

Configuring the PAC52XX CAFE for Different Control Topologies

Power Application Controller™

Marc D Sousa
Director, PAC Applications and Systems



www.active-semi.com
Copyright © 2015 Active-Semi, Inc.

Table of Contents

1 Overview	4
1.1 Block Diagram	5
1.2 Differential Programmable Gain Amplifiers	5
1.3 Single-Ended Programmable Gain Amplifiers.....	6
1.4 General Purpose Comparators	6
1.5 Phase Comparators.....	6
1.6 Protection Comparators.....	6
1.7 Push-Button.....	6
2 Configuring the CAFE for Sensorless BEMF	7
2.1 Application Block Diagram	7
2.2 Signal Manager Configuration.....	7
2.3 Configuring Differential Signal Inputs	8
2.4 Configuring Single Ended Signal Inputs.....	9
2.5 Configuring Phase Voltage Inputs.....	10
2.6 Enabling Comparator Hysteresis.....	10
2.7 Enabling the Signal Manager	10
3 Configuring the CAFE for FOC	12
3.1 Application Block Diagram	12
3.2 Signal Manager Configuration.....	13
3.3 Configuring Differential Signal Inputs	13
3.4 Configuring Single Ended Signal Inputs.....	15
3.5 Enabling Comparator Hysteresis.....	16
3.6 Enabling the Signal Manager	16
4 Configuring Current Measurement and Protection	17
4.1 Current Sense Block Diagram.....	17
4.2 Over-Current Functionality	17
4.3 Configuring Current Measurement and Over-Current Protection	18
5 Configuring the Push-Button.....	20
6 Firmware Configuration for BLDC BEMF.....	21
7 Firmware Configuration for FOC.....	22
About Active-Semi.....	23

1 OVERVIEW

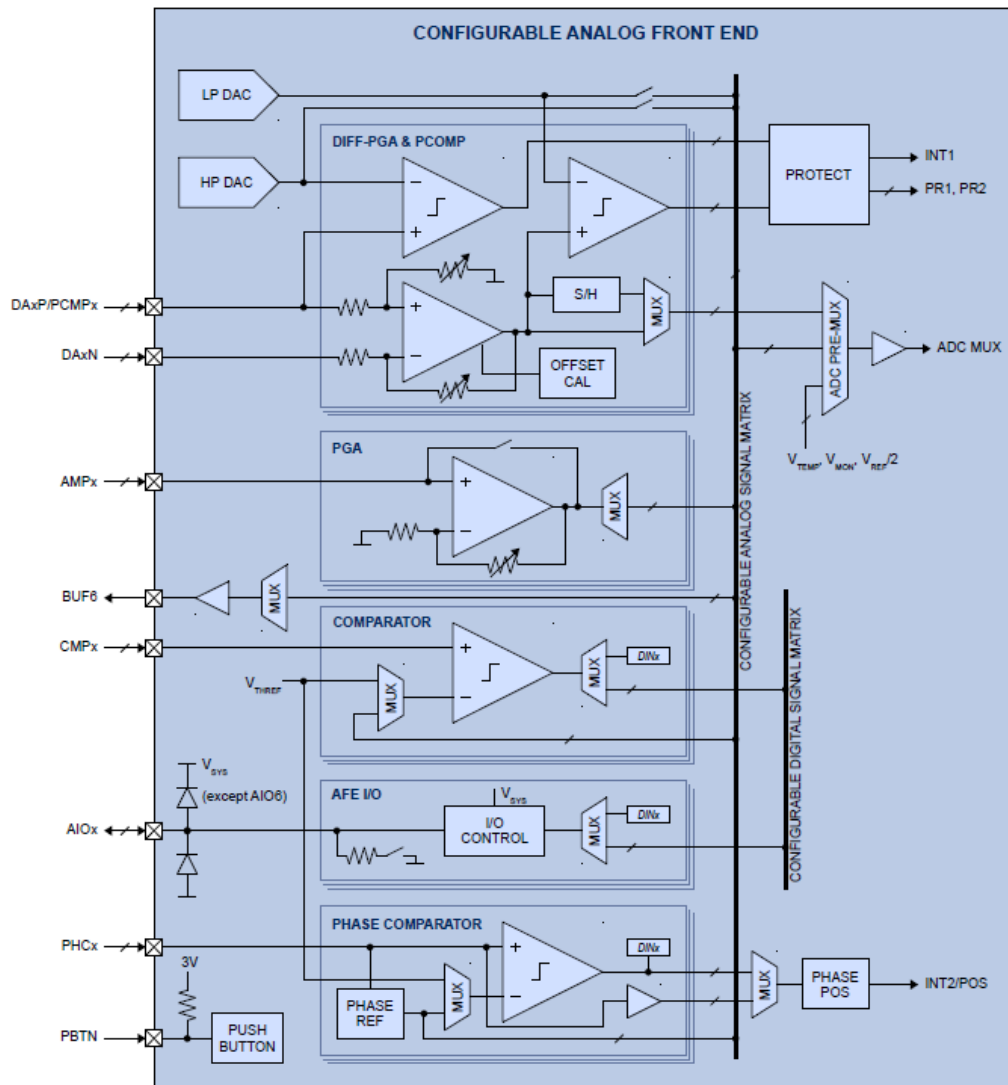
The family of Power Application Controller™ (PAC) is highly optimized for controlling and powering next-generation smart energy appliances, devices and equipment. The PAC family of controllers integrates management of many system functions for these applications, including power management and signal conditioning.

Each device in the PAC52XX family of controllers contains a Configurable Analog Front-End™ (CAFE). The CAFE contains a collection of differential and single-ended programmable gain amplifiers, comparators and other signal conditioning peripherals for use with various control applications.

The CAFE for the PAC52XX contains the following signal conditioning peripherals described below.

1.1 Block Diagram

Below is the block diagram for the Configurable Analog Front End (CAFE).



1.2 Differential Programmable Gain Amplifiers

Each device in the PAC52XX family has up to 3 differentially programmable gain amplifiers that are available to the application. The positive and negative signals on the device are connected to the input of the differential amplifiers. The differential gain on each of these amplifiers is programmable to a gain of 1X, 2X, 4X, 8X, 16X, 32X and 48X for zero ohm signal source impedance.

The differential programmable gain has -0.3V to 3.5V input common mode range, and its output can be configured for routing directly to the ADC pre-multiplexer, or through a sample-and-hold circuit synchronized with the ADC auto-sampling mechanism.

Each differential amplifier is accompanied by offset calibration circuitry, and two protection comparators for protection and event monitoring (such as over-current protection).

1.3 Single-Ended Programmable Gain Amplifiers

Each device in the PAC52XX family has up to 4 single-ended programmable gain amplifiers that are available to the application with the signal relative to V_{SSA} . The amplifier gain can be programmed to be 1X, 2X, 4X, 8X, 16X, 32X and 48X or as analog feed-through.

1.4 General Purpose Comparators

Each device in the PAC52XX family has up to 4 general purpose comparators that can be used to compare the input to a programmable threshold voltage.

1.5 Phase Comparators

Some devices in the PAC52XX family have up to 3 phase comparators that can be used for applications such as back EMF (BEMF) zero-cross detection.

The phase comparator takes its input from the PHCx pin on the PAC52XX. The comparator reference may either be the programmable threshold voltage (V_{THREF}) or internally generated virtual center-tap voltage (phase reference, in block diagram).

The comparator has a 0V to V_{SYS} input common mode range, and its polarity-selectable output is routed to the MCU and can be used to generate an interrupt, or may be polled by the firmware program.

1.6 Protection Comparators

Each differential amplifier in the PAC52XX has two protection comparators that can be used to trigger protection events. The state of these comparators is also accessible by the MCU by status registers.

The High-speed Protection (HP) comparators compare the positive input of the differential signal to the programmable 8-bit HP DAC output voltage, with full scale voltage of 2.5V. The Limit-Protection (LP) comparators compare the differential voltage after amplifier gain has been applied to the programmable 10-bit LP DAC output voltage, also with a full scale voltage of 2.5V.

Each protection comparator has a mask bit to prevent or allow it to trigger the main MCU interrupt (INT1/PB0). Each protection comparator also has mask bits to prevent or allow it to activate protection events PR1 and PR2. These two protection events allow the output drivers to be disabled upon a system event, such as over-current.

1.7 Push-Button

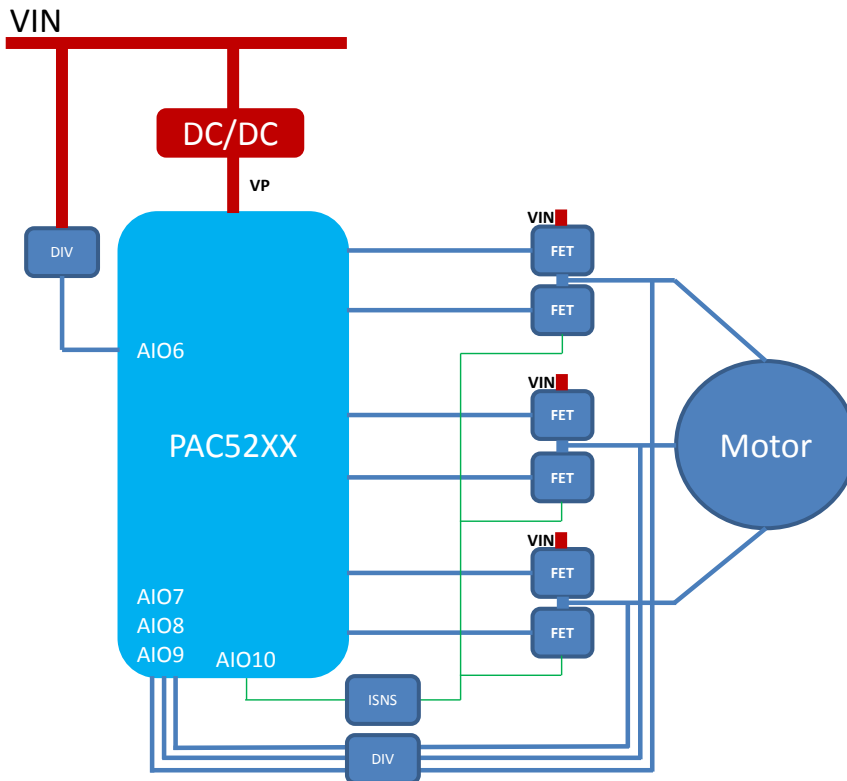
The push-button PBTN, may be used by the MCU to detect a user active-low push button event. When the system is in its ultra-low-power hibernate mode, the push-button may be used to wake up the system.

In addition, the push-button may be used as a hardware reset for the MCU when it is held low for longer than 8s during normal operation. The PBTN input is active-low and has a 55k Ω pull-up resistor to 3V.

2 CONFIGURING THE CAFE FOR SENSORLESS BEMF

2.1 Application Block Diagram

The diagram below shows a high-level block diagram of a BLDC Motor Controller that supports BEMF Control.



In this application, the motor phase voltages must be sampled to determine the BEMF zero-cross for motor commutation. Since these are high-voltage, they must be divided and then connected to AIO7, AIO8 and AIO9 for zero-cross detection. The full-scale amplifier and ADC range is 2.5V

The motor current is also sampled by the differential amplifier AIO10, and used for over-current protection and possibly also current control.

In addition, the user may want to sample another single-ended channel such as VIN on AIO6. Again, this may be a high voltage signal, so it must be divided before being sampled on AIO6. The full-scale amplifier and ADC range is 2.5V

2.2 Signal Manager Configuration

To configure the CAFE to support BEMF Sensorless Control, the following must be configured:

- Configure HPDAC and LPDAC for over-current protection
- Configure Differential Signal Inputs (AIO10) for Differential Comparator Mode
- Configure Single-Ended Inputs (AIO6) for Single-Ended Comparator Mode

- Configure Phase Voltage Inputs (AIO7, AIO8, AIO9) for Special Mode
- Enable Comparator Hysteresis
- Enable Signal Manager

For information on how to configure the PAC52XX for current measurement and over-current protection, see the section below.

2.3 Configuring Differential Signal Inputs

In order to sample current in this application, the most effective way is to use the integrated differential amplifiers on the PAC52XX.

In this application, the sources of the low-side FETs for all three motor phases are connected. A current sense resistor would be used to generate the positive and negative signals for the differentially-measured current sense. These signals would be connected to one of the differential signals on the PAC52XX, such as AIO10.

The Differential Input Signals should have the following configuration:

- Set AIO mode to Differential Amplifier Mode (**SOC.AIO0CFG.MODE10** = 01b)
- Set AIO Differential Amplifier Gain (**SOC.AIO.AIO0CFG.GAIN10**)
- Set LP comparator and blanking-time setting (**SOC.AIO.AIO0CFG.LP10EN**)
- Set HP comparator and blanking-time setting (**SOC.AIO.AIO1CFG.HP10EN**)
- Set HP and LP PR1 and PR2 Protection settings (see section later in this document)

To configure the Differential Amplifier Gain for any Differential Signals, see the table below:

SOC.AIO10CFG.GAIN10 SOC.AIO32CFG.GAIN32 SOC.AIO54CFG.GAIN54	Gain
000b	1X
001b	1X
010b	2X
011b	4X
100b	8X
101b	16X
110b	32X
111b	48X

To configure the LP comparator and blanking-time, see the table below:

SOC.AIO10CFG.LP10EN SOC.AIO32CFG.LP32EN SOC.AIO54CFG.LP54EN	Setting
00b	LP comparator disabled
01b	LP comparator enabled with 1 μ s blanking time
10b	LP comparator enabled with 2 μ s blanking time
11b	LP comparator enabled with 4 μ s blanking time

To configure the HP comparator and blanking-time, see the table below:

SOC.AIO10CFG.HP10EN SOC.AIO32CFG.HP32EN SOC.AIO54CFG.HP54EN	Setting
00b	HP comparator disabled
01b	HP comparator enabled with 1µs blanking time
10b	HP comparator enabled with 2µs blanking time
11b	HP comparator enabled with 4µs blanking time

2.4 Configuring Single Ended Signal Inputs

This application may need to sample other types of analog signals that do not require a differential amplifier. Examples of this would be sampling input voltage. For these signals, using one of the single-ended inputs can be configured as shown below.

In the example below, AIO6 is assumed to be used for sampling of a single-ended analog signal.

The Single-ended Input Signal should have the following configuration:

- Set AIO mode to Gain Amplifier Mode (**SOC.AIO6CFG.MODE6 = 01b**)
- Set AIO amplifier gain (**SOC.AIO6CFG.GAIN6**)
- Set AIO onto selected analog bus (**SOC.AIO6CFG.MUX6**)

To configure the Amplifier Gain for any Single-Ended Signals, see the table below:

SOC.AIO6CFG.GAIN6 SOC.AIO7CFG.GAIN7 SOC.AIO8CFG.GAIN8 SOC.AIO9CFG.GAIN9	Gain
000b	Direct Mode
001b	1X
010b	2X
011b	4X
100b	8X
101b	16X
110b	32X
111b	48X

To configure the Analog Bus for any AIO6, see the table below:

SOC.AIO6CFG.MUX6	Analog Bus
000b	AB6
001b	AB1
010b	AB2
011b	AB3
100b	AB4
101b	AB5
110b	AB6
111b	AB7

2.5 Configuring Phase Voltage Inputs

To support sensorless BLDC motor control applications using BEMF, the user needs to sample to when each of the motor phases has a zero-cross event, in order to correctly commutate the motor.

In order to do this, the PAC52XX family of devices allows inputs AIO7, AIO8 and AIO9 to be put into “Special Mode” so that the phase voltages can be compared to the virtual center-tap voltage to detect the zero-cross events.

To configure AIO7, AIO8 or AIO9 for “Special” Mode to support this feature, the following registers and fields need to be configured:

- AIO7:
 - Set **SOC.AIO7CFG.MODE6** to 11b (AIO7 Special Mode)
 - Set **SOC.AIO7CFG.ADMUX** to 01b (Analog Bus AB1)
 - Set **SOC.SMCFG.AIO7HYS** to 1b (AIO7 Comparator Hysteresis enabled)
- AIO8:
 - Set **SOC.AIO8CFG.MODE8** to 11b (AIO8 Special Mode)
 - Set **SOC.AIO8CFG.ADMUX** to 01b (Analog Bus AB1)
 - Set **SOC.SMCFG.AIO8HYS** to 1b (AIO8 Comparator Hysteresis enabled)
- AIO9:
 - Set **SOC.AIO9CFG.MODE9** to 11b (AIO9 Special Mode)
 - Set **SOC.AIO9CFG.ADMUX** to 01b (Analog Bus AB1)
 - Set **SOC.SMCFG.AIO9HYS** to 1b (AIO9 Comparator Hysteresis enabled)

The firmware to configure AIO7, AIO8 and AIO9 for Special Mode is shown below:

```
pac5xxx_tile_register_write(ADDR_CFGAIO7, 0xD0); // AIO9HYS, AIO8HYS, AIO7HYS
pac5xxx_tile_register_write(ADDR_CFGAIO7, 0xD0); // MODE7[1:0] = 11b (special mode), OPT7[1:0] = 01b (AB1 as COMP-)
pac5xxx_tile_register_write(ADDR_CFGAIO8, 0xD0); // MODE8[1:0] = 11b (special mode), OPT8[1:0] = 01b (AB1 as COMP-)
pac5xxx_tile_register_write(ADDR_CFGAIO9, 0xD0); // MODE9[1:0] = 11b (special mode), OPT9[1:0] = 01b (AB1 as COMP-)
```

2.6 Enabling Comparator Hysteresis

The HP, LP and special mode comparators in the PAC52XX Signal Manager have the ability to configure comparator hysteresis. To set the comparator hysteresis state, set the following registers to the values below.

To disable the comparator hysteresis, the value should be set to a 0b. To enable the comparator hysteresis, the value should be set to a 1b.

Comparator	Register
HP	SOC.SMCFG.HPROTHYS
LP	SOC.SMCFG.LPROTHYS
AIO6	SOC.SMCFG.AIO6HYS
AIO7	SOC.SMCFG.AIO7HYS
AIO8	SOC.SMCFG.AIO8HYS
AIO9	SOC.SMCFG.AIO9HYS

2.7 Enabling the Signal Manager

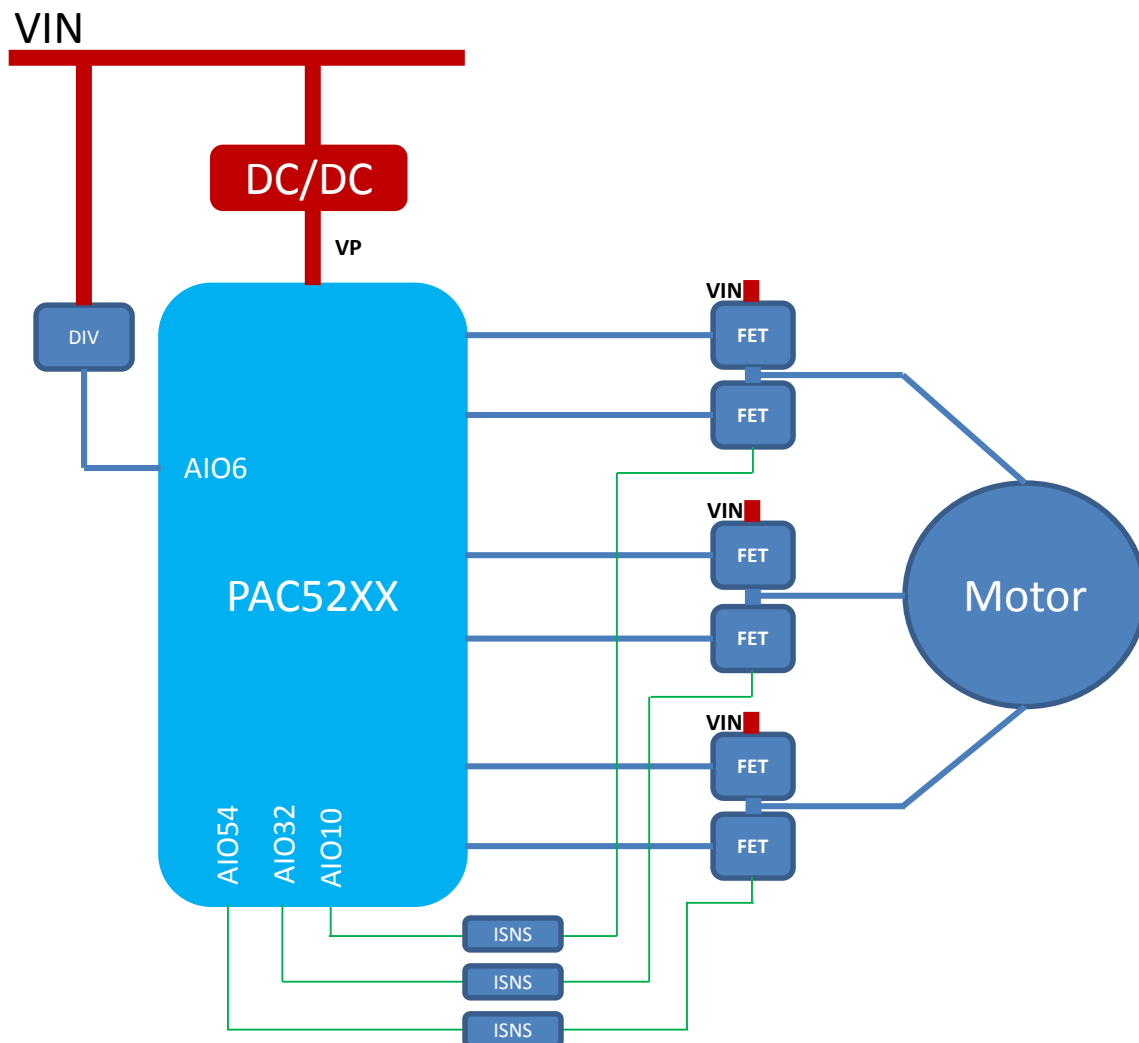
Once the signal manager is configured, it must be enabled before the application will function.

To enable the signal manager, set the **SOC.SMCTL.SMEN** bit to a 1b.

3 CONFIGURING THE CAFE FOR FOC

3.1 Application Block Diagram

The diagram below shows a high-level block diagram of a BLDC Motor Controller that supports FOC Control.



In this application, the controller must sample all three motor phase currents, which are done through individual current sense circuits, and then connected to three differential amplifier inputs on the PAC52XX. The phase currents are used for control of the motor, as well as over-current protection.

The full-scale range of the differential amplifier is 2.5V. Because this application performs sinusoidal motor drive, the phase current may be either positive or negative, depending on the position of the motor. The PAC52XX signal manager must be sure to configure these differential amplifier inputs so that they are referenced to $V_{REF}/2$, not GND.

For an application with three differential inputs, the user would want AIO10, AIO32 and AIO54 connected to the three differential signal pairs.

No position sensor or phase voltages are needed for control in this application.

In addition, the user may want to sample another single-ended channel such as VIN on AIO6. This may be a high voltage signal, so it must be divided before being sampled on AIO6. The full-scale amplifier and ADC range is 2.5V.

3.2 Signal Manager Configuration

To configure the CAFE to support Sensorless FOC Control, the following must be configured:

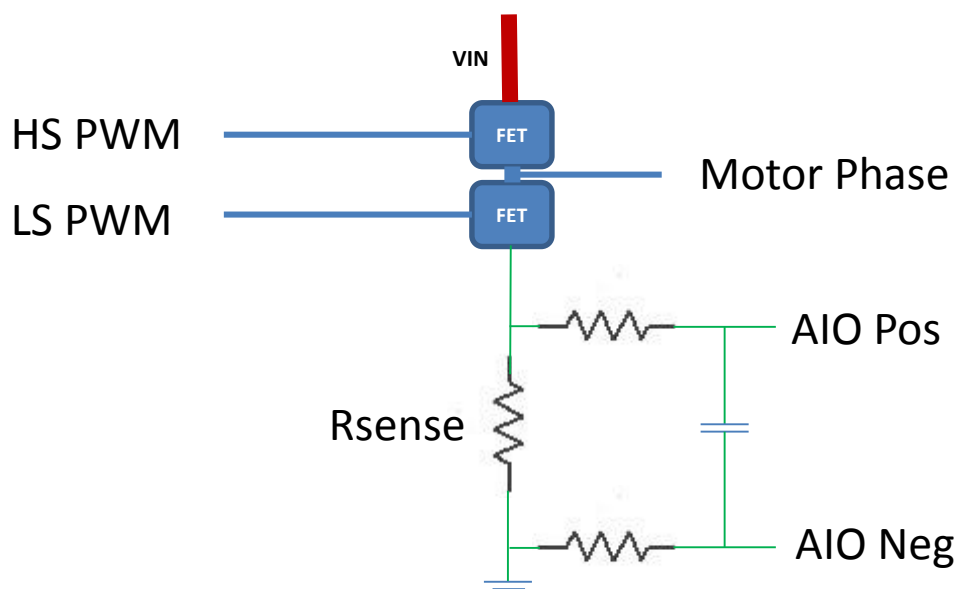
- Configure HPDAC and LPDAC for over-current protection
- Configure Differential Signal Inputs AIO10, AIO32 and AIO54 for Differential Comparator Mode
- Configure Single-Ended Inputs (AIO6) for Single-Ended Comparator Mode
- Enable Comparator Hysteresis
- Enable Signal Manager

For information on how to configure the PAC52XX for current measurement and over-current protection, see the section below.

3.3 Configuring Differential Signal Inputs

In this application, each motor phase current may be positive or negative at any point in time. The current must be accurately sampled for the control of the application, as well as for over-current protection. In order to support these two requirements, all three phase currents will be sampled using the three differential amplifier inputs on the PAC52XX.

For each motor phase, the current sense differential signals are generated as shown below:



The Positive and Negative terminals of the differential current are created from the source of the low-side FET, by using a sense resistor circuit as shown above. The positive and negative sides of the signal are then connected to the positive and negative terminal of the PAC52XX, such as AIO0 and AIO1.

The Differential Input Signals should have the following configuration:

- Set AIO mode to Differential Amplifier Mode:
 - **SOC.AIO0CFG.MODE10** = 01b
 - **SOC.AIO2CFG.MODE32** = 01b
 - **SOC.AIO4CFG.MODE54** = 01b
- Set AIO Differential Amplifier Gain:
 - **SOC.AIO.AIO0CFG.GAIN10**
 - **SOC.AIO.AIO2CFG.GAIN32**
 - **SOC.AIO.AIO4CFG.GAIN54**
- Set LP comparator and blanking-time setting:
 - **SOC.AIO.AIO0CFG.LP10EN**
 - **SOC.AIO.AIO2CFG.LP32EN**
 - **SOC.AIO.AIO4CFG.LP32EN**
- Set HP comparator and blanking-time setting:
 - **SOC.AIO.AIO1CFG.HP10EN**
 - **SOC.AIO.AIO3CFG.HP32EN**
 - **SOC.AIO.AIO5CFG.HP54EN**
- Set HP and LP PR1 and PR2 Protection settings (see section later in this document)
- Set Differential Amplifier Offset mode to enabled by setting these registers to 1b:
 - **SOC.AIO.AIO1CFG.OS10EN**
 - **SOC.AIO.AIO3CFG.OS32EN**
 - **SOC.AIO.AIO5CFG.OS54EN**

To configure the Differential Amplifier Gain for any Differential Signals, see the table below:

SOC.AIO10CFG.GAIN10 SOC.AIO32CFG.GAIN32 SOC.AIO54CFG.GAIN54	Gain
000b	1X
001b	1X
010b	2X
011b	4X
100b	8X
101b	16X
110b	32X
111b	48X

To configure the LP comparator and blanking-time, see the table below:

SOC.AIO10CFG.LP10EN SOC.AIO32CFG.LP32EN SOC.AIO54CFG.LP54EN	Setting
00b	LP comparator disabled
01b	LP comparator enabled with 1 μ s blanking time
10b	LP comparator enabled with 2 μ s blanking time

11b	LP comparator enabled with 4 μ s blanking time
-----	--

To configure the HP comparator and blanking-time, see the table below:

SOC.AIO10CFG.HP10EN SOC.AIO32CFG.HP32EN SOC.AIO54CFG.HP54EN	Setting
00b	HP comparator disabled
01b	HP comparator enabled with 1 μ s blanking time
10b	HP comparator enabled with 2 μ s blanking time
11b	HP comparator enabled with 4 μ s blanking time

3.4 Configuring Single Ended Signal Inputs

This application may need to sample other types of analog signals that do not require a differential amplifier. Examples of this would be sampling input voltage. For these signals, using one of the single-ended inputs can be configured as shown below.

In the example below, AIO6 is assumed to be used for sampling of a single-ended analog signal.

The Single-ended Input Signal should have the following configuration:

- Set AIO mode to Gain Amplifier Mode (**SOC.AIO6CFG.MODE6** = 01b)
- Set AIO amplifier gain (**SOC.AIO6CFG.GAIN6**)
- Set AIO onto selected analog bus (**SOC.AIO6CFG.MUX6**)

To configure the Amplifier Gain for any Single-Ended Signals, see the table below:

SOC.AIO6CFG.GAIN6 SOC.AIO7CFG.GAIN7 SOC.AIO8CFG.GAIN8 SOC.AIO9CFG.GAIN9	Gain
000b	Direct Mode
001b	1X
010b	2X
011b	4X
100b	8X
101b	16X
110b	32X
111b	48X

To configure the Analog Bus for any AIO6, see the table below:

SOC.AIO6CFG.MUX6	Analog Bus
000b	AB6
001b	AB1
010b	AB2
011b	AB3
100b	AB4
101b	AB5

110b	AB6
111b	AB7

3.5 Enabling Comparator Hysteresis

The HP and LP comparators in the PAC52XX Signal Manager have the ability to configure comparator hysteresis. To set the comparator hysteresis state, set the following registers to the values below.

To disable the comparator hysteresis, the value should be set to a 0b. To enable the comparator hysteresis, the value should be set to a 1b.

Comparator	Register
HP	SOC.SMCFG.HPROTHYS
LP	SOC.SMCFG.LPROTHYS

3.6 Enabling the Signal Manager

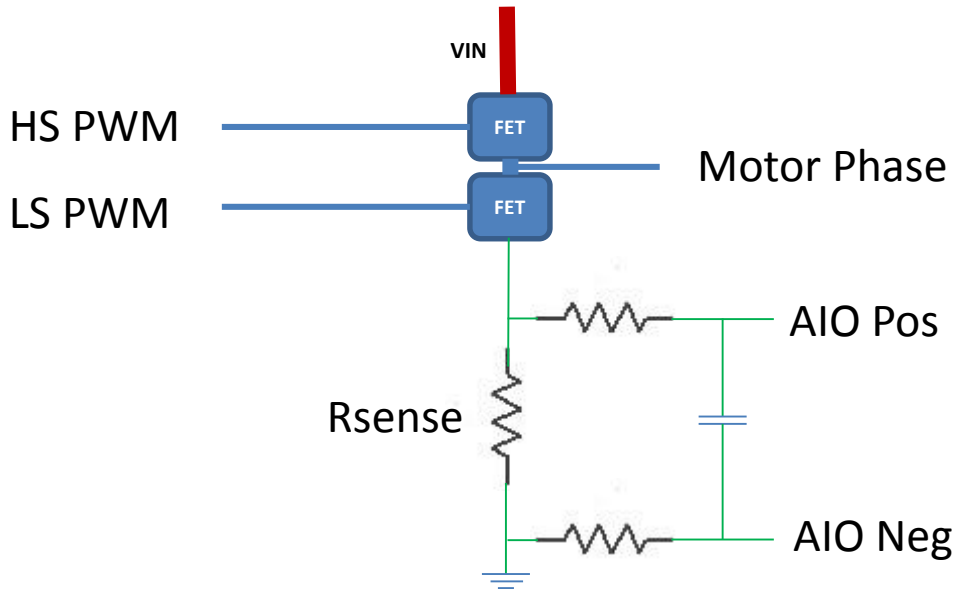
Once the signal manager is configured, it must be enabled before the application will function.

To enable the signal manager, set the **SOC.SMCTL.SMEN** bit to a 1b.

4 CONFIGURING CURRENT MEASUREMENT AND PROTECTION

4.1 Current Sense Block Diagram

Current may be measured in the PAC52XX by using one of the differential amplifier inputs as shown below:



For some applications, current must be measured and in most applications it is desirable to program an over-current threshold for application safety.

4.2 Over-Current Functionality

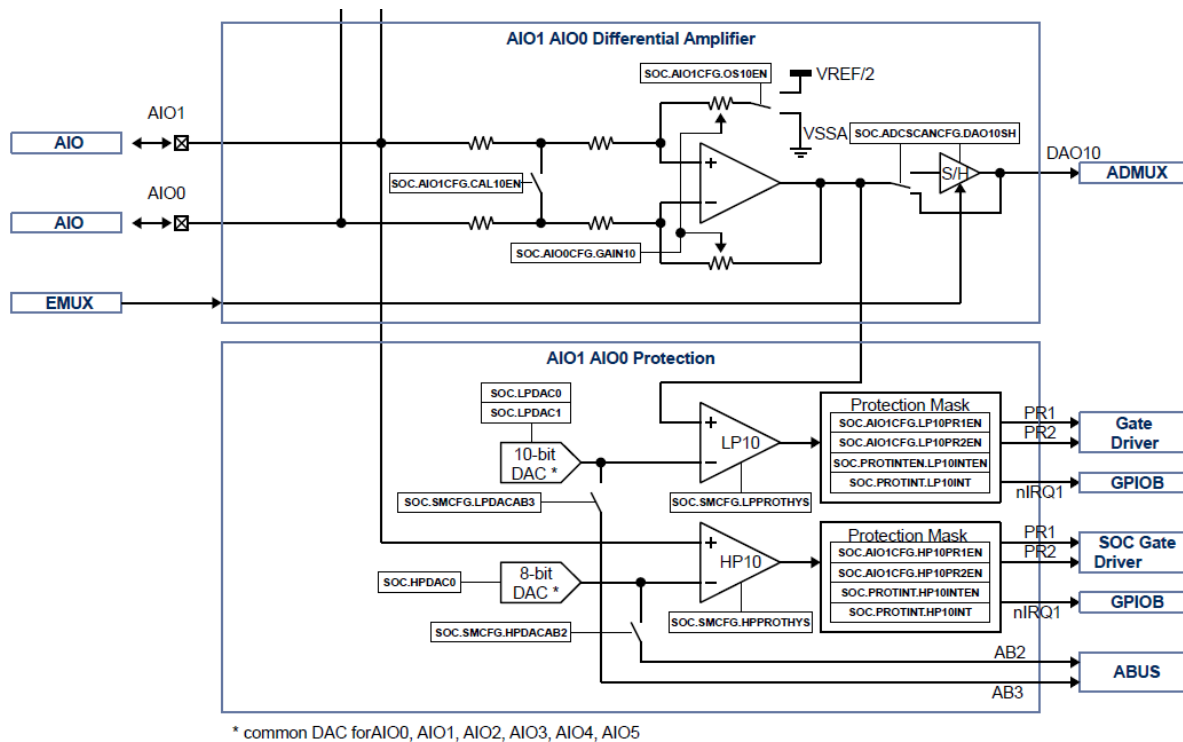
The PAC52XX has two different over-current comparators with programmable thresholds:

- HP (high-speed)
- LP (limit-protection)

Each comparator reference is user programmable via MCU firmware. The HP reference is set by a 8-bit DAC and the comparator compares the positive terminal of the differential signal against the reference.

The LP reference is set by a 10-bit DAC and the comparator compares the difference signal after the gain amplifier stage against the reference.

Below is a block diagram of the over-current protection circuitry for the PAC52XX's differential amplifiers:



Note that the LP and HP DACs used to set the comparator reference are for all three of the PAC52XX differential amplifiers.

If an over-current event is detected, the PAC52XX may take the following actions for both the PR1 and PR2 events:

- Disable the gate drivers
- Assert the nIRQ1/PB0 pin to the MCU

These will allow the user to have a high level of configurability for over-current, as well as having flexible notification options.

4.3 Configuring Current Measurement and Over-Current Protection

To configure current measurement and over-current support in the PAC52XX, the user should set the following registers:

- Enable Differential Mode for the Analog inputs for each of the desired inputs:
 - **SOC.AIO0CFG.MODE10** = 01b
 - **SOC.AIO2CFG.MODE32** = 01b
 - **SOC.AIO4CFG.MODE54** = 01b
- Set the Differential Amplifier Gain for each Signal:
 - **SOC.AIO0CFG.GAIN10**
 - **SOC.AIO2CFG.GAIN32**
 - **SOC.AIO4CFG.GAIN54**
- Set the LP comparator state and blanking time for each signal:
 - **SOC.AIO0CFG.LP10EN**

- **SOC.AIO2CFG.LP32EN**
 - **SOC.AIO4CFG.LP54EN**
- Set the PR1 and PR2 protection event enable masks to 1 for the HP comparator:
 - **SOC.AIO1CFG.HP10PR1EN**
 - **SOC.AIO1CFG.HP10PR2EN**
- Set the PR1 and PR2 protection event enable masks to 1 for the LP comparator:
 - **SOC.AIO1CFG.LP10PR1EN**
 - **SOC.AIO1CFG.LP10PR2EN**
- Set the HP comparator interrupt enable bits to a 1 for the AIO10, AIO32 and AIO54 comparators, for MCU interrupts:
 - **SOC.PROTINTEN.HP10INTEN**
 - **SOC.PROTINTEN.HP32INTEN**
 - **SOC.PROTINTEN.HP54INTEN**
- Set the LP comparator interrupt enable bits to a 1 for the AIO10, AIO32, AIO54 comparators, for MCU interrupts:
 - **SOC.PROTINTEN.LP10INTEN**
 - **SOC.PROTINTEN.LP32INTEN**
 - **SOC.PROTINTEN.LP54INTEN**

See the sections below to see for firmware examples of how to configure over-current protection and current sampling.

5 CONFIGURING THE PUSH-BUTTON

Some PAC52XX devices have a push-button input that can detect a user active-low push button event, and to put the system into an ultra-low-power hibernate mode. Once the system is in hibernate mode, PBTN can be used to wake up the system. In addition, PBTN can also be used as hardware reset for the MCU when it is held low for longer than 8 seconds during normal operation. The PBTN input is active-low and has a 55 k Ω pull-up resistor to 3V.

To configure the PAC52XX to use the push button, the user should use the following configuration:

- Set **SOC.PWRSTAT.PBEN** to 1b
- Set **SOC.PWRSTAT.PBINTEN** to 1b to enable the IRQ1/PB0 interrupt
- Set **SOC.PWRSTAT.PBINT** to monitor the interrupt status and write 1b to clear interrupt

6 FIRMWARE CONFIGURATION FOR BLDC BEMF

The following code shows an example of how to configure the analog front end for power, signal and driver control for this application.

Registers related to the signal manager described in this application note are highlighted in **red**.

```
void cafe_init(void)
{
    // Configure SOC Bridge for talking to MC02
    pac5xxx_tile_socbridge_config(1, 0); // SOC Bridge Enable, ~SOCB INT enable

    // Configure System and Power Manager
    pac5xxx_tile_register_write(ADDR_DEVID, 0x55); // Write DEVID to value other than 0x00, 0xFF

    if (pac5xxx_tile_register_read(ADDR_PWRSTAT)) // If any power manager error bits set on startup, clear them
        pac5xxx_tile_register_write(ADDR_PWRSTAT, 0xFF);

    pac5xxx_tile_register_write(ADDR_PWRCTL, 0x40); // Set MCUALIVE
    pac5xxx_tile_register_write(ADDR_PSTATSET, 0x80); // Set UNLOCK bit to allow firmware to modify SCFG & CFGPWR0
    pac5xxx_tile_register_write(ADDR_SCFG, 0x30); // Set VCLAMPSEL (62V) and FMODE (18kHz to 500kHz buck)
    pac5xxx_tile_register_write(ADDR_IMOD, 0xFF); // Set current modulation to 100%
    pac5xxx_tile_register_write(ADDR_ENBBM, 0x01); // Enable make before break in driver tile

    //Configure DC/DC Switching Regulator

    #if NODC
    pac5xxx_tile_register_write(0x15, 0xE0); // CFGPWR0: Disable DC/DC, VP=15V (Direct Supply)
    #else
    pac5xxx_tile_register_write(0x15, 0x80); // CFGPWR0: Enable DC/DC, VP=12V (SEPIC or Buck Mode)
    #endif

    // Set HPROT and LPROT protection threshold
    pac5xxx_tile_register_write(ADDR_HPDAC, OC_LIMIT); // HPDAC: HPROT DAC (8b)

    // AIO54: Imotor
    pac5xxx_tile_register_write(ADDR_CFGAIO4, 0x78); // AIO54: DiffAmp, 48X gain, LPOPT disabled
    pac5xxx_tile_register_write(ADDR_CFGAIO5, 0x81); // AIO54: nHP54PR1M set, HPOPT enabled (lus blanking)

    // Enable protection interrupt mask
    pac5xxx_tile_register_write(ADDR_PROTINTM, 0x40); // PROTINTEN: HP54INTEN

    pac5xxx_tile_register_write(ADDR_CFGDRV1, 0xA0); // Disable both HS and LS drivers on PR1 event (nHSPR1M=1, nLSPR1M=1)
    pac5xxx_tile_register_write(ADDR_SIGSET, 0xE8); // SMCFG: Comparator hysteresis (HPROTHYS, AIO7HYS, AIO8HYS, AIO9HYS)
    pac5xxx_tile_register_write(ADDR_SYSTAT, 0x01); // Turn on nINTM interrupts to MCU

    // Configure single-ended AIO for Vin
    pac5xxx_tile_register_write(ADDR_CFGAIO6, 0x61); // MODE6 = gain amp mode, GAIN6=8X, MUX6=AB1

    // Configure Sensorless Comparators
    pac5xxx_tile_register_write(ADDR_CFGAIO7, 0xD0); // MODE7 = special mode, OPT7 = AB1
    pac5xxx_tile_register_write(ADDR_CFGAIO8, 0xD0); // MODE8 = special mode, OPT8 = AB1
    pac5xxx_tile_register_write(ADDR_CFGAIO9, 0xD0); // MODE9 = special mode, OPT9 = AB1

    // Enable protection interrupt mask
    pac5xxx_tile_register_write(ADDR_ADCSCAN, 0x18); // ADSCANCFG: SCANEN, ADCBUFEN

    // Enable signal manager
    pac5xxx_tile_register_write(ADDR_ENSIG, 1);

    // Enable manager
    pac5xxx_tile_register_write(ADDR_ENDRV, 1);
}
```

7 FIRMWARE CONFIGURATION FOR FOC

The following code shows an example of how to configure the analog front end for power, signal and driver control for this application.

Registers related to the signal manager described in this application note are highlighted in **red**.

```

void cafe_init(void)
{
    // Configure SOC Bridge for talking to MC02
    pac5xxx_tile_socbridge_config(1, 0); // SOC Bridge Enable, ~SOCB INT enable

    // Configure System and Power Manager
    pac5xxx_tile_register_write(ADDR_DEVID, 0x55); // Write DEVID to value other than 0x00, 0xFF

    if (pac5xxx_tile_register_read(ADDR_PWRSTAT)) // If any power manager error bits set on startup, clear them
        pac5xxx_tile_register_write(ADDR_PWRSTAT, 0xFF);

    pac5xxx_tile_register_write(ADDR_PWRCTL, 0x40); // Set MCUALIVE
    pac5xxx_tile_register_write(ADDR_PSTATSET, 0x80); // Set UNLOCK bit to allow firmware to modify SCFG & CFGPWR0
    pac5xxx_tile_register_write(ADDR_SCFG, 0x30); // Set VCLAMPSEL (62V) and FMODE (18kHz to 500kHz buck)
    pac5xxx_tile_register_write(ADDR_IMOD, 0xFF); // Set current modulation to 100%
    pac5xxx_tile_register_write(ADDR_ENBBM, 0x01); // Enable make before break in driver tile

    //Configure DC/DC Switching Regulator
    pac5xxx_tile_register_write(0x15, 0x80); // CFGPWR0: Enable DC/DC, VP=12V (SEPIC or Buck Mode)

    // Set HPROT and LPROT protection threshold
    pac5xxx_tile_register_write(ADDR_HPDAC, OC_LIMIT); // HPDAC: HPROT DAC (8b)

    // AIO10: Iu
    pac5xxx_tile_register_write(ADDR_CFGAIO0, 0x48); // AIO0CFG: MODE10 = DiffAmp, GAIN10 = 1X, LP10EN = disabled
    pac5xxx_tile_register_write(ADDR_CFGAIO1, 0x09); // AIO1CFG: HP10EN = HP enabled (lus blank), OS10EN (offset)

    // AIO32: Iv
    pac5xxx_tile_register_write(ADDR_CFGAIO2, 0x48); // AIO2CFG: MODE32 = DiffAmp, GAIN32 = 1X, LP32EN = disabled
    pac5xxx_tile_register_write(ADDR_CFGAIO3, 0x09); // AIO3CFG: HP32EN = HP enabled (lus blank), OS32EN (offset)

    // AIO54: Iw
    pac5xxx_tile_register_write(ADDR_CFGAIO4, 0x48); // AIO4CFG: MODE54 = DiffAmp, GAIN54 = 1X, LP54EN = disabled
    pac5xxx_tile_register_write(ADDR_CFGAIO5, 0x09); // AIO5CFG: HP54EN = HP enabled (lus blank), OS54EN (offset)

    // Enable protection interrupt mask
    pac5xxx_tile_register_write(ADDR_PROTINTM, 0x70); // PROTINTEN: HP54INTEN, HP32INTEN, HP10INTEN

    pac5xxx_tile_register_write(ADDR_CFGDRV1, 0xA0); // Disable both HS and LS drivers on PR1 event (nHSPR1M=1, nLSPR1M=1)
    pac5xxx_tile_register_write(ADDR_SIGSET, 0x08); // SMCFG: Comparator hysteresis (HPROTHYS)
    pac5xxx_tile_register_write(ADDR_SYSTAT, 0x01); // Turn on nINTM interrupts to MCU

    // Configure single-ended AIO for Vin
    pac5xxx_tile_register_write(ADDR_CFGAIO6, 0x61); // MODE6 = gain amp mode, GAIN6=8X, MUX6=AB1

    // Enable protection interrupt mask
    pac5xxx_tile_register_write(ADDR_ADCSCAN, 0x18); // ADSCANCFG: SCANEN, ADCBUFEN

    // Enable signal manager
    pac5xxx_tile_register_write(ADDR_ENSIG, 1);

    // Enable manager
    pac5xxx_tile_register_write(ADDR_ENDRV, 1);
}

```

ABOUT ACTIVE-SEMI

Active-Semi, Inc. headquartered in Dallas, TX is a leading innovative semiconductor company with proven power management, analog and mixed-signal products for end-applications that require power conversion (AC/DC, DC/DC, DC/AC, PFC, etc.), motor drivers and control and LED drivers and control along with ARM microcontroller for system development.

Active-Semi's latest family of Power Application Controller (PAC)[™] ICs offer high-level of integration with 32-bit ARM Cortex M0, along with configurable power management peripherals, configurable analog front-end with high-precision, high-speed data converters, single-ended and differential PGAs, integrated low-voltage and high-voltage gate drives. PAC IC offers unprecedented flexibility and ease in the systems design of various end-applications such as Wireless Power Transmitters, Motor drives, UPS, Solar Inverters and LED lighting, etc. that require a microcontroller, power conversion, analog sensing, high-voltage gate drives, open-drain outputs, analog & digital general purpose IO, as well as support for wired and wireless communication. More information and samples can be obtained from <http://www.active-semi.com> or by emailing marketing@active-semi.com

Active-Semi shipped its 1 Billionth IC in 2012, and has over 120 in patents awarded and pending approval.

LEGAL INFORMATION & DISCLAIMER

Copyright © 2012-2015 Active-Semi, Inc. All rights reserved. All information provided in this document is subject to legal disclaimers.

Active-Semi reserves the right to modify its products, circuitry or product specifications without notice. Active-Semi products are not intended, designed, warranted or authorized for use as critical components in life-support, life-critical or safety-critical devices, systems, or equipment, nor in applications where failure or malfunction of any Active-Semi product can reasonably be expected to result in personal injury, death or severe property or environmental damage. Active-Semi accepts no liability for inclusion and/or use of its products in such equipment or applications. Active-Semi does not assume any liability arising out of the use of any product, circuit, or any information described in this document. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of Active-Semi or others. Active-Semi assumes no liability for any infringement of the intellectual property rights or other rights of third parties which would result from the use of information contained herein. Customers should evaluate each product to make sure that it is suitable for their applications. Customers are responsible for the design, testing, and operation of their applications and products using Active-Semi products. Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products. All products are sold subject to Active-Semi's terms and conditions of sale supplied at the time of order acknowledgment. Exportation of any Active-Semi product may be subject to export control laws.

Active-Semi[™], Active-Semi logo, Solutions for Sustainability[™], Power Application Controller[™], Micro Application Controller[™], Multi-Mode Power Manager[™], Configurable Analog Front End[™], and Application Specific Power Drivers[™] are trademarks of Active-Semi, I. ARM[®] is a registered trademark and Cortex[™] is a trademark of ARM Limited. All referenced brands and trademarks are the property of their respective owners.