TI Designs
Portable Audio Amplifier With Auto Audio Control 1S1P BMS Reference Design

Description

This TI Design is a high performance, 10-W (5 W per speaker) portable audio amplifier that includes everything needed to implement a battery management solution (BMS) for a portable audio amplifier (including a charger, fuel (gas) gauge, and protection for a 1S1P 18650 2400-mAh Lithium battery) for a Class D Audio Amplifier. Get extended operating time by using efficient power regulators, a Class D amplifier, and proper battery management.

Resources

TIDA-01182	Design Folder
bq25896	Product Folder
bq27426	Product Folder
bq29705	Product Folder
LM3481	Product Folder
MSP430FR5735	Product Folder
TPS78225	Product Folder
TPA3004D2	Product Folder
DAC081C085	Product Folder
PCA9306	Product Folder
CSD16323Q3	Product Folder
CSD17578Q5A	Product Folder

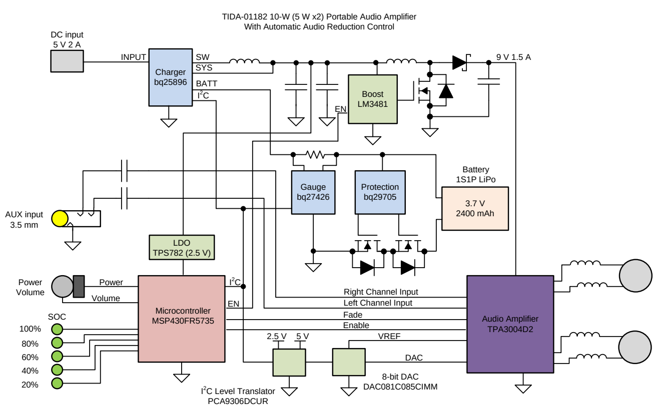
TI E2E™ Community — ASK Our E2E Experts

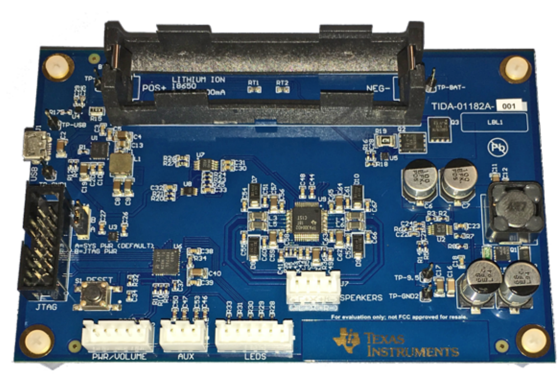
Features

- High Performance, 10-W (5 W per Speaker) Portable Audio Amplifier
- Includes Everything Needed for a BMS for a Portable Audio Amplifier (Charger, Fuel (Gas) Gauge, and Protection for a 1S1P 18650 Lithium Battery)
- Class D Audio Amplifier 0.01% THD
- High Efficiency Boost Regulator and Ultra-Low-Power LDO Regulator
- Ultra-Low-Power MSP430™ Microcontroller
- Extended Play Time Due to High Efficiency and Proper Battery Management Techniques

Applications

- Portable Audio Amplifiers
- Portable MP3 and DVD Players





An IMPORTANT NOTICE at the end of this TI reference design addresses authorized use, intellectual property matters and other important disclaimers and information.

1 System Overview

1.1 System Description

This is a system core design for any high-performance, 10-W (5 W per speaker) portable audio design that includes everything needed to implement a battery management solution (BMS). This TI Design includes a charger, gauge, and protection for a 1S1P 18650 2400-mAh Lithium battery. This core design can be used in many portable audio designs as a standalone portable audio amplifier or as a portable MP3 player or a portable DVD player. The Class D Audio Amplifier has enough power for almost any design that requires quality, high performance, portability, and long life. This TI Design offers an extended operating time by using efficient power regulators, the Class D amplifier, and proper battery management.

1.2 Key System Specifications

Table 1. Key System Specifications

PARAMETER	VALUE	UNIT
Minimum input voltage limit for charger	4.5	V_{MIN}
Maximum input voltage limit for charger	14	V_{MAX}
Battery SOC (at the time of test)	60 to 80	%
Max charge current	2.5	A
Charge termination current	250	mA
Battery voltage with audio amp on (no audio)	3.84	V
Battery current with audio amp on (no audio)	240	mA
Battery voltage with audio amp loaded	3.15	V
Battery current with audio amp loaded	2.93	A
Battery Impedance	235	m
Protection current limit	5.25	A
Boost regulator output voltage—audio amp off	9.6	V
Boost current—audio amp on (no audio)	82	mA
Boost regulator output voltage—audio amp on	9.59	V
Boost regulator output current—audio amp on	750	mA
Boost regulator output voltage under load	9.55	V
Boost regulator maximum load current	1.6	A_{MAX}
Boost regulator efficiency	85	%
Boost load regulation	160	mV P-P
Audio amplifier—Load test—Boost regulator output current	1.2	A
Audio amplifier—Load test—Boost regulator input current	2.93	A
Audio amplifier—Peak load test—Boost regulator input current	5	A
Audio amplifier Class D average current, low battery	320	mA
Audio amplifier Class D average current, high battery	230	mA
Power output	10.2	W
Output test the audio amplifier	84	dB
Maximum audio input level	2	V P-P
Total circuit current in low power shutdown mode	0.0007	A
Total circuit idle current	17	mA
Maximum temperature rise in the boost regulator	31	°C
Maximum temperature rise in the charger	38	°C
Maximum temperature rise in the audio amp	34	°C

1.3 Block Diagram

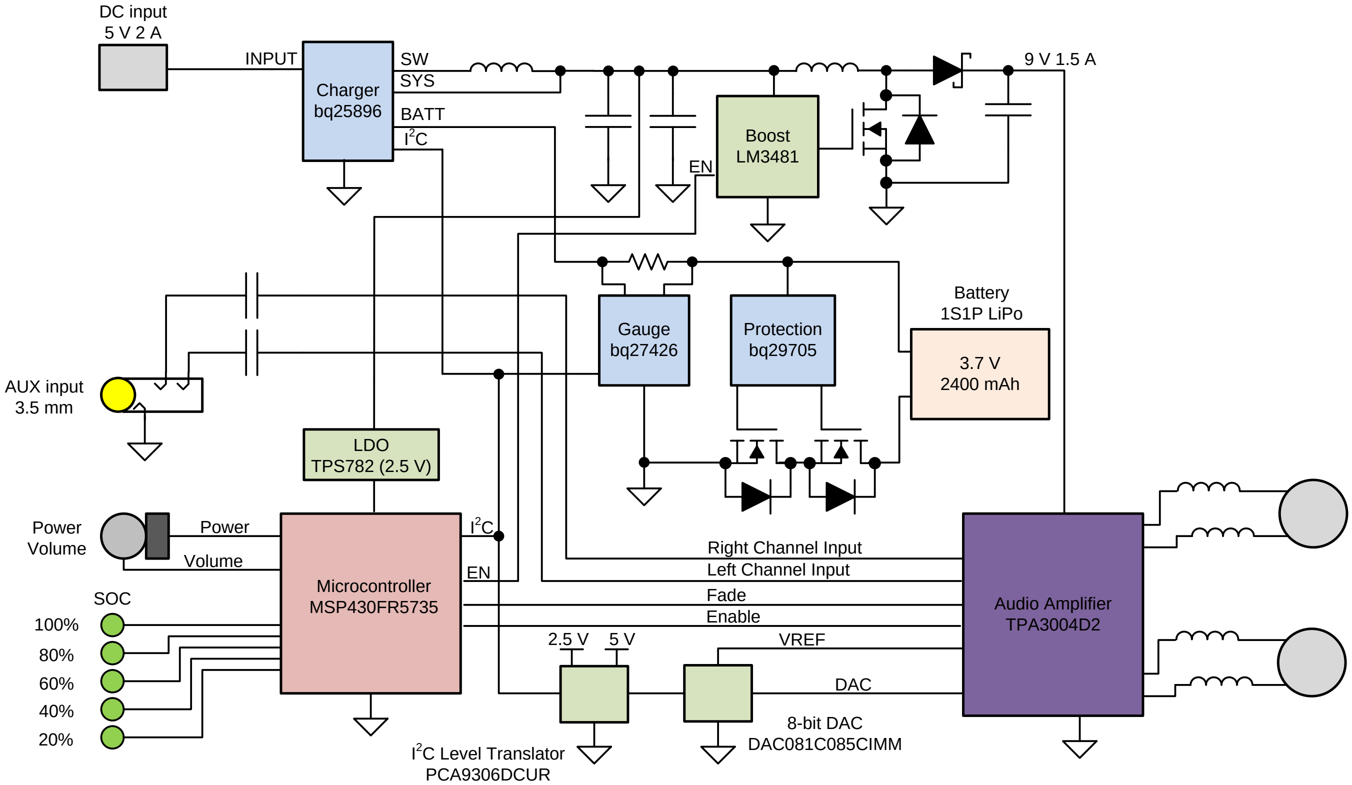


Figure 1. Block Diagram

1.4 Highlighted Products

1.4.1 bq25896 Charger

The bq25896 is a highly-integrated 3-A switch-mode device for battery charge and system power path management for a single-cell Li-Ion and Li-polymer battery. This device supports high input voltage fast charging. The low impedance power path optimizes switch-mode operation efficiency, reduces battery charging time, and extends battery life during discharging phase. The I^2C serial interface with charging and system settings makes the device a truly flexible solution.

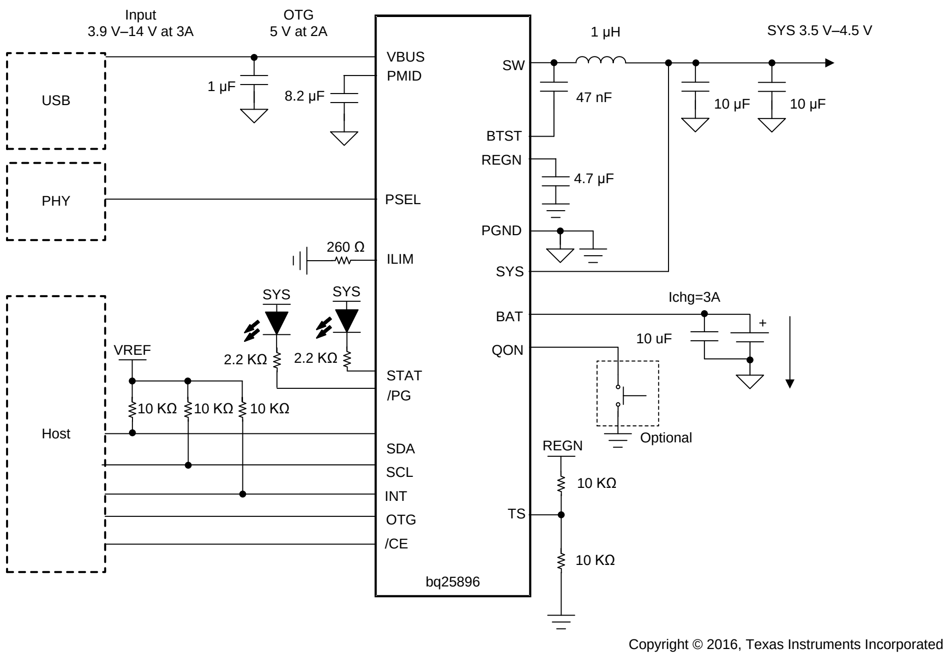


Figure 2. bq25896 Typical Application Diagram

1.4.2 bq27426 Gauge

The Texas Instruments bq27426 battery fuel gauge is a single-cell gauge that requires minimal user configuration and system micro-controller firmware development, leading to a quick system bring-up. Three chemistry profiles are pre-programmed to enable minimum user-configuration, and to help manage customer inventory across projects with different battery chemistries. The bq27426 battery fuel gauge has very low sleep power consumption leading to longer battery run time. Configurable interrupts help save system power and free up the host from continuous polling. Accurate temperature sensing is supported via an external thermistor. The bq27426 battery fuel gauge uses the patented Impedance Track™ algorithm for fuel gauging, and provides information such as remaining battery capacity (mAh), state of charge (%), and battery voltage (mV).

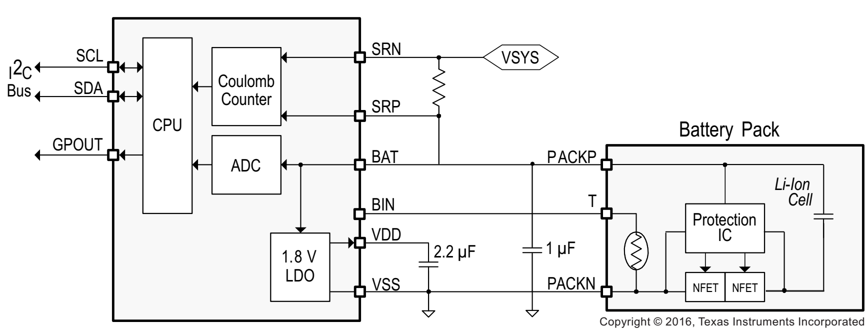


Figure 3. bq27426 Typical Application Diagram

1.4.3 bq29705 Protector

The bq2970 battery cell protection device provides an accurate monitor and trigger threshold for overcurrent protection during high discharge or charge current operation or battery overcharge conditions. The bq2970 device provides the protection functions for Li-Ion/Li-Polymer cells and monitors across the external power FETs for protection due to high charge or discharge currents. In addition, there is overcharge and depleted battery monitoring and protection. These features are implemented with low current consumption in normal mode operation.

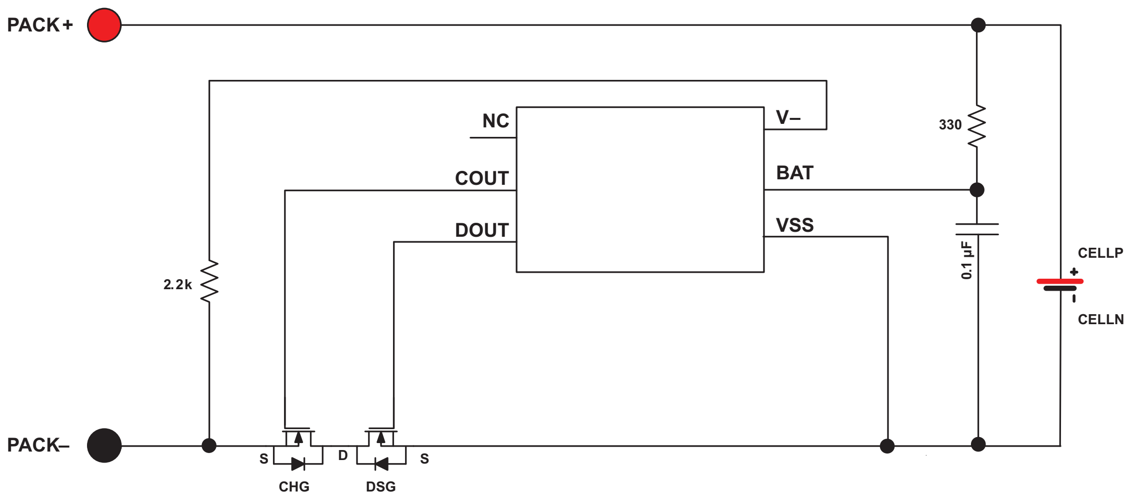


Figure 4. bq29705 Typical Application Diagram

1.4.4 LM3481 Boost Regulator

The LM3481 device is a versatile low-side N-FET high-performance controller for switching regulators. The device is designed for use in boost, SEPIC, and flyback converters and topologies requiring a low-side FET as the primary switch. The LM3481 device can be operated at very high switching frequencies to reduce the overall solution size. The switching frequency of the LM3481 device can be adjusted to any value between 100kHz and 1MHz by using a single external resistor or by synchronizing it to an external clock. Current mode control provides superior bandwidth and transient response in addition to cycle-by-cycle current limiting. Current limit can be programmed with a single external resistor. The LM3481 device has built-in protection features such as thermal shutdown, short-circuit protection and overvoltage protection. Power-saving shutdown mode reduces the total supply current to 5 uA and allows power supply sequencing. Internal soft-star limits the inrush current at start-up. Integrated current slope compensation simplifies the design and, if needed for specific applications, can be increased using a single resistor.

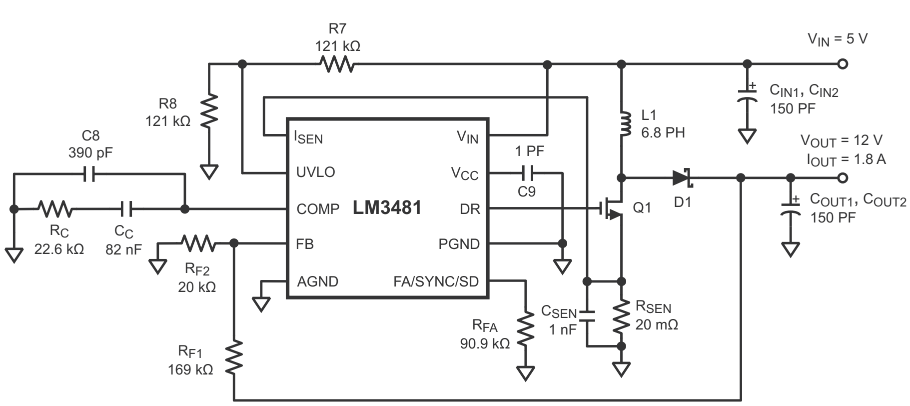


Figure 5. LM3481 Typical Application Diagram

1.4.5 MSP430FR5735 Microcontroller

The TI MSP430FR573x family of ultra-low-power microcontrollers consists of multiple devices that feature embedded FRAM nonvolatile memory, ultra-low-power 16-bit MSP430™ CPU, and different peripherals targeted for various applications. The architecture, FRAM, and peripherals, combined with seven low-power modes, are optimized to achieve extended battery life in portable and wireless sensing applications. FRAM is a new nonvolatile memory that combines the speed, flexibility, and endurance of SRAM with the stability and reliability of flash, all at a lower total power consumption. Peripherals include a 10-bit ADC, a 16-channel comparator with voltage reference generation and hysteresis capabilities, three enhanced serial channels capable of I²C, SPI, or UART protocols, an internal DMA, a hardware multiplier, an RTC, five 16-bit timers, and digital I/Os.

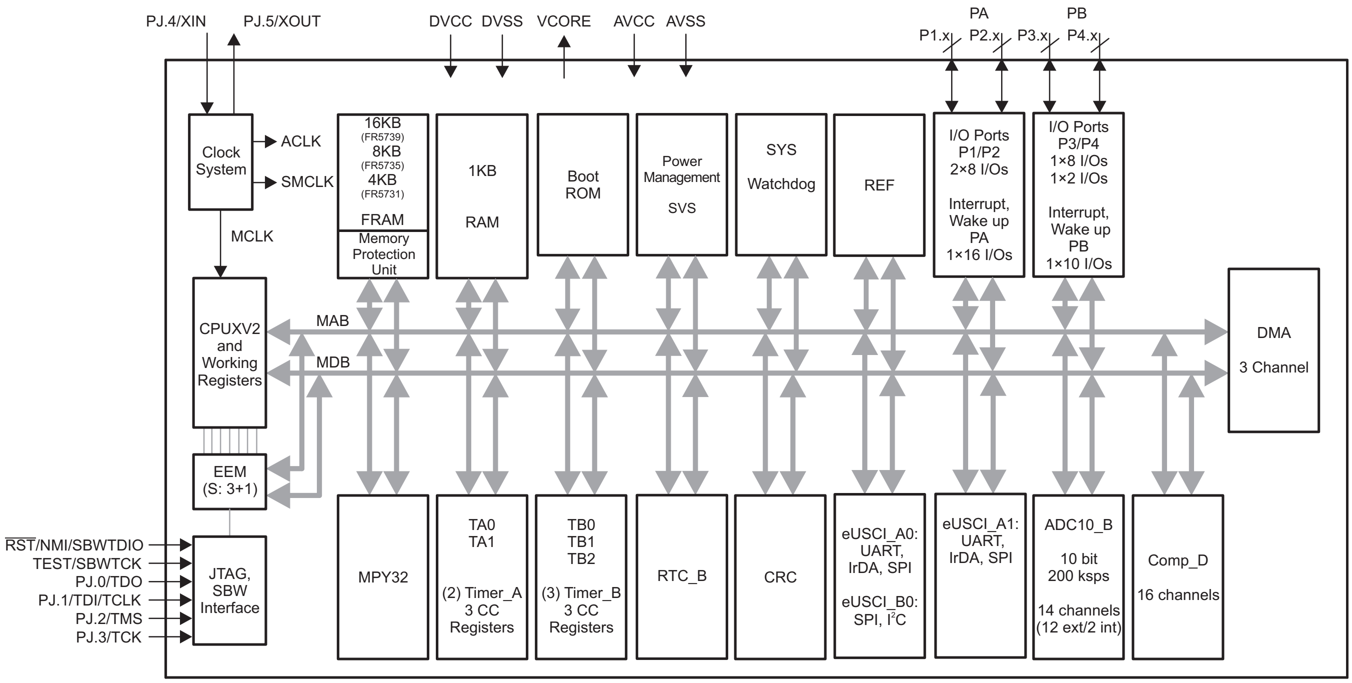


Figure 6. MSP430FR5735 Typical Block Diagram

1.4.6 TPS78225 LDO Regulator

The TPS782 family of low-dropout regulators (LDOs) offers the benefits of ultra-low-power and miniaturized packaging. This LDO is designed specifically for battery-powered applications where ultra-low quiescent current is a critical parameter. The TPS782 with ultra-low IQ (500 nA) is ideal for microprocessors, microcontrollers, and other battery-powered applications.

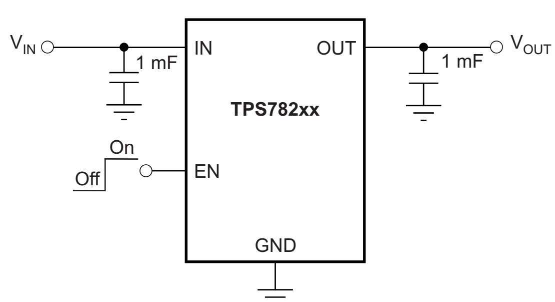


Figure 7. TPS782 Typical Application Diagram

1.4.7 TPA3004D2 Class D Audio Amplifier

The TPA3004D2 is a 12-W (per channel) efficient, Class-D audio amplifier for driving bridged-tied stereo speakers. The TPA3004D2 can drive stereo speakers as low as 4 Ω. The high efficiency of the TPA3004D2 eliminates the need for external heatsinks when playing music. Stereo speaker volume is controlled with a DC voltage applied to the volume control terminal offering a range of gain from –40 to 36 dB. Line outputs, for driving external headphone amplifier inputs, are also DC voltage controlled with a range of gain from –56 to 20 dB. An integrated 5-V regulated supply is provided for powering an external headphone amplifier.

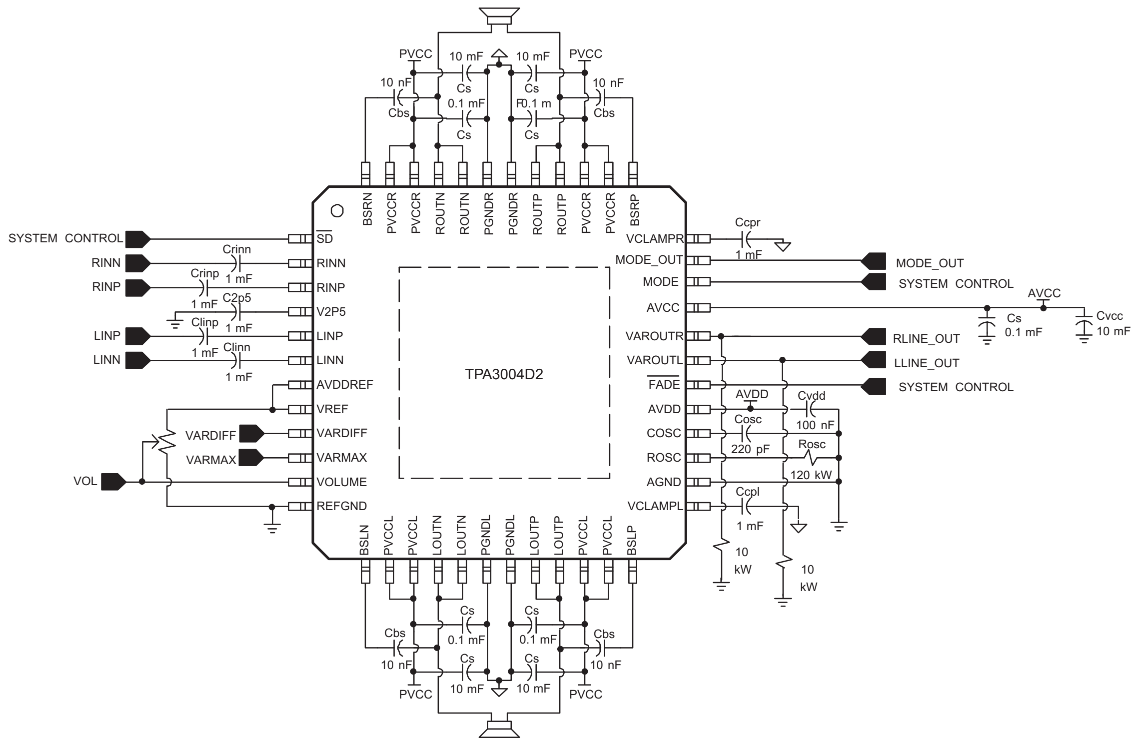


Figure 8. TPA3004D2 Typical Application Diagram

1.4.8 DAC081C085 8-Bit DAC

The DAC081C08x device is an 8-bit, single-channel, voltage-output digital-to-analog converter (DAC) that operates from a 2.7- to 5.5-V supply. The output amplifier allows rail-to-rail output swing and has a 4.5-µs settling time. The DAC081C081 uses the supply voltage as the reference to provide the widest dynamic output range and typically consumes 132 µA while operating at 5 V. It is available in 6-lead SOT and WSON packages and provides three address options (pin selectable). The DAC081C081 and DAC081C085 use a 2-wire, I²C-compatible serial interface that operates in all three speed modes, including high-speed mode (3.4 MHz). An external address selection pin allows up to three DAC081C081 or nine DAC081C085 devices per 2-wire bus. Pin-compatible alternatives to the DAC081C081 are available that provide additional address options.

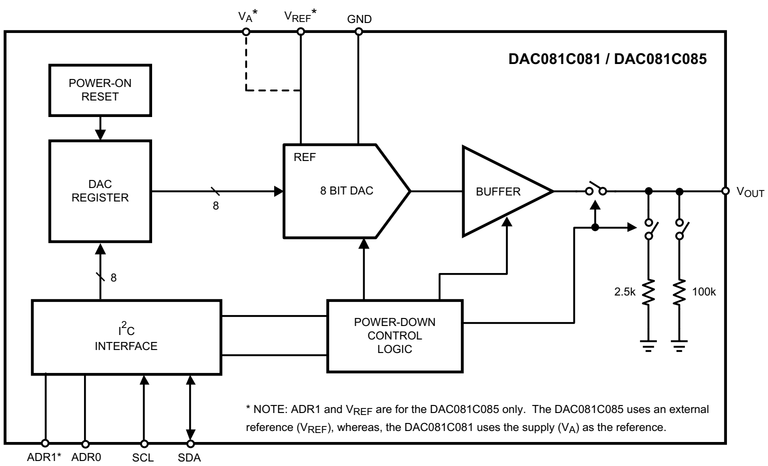


Figure 9. DAC081C085 Functional Block Diagram

1.4.9 PCA9306 Level Translator

The PCA9306 device is a dual bidirectional I^{2}C and SMBus voltage-level translator with an enable (EN) input and is operational from 1.2- to 3.3-V VREF1 and 1.8- to 5.5-V VREF2. The PCA9306 device allows bidirectional voltage translations between 1.2 and 5 V without the use of a direction pin. The low ON-state resistance (RON) of the switch allows connections to be made with minimal propagation delay. When EN is high, the translator switch is ON, and the SCL1 and SDA1 I/O are connected to the SCL2 and SDA2 I/O, respectively, allowing bidirectional data flow between ports. When EN is low, the translator switch is off, and a high-impedance state exists between ports.

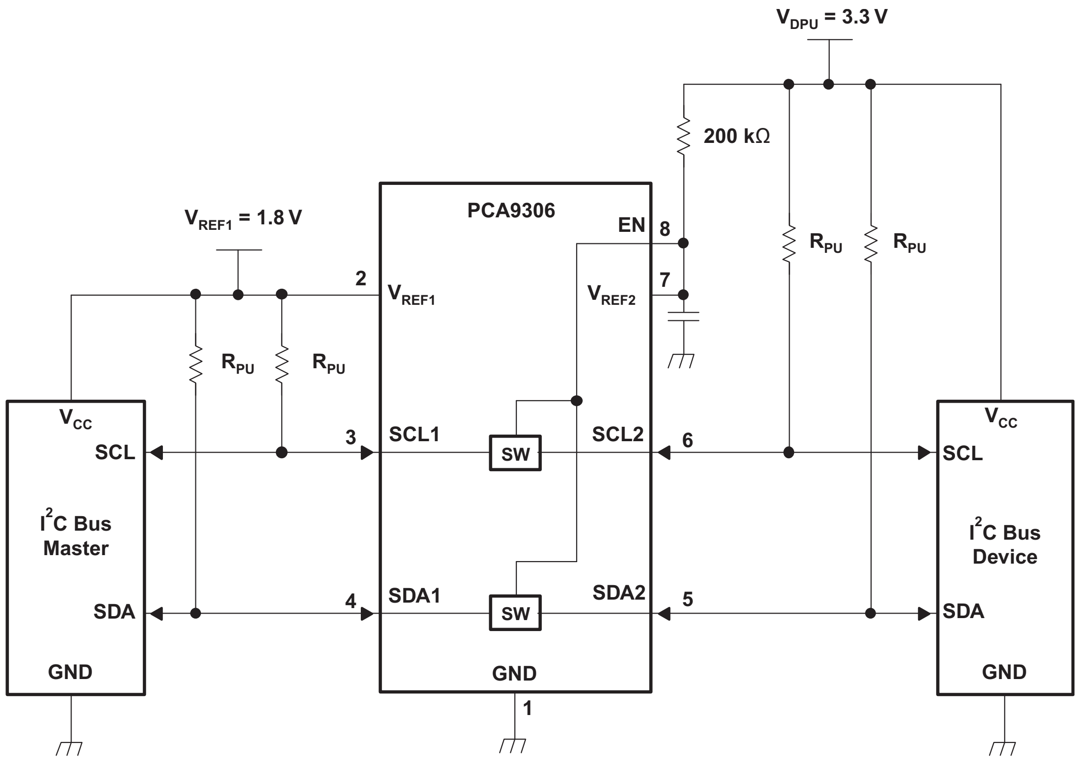


Figure 10. PCA9306 Typical Application Diagram

1.4.10 CSD16323Q3 MOSFET

This 25-V, 4.4-mΩ, SON 3.3-mm×3.3-mm NexFET™ Power N-Channel MOSFET has been designed to minimize losses in power conversion and optimized for 5-V gate drive applications.

1.4.11 CSD17578Q5A MOSFET

This 30-V, 5.9-mΩ, SON 5-mm×6-mm NexFET™ Power N-Channel MOSFET is designed to minimize losses in power conversion applications.

2 Getting Started Hardware and Software

2.1 Operations

Powering on and off is controlled by the knob in the center of the unit. Rotate to the left until it clicks for off. Rotate to the right to turn on and continue to rotate to the right to increase the volume. This unit never actually turns off or disconnects the power source. Turning this unit off disables the charger, boost regulator and the audio amplifier. The demo unit will enter into a very low power mode monitoring the USB port for power and the on/off switch for changing states. If power is supplied to the micro-USB port the unit will enter into charge mode. The LEDs will ramp from the bottom moving up to the LED that represents the charged state of charge (SOC). When the unit is on and power is supplied to the USB port, the unit will operate using power path technology that will charge the battery and power the unit at the same time.

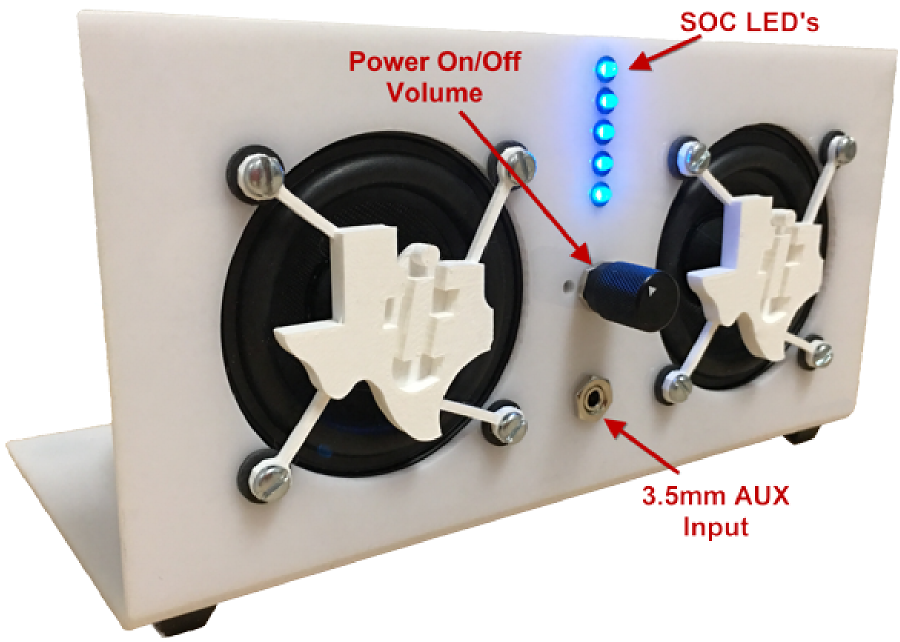


Figure 11. Demo Front View Markup

In the normal operating mode and while not charging, the LEDs will light up showing the SOC of the battery. Each LED represents 20% SOC from 0% to 20%, 20% to 40%, 40% to 60%, 60% to 80%, and 80% and 100%. When the SOC falls below 40% and the battery voltage goes below 3.4-V DC the audio will automatically begin reducing to prevent clipping and distortion as the battery begins a complete discharge. To better understand how this feature works, review Section 2.2. To charge, plug in a micro-USB cable to the input port on the board. This unit will charge at the appropriate rate based on the power that is available for the power source. Any standard USB output will work up to 2.5 A. To play audio, connect a 3.5-mm stereo auxiliary cord to the source and plug the other end into the portable amplifier AUX input. Set the output audio level of the source to max (do not exceed 2 Vp-p) and control the audio output level with the volume control on the front panel.

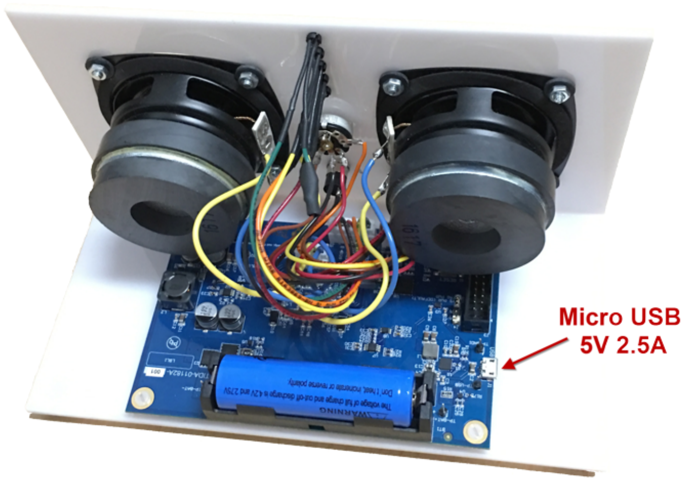


Figure 12. Demo Rear View Markup

2.2 Programming

The code was written in C using Code Composer Studio™ (CCS) and the MSP430 Driver library.

NOTE: No actual code will be provided for this TI Design.

This design guide provides a flow diagram and explains how the code was written. Some examples of the code are provided for reference only for areas that have been determined to be tricky or difficult. All of the following examples are for reference only. Choosing how to write the code may be different from these examples, but the process will be the same.

2.2.1 Using I²C To Write Data To The Gauge

The gauge uses 16-bit values internally. I²C uses 8-bit values to send and receive bytes. To write a byte to the gauge, separate the data value into a high byte and a low byte, then call the gauge_write_reg_1byte (G_REG_ADDR, G_REG_DATA_1) function as shown in Figure 13. Write the msb byte first, then the lsb byte second. Follow the instructions in the datasheet for details on reading and writing to the gauge. The gauge_write_reg_1byte(G_REG_ADDR, G_REG_DATA_1) function is not part of the driver library but is a standard I²C method to send data.

```
// convert design capacity to two 8 bit bytes to be able to transmit the data
      uint16_t qmax_msb = (qmax_cell_0 & 0xff00) >> 8;
      uint16_t qmax_lsb = (qmax_cell_0 & 0x00ff)     ;

      gauge_write_reg_1byte(0x40, qmax_msb);          // write the qmax msb
      gauge_write_reg_1byte(0x41, qmax_lsb);          // write the qmax lsb

//****************************************************************************
// Gauge Comms write address and one byte of data
//****************************************************************************
   // 1 data bytes write register routine for the gauges
   void gauge_write_reg_1byte(G_REG_ADDR, G_REG_DATA_1)
   {
   // Get the I2C port ready to send data
      init_I2C_TX();
   // specify slave address
      EUSCI_B_I2C_setSlaveAddress(EUSCI_B0_BASE, 0x55);
   // Set to transmit mode
      EUSCI_B_I2C_setMode(EUSCI_B0_BASE, EUSCI_B_I2C_TRANSMIT_MODE);
   // Enable I2C Module to start operations
      EUSCI_B_I2C_enable(EUSCI_B0_BASE);
   // Delay before this transaction
      __delay_cycles(100);
   // sends the first byte of the packet
      EUSCI_B_I2C_masterSendMultiByteStart(EUSCI_B0_BASE, G_REG_ADDR);
   // sends the next byte in the packet
      EUSCI_B_I2C_masterSendMultiByteNext(EUSCI_B0_BASE, G_REG_DATA_1);
   // ends the packet
      EUSCI_B_I2C_masterSendMultiByteStop(EUSCI_B0_BASE);
      __delay_cycles(100);
   }
```

Figure 13. Code Using I²C to Write Data to Gauge

2.2.2 Using I²C to Read Two Bytes From the Gauge

The same as in writing to the gauge, the data value are 16-bit values for reading as well. So, read the entire data value as two 8-bit values (msb and lsb) from the gauge, combine these values into one 16-bit value, then return the value to the calling code section.

```
// get state of charge
StateOfCharge = gauge_read_16_reg_LSB(0x1C);
//*************************************************************************
// read two bytes from the gauge and combine them for a 16bit value
//*************************************************************************
uint16_t gauge_read_16_reg_LSB(G_REG_ADDR)  // 2 byte read register routine for the
gauges
{
      uint16_t temp_read;
      uint8_t temp_read_l;
      uint8_t temp_read_h;
// get the I2C port ready to transmit
      init_I2C_TX();
// specify the slave address
      EUSCI_B_I2C_setSlaveAddress(EUSCI_B0_BASE, 0x55);
// Set to transmit mode
      EUSCI_B_I2C_setMode(EUSCI_B0_BASE, EUSCI_B_I2C_TRANSMIT_MODE);
// Enable I2C Module to start operations
      EUSCI_B_I2C_enable(EUSCI_B0_BASE);

      __delay_cycles(100);
// Send single byte data.
      EUSCI_B_I2C_masterSendSingleByte(EUSCI_B0_BASE, G_REG_ADDR);

      __delay_cycles(100);
// get the I2C port ready to receive
      init_I2C_RX();
// specify the slave address
      EUSCI_B_I2C_setSlaveAddress(EUSCI_B0_BASE, 0x55);
// Set to receive mode
       EUSCI_B_I2C_setMode(EUSCI_B0_BASE, EUSCI_B_I2C_RECEIVE_MODE);
// Enable I2C Module to start operations
       EUSCI_B_I2C_enable(EUSCI_B0_BASE);
// Clear the interrupt
      EUSCI_B_I2C_clearInterrupt(EUSCI_B0_BASE, EUSCI_B_I2C_RECEIVE_INTERRUPT0 +
EUSCI_B_I2C_BYTE_COUNTER_INTERRUPT);

      __delay_cycles(100);
// I2C start condition
      temp_read_l = EUSCI_B_I2C_masterReceiveSingleByte(EUSCI_B0_BASE);
// I2C start condition
      temp_read_h = EUSCI_B_I2C_masterReceiveSingleByte(EUSCI_B0_BASE);

      __delay_cycles(100);
// combine the msb and lsb
      temp_read = ((temp_read_h << 8) | temp_read_l);
// return the value
      return(temp_read);
}
```

Figure 14. Code Using I²C to Read Two Bytes From Gauge

2.2.3 Using I²C to Write Data to the DAC

The DAC requires that the 8-bit value to set the DAC to needs to be inserted into the middle of a 16-bit value. Figure 15 shows a completed example. The MSP430 ADC is a 10-bit value; this value is averaged over several samples to prevent it from bouncing around. The average value is sent to the DAC_write_reg (volume_average) function.

```
// call the DAC write routine and send the averaged volume to the DAC
    DAC_write_reg(volume_average);
//*****************************************************************************
// DAC Comms  //DAC output value
//*****************************************************************************
void DAC_write_reg(DAC_value)
{
// Get the I2C port ready to send data
      init_I2C_TX();
// take the 10bit raw adc value, shift right to make it 8bit. Shift left 4 bits to
create the 16bit I2C value
      set_DAC_value = (DAC_value >> 2) << 4;
// msb - convert the new 16bit with/volume into two 8bit bytes to be able to
transmit the data
      dac_msb = (set_DAC_value & 0xff00) >> 8;
// lsb
      dac_lsb = (set_DAC_value & 0x00ff)      ;
// specify the slave address
      EUSCI_B_I2C_setSlaveAddress(EUSCI_B0_BASE, DAC_SLAVE_ADDRESS);
// Set to transmit mode
      EUSCI_B_I2C_setMode(EUSCI_B0_BASE, EUSCI_B_I2C_TRANSMIT_MODE);
// Enable I2C Module to start operations
      EUSCI_B_I2C_enable(EUSCI_B0_BASE);
// Delay before this transaction
      __delay_cycles(100);
// begins the packet by sending the slave address
      EUSCI_B_I2C_masterSendStart(EUSCI_B0_BASE);
// sends the first byte of the packet
      EUSCI_B_I2C_masterSendMultiByteStart(EUSCI_B0_BASE, dac_msb);
// sends the next byte in the packet
    EUSCI_B_I2C_masterSendMultiByteNext(EUSCI_B0_BASE, dac_lsb);
// ends the packet
    EUSCI_B_I2C_masterSendMultiByteStop(EUSCI_B0_BASE);
}
```

Figure 15. Code Using I²C to Write Data to DAC

2.2.4 Reading the Volume Potentiometer and Setting the Auto Volume Control

The first step is to read the voltage drop across the volume potentiometer. This voltage is used to set the audio output level of the Class D amplifier. The raw ADC value from the MSP430 is 10 bits. Average this value, then determine the battery state, and determine what the maximum level the amplifier can be set to in order to prevent clipping and distortion. Then set the DAC level to control the output of the amplifier.

```
//****************************************************************************
// read the volume setting with the ADC (0-1023bits) (0-3.3V)
//****************************************************************************
void read_ADC(void)
{
    int i = 0;          // set to 0
    int sum_array = 0;  // set to 0
    int volume_max = 0;
//Enable and Start the conversion in Single-Channel, Single Conversion Mode
    ADC10_B_startConversion(ADC10_B_BASE, ADC10_B_SINGLECHANNEL);
    __delay_cycles(100);            // wait for conversion
// The latest ADC value as read from the MSP430
    volume_read_new = ADC10_B_getResults(ADC10_B_BASE);
//FIFO averaging filter for the volume ADC_CH1 ------------------------------
      i = 0;            // reset to 0
      sum_array = 0;    // reset the last averaged value to zero to recalculate the
new average value
// Vdc_1_length = the amount of samples to average across, this number will set the
array length
// Vdc_1_array = the size of the array that will store the FIFO values for averaging
      for(i = 0; i < Vdc_1_length; i++) // shift the array as a FIFO
      {
// Makes all the array indexes equal the number after them (Moves all of the values
in the array up one)
            Vdc_1_array[i] = Vdc_1_array[i + 1];
      }
      Vdc_1_array[Vdc_1_length - 1] = volume_read_new;      // add the new value to
the end of the array
      for(i = 0; i < Vdc_1_length; i++)   // sum the array
      {
// add all of the array elements each time you add a new element
            sum_array += Vdc_1_array[i];
      }
// divide the sum of the array to get an average, then divide by 2
      volume_average = ((sum_array / Vdc_1_length)/2);
// If the battery voltage is below the threshold recalculate to get the auto reduced
value
      if (m_Voltage_1 <= 3400)
      {
            volume_max = (((1023-(3400-m_Voltage_1))-24)/2);
// if the actual averaged values is greater than the calculated limit use the limit
            if (volume_average >= volume_max)
            {
                  volume_average = volume_max;
            }
// if the actual averaged value is below the calculated limit then use the averaged
value
      }
// call the DAC write routine and send the averaged volume to the DAC
    DAC_write_reg(volume_average);
      __delay_cycles(10);
}
```

Figure 16. Code Reading Volume Potentiometer and Setting Auto Volume Control

2.2.5 Flowchart

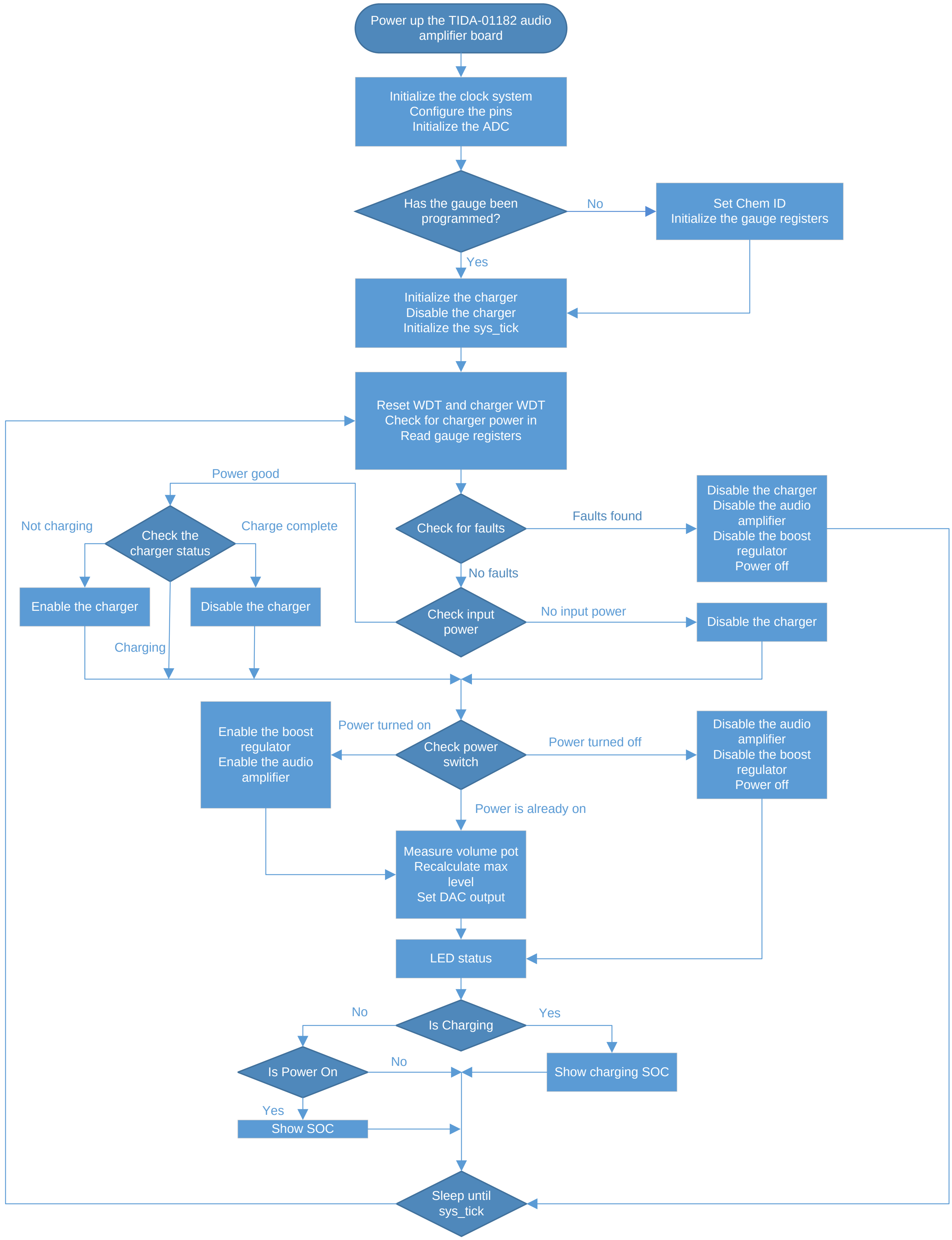


Figure 17. Code Flowchart

3 Testing and Results

Figure 18 shows the TIDA-01182 test bench.

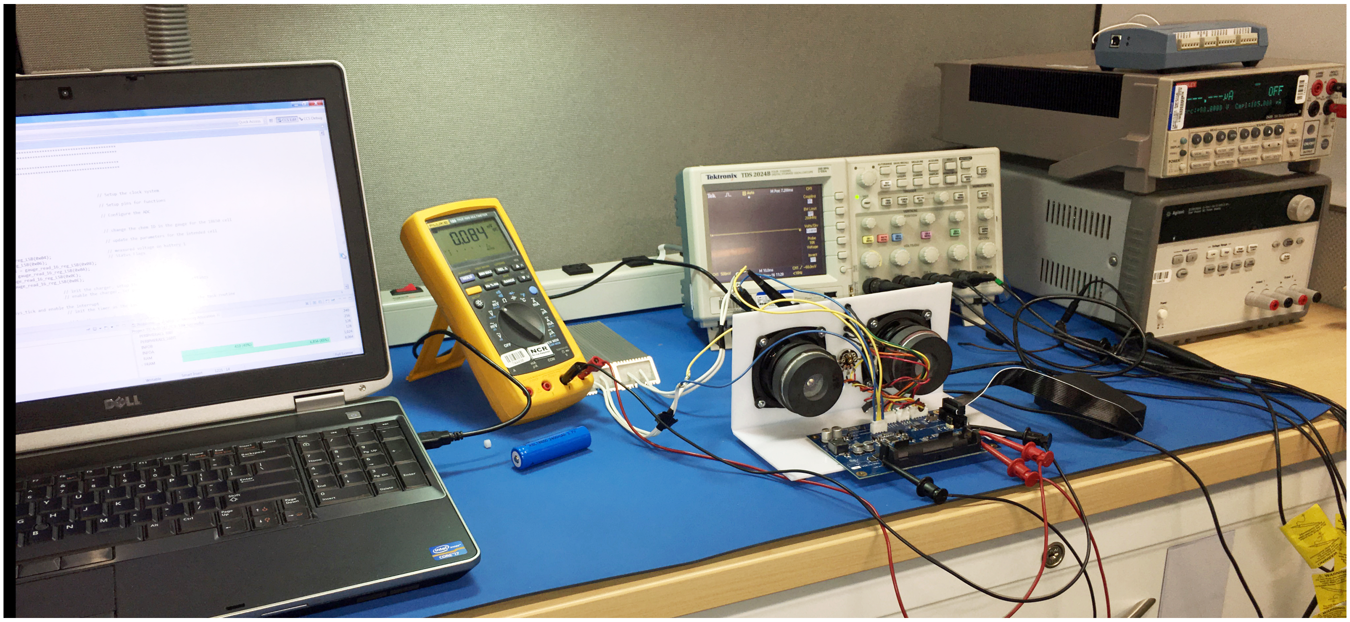


Figure 18. Test Bench

The test bench setup includes the following:

- Dell® laptop running CCS
- MSP-FET Flash Emulation Tool
- Tektronix® TDS 2024B Oscilloscope
- Fluke™ 189 Multimeter
- Keithley™ 2812 SourceMeter®
- BK Precision 8502 Electronic Load
- Agilent™ E3649A Power Supply
- Flir™ i7 Thermography Camera
- Measurement Computing™ USB-2408 Data Acquisition Module
- LMEM™ MMML100W 8-Ω Load, Resistive Load
- Nady ASM-2 Analog Sound Pressure Level Monitor

Figure 18 is a picture of the basic test bench setup. High-power 8-Ω resistive loads were used on the speaker outputs for most of the testing to minimize the noise during test. Once the power section was verified, the balance of the tests were completed using a battery. The boost, audio amplifier, and power path tests were tested using a battery.

3.1 CCS Variable Data

Most of the test results for the gauge and charger were provided by the CCS software while writing the code and debugging the firmware. Create a list of variables to be read from the gauge IC and a set of flag variables to be read from the charger, read the gauge and charger, and then displayed the results in these lists. See Figure 19 for a preview of the list variables that were used while testing the TIDA-01182.

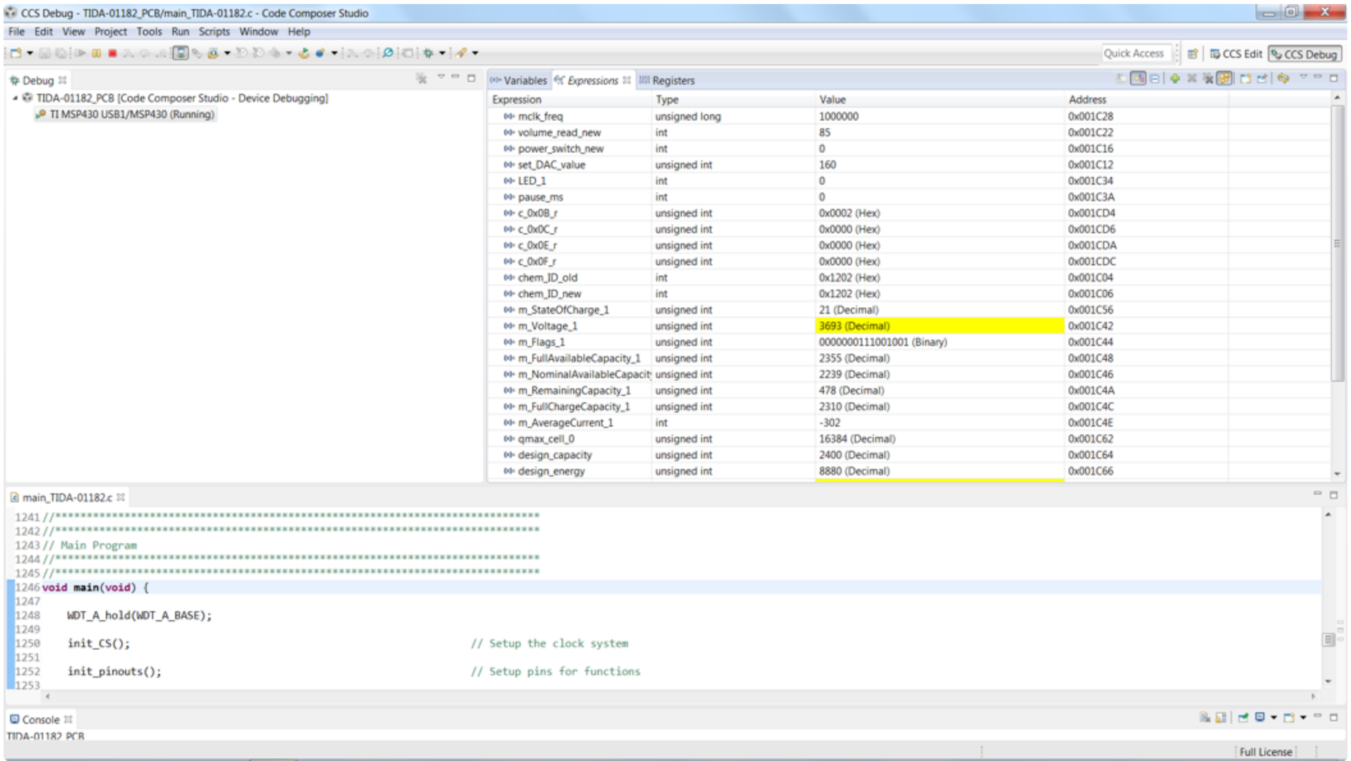


Figure 19. CCS Screenshot Variables

Figure 20 is a picture of the basic debugging test setup used with CCS to write the test code.

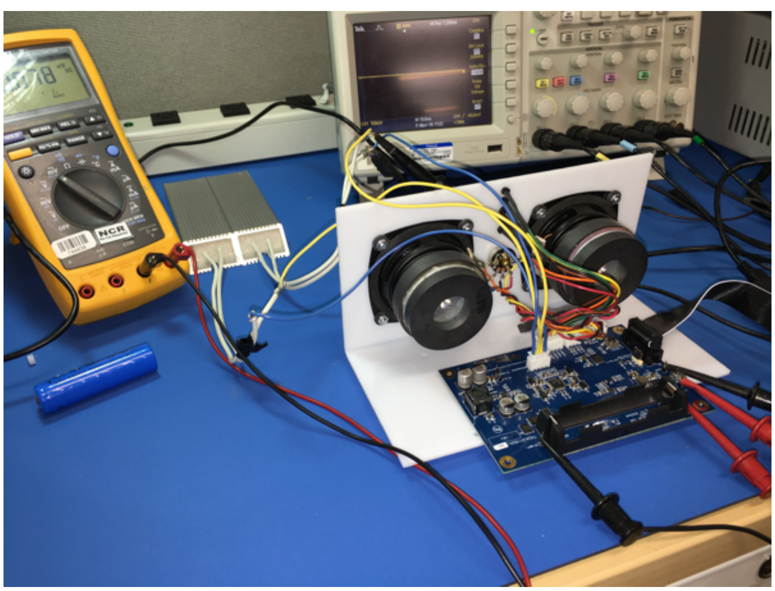


Figure 20. TIDA-01182 Test Bench Debug

3.2 Scope Shots

3.2.1 Boost Regulator

The boost regulator output was clean and varied by about ±4% across several units. In the scope shot in Figure 21, channel 1 in yellow is the boost regulator DC image and channel 2 is the same boost regulator using an AC view to look at the output ripple. The ripple was better than the expected 20 mV, providing a quiet clean audio power source. The peak-to-peak spikes were developed from the current that was sources to the Class D amplifier showing a good load regulation.

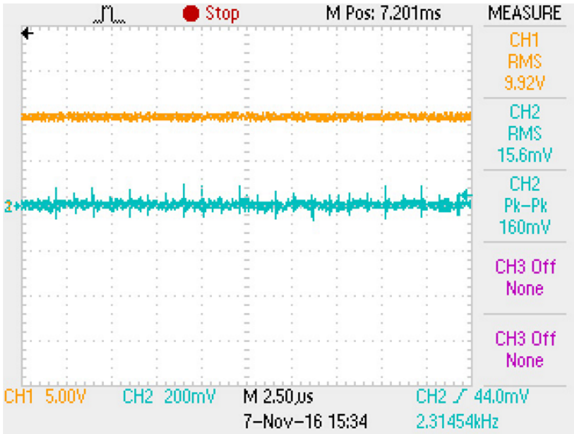


Figure 21. Boost Regulator Output

3.2.2 Class D Output

The output of the Class D amplifier when referenced to ground looks like a 1-MHz square wave. The overshoot spikes are from the inductance of the traces on the PCB and the wires to the load resisters.

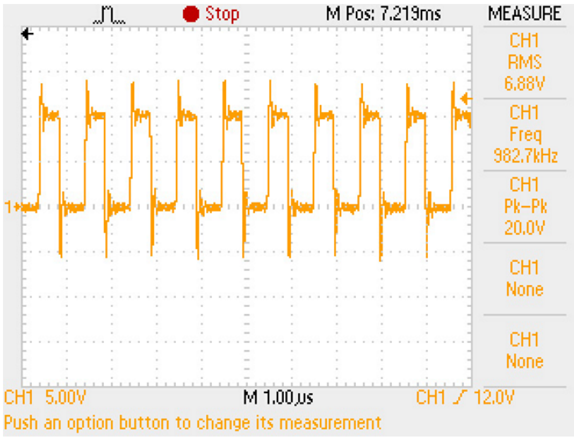


Figure 22. Class D Output Ground Reference

Figure 23 is a view of the Class D amplifier when connected differentially across the speaker. The square wave is gone and only a few high frequency spikes are present. These spikes are from the overshoot and ringing that occurs during the rising and falling edges of the Class D square waves. At 1 MHz these spikes are not audible, but may cause radiated EMI. Pay extra attention to the board layout to minimize radiated EMI.

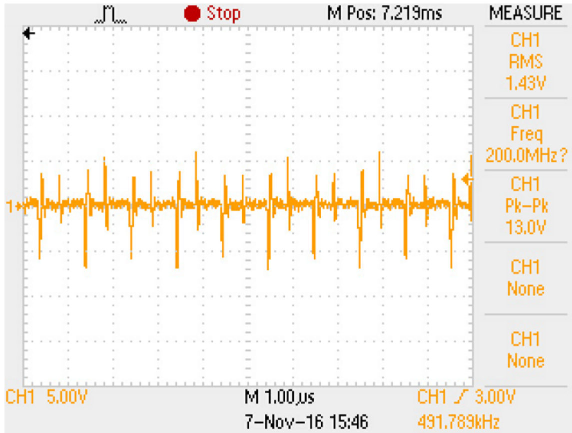


Figure 23. Class D Output Differential

If the traces on the PCB are kept short and the wires to the speakers are kept to a minimum, then the standard ferrite bead and capacitor filters are sufficient to keep the noise to a minimum and prevent unwanted radiated emissions as in Figure 24.

Figure 24. Class D Differential FFT Output

3.2.3 Charge Cycle

This charge cycle used a 1-A power supply. The charger opted to charge at 800 mA based on the voltage drop on the output of the wall power supply. The charge time was 1.6 hours to a full charge. The charge voltage for this battery is set to 4.2-V DC. The max charge current is 2.4 A. The charge termination is set to 240 mA and terminated as expected.

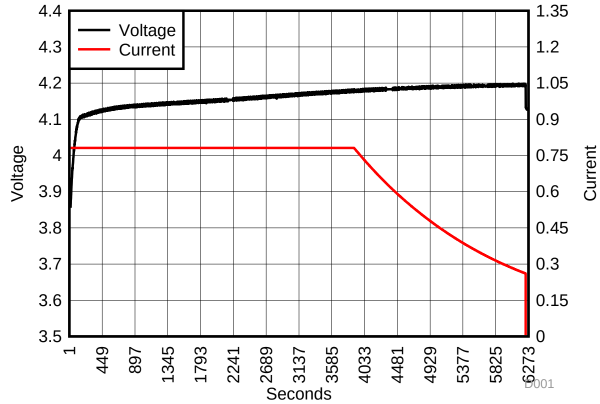


Figure 25. Charge Cycle Plot

3.2.4 Thermal Results

The charger thermal profile shows good thermal dispersion and a sufficient amount of heat dissipation. While charging at 2.5 A, the charger maintained a well-controlled thermal profile. The extra copper used in the flooded ground planes helps to provide the ability to sink the heat but also to spread the heat evenly throughout the PCB. The extra copper on the top and bottom layers also help to dissipate the heat so it does not build up in the circuit board.

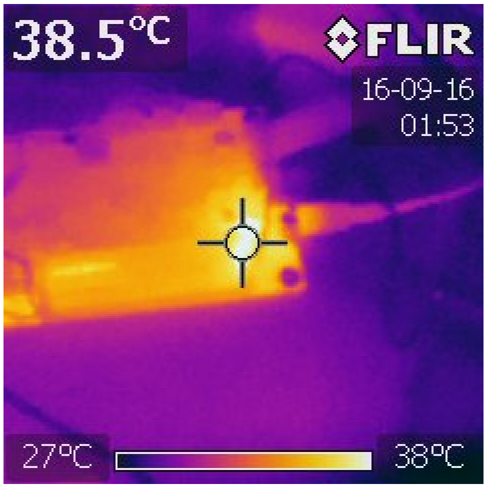


Figure 26. Charger Thermal

The Class D amplifier is very efficient and produces minimal heat in the actual IC. The separation of the power and analog grounds did not create a localized hot spot due to poor thermal dispersion. The boost regulator saw a small amount of localized heating. The ground planes were good at drawing the heat from the IC, thereby preventing thermal stress on the IC. Overall, the thermal dispersion and heat dissipation was very good on all parts of the circuits. There were no excessive heat spots and all circuits operated well within thermal limits.

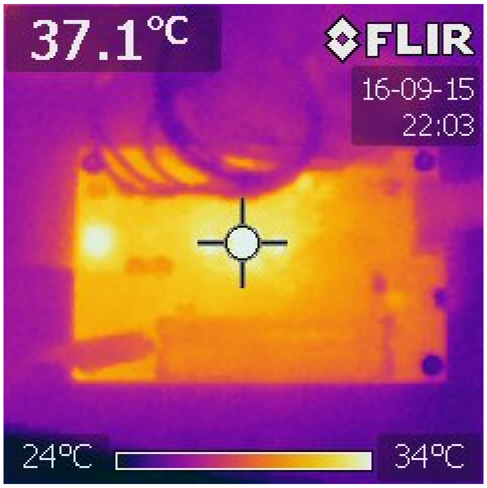


Figure 27. Boost and Class D Thermal

4 Design Files

4.1 Schematics

To download the schematics, see the design files at TIDA-01182.

4.2 Bill of Materials

To download the bill of materials (BOM), see the design files at TIDA-01182.

4.3 PCB Layout Recommendations

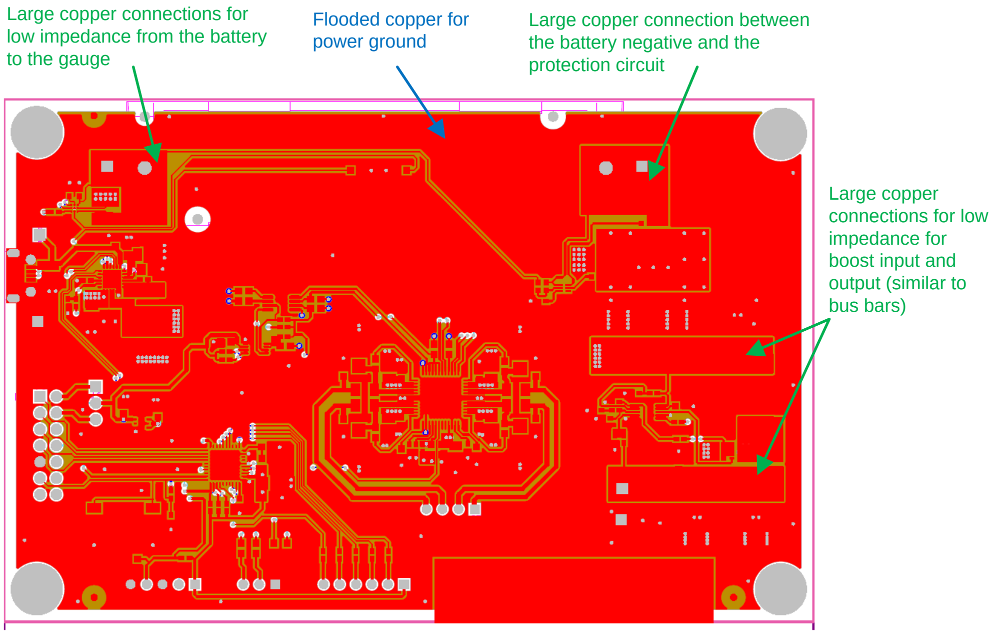


Figure 28. PCB Top Layer 1 (Component Layer) Layout

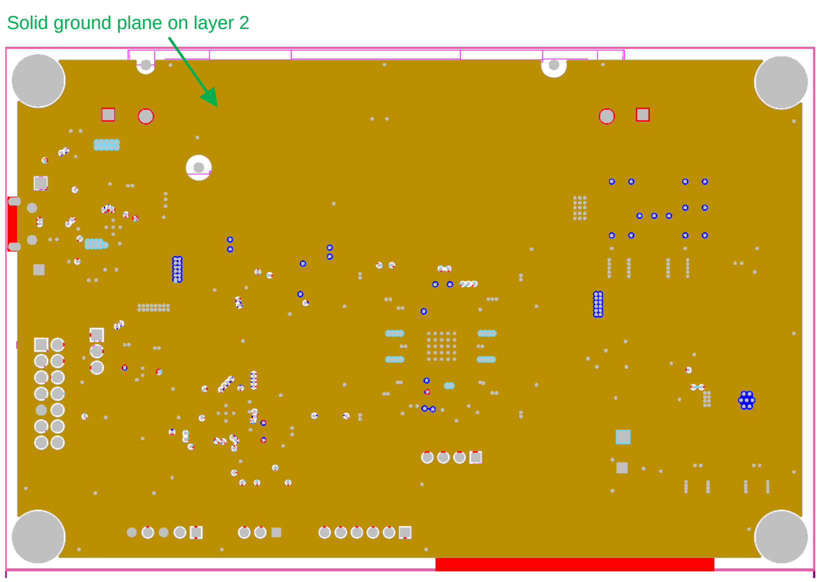


Figure 29. PCB Inner Layer 2 (Ground Plane Layer) Layout

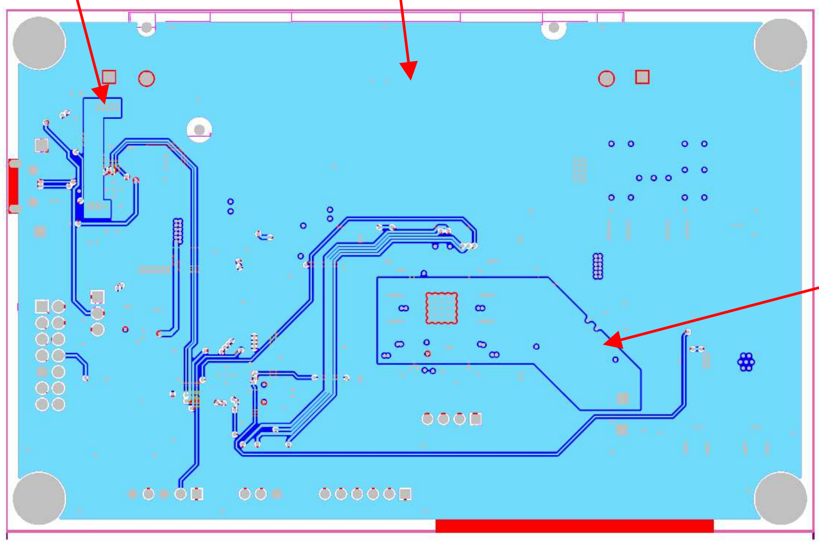


Figure 30. PCB Inner Layer 3 (Power and Ground Plane Layer) Layout

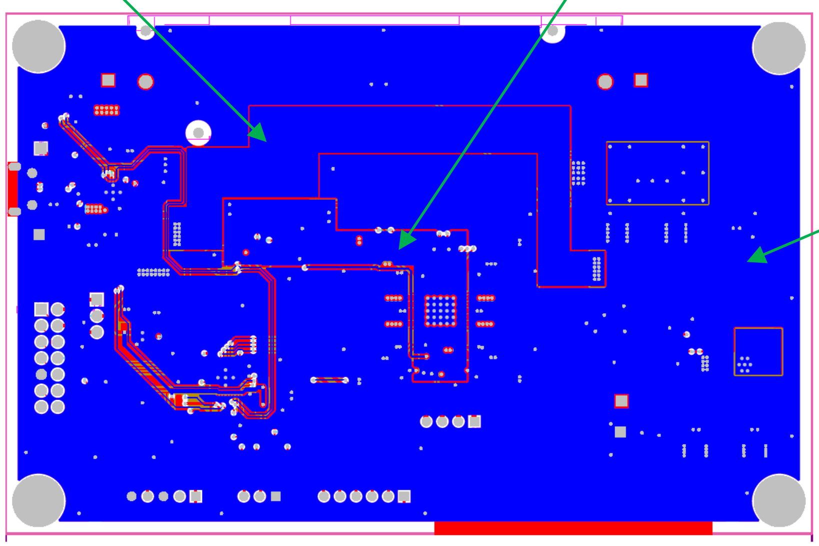


Figure 31. PCB Bottom Layer 4 (Analog Ground and Ground Plane Layer) Layout

4.3.1 Layout Prints

To download the layer plots, see the design files at TIDA-01182.

4.4 Altium Project

To download the Altium project files, see the design files at TIDA-01182.

4.5 Gerber Files

To download the Gerber files, see the design files at TIDA-01182.

4.6 Assembly Drawings

To download the assembly drawings, see the design files at TIDA-01182.

5 About the Author

GORDON VARNEY is a Senior Systems and Applications Designer at TI, where he is responsible for developing reference design solutions and demos for the BMS (Battery Management Solutions) Group. Gordon brings to this role his extensive experience in battery management, power, analog, digital, microcontrollers, and energy harvesting. Gordon is an expert in circuit design and board layout as well as programming in several languages. Gordon earned his bachelor of science in electrical engineering (BSEE) from KWU in 1989 and is a member of the Institute of Electrical and Electronics Engineers (IEEE) and the Society of Automotive Engineers (SAE). Gordon is on the Board of Advisors for UTA's Engineering Department and has lectured more than a dozen times at several Colleges and Universities in his 27 years as an engineer.