
9	SW1	Operation Button
10	JTAG1	JTAG interface for external emulator
11	PLC AFE Systems Module	Not used in this version
12	JP6	TRST Jumper
13	JP5	-15V Power Jumper
14	CN6	RS232 Port
15	U2	The DIM100 28035 Control Card Port

Table 2.1 The Key Points

2.1.2 The Auxiliary Power Supply

The auxiliary power of the kit can be available by two ways. The one is using the external +15V adapter. Insert the adapter to the J1(See the Table 2.1), then switch the S1 to power on. The other way is using the power module on board. Please see the jump configuration

in the table below.

	External +15V Adapter	Onboard +15V
JP1	×	√
JP2	√	√
JP3	×	×
JP6	Unaffected	×

Table 2.2 The Jumper Setting for the board

2.1.3 The signal sensing

There are 3 key signals are used in the controller loop:

1 The DC BUS voltage.
2 The inductor current.
3 The grid voltage.

The DC BUS voltage sensing is very simple. From the circuit, we can get the sample ratio of the signal can be calculated as the following.

$$K_{ratio\_DCBUS} = \frac{R_6}{R_4 + R_5 + R_{14} + R_6} = \frac{10}{3010} = 0.003322$$

For the inductor current sensing, there is a hall sensor, whose sample ratio is 4/5. Besides, the differential circuit is used to get a appropriate ratio. The current sample ratio if the current is calculated below:

$$K_{ratio\_current} = K_{hall} \frac{R_{41}}{R_{35} + R_{15}} = 0.15974$$

For the utility voltage, just the differential ciucuit is used, the current is calculated below:

$$K_{ratio\_grid\_voltage} = \frac{R_{59}}{R_{26} + R_{27} + R_{28} + R_{54}} = 0.003311$$

Note: For the inductor current and grid voltage, there is a 1.65V offset in the sample circuit, this need to be subtracted in the firmware.

2.1.4 Zero Crossing Detection

The zero crossing detection is used to detect the frequency, besides, it is very convenient to detect the islanding conditions. The kit uses a comparator to get a falling edge in every positive zero crossing. Besides, a positive feedback of the comparator is used to get a sharp edge. Please refer to the schematic for the detail information.

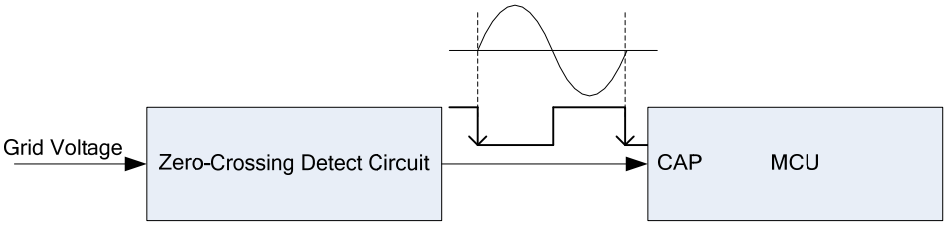


Fig 2.3 Zero Crossing

The CAP of the MCU will capture the falling edge of the input signal and save the capture value, which represent the positive zero crossing time of the grid voltage. In the firmware design, there will be a interrupt for the capture event, the frequency can be calculated by the following method.

$$f_{grid\_freq} = \frac{f_{cpu\_clk}}{(CAP_0 - CAP_1)}$$

The f_{grid_freq} is the grid frequency, the f_{cpu_clk} is the MCU CPU clock, the CAP0 is the capture value this time, the CAP1 is the capture value saved last time.

2.1.5 The IGBT driver

The kit has 4 IGBT driver modules, whose function is isolation and amplify the driving capacity. The functional diagram for the driver is below.

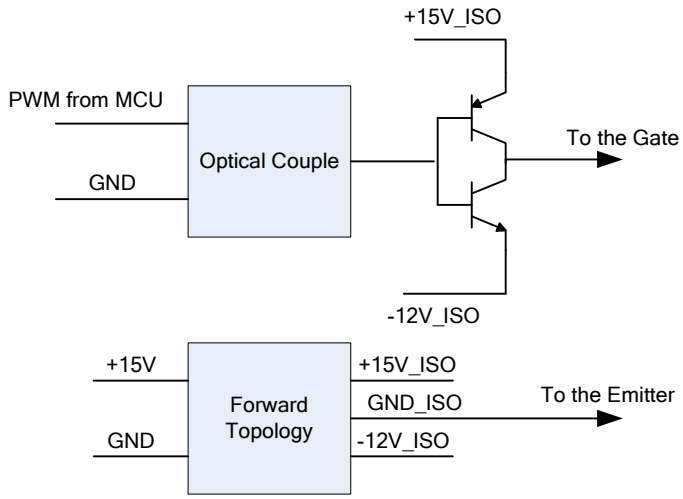


Fig 2.4 IGBT driver diagram

The driver can output +15V for the turning on status and -12V for the turning off status.

2.1.7 The Inductance

In order to smooth the current ripple, there is a inductor in the main circuit. The inductance is determined by the switching frequency f_s, the DC bus voltage V_{bus}, and the requirement of current ripple ΔI.

In a certain switching period, the inductor current can be described as below:

$$L\frac{\Delta I}{\Delta T}=V_L \qquad (1\text{-}2)$$

Where the L is the inductance, and the

Assume in the single switching period, the rise of the current ΔI_r is equal to the fall of the current ΔI_f :

$$\Delta I_r \;=\; \Delta I_f \qquad (1\text{-}3)$$

Then we can get:

$$L\frac{\Delta I_r}{\Delta T}=L\frac{\Delta I}{dT_s}=V_{bus}-V_o \quad (1\text{-}4)$$

$$L\frac{\Delta I_f}{\Delta T_f}=L\frac{\Delta I}{(1-d)T_s}=-V_o \qquad (1\text{-}5)$$

Link the (1-4),(1-5), (1-1)we can get:

$$\Delta I=\frac{V_{bus}}{Lf_s}[-(d-0.5)^2+0.25] \qquad (1\text{-}6)$$

From (1-6) we know the maximum ΔI occurs at the $d=0.5$, and the maximum value is below:

$$\Delta I_{\max}=\frac{V_{bus}}{4Lf_s} \qquad (1\text{-}7)$$

According to (1-7), we can caculate the inductance requirement of full bridge inverter with the single polar modulation.

2.2 Firmware

2.2.1 The firmware structure

The typical front and background system is used in the firmware design. For the background, 3 different timer based tasks are scheduled to deal with the non-urgent tasks. Besides, 3 interrupt service routines are used as the front to deal with the urgent things, such as the close loop controllers, the capture event, and the SCI receiving.

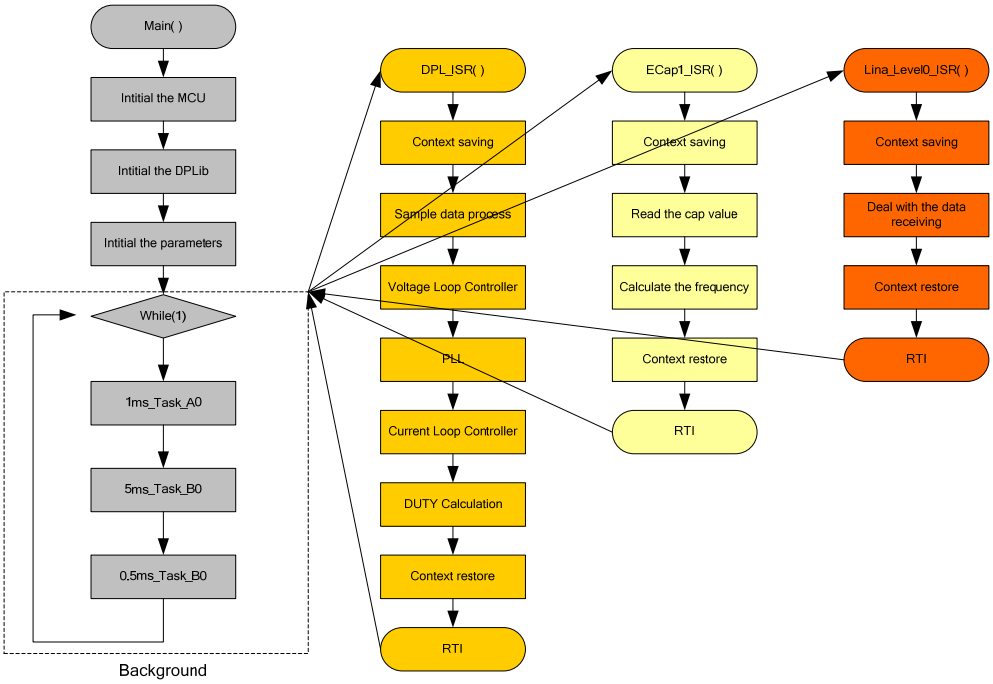


Fig 2.1 The firmware structure

2.2.2 The status machine

In order to distinguish the different status of the system, the status machine is used. Different status represents the different running mode. According to the mode, the other tasks can take the appropriate action.

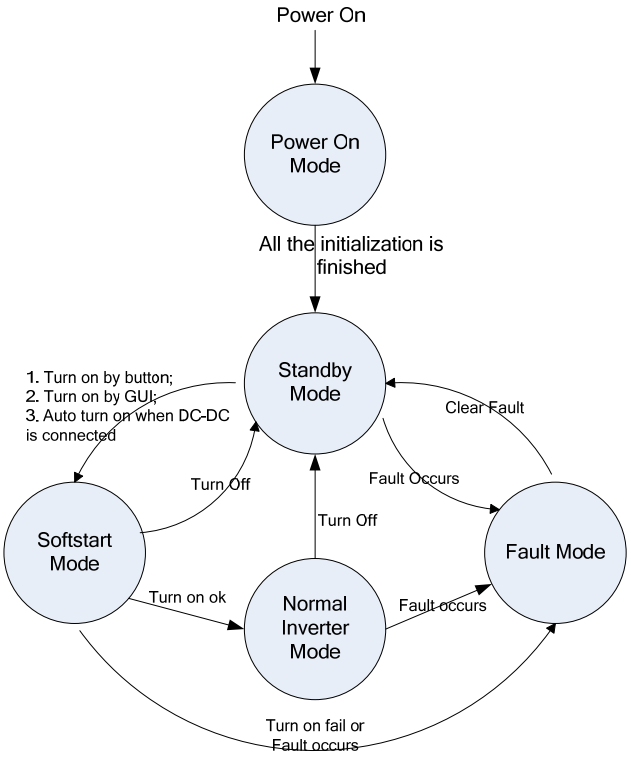


Fig 2.2 Status Machine

There are 5 different running mode in the firmware.

1 **Power On Mode**. When the board powered up, it will get into the power on mode first, the MCU will initialize itself. When the initialization is finished, the system will transfer to standby mode automatically.

2 **Standby Mode**. When the system is in standby mode, all the PWM and Relay is off. The system is waiting for the command to turn on. Besides, it will detect if the fault occurs.

3 **Soft Start Mode**. When there is a turning on command, the system will go to the soft start mode first, the PWM and relay will be turned on. When the turning on is OK and no fault occurs, the system will get into the normal inverter mode automatically.

4 **Normal Inverter Mode**. When the system is in normal inverter mode, it means the system is feed the energy out. If there is no fault and turning off command, the system will stay in this mode.

5 **Fault Mode**. When there is a fault, for example bus over voltage, the system will transfer to the fault mode immediately. All the PWM is off, the output relay is cut off from the output. The fault can be cleared by the button or the GUI. When the fault is cleared, it will return to standby mode.

2.2.3 The LED flashing design

The LED on the control card will flashing in different ways according to the running mode defined in 2.2.2. Please refer to the table below.

The LD2 is defined as the mode LED, and the LD3 is defined as the fault LED.

System Mode	LD2	LD3
Power On Mode	Always On	Always On
Standby Mode	Flashing in every 0.5s	Always Off
Standby Mode(with warning**[2]**)	Flashing in every 0.5s	Flashing in every 0.5s
Soft Start Mode	Flashing fast	Always Off
Normal Inverter Mode	Always On	Always Off
Fault Mode	Always Off	Always On**[1]**

Table 2.3 The LED flashing definition

User can get the running mode by the LED flashing quickly.

Note:

[1] If the LD3 is flashing or always on, please power off and check the hardware.

[2] When the LD3 is flashing, please press the button on the board or click the turn on button in the GUI to clear the warning. The system can be turned on only if there is no fault or warning. The warning can be generated by the following conditions: turning off, grid voltage out of range or the DC bus voltage abnormal. Please check the firmware for the warning generation details. The flag named ***FSuperFlag.BIT.FwWarning*** represent the warning status.

2.2.4 The Tasks

In the background, there are 3 main tasks are used.

1. Task_A0. The 1 millisecond task, it has 4 sub-tasks, only task A1 and task A3 are used in the system.

The sub task **A1** deals with the status machine transition. The status will be checked in every 20ms, so when the running mode is changed, the new running mode will take effect after 20ms.

The sub task **A3** deals with the onboard button detection and the LED flashing control.

2. **Task_B0**. The 4 milliseconds task, it has 4 sub-tasks.

The sub task **B1** deals with the fault detection, including the short circuit check, over current check, grid voltage and frequency check, as well as the DC bus voltage check.

The sub task **B2** deals with the measurement calculation, it will calculate the grid voltage RMS and output current RMS, the active power, the DC bus voltage, as well as the zero crossing check.

The sub task **B3** deals with the turning on check.

The sub task **B4** deals with the GUI command processing and board to board communication.

3. **Task_C0**. It is the 0.5 millisecond task. Only the C0 is used to check the SCI communication.

2.2.5 The Interrupt

There are 3 interrupts are used to deal with the real-time event.

1. The ADCINT1. The interrupt is generated by the ADC EOC. When the ADC sampling is finished, the interrupt will be triggered. The ISR will execute the controller algorithm.
2. ECAP1_INT. The interrupt is generated by the capture event. When the zero crossing happens, the falling edge will trigger the capture event.
3. LIN0INTA. The interrupt is generated by the RXD event of the LINA, the LIN is used as the SCI port to communicate with the DC-DC board.

3 How to Build the Firmware.

3.1 The file structure of the project

There are many files in the software project, including the c files, the assembly files, the head files as well as the cmd files.

C files Name	**Description**
ADC_SOC_Cnf.c	Initialize the ADC
SciCommsGui.c	Communication with GUI
SolarHv_DCAC-DevInit_F2803x.c	MCU Device initialization
SolarHv_DCAC-CAP_Cnf.c	Cap initialization
SolarHv_DCAC-Lin.c	Communication with DC-DC board
SolarHv_DCAC-main.c	The background
SolarHv_DCAC-PWM_Cnf.c	ePWM initialization

Table 3.1 C files in project

asm files Name	**Description**
SolarHv_DCAC-CNTL_2P2Z.asm	The 2P2Z controller for the current
SolarHv_DCAC-ADCDRV_5CH.asm	ADC sample
SolarHv_DCAC-DLOG_4CH.asm	Get the real time data
SolarHv_DCAC-GEN_SIN_COS.asm	Generate the sine and cosine wave
SolarHv_DCAC-INV_ICMD.asm	Calculate the current loop reference
SolarHv_DCAC-ISR.asm	The ADC interrupt ISR for the controller
SolarHv_DCAC-PWMDRV.asm	Calculate the CMPR and update the duty

Table 3.2 asm files

Other files Name	**Description**
SolarHv_DCAC-Settings.h	The project build setting
SolarHv_DCAC-f28035_FLASH.CMD	Cmd file for code running in Flash
SolarHv_DCAC--f28035_RAM.CMD	Cmd file for code running in RAM

Table 3.3 Other Files

3.2 The blocks introduction

There are some blocks which are used to realize the specified function. Users can use the blocks in their own projects.

3.2.1 The ADCDRV_5CH m n p q s

The block named ADCDRV_5CH is the ADC sampling driver module, which can be used to get 5 sample channels.

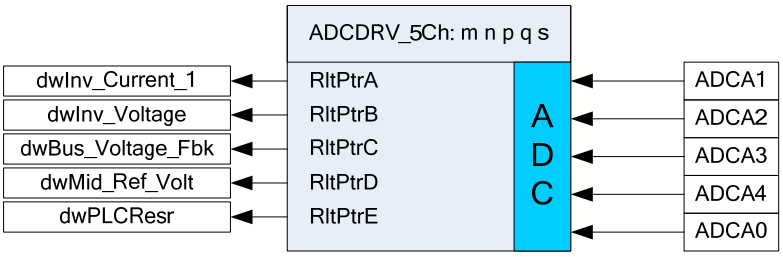


Fig 3.1 ADCDRV_5CH block

There are 5 channels ADC are used:

ADCA1 is assigned for the inductor current sensing, the **dwInv_Current_1** is named for this channel in the software, the format of dwInv_Current_1 is **Q24**.

ADCA2 is assigned for the grid voltage sensing, the **dwInv_Voltage** is named for this channel in the software, the format of **dwInv_Voltage** is **Q24**.

ADCA3 is assigned for the DC BUS voltage sensing, the **dwBus_Voltage_Fbk** is named for this channel in the software, the format of **dwBus_Voltage_Fbk** is **Q24**.

ADCA3 is assigned for the 1.65V reference sensing, the **dwMid_Ref_Volt** is named for this channel in the software, the format of **dwMid_Ref_Volt** is **Q24**.

ADCA0 is reserved for the PLC application in the future.

3.2.2 The GEN_SIN_COS: n

The GEN_SIN_COS: n is used to generate the sine wave and cosine wave.

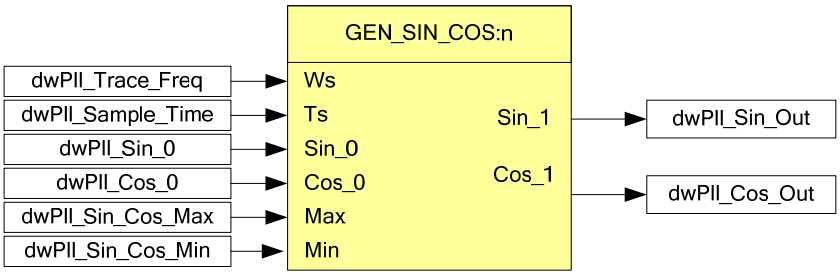


Fig 3.2 The GEN_SIN_COS: n

The **Ws** is the frequency input of the generator. The **dwPll_Trace_Freq** is assigned for this input, the format is **Q20**. For example:

dwPll_Trace_Freq = _IQ20(376.9911) represents the 60Hz.

The Ts is the sample frequency of the generator. The dwPll_Sample_Time is assigned for this input, the format is Q24. For example:

dwPll_Sample_Time = _IQ(0.000052) represents the 52e-6 seconds

The **Sin_0** is the initial value of the sine value. The **dwPll_Sin_0** is assigned for this input, the format is **Q22**. The default value of the **dwPll_Sin_0 is 0.**

The **Cos_0** is the initial value of the sine value. The **dwPll_Cos_0** is assigned for this input, the format is **Q22**. The default value of the **dwPll_Cos_0 is _IQ22(0.99).**

The **Max** is the maximum value of the output value. The **dwPll_Sin_Cos_Max** is assigned for this input, the format is **Q22**. The default value of the **dwPll_Sin_0 is _IQ22(0.99)**

The **Min** is the minimum value of the output value. The **dwPll_Sin_Cos_Min** is assigned for this input, the format is **Q22**. The default value of the **dwPll_Sin_Cos_Min is 0**

3.2.3 INV_ICMD:n

The INV_ICMD:n is used to calculate the current reference.

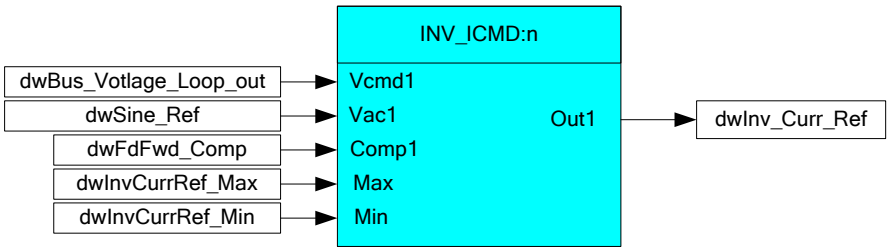


Fig 3.3 INV_ICMD:n

The Vcmd1 is the amplitude of the reference current, which is usually the voltage loop controller output. The dwBus_Voltage_Loop_Out is assigned as the interface. The format is Q24.

The Vac1 is the unit sine wave, which represent the reference angle of the current, which is usually the sine generator's output. The dwSine_Ref is assigned as the interface. The format is Q24.

The Comp1 is the compensation for the change of the grid voltage. The default value is 1.

The Max, Min is the limitation of the output.

Out1 is the output of the block, the dwInv_Curr_Ref is assigned as the interface. The format is Q24.

3.2.4 PWMDRV:n

The PWMDRV: n is used to calculate the CMPR according to the controller's output, besides, it will update the CMPR register when it finishes the calculation.

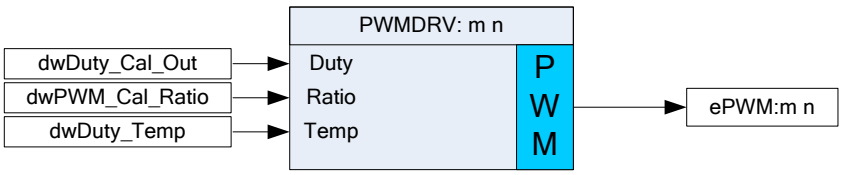


Fig 3.4 PWMDRV:n

The Duty is the output of the controller, which is usually the current loop controller output. The dwDuty_Cal_out is assigned for this input, the format is Q24.

The Ratio is the conversion ratio between the duty and the CMPR value. The format is Q8. The ratio can be calculated by the following method.
Ratio = Period * 1000 / Vdc.
The Temp is reserved for debug.

3.2.5 CNTL_2P2Z:n

This is same to the blocks defined in Digital Power Library.

3.2.6 DLOG_4CH:n

This is the similar to the blocks defined in the Digital Power Library, but the start of the log is different. In this project, when the variable wDataEnable is 1, the block starts the data log.

3.3 The Build Step

The following chapter will discuss the incremental build step, this is useful for the user to realize all the function step by step.

The build step can be set by the pre-define macro INCR_BUILD in the head file named SolarHv_DCAC-Settings.h, please see the setting in the table below.

INCR_BUILD == 1	Open Loop Build
INCR_BUILD == 2	Close Loop without PLL
INCR_BUILD == 3	Close Loop with PLL

Table 3.4 Incremental build option

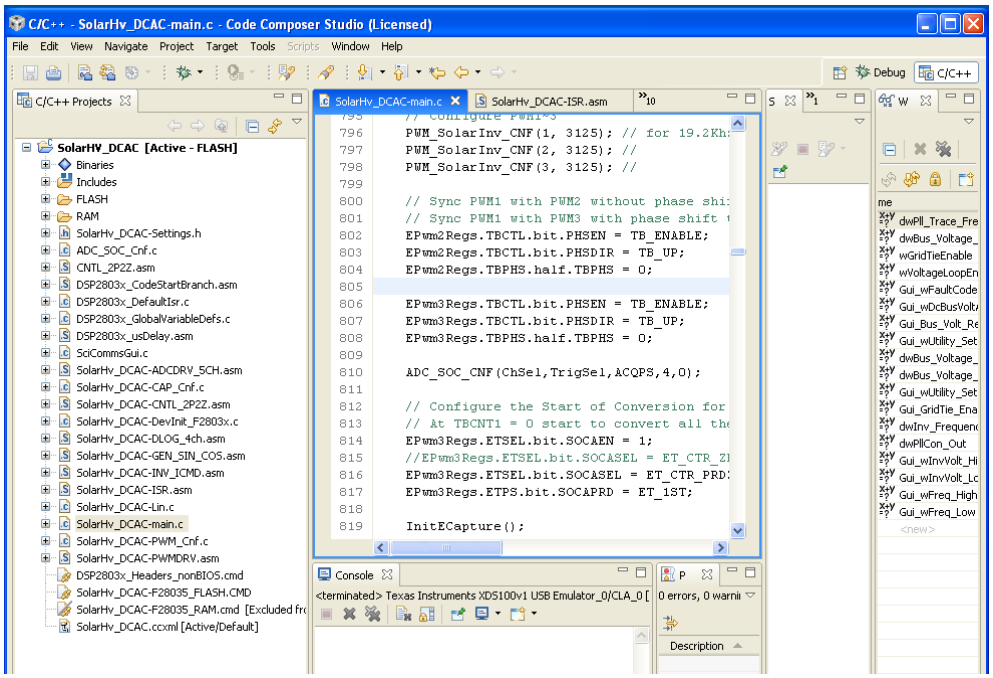
3.3.1 Start the CCS Project

1. Connect the USB cable to the ISO PiccoloB control card. Short the jumper JP2,JP4,JP5,JP6, open the jumper JP3,JP1.

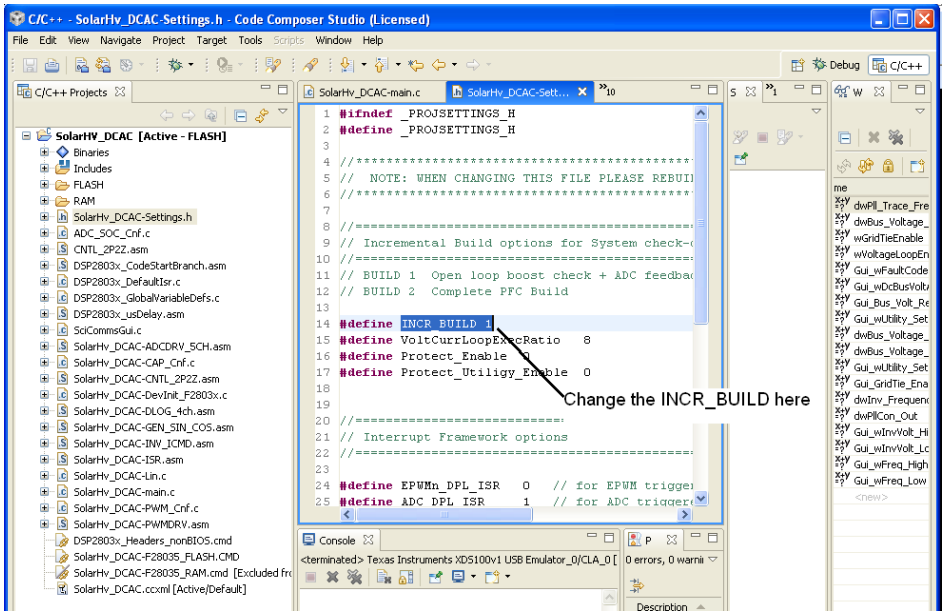
2. Insert the +15V adapter to J1, then switch the S1 to power on the auxiliary power.

3. Start the CCSV4, create a new workspace. When the IDE is opened, please click the menu: Project \Import Existing CCS \CCE Eclipse Project and under Select root directory navigate to and select ..***controlSUITE\development_kits\Solar HV Kit\DC-AC board.***

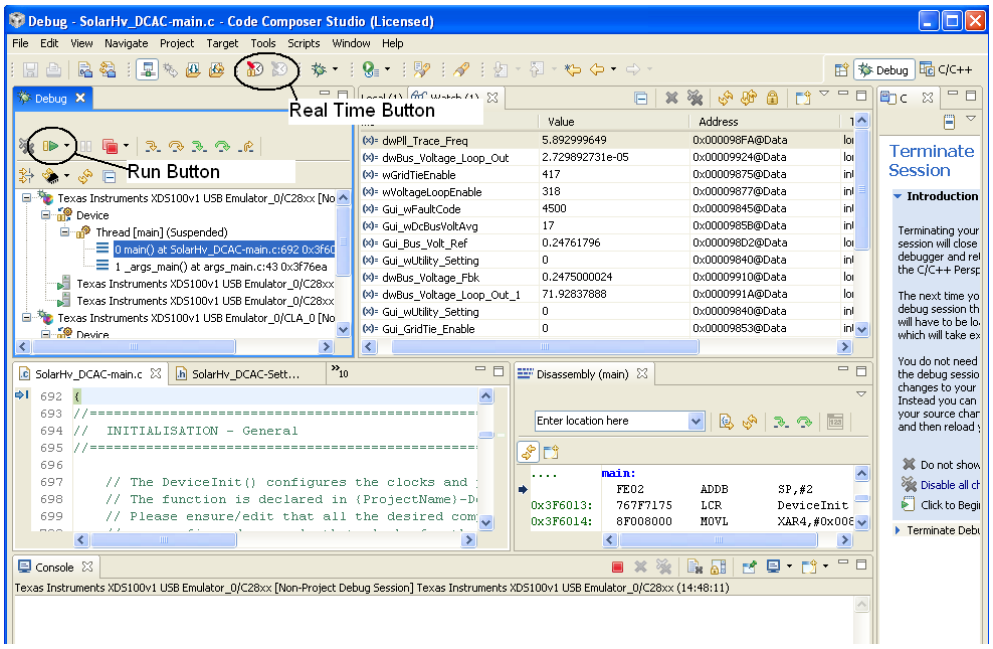
4. When the project is opened successfully, you can see the following workspace.



5. Change the incremental build option by set a value to INCR_BUILD.



6. Set the build configuration by clicking the menu: Project\Active Build Configuration. If user want to run the code the in RAM, please choose the RAM option, or the FLASH option need to be chose.
7. Rebuild the project by clicking the menu: Project\Rebuild All. If there is no error, the new .out file will be created.
8. In the .ccxml file that open up select Connection as “Texas Instruments XDS100v2 USB Emulator” and under the device, scroll down and select “TMS320F28035”. Click Save.
9. Start the TI debugger by clicking the **Target\Debug Active Project**.
10. When the code is loaded successfully, you will see the debug window as below.



11. Using the real-time debug option by clicking the button in the tool bar.
12. Run the code by clicking the “run” button in the tool bar.

3.3.2 Open Loop Build

The first step is the open loop build, let the board output a sine wave. In this step, the GEN_SIN_COS and the PWMDRV block are used to generate the SPWM. Besides, the DLOG_4CH and ADCDRV_5CH are also used, the user can check the sample data by the real time or the GUI(If the GUI is used, the user must run the code in the flash).

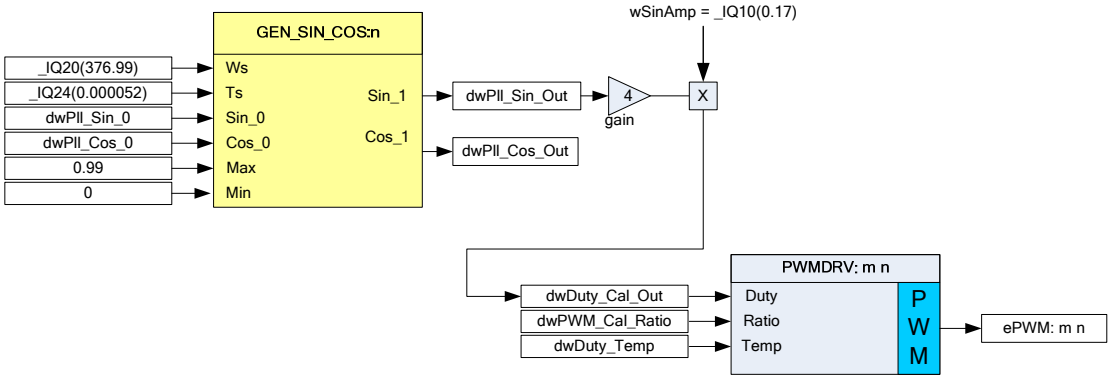


Fig 3.5 The open loop build

The open loop build can be available when set the **INCR_BUILD = 1** in the **SolarHv_DCAC-Settings.h** file.

When the code is running, set the DC source input to about 400V, then please press the SW1 to turn on the board.

3.3.2 Close Loop Build without PLL

When the grid is not connected to the board, the board can run the close loop without the PLL. It will output a constant current to the load.

Before build this step, make sure you do the open loop test successfully. Besides, it must connect a resistor load to the output. The suggested resistor load is 25ohm/1000W.

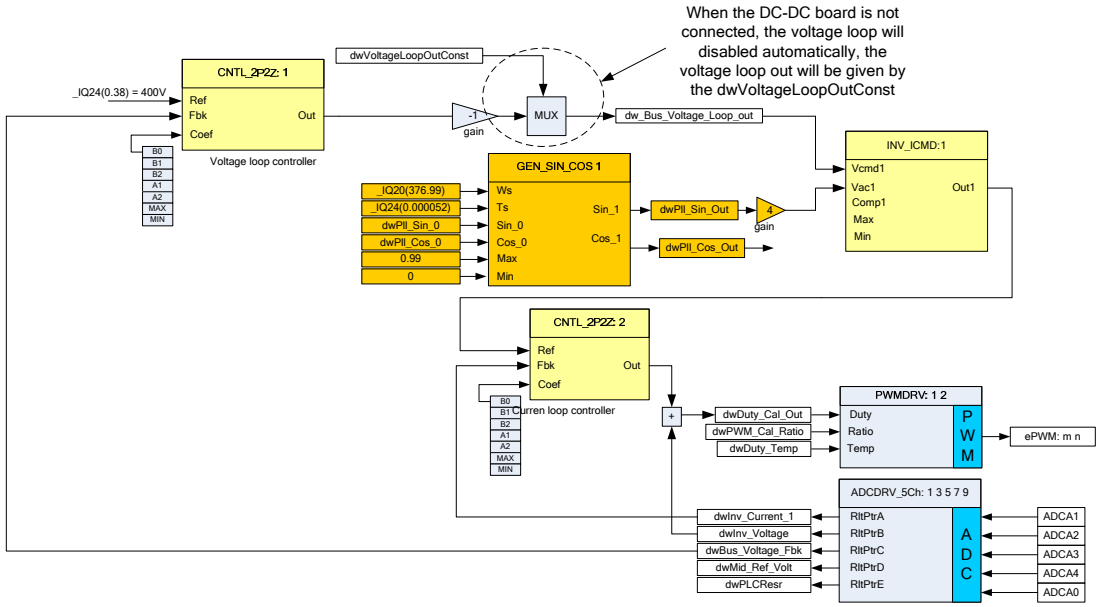


Fig 3.6 Colse Loop Build without PLL

The close loop without PLL build can be available when set the **INCR_BUILD = 2** in the **SolarHv_DCAC-Settings.h** file.

Please note that when the DC-DC board is not connected, the voltage loop will be

disabled automatically. The **dw_Bus_Voltage_Loop_Out** will be given by the dwVoltageLoopOutConst directly. User can modify the **dwVoltageLoopOutConst** in the real time to get the different output current value.

3.3.3 Close Loop Build with PLL

If all the above build is finished, user can do the final build step for the grid tie test.
The user must connect the test tool to the board like the following diagram.

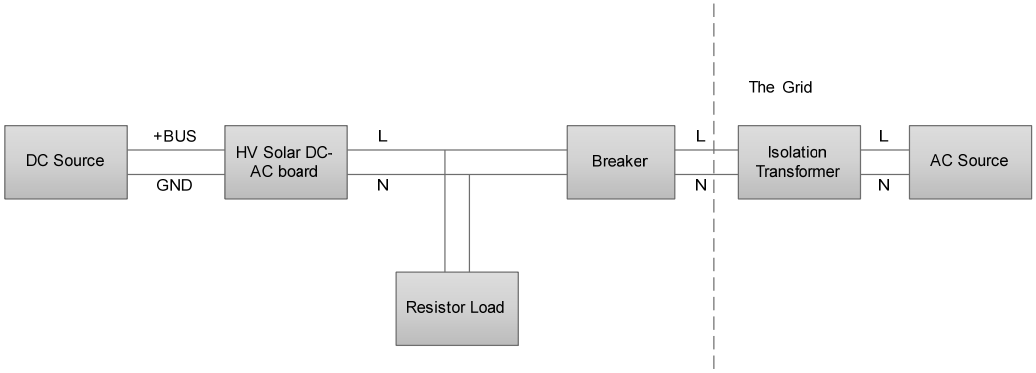


Fig 3.7 The connection of the test

For the safety, TI strongly sugget the user use a breaker between the grid and the inverter output.

Note: All the test should be done in lab, and the user must use the AC source to emulate the grid, we can not ensure the security when the user use this board to connect to the grid.

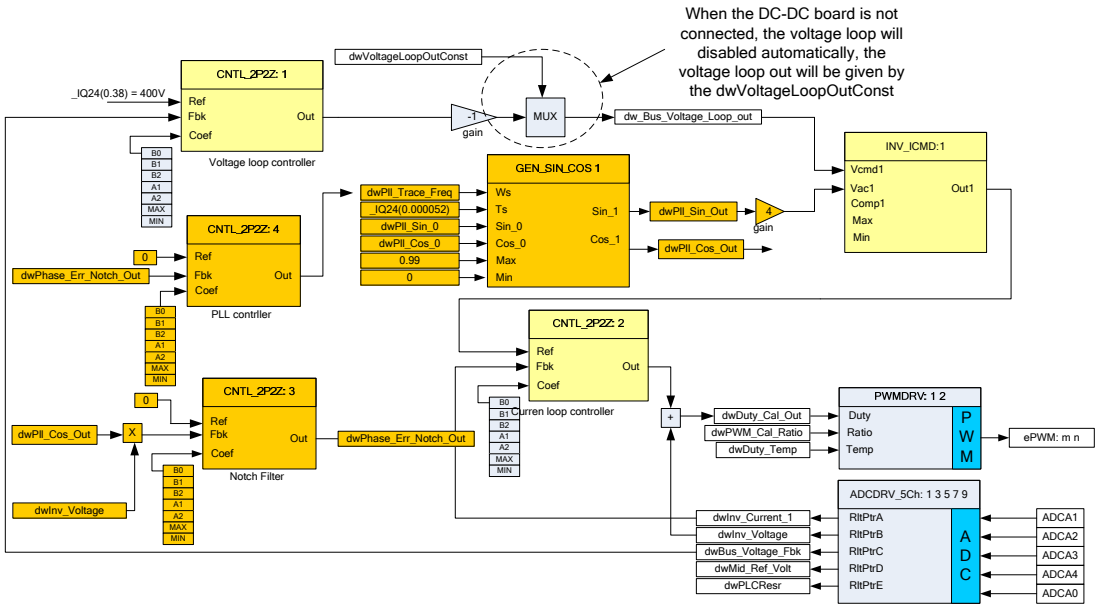


Fig 3.8 Colse Loop With PLL Build

The close loop with PLL build can be available when set the **INCR_BUILD = 3** in the **SolarHv_DCAC-Settings.h** file.

Please note that when the DC-DC board is not connected, the voltage loop will be disabled automatically. The **dw_Bus_Voltage_Loop_Out** will be given by the dwVoltageLoopOutConst directly. User can modify the **dwVoltageLoopOutConst** in the real time to get the different output current value.

1 Test Result

1.1 Specification

The system main spec is below:
Power Rating: 600W.
Norminal Grid Voltage: 120V/60Hz(RMS), 220V/50Hz.
Output Power Factor: 1.
THDi: <5%.
Panel Input Voltage Range: 400V.
Grid Tie.
Anti-islanding Protection.
Test Condition:
AC source connected, with 120VAC/60Hz;
DC bus voltage: 400V
Power range: 100-600W output;
Grid-Tie.
Room Temperature

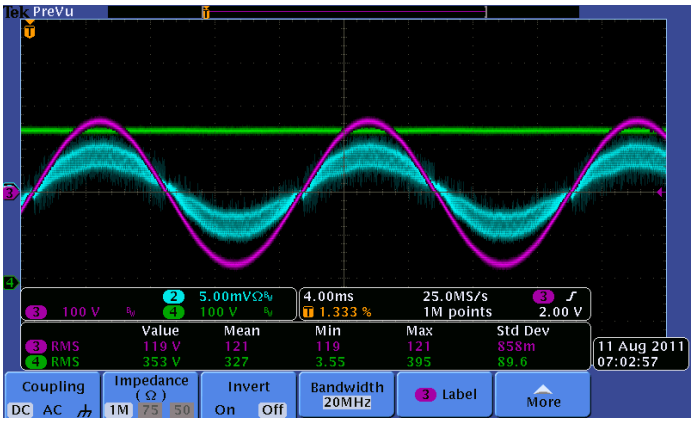
4.2 DC-AC Board Current Loop Grid-Tie Test Result

1. Light Load Current and Grid Voltage waveform

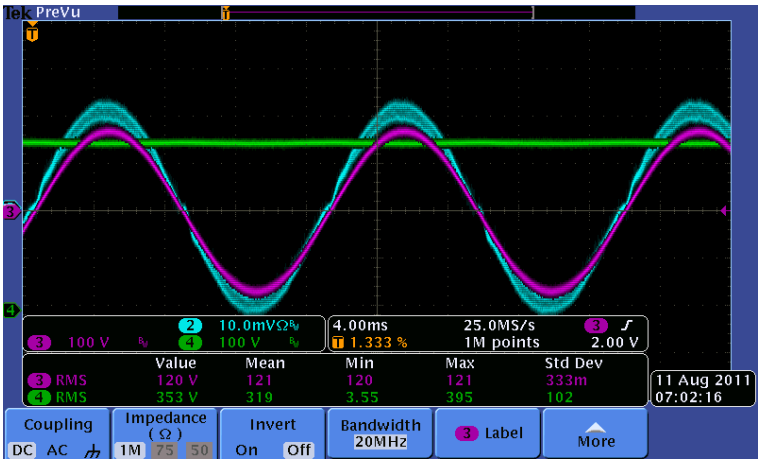
CH2: Output Current(Blue)

CH3: Grid Voltage(Red)

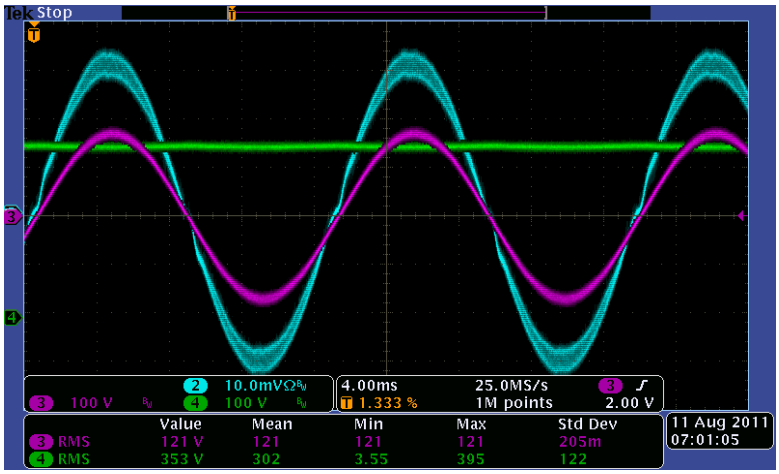
CH4: Bus voltage



2. Middle Load Current and Grid Voltage waveform



3. Full load current and grid voltage waveform



1.2 Output Power Factor and THDi

Inv V_out	Inv P_out	Output PF	THDi
119.5	100.3	0.983	12.60%
119.8	151.6	0.992	8.70%
119.2	198.4	0.995	6.80%
119.5	248.1	0.996	5.80%
119.8	297.7	0.997	5%
120.1	344.1	0.997	4.30%
119.6	391.7	0.997	3.90%
119.9	439.2	0.997	3.60%
120	464.2	0.997	3.40%

Table 4.1 The PF and THDi

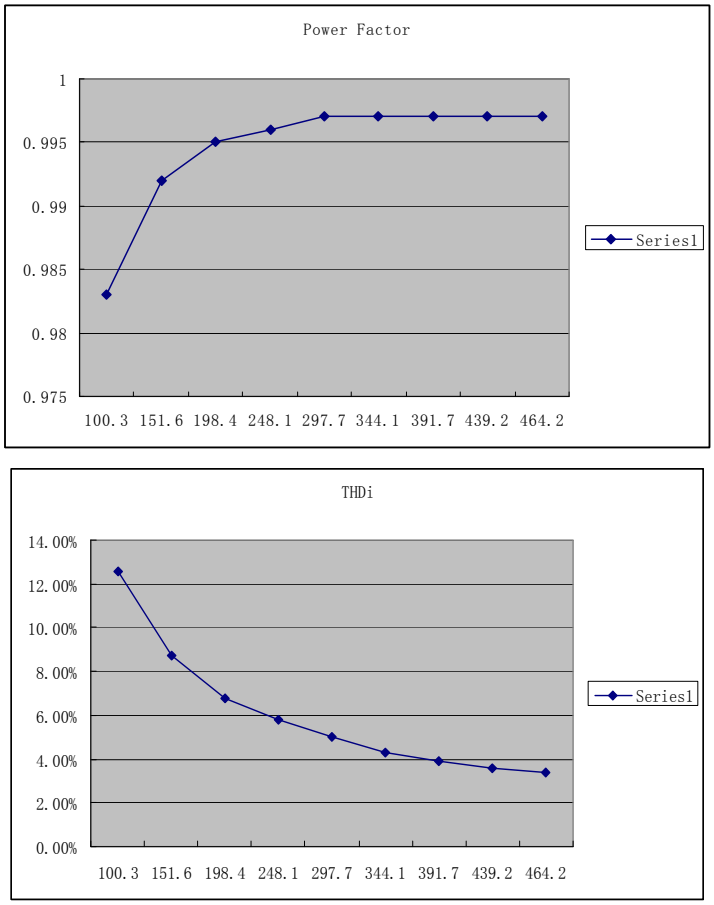
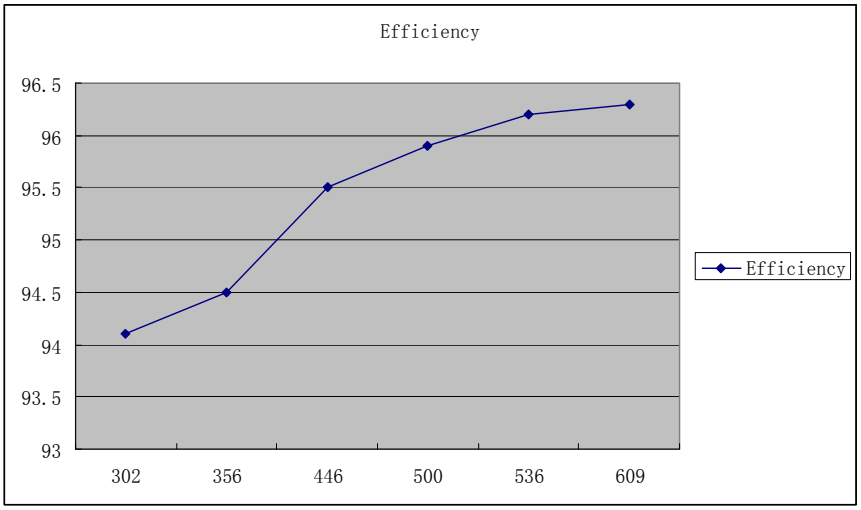


Fig 4.2 PF and THDi

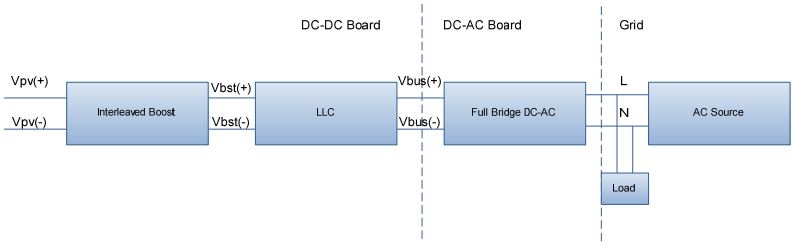
1.3 Efficiency

Item	DC_in(V)	AC_out(V)	Output(W1)	Input(W)	Efficiency(%)
1	400	120	609	632	96.3
2	400	120	536	557	96.2
3	400	120	500	521	95.9
4	400	120	446	467	95.5
5	400	120	356	376	94.5
6	400	120	302	321	94.1



The efficiency

4.4 The HV Solar System Test



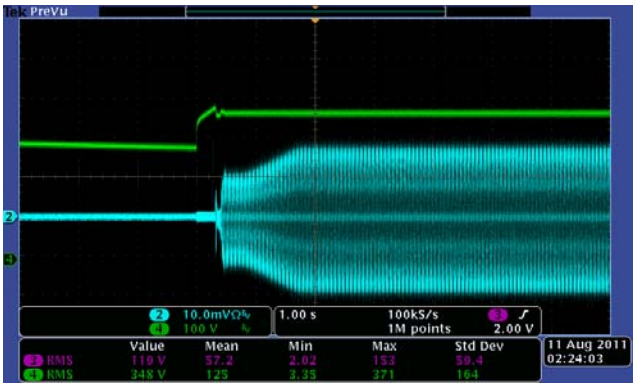
The System Structure and the Connection

1. 120VAC/60Hz, turning on

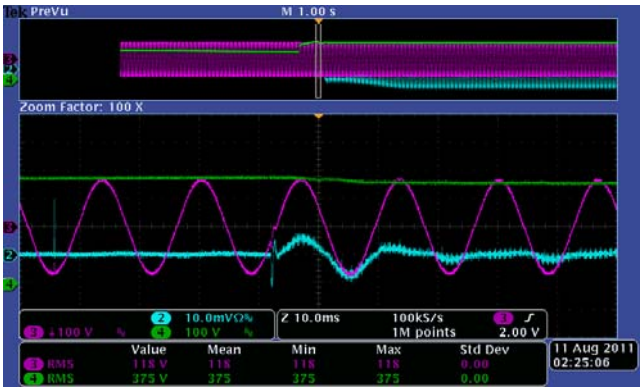
CH2: Output Current(Blue)

CH3: Grid Voltage(Red)

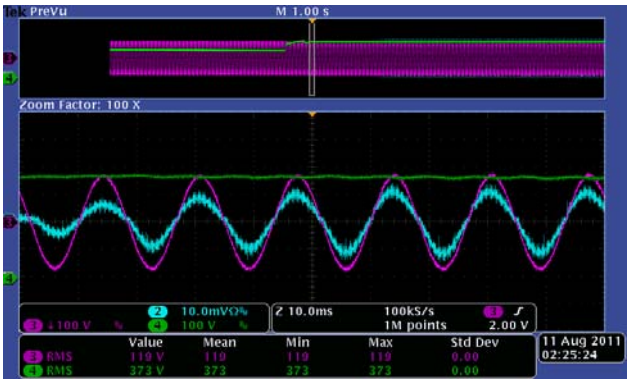
CH4: Bus voltage



The turning on overview



The DC-AC turn on the PWM



2. 120VAC/60Hz, 500W

CH2: Output Current(Blue)

CH3: Grid Voltage(Red)

CH4: Bus voltage