

# Using Fusion for Closed-Loop Power Supply Margining

---

## Overview

A growing number of embedded systems designers want the ability to dynamically alter the precise value of a power supply's voltage. Closed-loop power supply margining is a technique whereby a power rail is continuously monitored and the system can force the rail to go up or down in very small increments. Some of the benefits of doing this are as follows:

1. Better margin for the power supply rail. Dynamically adjusting the rail to maintain the ideal nominal value negates aging, tolerance, and temperature effects to the resistors that set the voltage value. The controller's bandwidth is generally kept well below the power supply's loop bandwidth so that it will not react to load changes and interfere with the power supply.
2. Ability to reduce the power supply's voltage in order to reduce overall power consumption. Since the closed-loop controller can make very small adjustments to the power supply's output value, the power supply can be kept at a lower than nominal value but still within the tolerance of all devices on that rail. This is generally not possible with open-loop control, since designers want to ensure maximum margin for long-term operation. Closed-loop control guarantees the margin over a long term, so reducing the voltage is now an option. This can also be done dynamically to adapt to traffic load conditions.
3. Ability to increase the power supply's voltage above nominal value. In this case, the devices on a given supply rail can benefit from a slight performance increase, and all devices are still guaranteed to operate within their maximum limits.
4. Some manufacturers currently ramp power supplies up and down during production testing to verify power supply margin. The values are generally fixed and provide a simple pass or fail. In addition, changing the increment requires trim resistors or other component changes. Closed-loop margining allows manufacturers to track the point at which failures occur, providing the manufacturer with records of the actual margin for each board that goes through the line. These records provide valuable reliability data to manufacturers.
5. Ability to compensate for speed and power consumption effects due to device temperature. For instance, as a device gets cooler (and faster), the voltage can be reduced in order to conserve power.
6. Ability to compensate for manufacturer device errata. It is not entirely uncommon for manufacturers to request a non-standard voltage on early production or prototype devices (e.g., using 3.0 V instead of 3.3 V to power an LVCMOS I/O bank). With a programmable voltage rail, a simple software change suffices instead of a board-level retrofit.

Several Power Management devices on the market support closed-loop margining; however, each has limitations in feature set and number of rails they can support. This document describes how to implement closed-loop margining using Actel's family of Fusion FPGAs. Using Fusion, it is now possible to take advantage of the superset of features available from an FPGA as well as implementing closed-loop trimming in a single device.

## A Typical Closed-Loop Margining System

Figure 1 shows the various blocks required to implement closed-loop trimming.

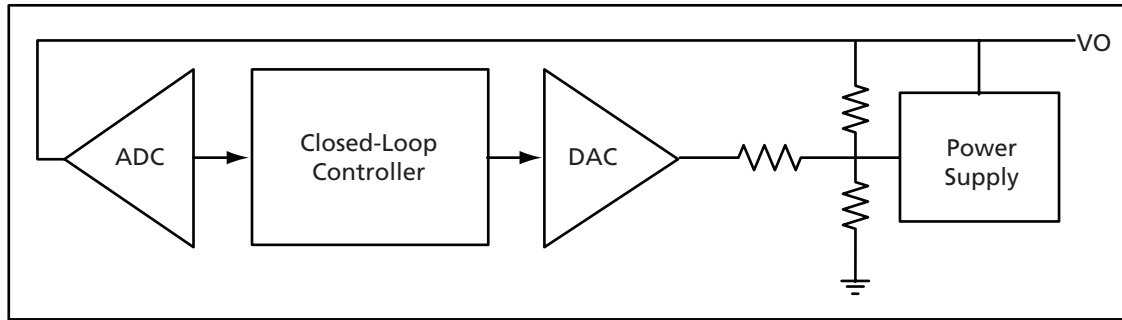


Figure 1 • Closed-Loop Trimming Block Diagram

The challenge in implementing this in an FPGA is the need for an analog-to-digital converter (ADC) and a digital-to-analog converter (DAC). Fusion addresses this problem by providing a built-in ADC and a specialized block for creating a DAC using an external RC low-pass filter.

## FPGA-Based DAC

Using a pulse width modulation (PWM) output feeding into a single-pole RC low-pass filter is a common and cost-effective technique for creating a DAC. This technique does not, however, lend itself well to applications where power supply margining is required. The main reason behind this is the high output ripple created by a PWM DAC. This document describes an alternate technique, still employing FPGA logic and regular digital I/O feeding into a single-pole RC filter, whereby a DAC can be created with extremely low output ripple. In addition, this DAC has lower output impedance, a higher update rate, and consumes fewer logic resources than the PWM-based DAC. For instance, in Actel's Fusion mixed-signal Programmable System Chip (PSC), a 12-bit low-ripple DAC design using a single-pole RC filter utilizes 51 logic tiles and can easily achieve  $\mu\text{V}$  ripple with update rates in excess of 1 kHz. This newer type of DAC will be referred to as "Low Ripple DAC" in this document.

The following sections describe the two approaches and discuss the performance advantages of the Low Ripple DAC. Both were implemented and tested on Actel's Fusion family of mixed-signal PSCs. The Low Ripple DAC was specifically designed for Fusion because of its inherent ability to act as a Power Supervisor, Sequencer, and Marginer, but could easily be used in a variety of other applications where low output impedance, low ripple, high bandwidth, and small size are important.

## PWM

First, let's look at how a PWM works. A PWM is essentially a clock or pulse train signal with fixed frequency and variable duty cycle.

Figure 2 shows what the output looks like for various duty cycle variations. The duty cycle can be varied over a number of discrete steps, depending on the particular resolution chosen. For example, an 8-bit PWM would have 256 different duty cycle possibilities. The PWM period in this case would be 256 times longer than the period of the reference clock used to drive the circuitry that generates the output. If the reference clock is 1 MHz (a 1  $\mu$ s period), the PWM period would be 256  $\mu$ s.

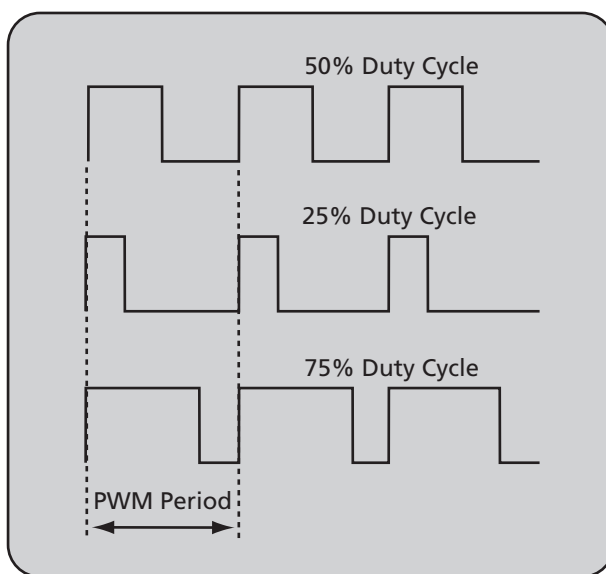


Figure 2 • Standard PWM

## PWM DAC

The energy in the PWM output is linearly proportional to the duty cycle. By feeding the output into a low-pass filter, the energy is converted to a DC level.

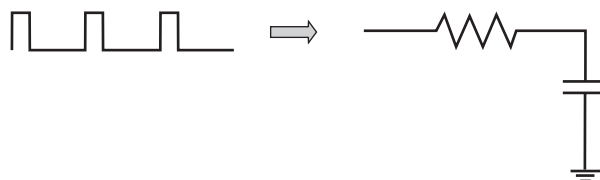


Figure 3 • Converting a PWM Output into a DC Level

This approach has limitations. The most important one has to do with the output ripple. As seen in [Figure 3](#), the filter reduces the size of the pulses significantly, but cannot eliminate them.

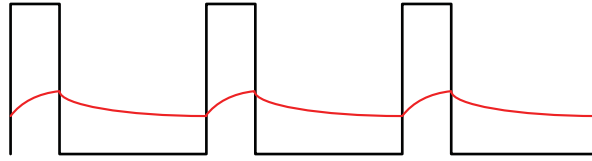


Figure 4 • PWM DAC Output Ripple

The output ripple will be a factor of several parameters:

1. RC time constant
2. PWM period
3. Size of the pulses

A larger RC time constant results in a slower rise time at the output of the filter, so that for a given period, the ripple will be smaller. It follows that a shorter period will also lead to a lower ripple. In other words, increasing the PWM frequency reduces ripple.

In the case of the single-pole RC filter in [Figure 1 on page 2](#), the output ripple is the rise time of a step input response, defined by [EQ 1](#).

$$V_{OUT} = V_{IN}(1 - e^{-T/t})$$

EQ 1

where

- $V_{OUT}$  = output voltage of the filter  
 $V_{IN}$  = input voltage to the filter  
 $e$  = natural base (2.71828...)  
 $T$  = high time during high pulse, low time during low pulse  
 $t$  = RC time constant ( $t = RC$ )

The worst-case ripple will occur at 50% duty cycle. "Example 1" applies [EQ 1](#) to a 50% duty cycle PWM output.

### Example 1

- Clock = 33 MHz  
Duty Cycle = 12 bits  
R = 1 k $\Omega$   
C = 220 nF

Ripple = ~406 mV

This is almost half a volt, which is much too large for margining applications. We can increase the RC time constant and the clock, but we encounter practical limitations.

- R needs to be small to keep the output impedance of the DAC low. That means we need to increase C. Bringing the ripple down to 0.5 mV would require a 220  $\mu$ F capacitor.
- Increasing the time constant reduces the bandwidth/response time of the DAC. To get reasonable ripple levels for the above design, the response time would be well below 1 update per second.
- Doubling the clock rate only halves the ripple. Limited results can be obtained before the clock frequency becomes impractical to implement. In addition, designs may have a limited number of clock frequencies available.

Adding a second- or third-order filter would help, but that increases the cost, complexity, and board space requirements. What we want is a way to reduce ripple without having to increase the clock or RC time constants.

## Low Ripple DAC

Low Ripple DAC was designed with low ripple output in mind. It processes the output pulses to minimize the resulting ripple at the output of the low-pass filter. As such, a single-order filter is the only requirement for achieving extremely low ripple while maintaining low source impedance.

For instance, if we use the same component values as in "Example 1" on page 4, instead of approximately 406 mV of ripple, we now get about 454  $\mu$ V of ripple. This is a reduction of ripple by approximately 3 orders of magnitude.

A more interesting example would be to bring the ripple down to below 50  $\mu$ V. Most supplies regulate to a few mV of ripple on the output, so to put things into perspective, this is 20 times smaller than 1 mV. One of the advantages of Low Ripple DAC is its inherent ability to increase precision without affecting ripple, so let's consider a 14-bit DAC example.

### Example 2

Clock	=	66 MHz
Precision	=	14 bits
R	=	250 $\Omega$
C	=	4.7 $\mu$ F

*PWM Ripple* = ~165 mV

*Low Ripple DAC Ripple* = ~43  $\mu$ V

### Example 3

Finally, if you can increase the resistance to 1 k $\Omega$ , for example (acceptable for most applications), then you can reduce the capacitance.

Clock	=	66MHz
Precision	=	14 bits
R	=	1 k $\Omega$
C	=	1 uF

*PWM Ripple* = ~193 mV

*Low Ripple DAC Ripple* = ~50 uV

In "Example 3", the full scale settling time is just under 10 ms, so the update rate easily achieves 100 updates per second.

## Ripple Comparison

Below is some real-world data.

### Example 4

R = 22.3 k $\Omega$   
C = 20 nF  
V<sub>CC</sub> = 3.3 V  
Clock = 18.431 MHz  
Duty Cycle = 12 bits (value set to 2,048 decimal – 50%)

PWM Ripple = 364 mV      Low Ripple DAC Ripple = 401  $\mu$ V

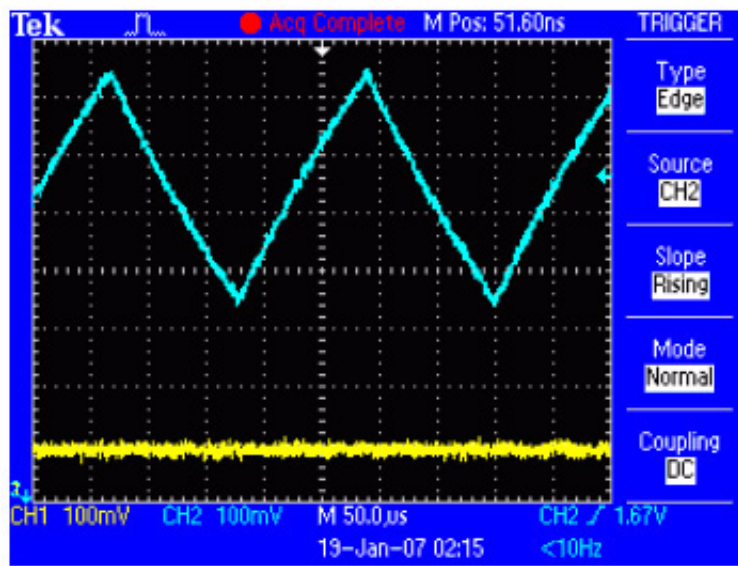


Figure 5 • Ripple Comparison Scope Plots

There is about 15 mV of digital noise on the power supply for the board used in this experiment, so with 401  $\mu$ V of ripple it is actually impossible to see the Low Ripple DAC ripple with this setup. The PWM ripple, however, at 364 mV is quite easy to measure.

## Bandwidth Comparison

The significant reduction in ripple allows us much more flexibility in choosing an RC time constant. This means we have more room to trade off ripple and bandwidth parameters when choosing the RC time constant. Take a look at Figure 6.

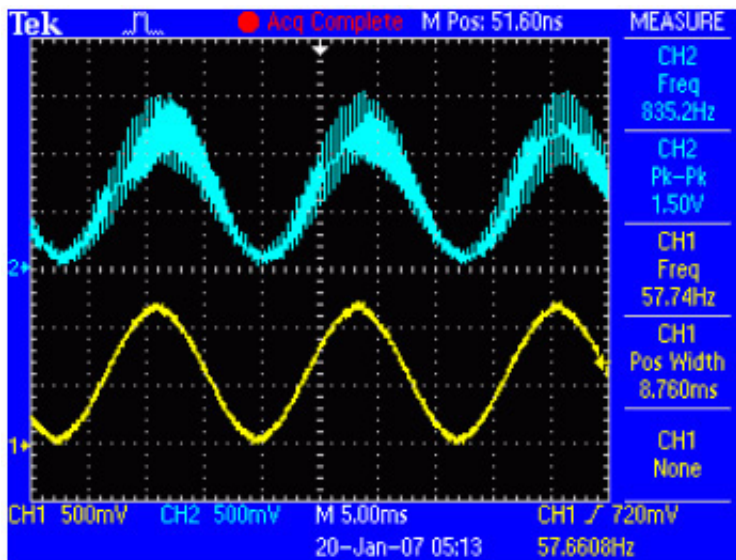


Figure 6 • Bandwidth Comparison

It shows a 60 Hz sine wave going through both types of DACs with identical filter parameters, taken from Example 2. (We will continue to use blue for PWM DAC and yellow for Low Ripple DAC, as in Figure 5 on page 6). Notice that the large ripple of the PWM DAC makes the sine wave rather jagged. To reduce the ripple, we increase the RC time constant of the PWM DAC filter (C is increased by a factor of 10). The problem is that we also reduce the bandwidth of the filter to the point where the actual signal is being attenuated. See Figure 7.

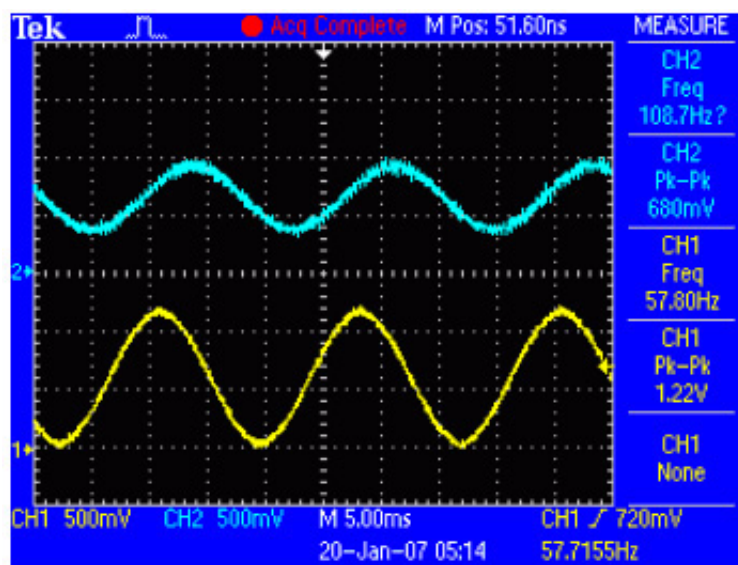


Figure 7 • Additional Filtering

Even here we can still see some ripple on the PWM DAC output, whereas the Low Ripple DAC output is smooth. Reducing ripple to acceptable levels by simply increasing the RC time constant results in a DAC with extremely poor bandwidth, which translates to slow response time.

### System Requirements

One key consideration for the Low Ripple DAC is that it must be monotonic. Accuracy is not critical, since the closed-loop controller does not really care what the actual DAC value is. It only needs to look at the voltage rail and decide if the DAC should go up or down. It is the accuracy of the ADC that sets the overall accuracy of the system. But if the output was not monotonic, a control signal telling the DAC to go up by one LSB might actually cause the output to go down by one LSB. The system would eventually self-correct, but the net effect is to introduce noise on the order of a few LSBs.

The plot in Figure 8 shows the actual measured response of the Low Ripple DAC versus an ideal response. The setup was as follows:

- R = 470  $\Omega$
- C = 1  $\mu\text{F}$
- V<sub>CC</sub> = 3.3 V
- Clock = 9.2 MHz
- I/O = 12 mA drive strength, low slew rate
- DAC Values = Sweep from 0 to 1,023

The 10-bit DAC output was measured using the 12-bit Fusion ADC. The plot clearly shows monotonic behavior. The data collected was carefully analyzed to ensure that each step is monotonic.

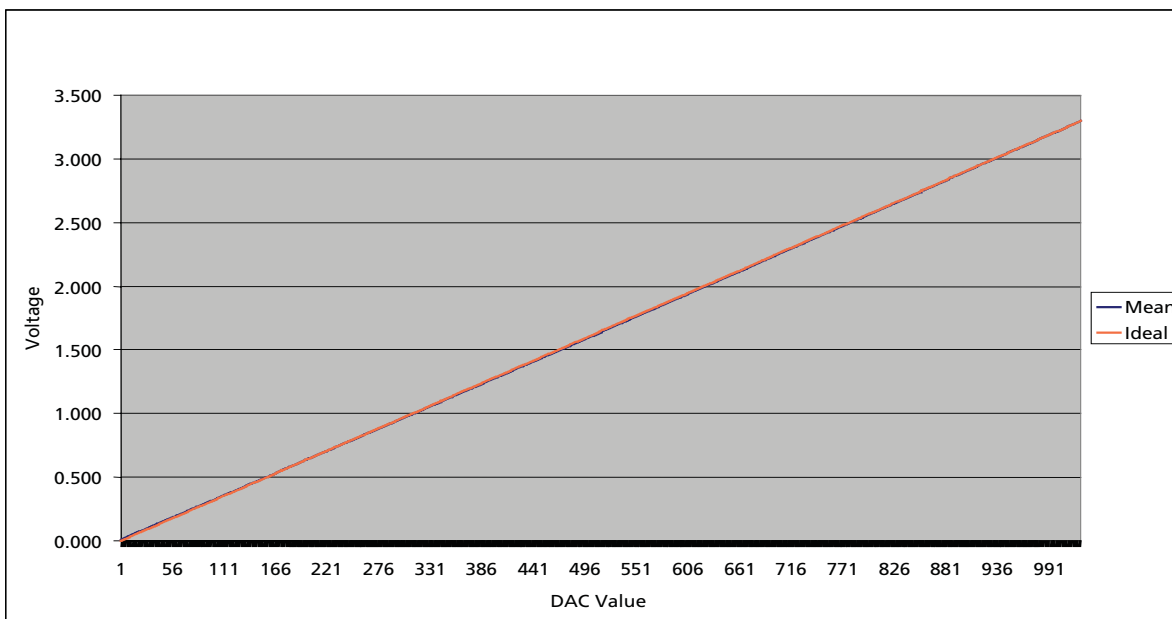


Figure 8 • Low Ripple DAC Plot

## Closed-Loop Margining Demonstration

With Actel's System Management Development Kit, you can use a Fusion device to control the output value of a power supply. In the example to follow, the DC1084A-A Demonstration Board from Linear Tech was used to connect Fusion to an LTM4062 switch mode power module. Connections on the System Management Demo Board are applied in the lower right corner, as shown by the red circle in [Figure 9](#).

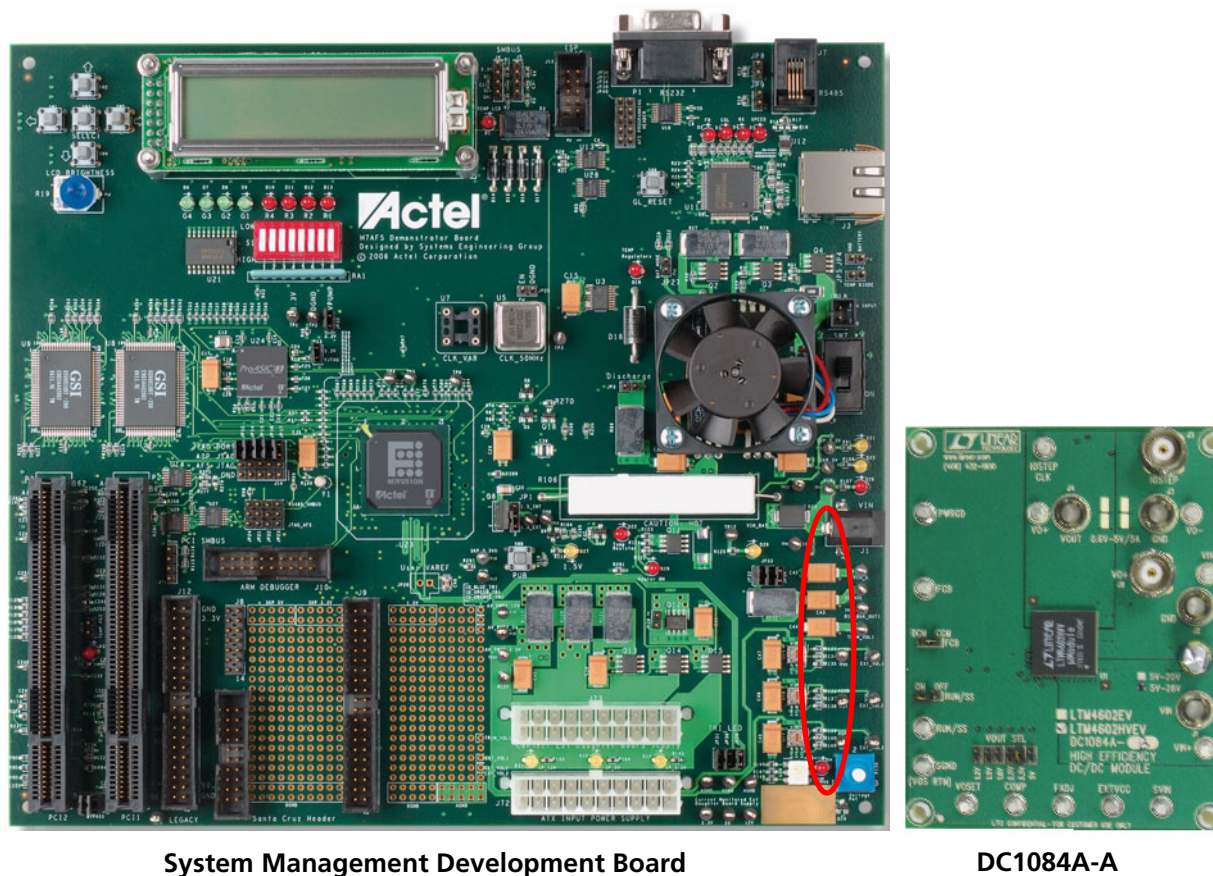


Figure 9 • Demonstration Boards

To perform closed-loop trimming, the following board modifications to the System Management Demo Board are necessary:

1. R207 ( $R_f$ ) = 4K7 (center, backside of the board)
2. R206 ( $R_{trim}$ ) = 100 k (center, backside of the board)
3. C36 ( $C_f$ ) = 220 nF (center, front side of the board)
4. Depopulate C44 (halfway between the fan and the blue potentiometer)
5. Program the AFS600 with the ClosedLoopTrimDemo.stp programming file.
6. The VOUT SEL header on the LTC board has 6 jumper locations. You should only populate the 3.3 V location. This essentially selects a  $R_{set}$  value of 22.1 k $\Omega$  ([Figure 10](#)).
7. Connect the two boards as shown in [Figure 10](#).
8. Connect the 9-pin serial port connector to a COM port on your PC.
9. Launch HyperTerm using 115200 8-N-1

The resistor values and ADC configuration are chosen so that the controller can adjust the power supply output to lie between 0.8 V and 2.5 V. Range and precision are decided by the  $R_{trim}$  and  $R_{set}$  resistor selection and the bit width of the DAC.

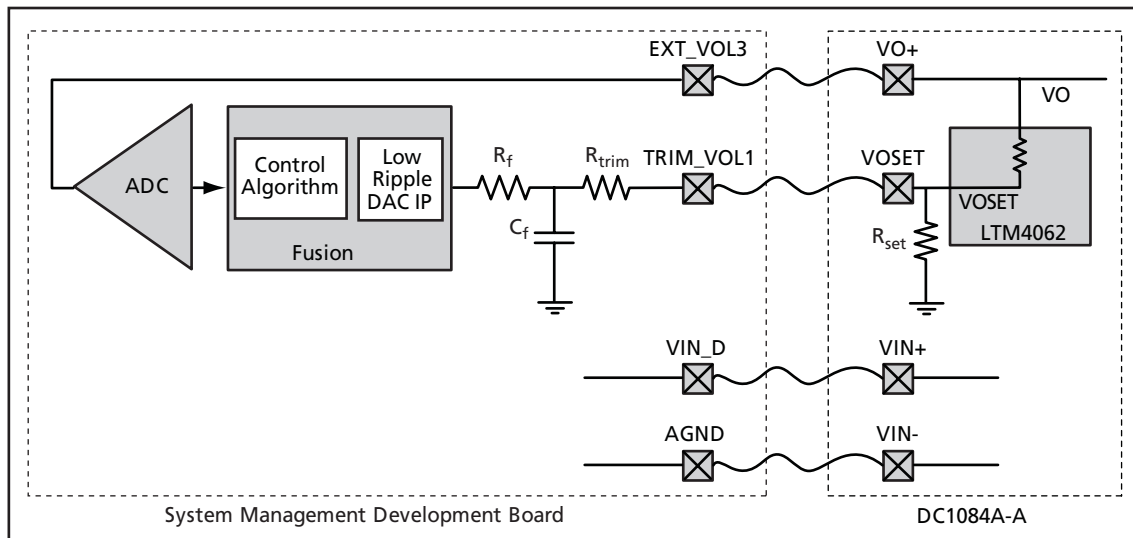


Figure 10 • Demonstration Board Connection

After programming, click the **GL\_RESET** button at the top right of the System Management Development Board (or toggle the power switch). You should see the following menu appear.

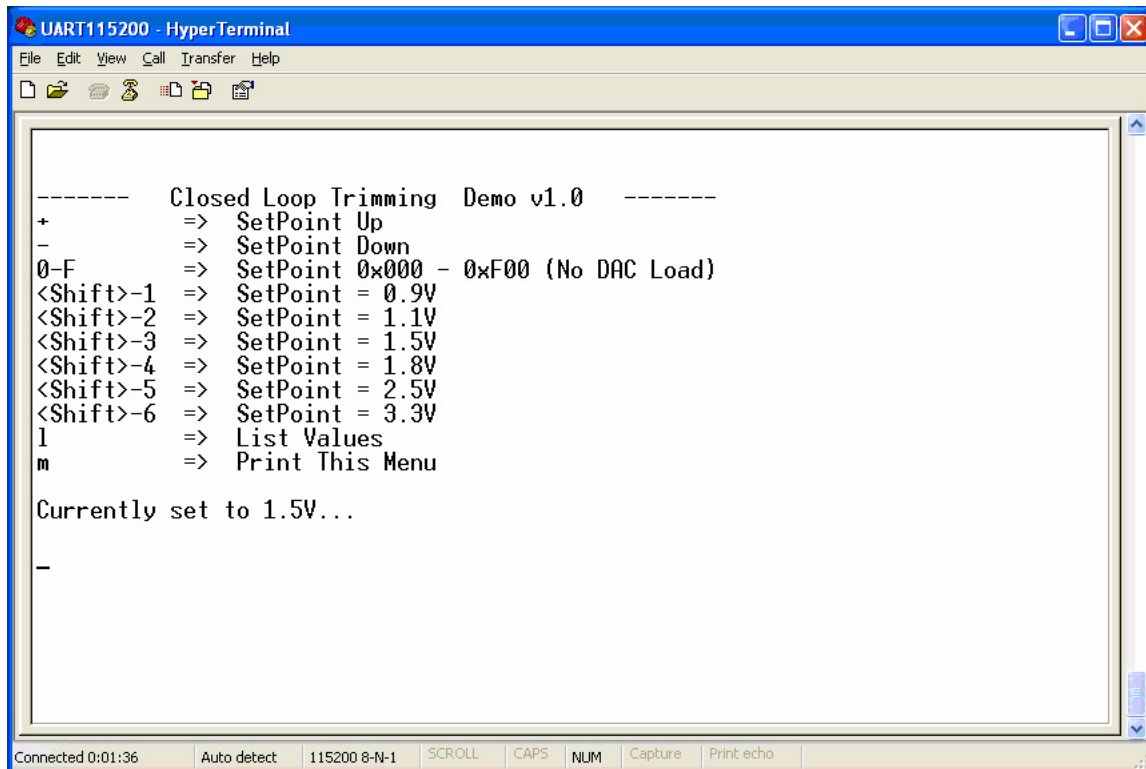


Figure 11 • Closed Loop Trimming Menu

## Command Description (case sensitive)

+	Increases the SetPoint voltage value by 1 LSB (press the "=" key)
-	Decreases the SetPoint voltage value by 1 LSB
0-f	Immediately sets the SetPoint value based on the actual key pressed 1 maps to 0x100 2 maps to 0x200, etc.
<Shift>-1	Immediately sets the SetPoint value to achieve 0.9 V. Also immediately sets the DAC value to match.
<Shift>-2	Immediately sets the SetPoint value to achieve 1.1 V. Also immediately sets the DAC value to match.
<Shift>-3	Immediately sets the SetPoint value to achieve 1.5 V. Also immediately sets the DAC value to match.
<Shift>-4	Immediately sets the SetPoint value to achieve 1.8 V. Also immediately sets the DAC value to match.
<Shift>-5	Immediately sets the SetPoint value to achieve 2.5 V. Also immediately sets the DAC value to match.
<Shift>-6	Immediately sets the SetPoint value to achieve 3.3 V. Also immediately sets the DAC value to match.
l	Lists the SetPoint value, the average (mean) value of the output of the power supply, and the current DAC value.
m	Prints the menu again.

## Operation

When the card first comes up, the SetPoint value will be 0x05DC. The algorithm will adjust the DAC value until it reads 0x05DC from the ADC (the ADC is set to 12-bit mode, and the Low Ripple DAC IP is configured for 12 bits as well). Since the analog input connected to the rail is set to use the 4 V prescaler range, a value of 0x05DC translates to exactly 1.5 V. EQ 2 is the formula used.

$$\text{ADCvalue} = 2^{\text{resolution}} \times \frac{V_{\text{IN}} \times \text{GAINprescaler}}{\text{VAREF}}$$

EQ 2

Where

$V_{\text{IN}}$	=	Voltage of power supply rail
GAINprescaler	=	0.625 (4 V range)
VAREF	=	2.56 V (internal reference)
Resolution	=	12 bits

If the ADC value is below the SetPoint value, the controller will decrement the DAC value by 1 LSB. Reducing the DAC output forces the power supply to raise the voltage (an inverse relationship exists between the DAC and the power supply voltage). If the ADC value is above the SetPoint value, then the controller will increment the DAC value by 1 LSB, thereby reducing the power supply voltage.

Before making another adjustment, the controller will average the voltage over several samples to get rid of in-band noise. The averaging process also ensures that the DAC update rate is much lower than the power supply's switching loop bandwidth so that no interaction will ever exist between the power supply's regulation and the closed-loop trimming function.

The Fusion ADC has a maximum accuracy better than 1%; therefore the power supply's output value will be within 1% of the desired value. You can simulate a calibration procedure by incrementing or decrementing the SetPoint value until the output reading is precisely the desired value. Reading the actual SetPoint (press the "l" key) and subtracting from the calculated SetPoint value will indicate the

offset required for that voltage rail. In a real system this process could be automated. The calibration offsets would be stored in Fusion's flash memory and subsequently used by the controller to compensate for the ADC error. Although most scenarios do not need better than 1%, calibration would allow one to achieve much better accuracy (0.1% is typical).

Note that when you adjust the SetPoint value up or down by 1 LSB (by pressing the "+" or "-" keys), the controller will adjust the DAC value by using the simple control algorithm described in the previous paragraph. However, when you select one of the five preset values, the SetPoint is changed and the DAC is immediately set to a value that will force the power supply output to be close to the desired voltage. From there, the control algorithm takes over and adjusts the DAC the rest of the way. If you press any key between "0" and "f", the SetPoint will be changed instantly, but the DAC will not incur a corresponding change. If you move the SetPoint by a considerable amount, it will take the algorithm some time to bring the power supply to match the SetPoint voltage. This is normal behavior. Any application that requires an immediate change in voltage can modify the DAC value simultaneously.

You can also simulate an external influence to the output supply by adding a jumper to the 1.2 V header position on the power supply board. This essentially reduces the  $R_{set}$  value, thereby forcing an instantaneous increase of the power supply's output. You can then observe how the controller will react by raising the DAC value and bringing the power supply output back to the SetPoint voltage.

## Control Algorithm

The control algorithm described in this demonstration is very simple and effective. Since it is implemented in a Fusion PSC, the choice of the actual algorithm used is entirely up to the user. More complex algorithms can easily be created.

In addition, the user can choose how to implement the algorithm. Some examples:

- Simple state machine designed using RTL (Verilog and VHDL)
- Simple interface to a processor or other control logic in the system. For example, an I<sup>2</sup>C interface could report the current voltage to a remote entity, and accept DAC changes from that entity.
- Using a processor directly in Fusion, such as CoreABC, Core8051, or the ARM<sup>®</sup> Cortex™-M1 processor core.

## Conclusion

By using a Low Ripple DAC block and an RC filter to create a DAC, it is now possible to margin power supplies from the digital output of Actel's Fusion PSCs at almost no extra cost or board space. The output impedance, bandwidth, and ripple more than satisfy the requirements for power supply margining. In addition, it is now possible to create several trim DACs because of the high number of outputs typically available on Fusion PSCs. Finally, with Fusion's 30 channels of ADC inputs, it is possible to supervise, sequence, and perform closed- or open-loop margining on several power supplies from a single chip.

No other solution in the industry provides the user with more flexibility: choice of algorithm, implementation style, and DAC precision. This added flexibility also significantly reduces risk in the event that the design requires unforeseen changes.

For your convenience, a ripple calculator is available to help analyze tradeoffs between clock frequency, RC time constant, bandwidth, and ripple magnitude. The "ripple calculator" and the "Low Ripple DAC block" are available along with the design files to implement the closed-loop margining application using Fusion.

Actel and the Actel logo are registered trademarks of Actel Corporation.  
All other trademarks are the property of their owners.



[www.actel.com](http://www.actel.com)

**Actel Corporation**

2061 Stierlin Court  
Mountain View, CA  
94043-4655  
USA

**Phone** 650.318.4200  
**Fax** 650.318.4600

**Actel Europe Ltd.**

River Court, Meadows Business Park  
Station Approach, Blackwater  
Camberley Surrey GU17 9AB  
United Kingdom

**Phone** +44 (0) 1276 609 300  
**Fax** +44 (0) 1276 607 540

**Actel Japan**

EXOS Ebisu Building 4F  
1-24-14 Ebisu Shibuya-ku  
Tokyo 150 Japan

**Phone** +81.03.3445.7671  
**Fax** +81.03.3445.7668  
<http://jp.actel.com>

**Actel Hong Kong**

Room 2107, China Resources Building  
26 Harbour Road  
Wanchai, Hong Kong

**Phone** +852 2185 6460  
**Fax** +852 2185 6488  
[www.actel.com.cn](http://www.actel.com.cn)

