CSM-BMS-SPI

User Guide

Current-Sensor-Module Demonstration Kit for Automotive BMS Applications

Overview

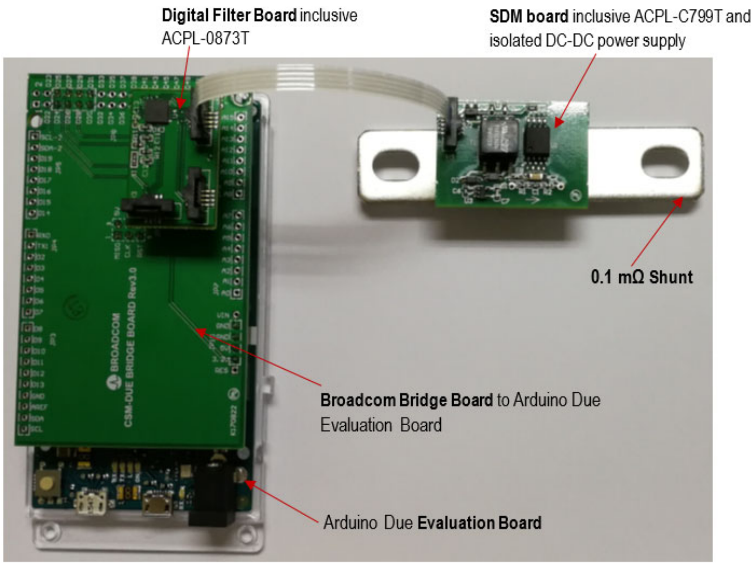
The Current-Sensor-Module Demonstration Kit (CSM-BMS-SPI) is designed to evaluate Broadcom digital filter ACPL-0873T and isolated Sigma-Delta Modulator ACPL-C799T with input voltage from an external shunt resistor and then output ADC data to the user's controller MCU, DSP, or microprocessor. The kit is specialized for the BMS (Battery Management System) battery pack's current measurements.

The Demonstration Kit mainly consists of the following items:

- Isolated Sigma-Delta Modulator (SDM) Board with ACPL-C799T
- Digital Filter (DF) Board with ASIC ACPL-0873T
- Broadcom Bridge Board to Arduino Due Evaluation Board

The key features of the Demonstration Kit are as follows:

- Programmable Digital Filter Mode
- Fast Over-range Detection
- Offset Calibration
- SPI Interface ADC Data Output
- Sensing range up to 500A current together with a 0.1 mΩ shunt, or a sensing range of up to 1000A current together with a 0.05 mΩ shunt



Ordering Information

The ordering code is W1202-475318 (CSM-BMS-SPI). Contact Broadcom sales or an authorized distributor for prices.

NOTE: Broadcom does not provide shunt products, but may offer shunt samples to customers by request until the limited stock is finished

Demonstration Kit Connections

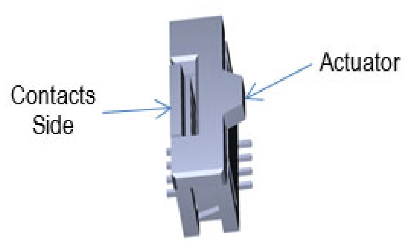
Refer to Evaluation Board Circuit Schematic for a detailed schematic.

Connection Map Between the Digital Filter Board and the SDM Board Using an FPC Cable

	SDM Board	Digital Filter
Signal	**H5**	**H1**
VDD	4	1
MCLK	3	2
MDAT	2	3
GND	1	4

FPC Connector

Same-side-contacts and 1-mm pitch Flexible-Printed-Circuit (FPC) cables are used to connect the Digital Filter Board and the Sigma-Delta Modulator Board. Lift the connector's actuator up before inserting or removing the cable. Press down on the actuator after inserting the cable into the connector.



Connection Map between the Digital Filter Board and the CSM-DUE Bridge Board

	Digital Filter Board	CSM-DUE Bridge Board		
Signal	**P2**	**P1**	**Pin Name**	**Connector**
/CS	1	1	D24	JP8
/INT	2	2	D26	JP8
SCLK	3	3	SCLK	P1
/RST	4	4	D28	JP8
MISO	5	5	MISO	JP1
OC	6	6	D27	JP8
MOSI	7	7	MOSI	JP1
DR	8	8	D25	JP8
GND	11	11	GND	JP10, JP1
VDD	12	12	3.3V	JP10

When other MCU evaluation boards are used to replace the Arduino Due, connect the respective MCU board I/O port to Digital Filter Board connector P2. A 3.3V power supply is required. See the following section for more information.

Power Supply VDD Requirements for the DF Board and the SDM Board

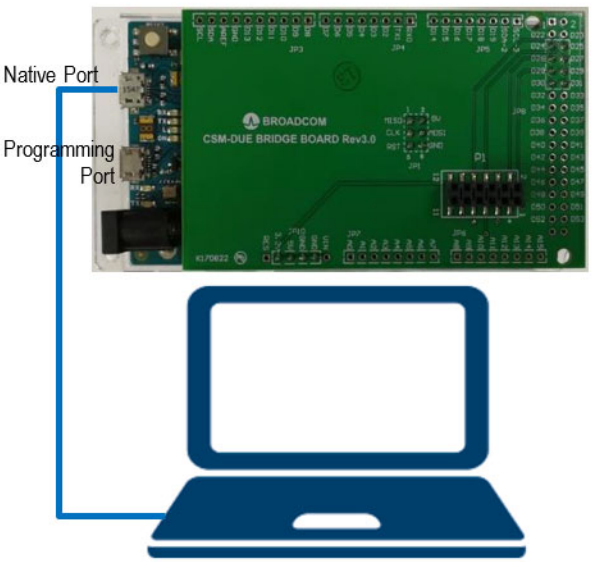
- Voltage: 3.3 VDC +/- 5%
- Current Minimum: 60 mA

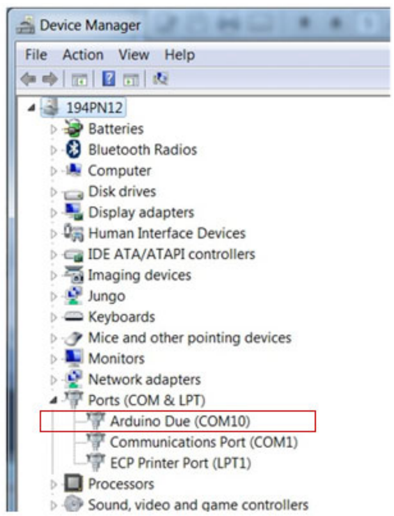
Software Installation and Operation

Installing the Arduino Due Board USB Driver

To install the Arduion Due Board USB driver, complete the following steps:

1. Install the Arduino Due board USB driver located in the Arduino IDE from https://www.arduino.cc/en/Main/Software. More technical information about the Arduino Due board is available at https://www.arduino.cc/en/Main/ArduinoBoardDue.
2. During Arduino IDE installation, connect the Micro-USB cable from the computer to the Arduino Due board Programming Port.
3. After Arduino IDE installation, connect the Micro-USB cable from the computer to the Arduino Due board Native Port.
4. Verify that the driver was successfully installed on the computer by navigating to **Control Panel > Device Manager > Ports (COM & LPT)**.





Broadcom GUI Software Requirements

To run the Broadcom GUI, the user's computer must meet the following requirements:

- Windows OS 7.0 or later.
- A display resolution of 1920 X 1080 pixels or greater.

The user must also copy the demonstration software, `DigitalFilter v1.5b.exe`, from a CD-ROM to the computer local drive.

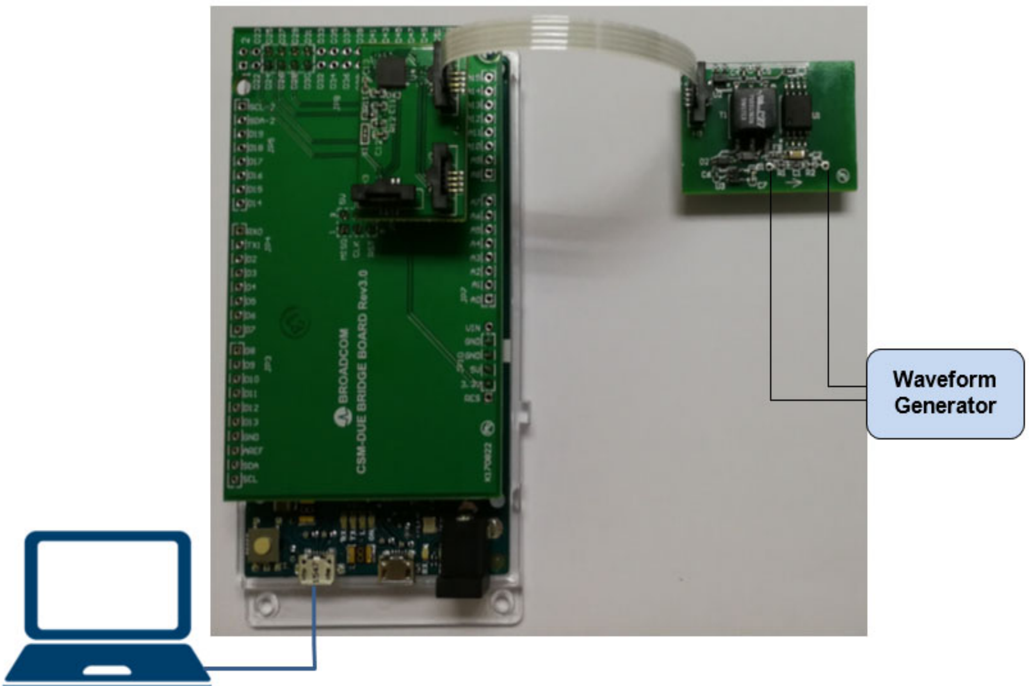
NOTE: Contact Broadcom local sales for the demonstration software.

Bench Verification of the GUI Software and Demonstration Kit Functions

Bench Verification

To perform bench verification, complete the following steps:

1. Connect a 1Ω-resistor as a shunt at the channel-1 ACPL-C799T input.
2. Send a General Waveform Generator signal to the 1Ω-resistor.
3. Connect the computer's USB port to the Arduino Due Board's Native Port.



Example of Saved Log Data

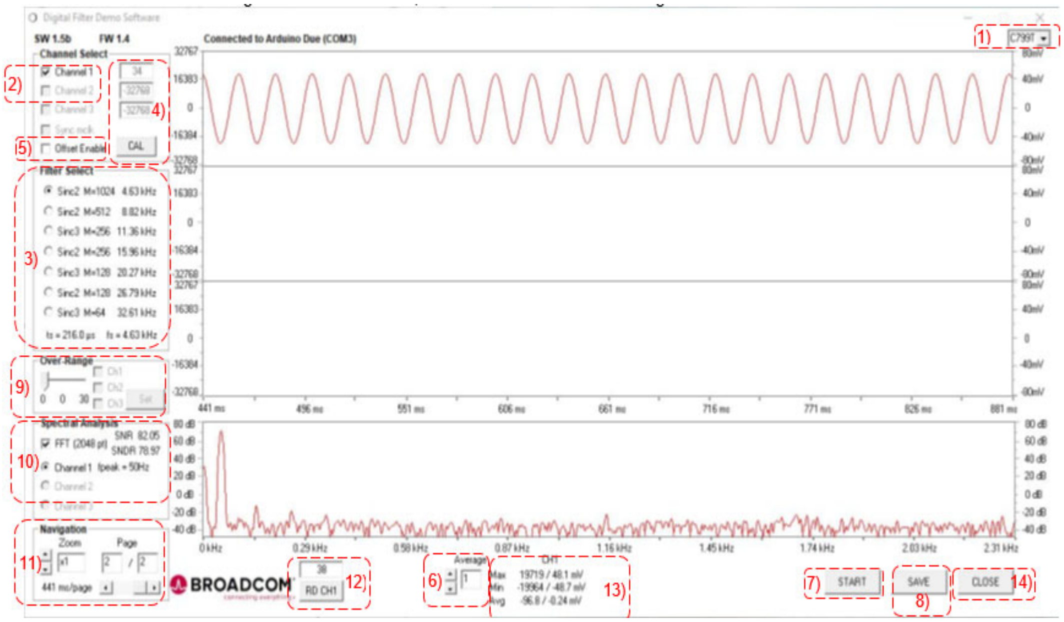
Channel-1 ADC Data in unsigned 16-bit decimal counts

Channel-1 ADC Fast-Fourier-Transformation (FFT) Data using one-sided power spectrum.

	Digital Filter Data Log SW 1.5b					
	ts = 88.0 µs fs = 11.36 kHz	Sinc3 M=256 F_CLK = 10 MHz	Sync off	Page 1/1	4080 Samples	
Chan 1 - Offset = 66	Chan 2	Chan 3	PS Chan 1 - peak 360 (999Hz)	PS Chan 2	PS Chan 3	Frequency(Hz)
53320			-105.276211			0
50395			-103.564487			2.8
42215			-104.075176			5.5
31214			-101.81834			8.3
20677			-98.867275			11.1
13747			-97.050011			13.9
12483			-97.430082			16.6
17263			-99.023142			19.4
26672			-98.255264			22.2
37894			-97.430813			25

GUI Software Operation

To open the GUI, double click on `DigitalFilter v1.5b.exe`. The following image shows the GUI.



To configure the GUI, complete the following steps:

1. Select the Sigma-Delta Modulator C799T mounted on the SDM Board.
2. Select applied input Channel 1. (Only SDM Channel 1 is connected.)
3. Select the Digital Filter Mode. For BMS battery current sensing, the Sinc2 M = 1024 filter mode is recommended.
4. Enter the Offset calibration. Before clicking on **CAL** to capture input offset value, short the Sigma-Delta Modulator C799T input (that is, remove the Waveform Generator source cable and use a wire to short the 1Ω resistor).
5. Enable or disable Input Offset Calibration as necessary.
6. Select a post-ADC Averaging interval from 1 to 100.

   Example: If 50 is selected, 50 ADC are averaged by the GUI software. This means that the final sampling time is multiplied by the averaging interval x50, and the final data output rate is divided by averaging interval
   4.63 kHz / 50 = 92.6 Hz. When selecting 1, there is no post-ADC averaging, and raw ADC data is directly outputted.
7. Start or stop capturing data as necessary. The Digital Filter output ADC data reconstruct input signal is displayed in the GUI window.
8. Save the last captured data batch to the local drive. See Example of Saved Log Data for an example of the generated data.
9. Set the Over-Range value for Channel 1.
10. Turn the FFT display On or Off, then select display Channel 1. The FFT data is only valid when input signal is fixed frequency sinusoidal waveform within input range.
11. Set the Zoom level for the waveforms displayed.
12. Read the single ADC output data for Channel 1.

13. Read the displayed Statistic Maximum, Minimum, and Average values of the ADC data pages.
14. Close the software.

NOTE:

- The ADC data shown in the GUI is bipolar 16 bit.
- If a channel ADC output is –32768 or 32767 continuously, there may be an input open.
- Post-ADC Averaging can suppress high-frequency AC noise from the input current or voltage signal by trading off the sampling rate. Post-ADC Averaging doesn't affect Over-Current (OC) output speed.

SPI Communication Software Implementation and Practice with the Arduino Due Board

The Arduino Due is configured in SPI Master Mode (SPI Mode 0 MSB first). The CS Pin is used for filter conversion start. The CS Pin is also used by SPI Chip Select to read and write registers. When CS is low, the registers on the digital filter can be read and written without having to toggle CS. This allows the Data Ready (DR) status on the ACPL-0873T Interrupt Status Register (0x02) to be read without restarting the filter conversion.

Sampling Methods

There are three main methods for reading data from the digital filter.

Method 1: Poll DR Status (Using a Timer)

Since the filter conversion time is known, a timer can be set to poll the DR status in Register 0x02 so that tasks can be done in between reads. Polling of the DR status is started just before the data is ready.

Method 2: Interrupt on DR Status on INT Pin

The ACPL-0873T Interrupt Enable Register (0x03) can be configured to output the DR status to the INT Pin, and the Arduino Due can be configured to interrupt on the INT pin H→L. The DR status is cleared when the Interrupt Status Register (0x02) is read, so the Interrupt Status Register needs to be read after the data has been read out from the filter in order to manually clear the DR status.

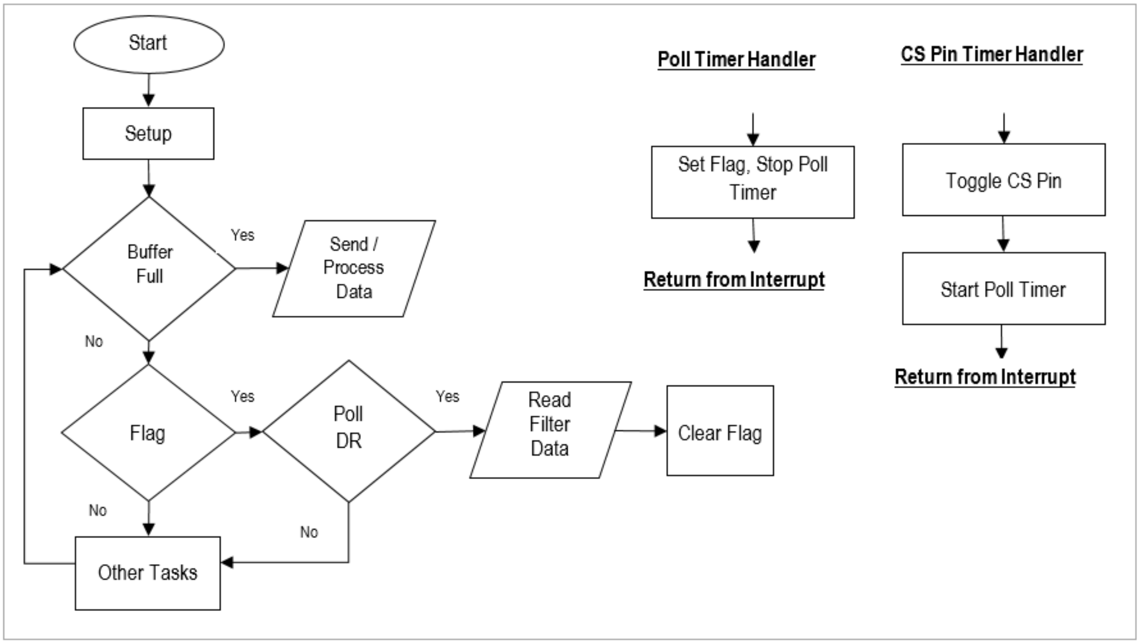
Method 3: Interrupt on DR Pin Signal

The DR pin signal is cleared automatically when CS is L→H, so the Arduino Due can be configured to interrupt on the DR pin signal L→H instead of using the DR status through the INT pin. The data can then be read out before the filter conversion is restarted by toggling CS, which clears the DR pin signal.

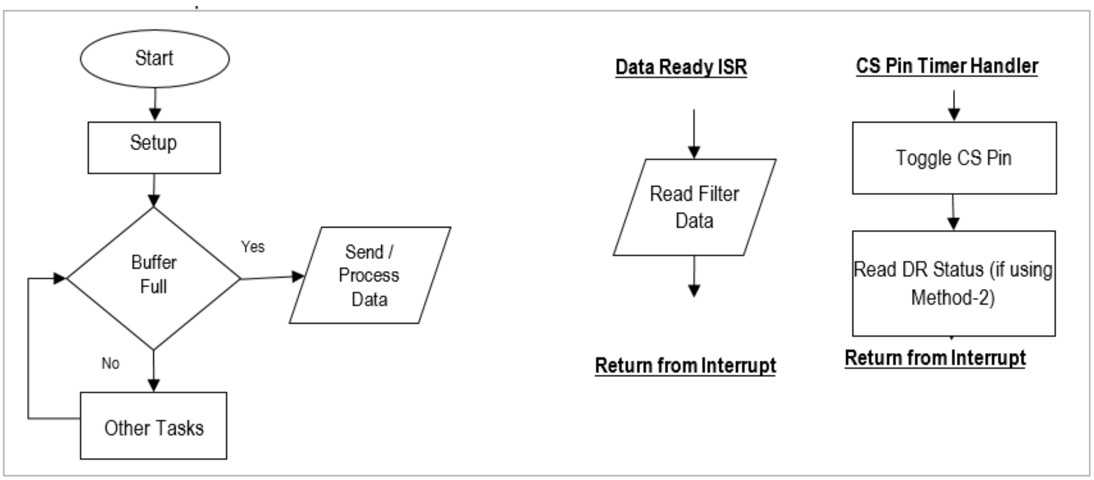
To achieve a constant sample rate, the CS pin can be toggled using either a timer or a PWM signal, but sufficient time must be allowed so the data can be sampled before the CS pin is toggled.

Flow Charts

Method 1: Polling Flow Chart



Methods 2 and 3: Interrupt Flow Chart



Sample Code

Timer Setup

```
// Delay = rc / (MCLK / 128)
void startTimer (Tc *tc, uint32_t channel, IRQn_Type irq, uint32_t rc) {
    pmc_set_writeprotect (false);
    pmc_enable_periph_clk ((uint32_t) irq);
    TC_Configure (tc, channel, TC_CMR_WAVE | TC_CMR_WAVSEL_UP_RC | TC_CMR_TCCLKS_TIMER_CLOCK4);
    TC_SetRA (tc, channel, rc >> 1);
    TC_SetRC (tc, channel, rc);
    TC_Start (tc, channel);
    tc->TC_CHANNEL[channel].TC_IER = TC_IER_CPCS;
    tc->TC_CHANNEL[channel].TC_IDR = ~TC_IER_CPCS;
    NVIC_EnableIRQ(irq);
}
void stopTimer(Tc *tc, uint32_t channel, IRQn_Type irq) {
    TC_Stop (tc, channel);
    NVIC_DisableIRQ (irq);
}
```

Method 1: Poll DR Status

```
void setup_polling() {
   // Set flags
   flag = false; obuffer = buffer = 0; counter = 0;
   // Start transfer
   setCS(); clearCS();
   // Start Poll timer
   startTimer (TC1, 1, TC4_IRQn, delay_poll [filter_type]);
   // Start CS Pin timer
   startTimer (TC1, 0, TC3_IRQn, delay_cs [filter_type]);
   // Read interrupt status register to clear DR
   spiTransfer (0x92);
   spiTransfer (0x00);
}

// Main Loop
if (flag)
{
    // Read interrupt status register to check DR
    spiTransfer (0x92);
    if ((spiTransfer (0x00)) & 0x01)
    {
        // Read 2 bytes to buffer
        spiRead ((uint8_t*) &spi_data [buffer] [counter], 2);
        // Increment counter and wrap when buffer full
        counter = (counter + 2) * ((counter + 2) < buffer_size);
        buffer = (buffer + (counter == 0)) & 0x01;
        // Clear flag
        flag = false;
    }
}
```

```
// CS Pin Timer Handler
void TC3_Handler() {
   TC_GetStatus (TC1, 0);
   // Start next transfer
   setCS(); clearCS();
   // Restart poll timer
   startTimer (TC1, 1, TC4_IRQn, delay_poll [filter_type]);
}

// Poll Timer Handler
void TC4_Handler()
{
  TC_GetStatus (TC1, 1);
  // Set flag
  flag = true;
  // Stop poll timer
  stopTimer (TC1, 1, TC4_IRQn);
}
```

Method 2: Interrupt Using DR Status on INT Pin

```
void setup_int_pin() {
   // Set flags
   obuffer = buffer = 0; counter = 0;
   // Start transfer
   setCS(); clearCS();
   // Set interrupt enable register DR_E
   spiTransfer (0xA3);
   spiTransfer (0x01);
   // Read interrupt status register to clear DR
   spiTransfer (0x92);
   spiTransfer (0x00);
   // Enable INT pin interrupt
   attachInterrupt (INT_PIN, data_ready_isr, FALLING);
   // Start CS Pin timer
   startTimer (TC1, 0, TC3_IRQn, delay_cs [filter_type]);
}

// Data Ready ISR
void data_ready_isr() {
   // Read 2 bytes to buffer
   spiRead ((uint8_t*) &spi_data [buffer] [counter], 2);
   // Increment counter and wrap when buffer full
   counter = (counter + 2) * ((counter + 2) < buffer_size);
   buffer = (buffer + (counter == 0)) & 0x01;
}

// CS Pin Timer Handler
void TC3_Handler() {
   TC_GetStatus (TC1, 0);
   // Start next transfer
   setCS(); clearCS();
   // Read interrupt status register to clear DR
   spiTransfer (0x92);
   spiTransfer (0x00);
}
```

Method 3: Interrupt Using DR Pin Signal

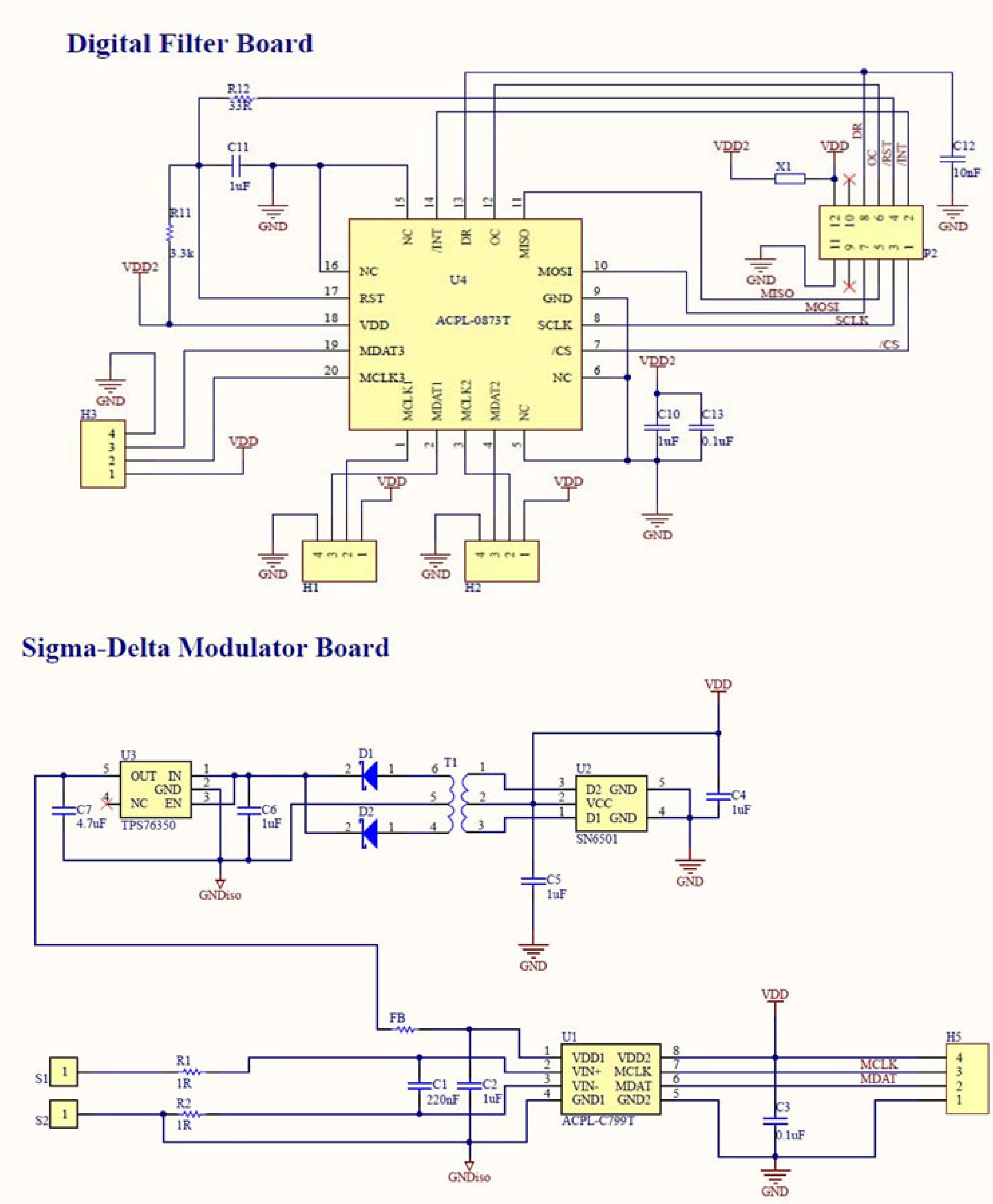
```
void setup_dr_pin() {
   // Set flags
   obuffer = buffer = 0; counter = 0;
   // Start transfer
   setCS(); clearCS();
   // Enable DR pin interrupt
   attachInterrupt (DR_PIN, data_ready_isr, RISING);
   // Start CS Pin timer
   startTimer (TC1, 0, TC3_IRQn, delay_cs [filter_type]);
}

// Data Ready ISR
void data_ready_isr() {
   // Read 2 bytes to buffer
   spiRead ((uint8_t*) &spi_data [buffer] [counter], 2);
   // Increment counter and wrap when buffer full
   counter = (counter + 2) * ((counter + 2) < buffer_size);
   buffer = (buffer + (counter == 0)) & 0x01;
}

// CS Pin Timer Handler
void TC3_Handler() {
   TC_GetStatus (TC1, 0);
   // Start next transfer
   setCS(); clearCS();
}
```

Evaluation Board PCB Design

Evaluation Board Circuit Schematic



Evaluation Board Component List

Sigma-Delta Modulator Board

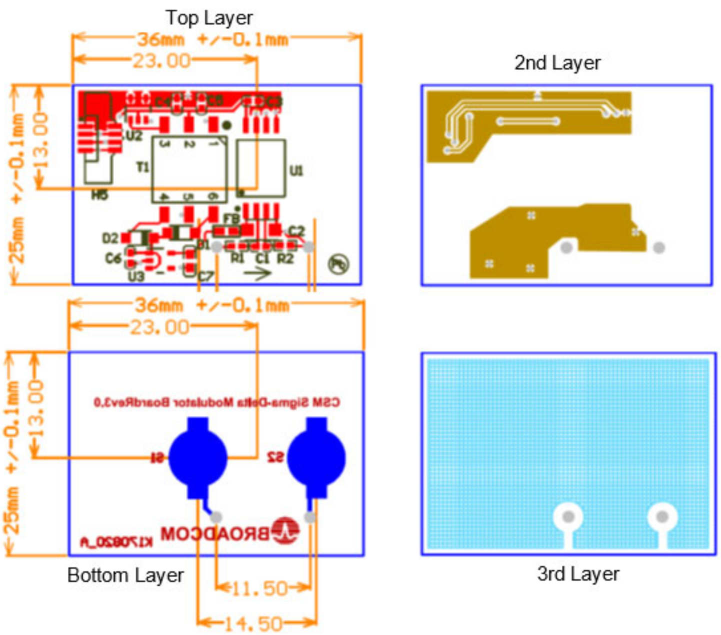
Land Pattern	Value	Designator	Description
C0402	220 nF	C1	C0402 220 nF /16V 10% X7R
C1206	1 µF	C2	C1206, 1 µF, 50 V, ± 0%, X7R
C0402	0.1 µF	C3	C0402, 0.1µF, 16V, X7R, +/-20%
C0402	1 µF	C4, C5, C6	C0402, 1 µF, 16 V, ±10%, X5R
C0603	4.7 µF	C7	C0603, 4.7 µF, 16V, ±10%, X5R
SOD-123	MBR0520L	D1, D2	Rectifier Diode
L0402	Ferrite Bead	FB	HZ0402A601R-10, 100 mA
—	TE 1734248	H5	FPC Connector, 1 mm pitch
R0402	1R	R1, R2	R0402, 1R, 50 V, 100 mW, ±1%
—	WE 7466203R	S1, S2	SMD Spacer, Blind hole M3
SO6	Viso 5kVac	T1	Transformer WE 750313626, or TTe HA00-17007LF
SSO8	ACPL-C799T	U1	Broadcom Iso Sigma-Delta Modulator
DBV	SN6501-Q1	U2	Transformer Driver
DBV	TPS76350-Q1	U3	5V LDO

Digital Filter Board

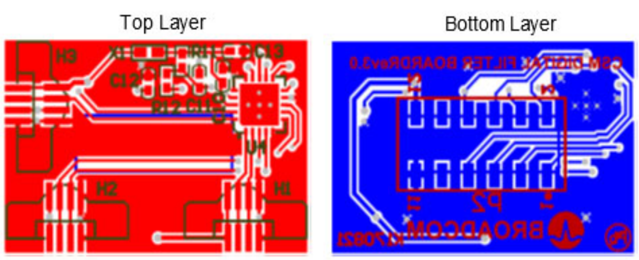
Land Pattern	Value	Designator	Description
C0402	1 µF	C10, C11	C0402, 1 µF, 25 V, ±10%, X5R
C0402	10 nF	C12	C0402, 10 nF, 25V, ±10%, X7R
C0402	0.1 µF	C13	C0402, 0.1µF, 16V, X7R, +/-20%
—	TE 1734248	H1, H2, H3	FPC Connector, 1 mm pitch
—	AMP 1241050-6	P2	Header 2X6P = 12P 2.54 mm
R0402	3.3 kΩ	R11	R0402, 3.3 kΩ, 50V, 125 mW, ±1%
R0402	33Ω	R12	R0402, 33Ω, 50V, 125 mW, ±1%
L0402	Ferrite Bead	X1	HZ0402A601R-10, 100 mA
QFN-20	ACPL-0873T	U4	Broadcom Digital Filter

Evaluation Board PCB Layout

Sigma-Delta Modulator BMS Board

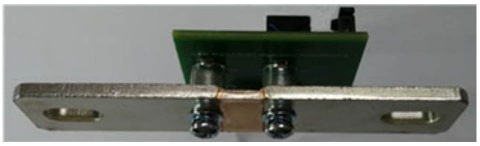


Digital Filter BMS Board



Mounting the Sigma-Delta Modulator BMS Board on the Shunt

The Sigma-Delta Modulator BMS Board can be mounted directly onto the shunt using two M3 screws (check the selected shunt dimensions). The recommended tightening torque is 0.5 Nm for M3 screws (pitch 0.5 mm and length 8 mm with flat and spring washers).



NOTE: If other shunt types cannot fit onto the Sigma-Delta Modulator BMS Board, connect a pair of twisted wires from the shunt to the SDM board. Keep the wires as short as possible.

Insulation Information between the Primary Side and the Secondary Side

- 5kVAC / 1 minute isolation voltage
- Clearance and Creepage distance: > 8 mm

Appendix

Digital Filter Typical Conversion Time

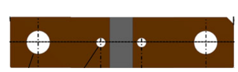
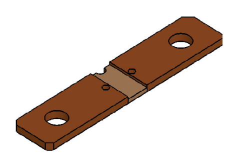
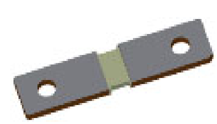
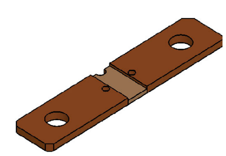
Filter Mode (K)	Decimation Ratio (D)	Filter Conversion Time t_C at 10 MHz MCLK ($1/t_C$)
SINC2	1024	205 µs (4.88 kHz)
SINC2	512	102 µs (9.76 kHz)
SINC2	256	51 µs (19.52 kHz)
SINC2	128	25 µs (39.04 kHz)
SINC3	256	77 µs (13.02 kHz)
SINC3	128	38 µs (26.04 kHz)
SINC3	64	19 µs (52.08 kHz)

NOTE: t_C is calculated as $t_C = 1 / f_{MCLK} * D * K$.

SPI Typical Timing

SPI Clock (MHz)	Time for 8-bit Write (µs)	Time for 8-bit Write and 8-bit Read (µs)	Time for 48-Bit Read (µs)
5	1.60	3.20	9.6
10	0.80	1.60	4.8
15	0.53	1.06	3.18
20	0.40	0.80	2.40

Associated Information: Shunt

Manufacturer	TT Electronics	KOA	Isabellenhutte	Vishay
0.1 mΩ Part Number	EBW-387-38-061	2017-c-818	BAS-M-R0001	WSBS8518L1000JTM4
				
0.05 mΩ Part Number	Available on request.	Available on request.	BAS-M-00005	WSBS8518L0500JTM4
				
Worldwide Contact	www.ttelectronicsresistors.com	www.koaglobal.com	www.isabellenhuette.de	www.vishay.com

Revision History

CSM-BMS-SPI-UG100; May 7

Initial release.

Important Notice

The Broadcom Current-Sense-Module Demonstration Kit's circuit schematic and PCB layout are reference designs made by Broadcom for evaluation purpose *only*. The verification was done at room temperature. Users may use the Demonstration Kit's circuit schematic and PCB design as a reference to evaluate Broadcom Current-Sense-Module functions *only* at room temperature on the condition that Broadcom holds neither liability on user's system performance with the Demonstration Kit applied nor on reliability.

The testing was done using a small sample size. The testing results presented in this document are *only* applicable to the circuit and component values, as well as other operating conditions only designated in this document. However, users may implement component value changes or circuitry modifications to achieve customized performance at their own discretion.

BROADCOM®

Lead (Pb) Free
RoHS 6 fully compliant
RoHS 6 fully compliant options available;
-xxxE denotes a lead-free product