

Analyzing Target System Energy Consumption in Code Composer Studio™ IDE

Software Development Systems

ABSTRACT

In the world of portable electronics, one of the key care-about for consumers is the time they can operate their devices on battery. A top priority therefore in the portable electronics industry is to develop “low-power” solutions. To enable our customers, the portable electronic device providers, to succeed in this endeavor, TI has developed tooling to measure power consumption on customer target systems and has integrated this capability into Code Composer Studio to allow users to understand and address power issues early in the product development lifecycle.

This application note introduces how to setup an environment for measuring power via Code Composer Studio; some helpful tips on designing target systems that are conducive to more accurate power measurement; how to instrument your application for power measurement and visualize energy/function utilization.

Contents

1	Introduction	3
2	Getting Started.....	4
2.1	Power Analysis Components	4
2.1.1	Code Composer Studio (CCS) v2.12	4
2.1.2	CCS Power Analyzer Plug-in and support software	4
2.1.3	Power Measurement and Capture Equipment	5
2.2	Modifying an Existing Target System for Power Measurement.....	5
2.3	Connecting Everything Up	6
3	Analyze Energy Consumption of Your Target System	6
3.1	Setting Up the Scope	6
3.2	Setting Up the Current Probe	7
3.3	Notation.....	7
3.4	Measuring Power Consumption	7
3.4.1	Flow of Operation.....	7
3.4.2	Using the Power Analysis	9
3.4.3	Selecting an Instrumentation Mode	10
3.5	Emulation Logic and Run Free	11
3.6	Saving Results	11
4	System Characteristics	11
4.1	Simulator Support	11
4.2	Scope Compatibility	11
4.3	System Power Considerations	11
Appendix A	Designing for Accurate Power Measurement.....	12

Figures

Figure 1.	Power Analysis Components	4
Figure 2.	Wire Loop and Capacitor Locations.....	5
Figure 3.	System Diagram.....	6
Figure 4.	Power analysis flow chart.....	9
Figure 5.	Board Modifications for Accurate Power Measurement.....	12

Tables

Table 1.	Mode Selection.....	10
-----------------	----------------------------	-----------

1 Introduction

This application report has been developed as a high-level introduction to guide users through the process of understanding how to measure and profile energy consumption on their own target systems. While the methodology is such that all analysis is done with respect to the user's target application, there is no restriction on where, physically, power is measured on your target board. That is to say, you can measure power with respect to the board supply, the digital signal processor (DSP) core supply, the input/output (I/O) supply, etc. The data that gets visualized though is always with respect to areas in your target application.

The two areas of concern for most are: understanding peak power (what is the maximum current drawn by the system at any particular time) and understanding average energy consumption (how much current is drawn over a period of time). The CCS Power Analyzer provides both these capabilities, though the accuracy of the information is dependant on where on the target you are measuring, the presence of decoupling capacitors, whether you use a wire loop and current probe for current measurement or a sense resistor, the length of the range of code you are measuring, etc.

This solution provides multiple modes of measurement. Each mode has its positives and negatives. Depending on where you are in your power understanding lifecycle, you must choose the appropriate mode. While some modes preserve the real-time constraints of your application, they may not provide the granularity of information you need (single-range real time mode). Whereas multifunction non-real time mode puts no restrictions on the number of functions instrumented and allows for higher sampling rates, it stops the target, and any real-time constraints your application has will be lost.

By incorporating measurement and profiling capability into the development environment, users can now understand energy consumption early in their application development lifecycle. You will know ahead of time whether the system meets its power requirements and thus can make adjustments for power in the application before the application development lifecycle ends.

2 Getting Started

2.1 Power Analysis Components

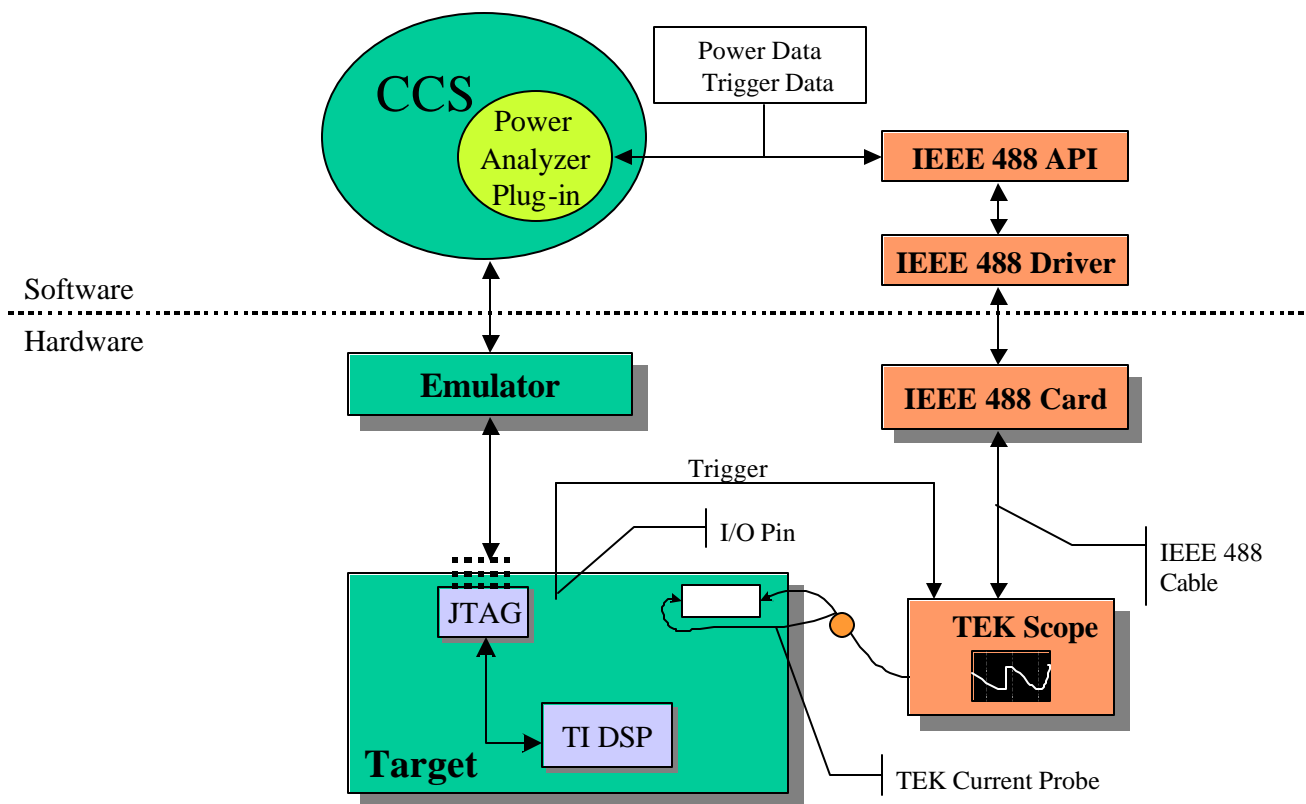


Figure 1. Power Analysis Components

2.1.1 Code Composer Studio (CCS) v2.12

At a minimum CCS 2.12.01 is required to use the Power Analyzer.

2.1.2 CCS Power Analyzer Plug-in and support software

The CCS Power Analyzer Plug-in is available via a CCS Update Advisor patch or in the TMS320C5510 DSP Starter Kit (DSK). It will automatically register into Code Composer Studio. The plug-in provides the capability to instrument your application in various modes and display the measured data. It also allows you to save the measured data to a Comma Separated Value (CSV) file that can be imported into standard spreadsheet and data analysis software packages.

Included with the plug-in are target specific support libraries that must be added to the project that is to be analyzed. This support software is responsible for coordinating code execution and power and peripheral data capture. You will need to use a different library, based on your choice of profiling method and memory model.

2.1.3 Power Measurement and Capture Equipment

To collect power data from your target, you will need the following equipment:

- Tektronix Oscilloscope with IEEE–488 Interface (with Probe cable)
- Tektronix Current Probe System
 - Tektronix A6312 10MHz AC/DC Current Probe
 - Tektronix AM5030 AC/DC Current Probe Amplifier
- IEEE–488 Card + Cable

2.2 Modifying an Existing Target System for Power Measurement

To perform power measurements, the target system may need to be modified, to allow a current probe (Tektronix A6312 in our setup) to be used to capture the current draw of the desired power supply (DSP core, I/O, System, etc.). This modification will consist of adding a loop of wire that the current probe is able to clip onto. The loop of wire should be placed between the output of the power supply and the load.

NOTE: The loop of wire should be kept as short as possible, yet be able to accommodate the current probe.

If there are any large capacitors on the output of the power supply, it is best to have them on the same side as the output of the power supply. These large capacitors filter the output of the power supply and slow down the response of the power supply for increases or decreases in the current demand of the load. By placing these capacitors on the input side of the current probe, the current probe will include any additional current that any large output capacitors add as the current demands change. See Figure below.

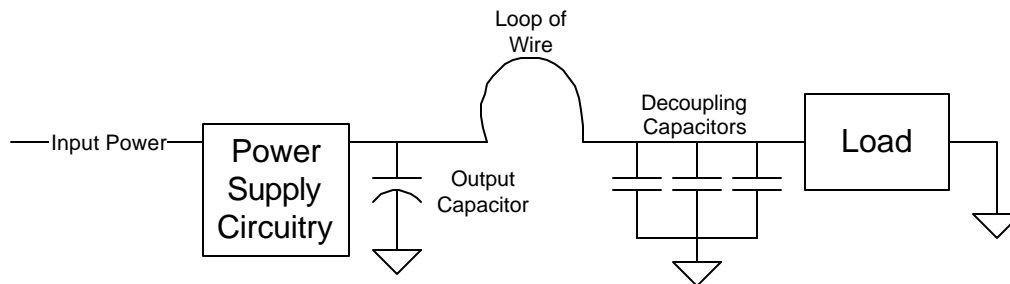


Figure 2. Wire Loop and Capacitor Locations

2.3 Connecting Everything Up

Figure 3 shows a block diagram example of the system, connected up to measure target system power consumption on an embedded system.

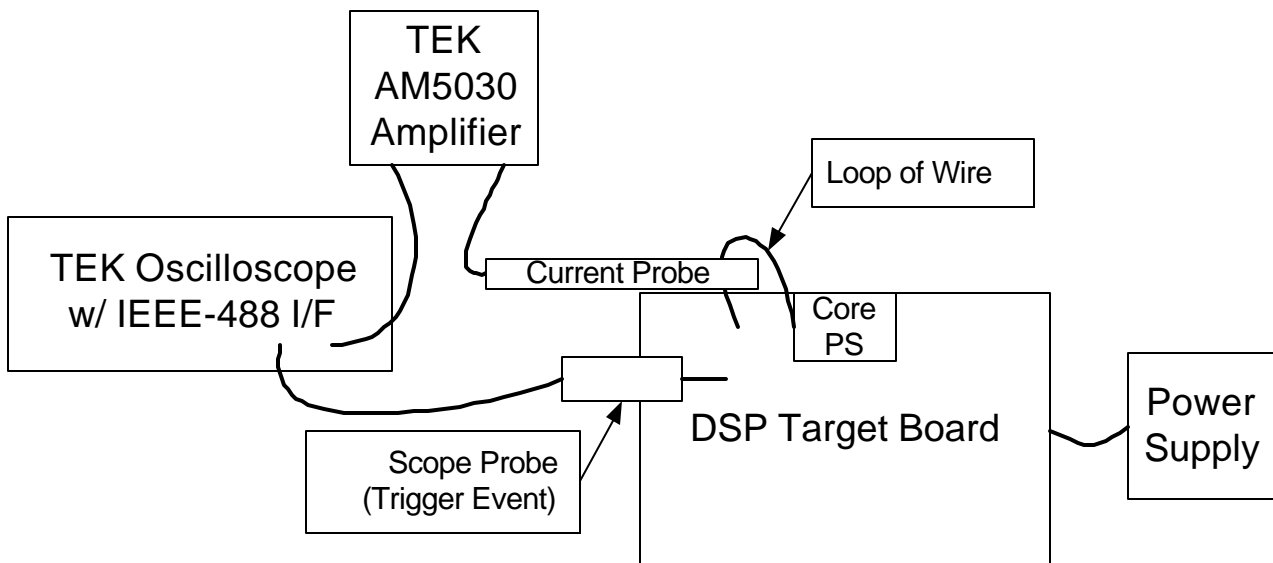


Figure 3. System Diagram

NOTE: If you wish to measure system power or I/O power, you may do so. There is no restriction on where power is measured. Just make sure to enter the correct voltage into the Power Analyzer Plug-in.

3 Analyze Energy Consumption of Your Target System

3.1 Setting Up the Scope

The oscilloscope should be set up in the following manner, for proper measurement of the current draw via the current probe:

1. The default configuration is to use channel 1 as the data channel (channel with the current probe from which we calculate the power), and channel 2 as the trigger channel (channel connected to an I/O pin that coordinates the beginning and end of the profiled function/range).
2. The voltage scaling of the channels should be 10mV/div for the data channel (Channel 1), and 2 V/div for the trigger channel (Channel 2).

3.2 Setting Up the Current Probe

The user should follow the setup procedure for the current probe and amplifier as described in the manufacturer's instruction manual. After completing the calibration, the user will need to adjust the output (current/div) setting of the current amplifier, to maximize the output of the current measurement. The current/division setting of the current amplifier must be input into the Power Analyzer Plug-In.

3.3 Notation

- Single range real time SR
- Multi-function real time MR
- Multi-function non real time MNR

3.4 Measuring Power Consumption

Once you have set up your PC (running CCStudio with the Power Analyzer Plug-In) to interface with your target board via an XDS-510/560 (or compatible) Emulator or onboard emulation, and to interface via an IEEE 488 card and cable to the oscilloscope, you are ready to start instrumenting your application for power measurement.

3.4.1 Flow of Operation

1. Connect the Power Capture equipment.
 - You need to have decided what pin you want to use for triggering and connect a probe from this pin to the trigger channel of your Power Capture equipment.
 - You need to decide which power source you want to measure (e.g. System, DSP, I/O, etc.) and connect a probe from this source to your Power Capture equipment.
2. Open Code Composer Studio and from the *Profiler* Menu, select *Power Analyzer*.
3. Select the *Options* button to adjust Settings (*General*, *Peripheral* and *Scope*) to match your measurement setup (pin used for a trigger, supply voltage being measured, current probe gain, etc.)
4. Open the project you want to analyze for power.
5. Decide if you want to measure over a single run or multiple runs. For a single run, you may choose from three of four modes available (Multi-Function Real-Time, Multi-Function Non-Real-Time, and Single Range Real-Time). These mode options allow you to profile a single range of code (which could be your entire application), or profile all (or some) functions in your application. For multiple runs, the *Frequency Sweep* mode is also available.

See [Selecting an Instrumentation Mode](#) on the differences and trade-offs of each mode. You must decide which one of these four modes is best for you. Based on this decision you have to perform some preparatory steps:

Select Mode, Multi-Function, Real-Time (MR). You must also add the correct profile support and RTDX libraries to your project, update your interrupt vector table, add the compiler option "-gpp", and add the power analysis multi-function real-time linker option "-u _TSPP_mr" to your project. In addition, if your application normally runs in an endless loop, you need to alter the code so that the execution completes.

-or-

Select Mode, Multi-Function, Non-Real-Time (MNR). You must also add a profile support library to your project, add the compiler option "-gpp", and add the power analyzer multi-function non-real-time linker option "-u _TSPP_mnr" to your project.

-or-

Select Mode, Single Range, Real-Time (SR). You must also add the correct profile support library to your project.

-or-

Select Mode, Frequency Sweep (SWP) feature. This option is quite different from the above three modes, as it will allow you to run your application with varying frequency settings.

For details on all modes, refer to the Online Help for the Power Analyzer.

6. *Rebuild* the application and *Load* it on to the target board (MR, MNR, SR modes only).
7. When profiling in one of the multi-function modes all of the functions in the application will be listed and will be configured for profiling by default. If you do not wish to have some functions profiled, uncheck the check box next to the function name. If a function is not profiled, the power used by a particular execution of that function is attributed to the nearest caller of the function that is being profiled..
8. *Run* the application.
9. If your application normally runs in an endless loop you must halt the application so the profiled data can be collected, collated and displayed. In the single-range mode, you must ensure that the application always halts AFTER the range of code to be measured has executed at least once. If profiling in multifunction non real-time mode, you can halt the target at any point.
10. After evaluating the data, you may save the data via *Options, Save Results* or *Options, Save Raw Data* buttons. To clear the results before the next run, you can select *Options, Clear Profile Results*.

See the next figure for details.

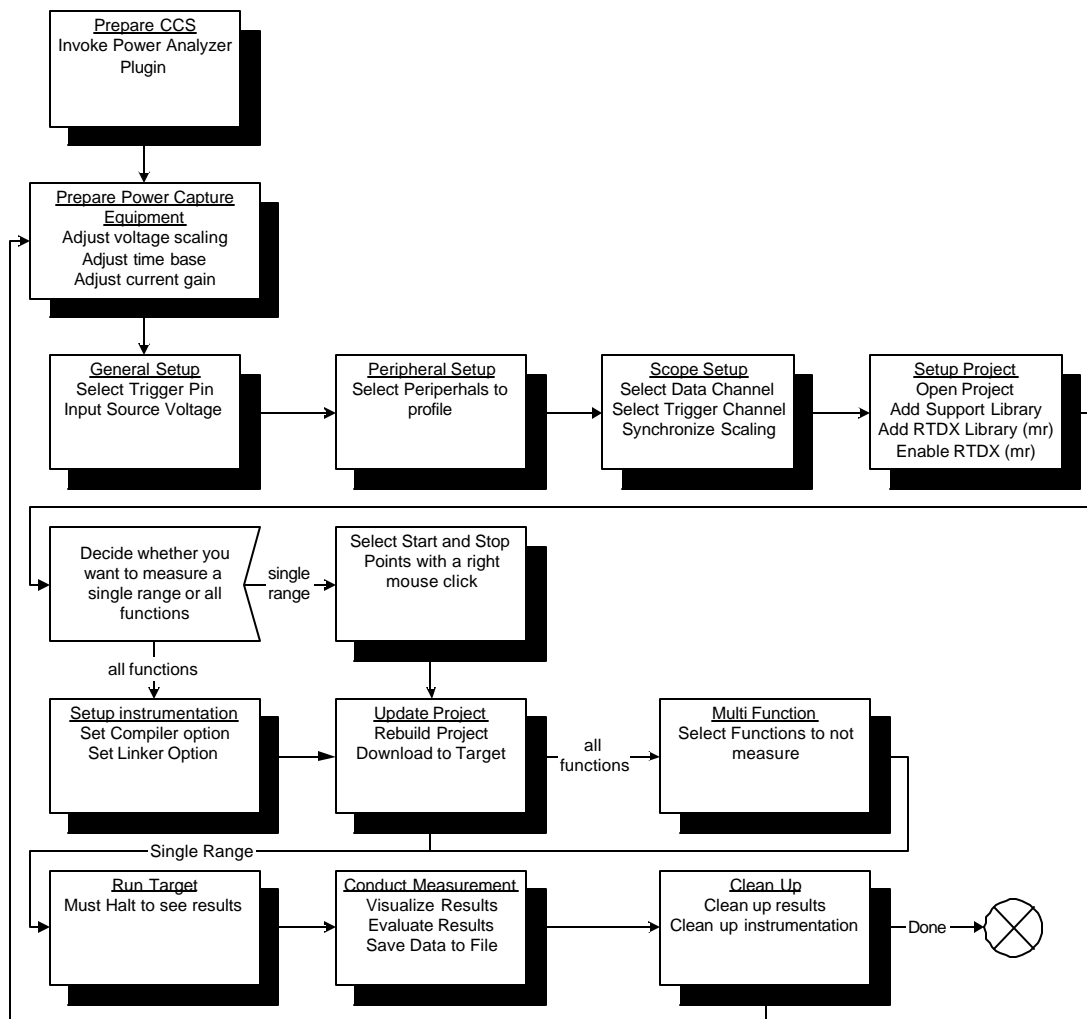


Figure 4. Power analysis flow chart

3.4.2 Using the Power Analysis

For detailed information on how to use the Power Analyzer plug-in see the Online Help for the Power Analyzer. This is accessible from the Code Composer Studio Online Help Contents page.

3.4.3 Selecting an Instrumentation Mode

The plug-in provides four modes of instrumentation, accessible through the *Mode* button.

1. **Multi-Function, Real-Time (MR) mode:** Instruments every function on the target application to be measured, and will record the power consumed by each function without ever stopping the target. The power consumption for all the functions is displayed when the application completes and the target is halted. Currently, C, C++, and assembly functions can be profiled. Any interrupt handlers that are implemented in C are not instrumented for profiling.

The multifunction real-time profiling support uses the RTDX Data log interrupt, `_datalog_isr`. This handler must be associated with interrupt 25. You will have to modify your interrupt vector table accordingly for multifunction real-time profiling for use with the c54x and c55x. The TMS470 does not require any modifications to the interrupt vector table. For RTDX transfer to occur, you must also enable interrupts early in your application. See the RTDX documentation for complete details.

2. **Multi-Function, Non-Real-Time (MNR) mode:** Instruments every function on the target application to be measured. It stops the target upon execution of each function; hence, the non-real-time name. The power consumed by each function is displayed each time the target halts at a function. Currently, C, C++, and assembly functions can be profiled. Any interrupt handlers that are implemented in C are not instrumented for profiling.
3. **Single Range, Real-Time (SR) mode:** Allows you to set a start and end point for measurement, and will display the power consumed while executing between those two points. If the range is executed more than a single time, the power for the range is averaged. **Note:** When profiling assembly code registers are not preserved before or restored after the CALL instruction to the power profiling support code. Users must save and restore any necessary registers that may be in use around the CALL.
4. **Frequency Sweep (SWP) mode:** A Frequency Sweep allows you to run one or more out files at a variety of frequencies to compare the power consumption across frequencies. The Frequency Sweep mode is only available to C55x targets.

The mode selection depends on the type of information you want to collect, the real-time constraints of your application, and the amount of intrusiveness and latency you are willing to accept during profiling. See the information below:

Mode	Run-Time Intrusiveness	Memory Overhead	Accuracy
Single range	Less [†]	Less	Most, Best for ranges > 5 μ s
Multifunction non-real-time	More	Less	Best for functions > 5 μ s (error < 10%)
Multifunction real-time	Less [†]	More	Best for functions > 5 μ s (error < 10%)

[†] Support Library (code and data) should be placed in internal memory, to reduce the amount of intrusiveness.

Table 1. Mode Selection

3.5 Emulation Logic and Run Free

Emulation logic on the target system consumes some power. This power is measured with the applications power in MR and MNR modes. These modes need the emulation logic because they use breakpoints (MNR) and RTDX (MR). Since the power consumed by the emulation logic is included with the power measured by the Power Analyzer, these measurements cannot be considered absolute, rather they are relative.

It is possible to turn off the emulation logic in SR mode. After instrumenting and loading your program, run it with the run free command on the debug menu instead of the normal run command. The run free command turns off the emulation logic and then instructs the target processor to run. There is one minor caveat, it takes some time for the emulation logic to be completely shut off. Therefore if the range you're measuring is at the very beginning of the program you should include a short delay loop before the range you're measuring begins.

3.6 Saving Results

The Power Analyzer allows you to save captured measurements (results and raw data) to a Comma Separated Value (CSV) file. This file can then be imported into Excel, Lotus, and many other applications.

4 System Characteristics

4.1 Simulator Support

This tool is NOT to be used with a Simulator.

4.2 Scope Compatibility

This tool has been validated with Tektronix TDS scopes only.

4.3 System Power Considerations

1. Location of capacitance, with respect to the current measurement, has a large impact on the response of the measurement.
2. Longer wire loops sometimes cause ringing on the measurement. It is best to make the wire loop as short as possible.
3. The best response with this test setup and our boards (C5510 TEB) was approximately 1 to 2 MHz.
4. Based on the test results of the TMS320C5510 TEB, a sense resistor setup should be explored to eliminate the ringing and minimize the noise.
5. External environment temperature can have a direct affect on the current probe, producing varied power results.

Appendix A Designing for Accurate Power Measurement

For users new to power measurement this appendix has been included to describe items that can be done to improve the access of making the current measurement of a new printed circuit board (PCB). These changes should have minimum effect on the cost or final design of the boards.

Add two pads to your design to accommodate a large surface mount resistor or a large diameter trace that could be cut to add the loop of wire to the design.

The addition of mounting holes for a two-pin header will allow for easy mounting of the loop of wire in this current configuration. However, this may be used with the addition of R1 and could be a large 0 ohms SMT resistor, or could be two pads with a large trace in-between that could be cut to enable power profiling. See Figure below.

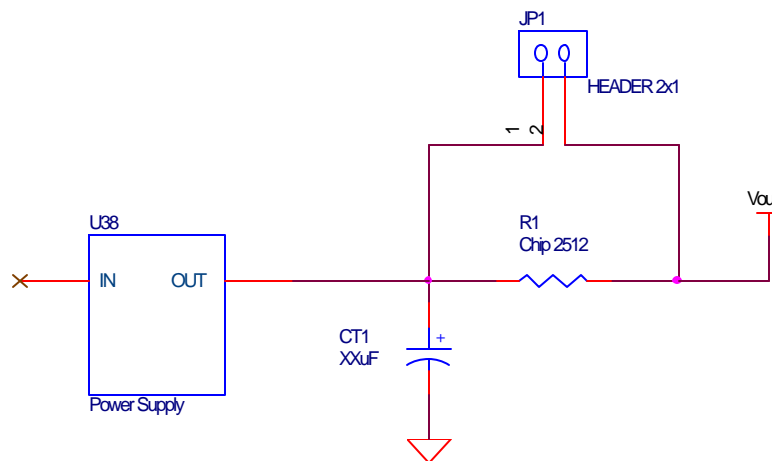


Figure 5. Board Modifications for Accurate Power Measurement