DSP Designer Getting Started With Microchip

1. Introduction: Digital Signal Processor, DSP, Design requires integrating both software and hardware. This combination dramatically increases complexity; consequently increasing time to market. DSP Designer attacks the problem by providing an advanced integrated development system that extends the analytical procedures of ICAP/4, the Intusoft SPICE simulation package, to include DSP hardware modeling and real time control. Included in this package are the following:

1. Solar2TiM Dual Buck/Boost SMPS Board
2. Microchip 16-bit x 28 pin Starter Board
3. Real Time AC, DC and TRAN
4. Tektronix Oscilloscope interface
5. Simulation Driven Code Generation
6. Control System Examples

The basic idea behind DSP Designer is to translate the world of simulation from analog to digital and provide real time hardware equivalents to the SPICE, AC, DC and TRAN analysis results. In the first part, the analog control system is changed to digital using the Z -Transform. A new model for the Z-Transform has been added to IsSpice4. Then a low speed real time communication; RTCOM, was added to communicate with various DSP's. To make the environment work similarly for different DSP's, a hardware board is included along with software that implements the AC, DC and TRAN interface using the ICAP/4 IntuScope waveform viewer. DSP software for Microchip is included, along with a modified 16 bit x 28 pin started evaluation kit.

DC analysis uses a special linkage to the schematic to display average voltages using a virtual instrument format. AC and TRAN use scripts in the IntuScope Calculator to display step load transient results and control system Bode plots. An interface to Tektronix Oscilloscopes lets you compare results for observable nodes. But most nodes fall into the unobservable category because they are mathematical representations inside the DSP control system. That's where the real power comes from, letting you check out everything!

Unique to lower voltage SMPS regulators is the nearly universal adoption of synchronous switching. The ON resistance for low voltage MOSFETS results in the ON state forward drop being much less than a diode drop. That improves overall efficiency with practically no cost impact. Synchronous switching makes power transfer bi-directional. Bi-directional power transfer expands the application universe to encompass both buck and boost applications with one circuit. The negative current in bi-directional power transfer presents special challenges for conventional current sense circuitry. For the SPICE simulator, synchronous switching makes the average switch model revert to its oldest and simplest form because the inductor current is continuous.

But, is DSP Designer just for power supplies? No! Motor control and other applications can easily fit within the framework. Intusoft will add other target application examples based on user requests. The ICAP/4 model library already includes a number of motor and "mechatronic" models.

2. Hardware Setup: The Intusoft Solar2TiM board is a dual synchronous buck/boost regulator that plugs directly into Microchip evaluation board. Many other DSP evaluation boards can be wired into the Solar2TiM board.

2.1 Microchip 16-bit x 28-pin Starter: There are 3 pieces of hardware needed to run your code. First is the evaluation board that contains the DSP. We supply the 16-bit 28-pin starter, DM300027 with the DSPIC33FJ16GS502-I/SP. Then you need the MPLAB ICD 3, the In Circuit Debugger interface DV164005, selling for $189.99. Figure 1 shows the Microchip dsPIC33 setup.

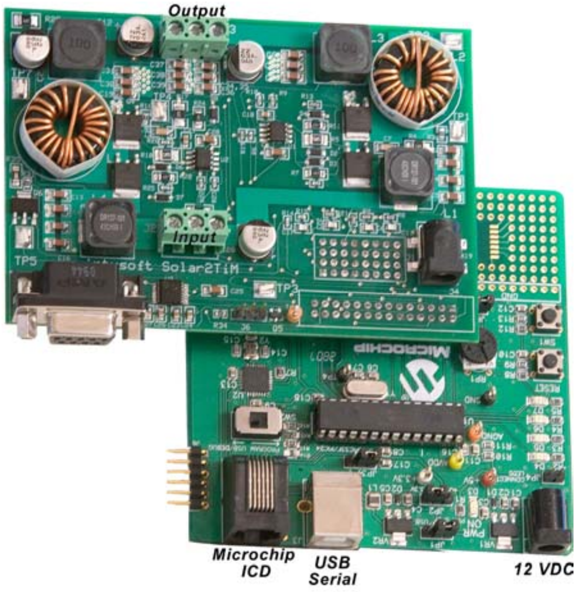


Figure 1, 16-bit x 28-pin starter setup shown with a dsPIC33 DSP connected to a Solar2TiM board.

This board plugs directly into the Solar2TiM Dual Buck-Boost Demo Board as shown in Figure 1. Start-up power is supplied from the Microchip board from the barrel connector a the lower right shown in Figure 1, and must exceed the under voltage lockout set point of the UCC27201 MOSFET driver(7.4V). To meet the FDMS5672, MOSFET switch, and the UCC27201 requirements; the voltage supplied to the Solar2TiM board via the barrel connector must be between 10V and 14V. Input voltage is supplied through the input connector at the bottom of the Solar2TiM board. The regulator outputs are connected to the output connector at the top right. Regulator 1 is the one on the right side and regulator 2 is on the left side. A clip lead connected to Q5 at the bottom of the Solar2TiM board is used to connect a 12 ohm resistor to one of the outputs. The signal on TP5 at the lower left drives the Q5 switch in order to provide a load-switching transient. TP3 on the left center of the board provides marker signals that allow you to measure control system execution times using an oscilloscope. Test points TP4 and TP1 on the left-center and right side of the board let you view the PWM switching signal. It's buffered by a 150 ohm resistor in order to reduce transient current which could give incorrect noise measurements if the signal was monitored directly.

2.2 Rework

(skip if shipped with DSP Designer)

The 16-bit x 28-pin starter board supplied with DSP Designer has been reworked in accordance with the following instructions for the dsPIC33 configuration. The spic33FJ16GS504 part has been installed. If you purchased the 16-bit x 28-pin starter board from another vendor, then the following rework is necessary.

The following instructions are also found in Microchip DS51656. "16-Bit 28-Pin Starter Development Board User's Guide"

Microchip Board Modification

• Remove resistors R14 and R15 (to control Buck 2 or Boost converter)

De-solder using solder wick and pop the resistors off with tweezers.

• Add diode, 1n4148 or equivalent From 9V power(anode) to J2-28 (cathode).

Some boards provided with a jumper wire on the back that can be unsoldered in order to add the diode, if not then place the diode in series with 9V power.

Jumper Configuration for dsPIC30 DSP (Previous DSP), 30mip

• JP1 in pin 1-2 position (supply)
• JP2 in pin 1-2 position (+5V)
• JP3 in 2-3 position (dsPIC33F/PIC24)
• JP4 open
• SW2 in USB/DEBUG mode

Jumper Configuration for dsPIC33 DSP (New DSP, 40mip)

• JP1 in pin 1-2 position (supply)
• JP2 in pin 2-3 position (+3.3V)
• JP3 in 1-2 position (dsPIC33F/PIC24)
• JP4 open
• SW2 in USB/DEBUG mode

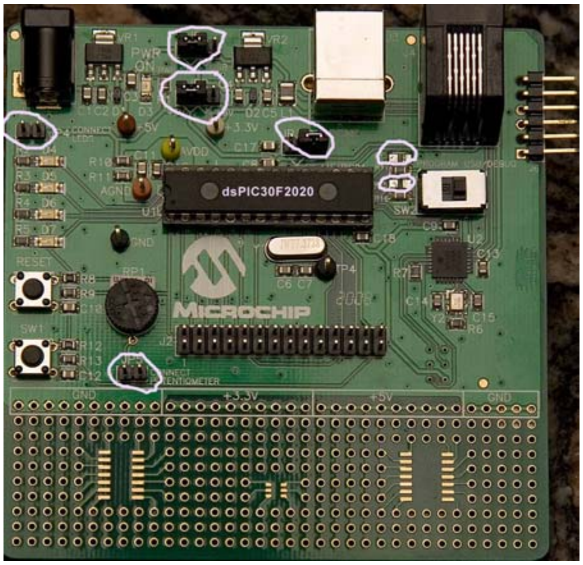


Figure 2, 16 bit, 28 pin starter board top view; jumpers shown for the dspic30 DSP.

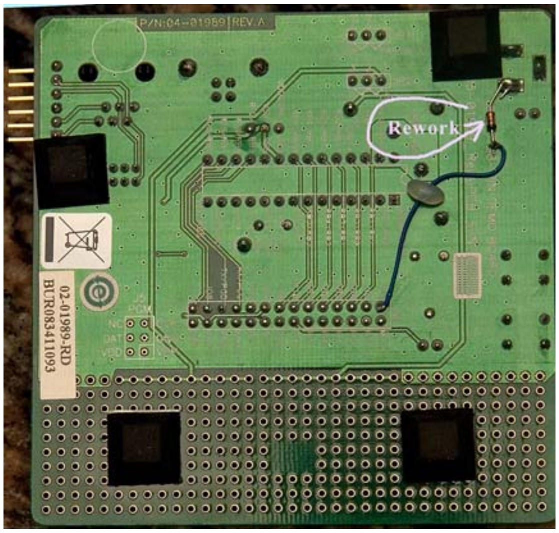


Figure 3, 16 bit, 28 pin starter bottom view shows rework.

2.3 Other DSP Vendors: There are a number of other DSP vendors making products with similar speed and cost. You must make an adapter for interconnecting with these vendors' evaluation boards.

The easiest way to do this is to wire either a 28 pin or 32 pin connecter to the available user breadboard area that plugs into the Solar2TiM board. Then connect the DSP to the pins required and plug it into the Solar2TiM board

3. Software Setup

DSP Designer comes as an add-in for Intusoft's ICAP/4 products (Demo and Consumer versions excluded). The software is already part of Build 3684 and subsequent builds. It is activated using the license manager in the icap4dsp.lic file.

3.1 Installing DSP Designer: After installing the most recent ICAP/4 package (Build 3684 or greater); install the collateral material using the DSP Designer CD. This is shipped with a license file that recognizes your copy of the software, and does not work on other computers. Collateral material includes drawings, libraries, models and DSP source code.

3.2 Integrated Development Environment (IDE): Microchip provides a free IDE for their DSP's that you must install. Microchip updates their web sites too frequently to provide reliable links so that Google search is your best bet.

Microchip: MPLAB version 8.4.xxx or greater, please use Google to search for "MPLAB 8.4 download". Scroll to the bottom and download the appropriate file.

3.3 C Compilers: Microchip provides a free C-Compiler that is adequate for their DSP's. The Microchip MPLAB C30 "LITE" version is included with your installation DSP Designer installation CD. The LITE version lacks some optimization capability, but the software we provide already uses assembly language to speed critical portions so you don't really need to pay for the C Compiler upgrade.

3.4 File Structure: Files that contain your source code are as follows, assuming everything went into default drives.

Intusoft: (Default Install)
C:\Spice8\
  Circuits
    Solar
      Solar2TiM.dwg, …

Microchip: (C:\DSP install)
C:\DSP\
  MCHP 12x28starter
    Solar2
      Drawing, Source Code,…

4. Using DSP Designer Software:

DSP Designer allows you to simulate your control loop with it embedded into a “real-life” simulated circuit and system. Source and load interactions, along with non-linear properties can be included. Refer to the “Lessons” documentation to learn how to run the software/hardware combination. Then, you can automatically generate DSP C and assembly code and validate your models using DSP designers RTCOM interface to compare real time data with simulated predictions. Iterating this process until the hardware performance matches predictions makes your design robust and allows you to select parameters that center its performance to maximize production yield.

4.1 ICAP/4 Average Models:

DSP controllers generally operate at a fixed iteration rate, governed by an Interrupt Service Routine, ISR, synchronized to a clock. Conventional analog implementations make a solution for each switching pulse; however, DSP based solutions can use any frequency less than the switching pulse rate. The controller is described in frequency domain up to ½ the ISR frequency using Z-Transform difference equations. In this realm, the detailed switching waveforms are unimportant. Simulators spend an inordinate amount of time making the switching signal simulation accurate. A number of years ago, “average” models were introduced that produce accurate results below the Nyquist frequency (1/2*ISR frequency), thereby removing the need for cycle-by-cycle switch modeling. These models are made for 2 domains, continuous conduction and discontinuous conduction modes. Crossover circuitry allows for a single, albeit, more complex model. Modern switch mode modulators replace the freewheeling diode with a synchronous switch, allowing the simplest of all models, the continuous current mode, CCM, model to be used. Figures 5 and 6 show the progression from a freewheeling diode to a synchronous switch, and the resulting 2-equation model.

$$
\begin{aligned}
Vout &= D*Vin \\
Iin &= D*Iout
\end{aligned}
$$

This simple model results in SPICE simulations that are hundreds of times faster than the cumbersome switching method. Moreover, the average model can be used in an AC analysis, resulting in automatic linearization of the system about its operating point. With this approach, the small signal analysis is automatically performed, yielding information to make a Bode plot. The Bode plot is then used to characterize system stability using gain and phase margins.

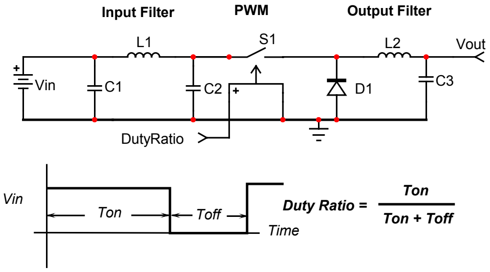
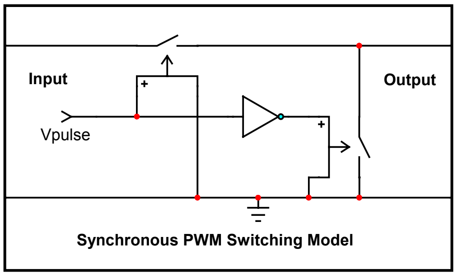


Figure 5, Classical Buck Regulator



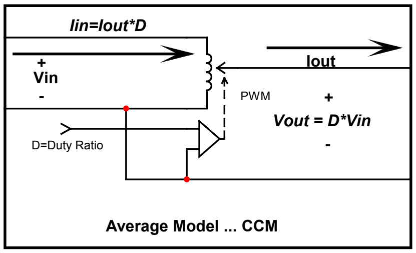


Figure 6, Average Model Derived.

In ICAP/4 this model is called PWM and symbolized as a variable ratio transformer. It is in fact the same model that is used for Gear Ratios and Transformers. The input duty ratio must be limited between 0 and 1. Typically the limits are .03 to .98 for IC operational reasons. The average model is slightly different than a pure Z-Transform model because it models the average value over a switching cycle rather than the instantaneous value. If we assume the Z-Transforms instantaneous value is sampled at the average point in the waveform, then the 2 models are exact. This actually isn’t a bad assumption because the average point tends to be away from the switching transients and is a good point to take the digital sample.

4.2 ZDELAY Model:

DSP controllers are characterized using Z-Transform analysis. The basic building block for these difference equations is a unit delay,

$$z^{-1}=e^{-sT}$$

In SPICE this is exactly the same as a transmission line. But, the SPICE transmission line runs in continuous time where the Z-Transform is valid only at sampling instants. Moreover, the transmission line's DC gain is unity and that may not result in a stable operating point calculation. A new IsSpice4 model, ZDELAY, was created using the code model software development kit (CMSDK). The ZDELAY model has the same AC gain as the unit delay; however, its DC operating point is usually given by its initial condition. In a transient analysis, the input is sampled every "tsample" seconds beginning after "tdelay" seconds, and then it is propagated to the output after "tprop" seconds. Optional limiting is available at the output. Values for the properties "tsample" and "tprop" are required. The user finds the initial conditions by running a transient solution to steady state. In the absence of non-linearity, the DC solution is guaranteed with no iterations; this important property is used in code generation. The ZDELAY code model cannot output frequencies greater than the Nyquist frequencies, thus eliminating spurious high frequency oscillation.

4.3 Application Examples:

Drawings vcur.dwg and mpid2.dwg illustrate the use of ICAP/4 in design and analysis of DSP controllers for virtual current control, and a modified proportional, integral, differential, MPID, controller. They use the Solar2TiM.dwg PCB running both of these algorithms simultaneously, one for each of the 2 channels. Intusoft's layered configurable schematic is used extensively to differentiate configurations for AC, TRAN and code generation. Each configuration is made up from a set of layers, some of which are shared with other configurations. A configuration is then paired with a simulation setup to provide the desired simulation. Pairings for vcur.dwg are in Table 1 and for mpid2.dwg are in Table 2.

Table 1: Important vcur.dwg configuration pairings

Configuration	Setup	Purpose
vcurac	ac	Bode plot
start	tran 10m	Startup transient
MPIDv1	.DSP!	MPID loop code generation
Plant	.DSP Plant	Plant loop code generation

To make the Bode plot, adjust the configuration, setup pairing, run the simulation using the Actions menu "Simulate". Then open IntuScope with a new plot and press 'b'. For transient simulation, adjust the pairing, run the simulation and use the probe tool to transfer waveforms into IntuScope. For code generation, select the pairing and select "Export DSP…"" from the file menu.

Table 2: Important mpid2.dwg configuration pairings

Configuration	Setup	Purpose
mainAC	ac	Bode plot
mainTran	tran 10m	Steadystate transient
start	tran 10m	Startup transient
MPID2	.DSP2	MPID2 loop code generation

The virtual current controller requires a nonlinear computation: the product of input voltage and duty ratio. The loop was broken into 2 linear parts for code generation, and the nonlinear product was linearized by using the previously calculated duty ratio. The calculations connecting the loops were done is assembly language using assembly code Macros in isr_asms.s.

4.4 DSP Code Generation:

As mentioned previously, a linear set of DSP equations can be solved without iteration. DSP Designer takes advantage of that property to have IsSpice4 build a matrix. When the equations are ordered properly, the matrix coefficients are constant. Proper ordering requires that the first rows are solutions for unwanted states. Next are the states that require a solution, followed by trivial states, which are inputs and ZDELAY outputs. This matrix ordering is enforced by several rules. First, unwanted solutions use numbered nodes. Then inputs are connected to grounded V-Source elements. The matrix is formed by IsSpice4 when a .DSP analysis and only a .DSP analysis is requested. The .DSP analysis directive can have arguments, which are nodes to be solved in reverse order. This is useful if limiting is required early on in the solution. IsSpice4 uses simple Gauss elimination to solve the matrix, so that changing the ordering of equations can impact the min-max coefficient ratio. SpiceNet reads this matrix and generates the .c, .h and assembly code that solves the matrix using backward substitution. Assembly coding is needed because the current crop of C compilers do not take advantage of the powerful multiply-accumulate (MAC) instruction available to DSP's. The schematic configuration used for code generation has a number of rules to assist the code generator. These rules include:

File and Struct Name
> *The various files use the schematic configuration name*
> *States instances are named using the following syntax*
> *<config name>_Statei.<node name>*

State Variables are identified as SPICE node names

Parts: Use only V-source, B-element, ZDELAY and X-Subcircuit components. Subcircuits have only B-elements

Nodes

Numbered, these states will not be calculated
zi prefix for ZDELAY inputs
zo prefix for ZDELAY outputs
embedded pwmx, identifies a pwm output for channel x =(1,2,3,4)
indx postfix, identifies an ADC input x=(0,1,2,…)

Parameters
PWMPERIOD, integer size of PWM for unity duty ratio
radix, number of fractional bits in MAC computation
refx, reference for indx (refx-ADCx)
LIMIT_HI+node, Hi Limit
LIMIT_LO+node, Lo Limit
<config name>_Statei.<node>= Initial Condition

Parameters are algebraic calculations done in the order present in the Parameters block shown on the schematic. The .par files shows their evaluation and is used by the code generator to define state variable limits and the reference for the input error signal and the PWM scale factor.

Special cases: Connecting a ZDELAY output to another ZDELAY input needs to use a B-element with gain = 1 so that the prefix rules are obeyed. If an indx postfix is used, the result is (refx - <prefix>indx) and limiting is applied to that result. These rules are error checked prior to running the code generator. If errors exist, a dialog will explain problems with error saved in the .err file and the code generation will be aborted.

A Typical Example Use drawing MPID2.dwg and select the code generator pairing; MPID2+.DSP2. This will generate code for channel 2 to produce a 5 volt output using a modified PID control law.

When completed successfully, there will be 3 files added to your project folder. The base name is derived from the configuration name so for this case the files are called MPID2.c, MPID2.h and and MPID2s.s . To use this code you must do the following:

Add include “MPID2.h” to main.c
Add init_MPID2() in main.c initialization
Add re_init_MPID2(vin2,vout2) in main.c, error recovery section.
Add call MPID2 in the isr_ams.s file
Add MPID2.c and MPID2s.s to your project.
isr_asms.s connects the automatically generated assembly code to the interrupt service routine, study the example carefully if you make different controllers

These additions have already been incorporated for this example, but it’s a process you must follow for different code generator files. Remember, the code generator will overwrite these files so that you may want to rename them. Using the wrong code generator could damage your hardware.

Source Code for DSPCOM: The following files are used for DSPCOM implementation:

tran.c	used for TRAN script
tran.h	
uart.c	used for serial communication
uart.h	
AC.h	used for AC script
AC.h	
singen.c	used for sine-cosine generation
singena.asm	
configure.h	sets up test configurations
configure.inc	
hardware.h	defines for hardware specific functions

You can follow the comments in the code to see how it works.

5. Using DSP Designer Hardware:

The minimum hardware requirement needed for the exercises presented here consists of:

1. Solar2TiM board ($200), included
2. 16-bit x 28-pin Starter eval, included
3. MPLAB ICD 3, you must purchase this
4. Tektronix Digital Oscilloscope
5. Intel powered PC (Windows XP preferred)
6. Cabling
   a. Microchip
      i. 2 USB A,B, supplied with MCHP products

   b. Tektronix
      i. USB or RS232 depending on scope model
7. Bench power supplies
   a. 60V x 6 amp
   b. 12v 200ma
8. Loads Resistors
   a. 12 ohm 5 watt, provided

You can use a lower voltage, lower current bench power supply than the 60 volt x 3 amp one listed above. The solar2TiM board and controller algorithms work at 24 volt input.

5.1 Real Time Communication(RTCOM):

RTCOM is at the heart of DSP Designer’s hardware interface. It has a small software footprint that enables software on either side of the interface to cooperatively perform measurements while the DSP is operating in real time. The hardware interface is through a UART connected to the PC via a USB to RS232 Bridge, operating at 9600 Baud. The choice of an RS232 interface stems from the need to keep it simple and slow. Other interfaces are very complex and offer higher speed with higher complexity. The complexity is often hidden the interface hardware, but it's still there, making the likelihood of interface errors much greater. RTCOM is established using the uart.c and uart.h modules. They establish a state-machine interface to send and receive data. The state machine is only allowed to write into 8 16-bit integer locations. That limitation prevents data from being transmitted to the DSP that could damage the hardware. The transmitted data is interpreted as commands and used for variable initialization. Any location can be "probed" via the interface. When a location outside of the shared region is read, it is actually averaged over a number of sample times (up to 1024 selected in the software) before it is returned.

5.2 Hardware AC, DC and TRAN: These are the basic SPICE simulation capabilities. By performing these operations in Hardware, you can directly compare simulation results with hardware performance. Software is written in the C-language for low speed operations. That allows the compiler to manage and initialize memory.

If you make the main program in assembly language, then you must manually manage the C program memory. That's quite cumbersome and error prone.

DC Analysis: The simplest operation is DC. For this mode the steady state value of a vector is returned. The vector name is associated with a DSP address using the map file produced by the compiler. Note that the map file doesn't extend into struct members, so that you must add these associations to C code by adding dummy variables. To access that dialog, press "Browse" in the "Add Waveform" dialog and select "DSP Communication Filter". The first time you do this, you need to locate your projects ".map" file. Once that's done, the ".map" file auto loads until you want to do another project. You can then add vectors and scale factors.

After completing this step, and if the DSP is running, you can press "view all" in IntuScope to display the average values in the output window. There is a special interface to SpiceNet for displaying DSP OP values. The SpiceNet voltage testpoint is used to activate this feature. If the IntuScope setup dialog contains the vectors node name, then it will be retrieved and displayed as a virtual instrument, complete with the units provided in the test point parameters. This effectively replaces multiple DVMs on your workbench. The DSP OP setup dialog lets you select various parameters, including the refresh rate, display fonts and virtual instrument skin. Remember, the DSP must be connected and running for the virtual instrument interface to work.

Transient Analysis: This is based on a simple concept. A pulsed signal is connected to a MOSFET on the Solar2TiM PCB. You connect it through a resistor to an output in order to get a transient response. You can sync to the pulse using TP5 and watch its response using your oscilloscope. The pulse is in sync with the PWM so you won't be able to average out the switching ripple.

To make measurements using IntuScope you use the TRAN script (Calculator\DSP\TRAN). You will be prompted for the vector that you want to view. What happens after that is that IntuScope will get the DSP data, one word for each pulse. It does this by incrementing the time index to find the correct data slot. So it takes N times the pulse period to assemble a single data record, where N is the number of samples. This process is repeated j times and averaged. N and j can be altered in the script. This takes awhile, but it makes data available that is internal to the DSP, which can't be viewed externally.

On the DSP side, "tran.h" handles the problem as inline code.

AC Analysis: This is the most complex measurement with the biggest payoff. It is accessed by using the AC script, <Calculator\DSP\AC>. This script sets up the DSP signal generator beginning at 200 Hz and stepping at "1.2 x" intervals up to ".35/T" Hz. If you are running from an appropriate project, the script will look up "T" in the ".par" file; otherwise you will be prompted to input the sample period, T. Data is collected for 50 ms following a 50ms period to achieve steady state. The frequencies are rounded in order for there to be an integer number of cycles each 50ms. The "configure.h" and "configure.inc" files contain the signal generator injection levels. These levels are injected into the loop in the appropriate assembly code. DSP Designer inserts the code during code generation and you activate it using the header files. At each ISR sample, the real and imaginary values are summed according to the following:

```
resum += Vsig * sinGenVars.cosprev;
imsum  += Vsig * sinGenVars.sinprev;
```

After the 50ms data acquisition period, these values are loaded into share_name02, 03, 04, 05. Shar_name01 is set to 1, signaling the IntuScope script that data is ready. When IntuScope has all of the data from the sine sweep, it calculates the transfer function and makes a Bode plot. The calculations are based on the single injection GFT model described by Middlebrook, which can be found in the SpiceNet "script syntax" help menu. Only the signal injection output needs to be measured because the input is simply the output less the injection level. This technique assumes that frequencies greater than the Nyquist frequency will be rejected because they are either incoherent or filtered. The fundamental spectrum, Fs, sweeps from 200 Hz to .35/T. But an alias spectrum sweeps from 1/T – Fs at the same time. If these 2 frequencies are coherent, then data anomalies may occur.

When *Fs*3 = (1/T – Fs)*, the strongest potential anomalous data will be viewed. That occurs at

*Fs=1/(4*T) = Fn/2*

Where Fn is the Nyquist or folding frequency. Spurious data will actually occur for all harmonics, however the largest artifact is at Fn/2. The digital filters in the control system will shape these effects as a function of frequency.

5.3 The Tektronix Oscilloscope Interface:

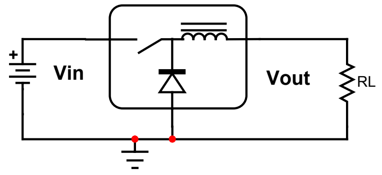
Tektronix Oscilloscopes communicate with Windows PC's using their TekVISA interface software. You must download it from the Tektronix web site. The first version that works for Windows platforms through Windows 7 is V3.3.4. If you are using a USB-RS232 adapter, then the Prolific driver version 3.3.10.140 or later is required. Windows download doesn't find it so you need to download it from the Prolific web site. Then when running IntuScope, connect to the Tektronix scope using the Add Waveform dialog, <Browse-Oscilloscope Input Filter>. Any change to the input port requires a reboot for the scope to be recognized. You can independently verify operation by downloading and running the Tektronix "Open Choice Desktop". The interface is slow, so you probably want to choose one waveform at a time. Waveform colors match the scope colors. The trigger time is set to 0, so you should trim the observed waveform using the "[]" button followed by the "←" button. Then adjust the trigger time to agree with your simulation time by running a script, "time=time+???" , where "???" is the desired offset. Then similarly trim the transient simulation or the "DSP\TRAN" acquired waveform to match scales. Then you can copy the waveforms onto the same plot for comparison. Remember, the Tek-scope waveform has the DC offset removed. If you are using the Solar2TiM board with a TPS series scope, you can connect the ground to the channel 2 output in order to DC couple the probe. Be aware, this ONLY works for scopes with floating probe grounds. Otherwise use AC coupling.

6. Solar2TiM Board:

The board has 2 synchronous buck/boost regulators.

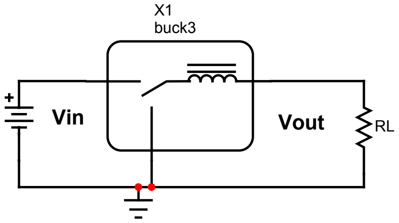
6.1 Theory:

The buck regulator is the simplest of all switch mode regulator topologies. It switches between an input voltage and ground at a periodic rate. Originally the switch to ground used a free-wheeling diode.



Buck Regulator with Free-wheeling Diode

At high load current, the diode conducts all of the time …Continuous Mode Conduction, CCM. At low currents, the inductor current drops to zero and the regulator operates in discontinuous current mode, DCM. Equations for these 2 modes are different and require a mode switch capability in the simulator. Synchronized switching occurs when the free-wheeling diode is replaced by a switch that is synchronized with the series switch.

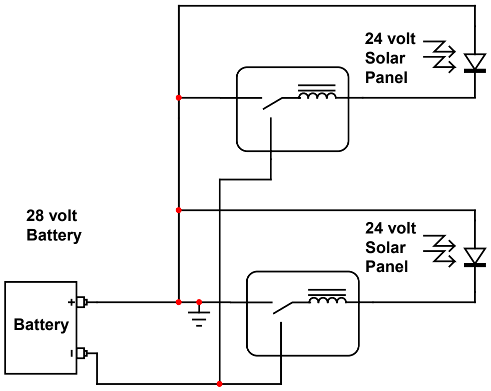


Synchronous Buck Regulator

Then a synchronous buck regulator always operates in CCM. For SPICE Modeling. That means that the average voltage output is proportional to the product of Duty Ratio and input voltage and the current reflected from the switched inductor to the input by the Duty Ratio. The most interesting property is that power conversion becomes bi-directional. The buck regulator becomes a boost regulator when converting power from Vout to Vin.

The board was initially designed for a solar PV application. However, it is capable of operating in many other applications, a few of which are illustrated here:

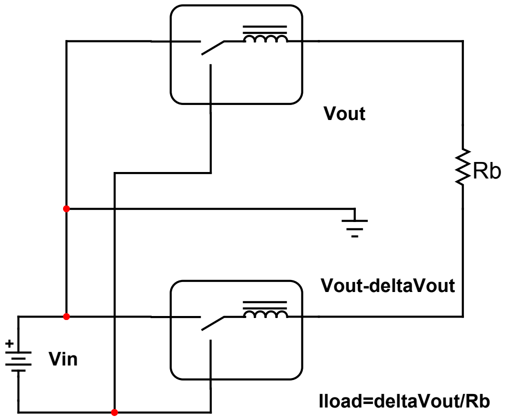
6.2 Solar Array Battery Charger: For this application, the Battery is connected to the Buck input and the soar panel is connected between the High side of the battery and the Buck output.



Dual BuckBoost Connected as an "Auto Transformer"

The Buck output would be typically 10 VDC when the battery voltage is 48 volts and the solar panel is 38 volts. This is an SMPS version of an auto transformer in which only 26% of the array power passes through the regulator, allowing the charger to exceed 97% efficiency. Each regulator can control one 210 watt panel in parallel, while performing peak power tracking for two panels at a time. Notice that by grounding the positive terminal that lightening protection for the system is possible.

6.3 Active Load: When operating as a dual Buck-Boost regulator, one side can be a source (Buck) and the other an active load (Boost).



Dual BuckBoost Connected as an Active Load

The advantage of this configuration is that only losses need be supplied from an external power supply and no load bank is needed. Connecting the outputs through 12 in. of AWG 20 wire makes a .01 ohm resistor that can be used to measure load current. For loads in excess of 10 amps, the temperature coefficient of the copper resistor must be taken into account. This configuration reduces the cost of the bench power supplies and load resistors.

Subtracting output power from input power is error prone and is the major source of inaccuracy in computing efficiency when active loading is not used. For example, in a regulator whose efficiency is 95%, the error in computing the loss with 1% instrumentation is 1% of the input power, which gives a 20% uncertainty in loss. For active loading, the power being controlled is the product of the 2 outputs and the load current, while the power lost is the input power plus the I-squared R loss in the ballast resistor. The accuracy of the loss measurement with the same instrumentation is 1%, a 20-fold increase.

6.4 Specification:

Input Voltage	0-60 volts
Input Current	0-2 amps
Output Voltage	0-60 volts
Output Current	0-6 amps w/o cooling
Switching frequency	100kHz to 200kHz 133kHz typical
Auxiliary Voltage	12 +-2 volts, 100ma

RCD Snubbers: A number of snubber configurations were tested to see if the MOSFET voltage could be reduced.

When the top switch turns on, the bottom switch will continue to conduct until its current becomes zero. At that time the voltage will switch to the input voltage, behaving as an L-C network with a step input. The inductance is the leakage inductance in the switch loop and the capacitance is the power MOSFET output capacitance.

The peak voltage is then twice the input voltage, requiring 120 volt MOSTFETs for the maximum, 60 Volt input. An RCD snubber could limit the peak voltage; however, its parasitic inductance must be much less than the network stray inductance, which is about 20nHy.

Diode plus capacitor lead inductance is on the order of 20nHy so that a significant reduction in overshoot, sufficient to use lower voltage MOSFET’s, is not possible.

6.5 Schematic: Please refer to the Solar2TIM.dwg. It can be opened using any ICAP/4 package including the demo version.

6.6 Bill of Materials: Please use a text editor (IsEd) to view the Solar2TiM.bom text file.

6.7 Layout: The Solar2TiM.pcb can be viewed using the free PCB download from expresspcb.com. Solar2TiM.net is an ICAP/4 generated PADS 2000 netlist that works with expresspcb to identify nodes that are connected to form nets.

6.8 Rework: Should it be necessary to perform rework to replace damaged parts, a solder wick, temperature controlled soldering iron, tweezers, liquid flux and a hot air station are needed,

SMT part removal is best accomplished using the hot air station. Replacing the power MOSFETs requires removal using the hot air station and pad cleaning with the solder wick.

Then new MOSFETs can be re-soldered using solder paste (DigiKey KE1507-ND), a plunger (10LL4-ND) and tip (KDS22TN25-ND) and the hot air station.

The solder paste must be refrigerated. It's poisonous so wrap carefully and segregate from food, kids and/or employees.

For other parts it's easiest to use a soldering iron with flux added to the cleaned pads. Clean the iron and load with a small amount of solder. Hold the part in place with tweezers and solder one lead. Continue to press down with the tweezers and solder the remaining leads. Clean up with alcohol or water. It's safe to immerse the board in water if it's completely dried before use.

The most common part failure is a blown power MOSFET resulting from excessive current. When drain current is large, the gate and drain voltage exceed device ratings and the device becomes a 3 terminal short. The most common cause of over current is when a software error turns both transistors on at the same time. That can destroy the resistor and diode connected to the gate. Avoid damage by turning off input power when stepping code through the debugger. The 12 volt power should be limited to about .2 amps to prevent cascade failures resulting from a blown power FET.

The DSP can also be damaged by static electricity. Spraying your floor and carpets with liquid Downey helps!

[1] **Thermal Runaway:** Power lost in the MOSFETs is proportional to I^2*Ron and temperature rise is proportional to the product of power and thermal resistance, J in Deg C per watt. The temperature coefficient of Ron is .006Deg C, the TC of silicon. Solving for Trise places the quantity 1- I^2*J*6m*Ron in the denominator. Setting this to zero, the condition for thermal runaway and solving for current gives: I = sqrt(1/(.006*.01*50)) = 18.25 when Ron=.010 and J=50 Deg C/watt. So additional cooling is not needed because the max current is limited to 10 amps.