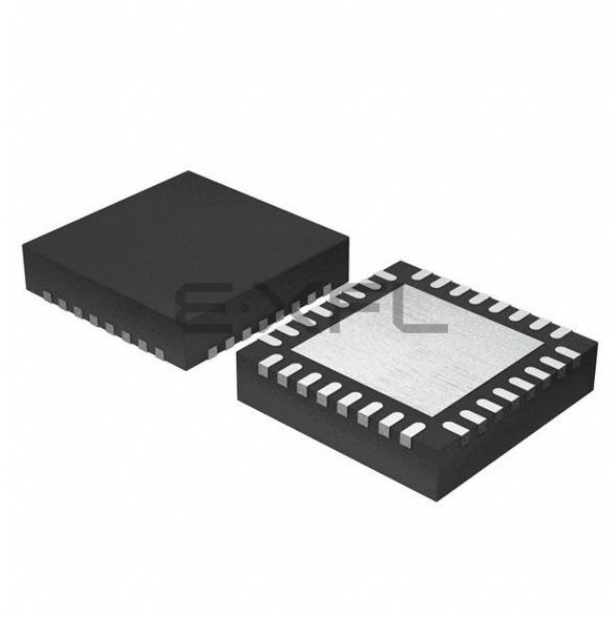

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	ARM7®
Core Size	16/32-Bit
Speed	20.48MHz
Connectivity	LINbus, SPI
Peripherals	POR, Temp Sensor, WDT
Number of I/O	6
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 32
Voltage - Supply (Vcc/Vdd)	3.5V ~ 18V
Data Converters	A/D 2x16b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 115°C (TA)
Mounting Type	Surface Mount
Package / Case	32-VFQFN Exposed Pad, CSP
Supplier Device Package	32-LFCSP-VQ (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/aduc7039wbcpz-rl

SPECIFICATIONS

ELECTRICAL SPECIFICATIONS

V_{DD} = 3.5 V to 18 V, V_{REF} = 1.2 V internal reference, f_{CORE} = 20.48 MHz driven from on-chip precision oscillator, all specifications T_A = −40°C to +115°C, unless otherwise noted.

Table 1.

Parameter	Test Conditions/Comments	Min	Typ	Max	Unit
ADC SPECIFICATIONS					
Conversion Rate[1]	ADC normal operating mode	10		1000	Hz
	ADC low power mode, chop on	10		650	Hz
Current Channel					
No Missing Codes[1]	Valid for all ADC update rates and ADC modes	16			Bits
Integral Nonlinearity[1, 2]			±10	±60	ppm of FSR
Offset Error[1, 2, 3]	Chop off, external short, after user system calibration, 1 LSB = (36.6/gain) µV	−10	±3	+10	LSB
Offset Error[1, 3]	Chop on, external short, low power mode, MCU powered down	+250	±50	−300	nV
Offset Error[1, 3]	Chop on, external short, after user system calibration Tested at CD = 0, VDD = 18 V	0		3.0	µV
Offset Error[1, 3]	Chop on, external short, after user system calibration Tested at CD = 0, VDD = 4 V		±0.5		µV
Offset Error Drift[1, 2, 4]	Chop off, gains of 8 to 64, normal mode		±0.03		LSB/°C
Offset Error Drift[1, 4]	Chop off, valid for ADC gain of 512		±30		nV/°C
Offset Error Drift[1, 4]	Chop on		±5		nV/°C
Total Gain Error[1, 3, 5, 6]	Factory calibrated at a gain of 4; normal mode	−0.5	±0.1	+0.5	%
	Low power mode	−1	±0.2	+1	%
Gain Drift[1, 7]			±3		ppm/°C
PGA Gain Mismatch Error			±0.1		%
Output Noise[1]	10 Hz update rate, gain = 512, chop enabled		100	150	nV rms
	1 kHz update rate, gain = 512, ADCFLT = 0x0007		0.6	0.9	µV rms
	1 kHz update rate, gain = 32, ADCFLT = 0x0007		0.8	1.2	µV rms
	1 kHz update rate, gain = 8, ADCFLT = 0x8101		2.1	4.1	µV rms
	1 kHz update rate, gain = 8, ADCFLT = 0x0007		1.6	2.4	µV rms
	1 kHz update rate, gain = 4, ADCFLT = 0x0007		2.6	3.9	µV rms
	ADC low power mode, 250Hz update rate, chop enable, gain = 512		0.6	0.9	µV rms
Voltage Channel[8]					
No Missing Codes[1]	Valid at all ADC update rates	16			Bits
Integral Nonlinearity[1]			±10	±60	ppm of FSR
Offset Error[1, 3]	Chop off, 1 LSB = 439.5 µV, after two point calibration	−10	±1	+10	LSB
Offset Error[1, 3]	Chop on, after two point calibration	−1	±0.3	+1	LSB
Offset Error Drift[4]	Chop off		±0.03		LSB/°C
Total Gain Error[1, 3, 5, 6]	Includes resistor mismatch	−0.25	±0.06	+0.25	%
Total Gain Error[1, 3, 5, 6]	Temperature range = −25°C to +65°C	−0.15	±0.03	+0.15	%
Gain Drift[1, 7]	Includes resistor mismatch drift		±3		ppm/°C
Output Noise[1, 9]	10 Hz update rate, chop on		60	90	µV rms
	1 kHz update rate, ADCFLT = 0x0007		180	270	µV rms
Temperature Channel					
No Missing Codes[1]	Valid at all ADC update rates	16			Bits
Integral Nonlinearity[1]			±10	±60	ppm of FSR
Offset Error[3, 10]	Chop off, 1 LSB = 19.84 µV (in unipolar mode)	−10	±3	+10	LSB
Offset Error[1, 3]	Chop on	−5	+1	+5	LSB
Offset Error Drift[1]	Chop off		0.03		LSB/°C
Total Gain Error[1, 3, 10]	VREF = (REG_AVDD, GND_SW)/2	−0.25	±0.06	+0.25	%
Gain Drift[1]			3		ppm/°C
Output Noise[1]	1 kHz update rate		7.5	11.25	µV rms

ABSOLUTE MAXIMUM RATINGS

T_A = −40°C to +115°C, unless otherwise noted.

Table 2.

Parameter	Rating
AGND to DGND to VSS to IO_VSS	−0.3 V to +0.3 V
VBAT to AGND	−22 V to +40 V
VDD to VSS	−0.3 V to +40 V
LIN to IO_VSS	−16 V to +40 V
LIN Short-Circuit Current[1]	200 mA
Digital I/O Voltage to DGND	−0.3 V to REG_DVDD + 0.3 V
ADC Inputs to AGND	−0.3 V to REG_AVDD + 0.3 V
ESD (HBM) Rating	
HBM-ADI0082 (Based on ANSI/ESD STM5.1-2007); All Pins Except LIN and VBAT	2.5 kV
LIN and VBAT	±6 kV
IEC61000-4-2 for LIN and VBAT	±7 kV
Storage Temperature	150°C
Junction Temperature	
Transient	150°C
Continuous	130°C
Lead Temperature	
Soldering Reflow (15 sec)	260°C

[1] 200 mA can be sustained on the LIN pin for 2 seconds. The active internal short circuit protection HVCFG[1] = 0 is required to be enabled on this device during LIN operation and is the default operation. This disconnects the LIN pin, if a short circuit event occurs, after the specified maximum period of 90 µs.

Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ESD CAUTION

ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

Pin No.	Mnemonic	Type[1]	Description
21	GPIO_0/$\overline{SS}$	I/O	General-Purpose Digital I/O 0, or SPI Interface. By default, this pin is configured as an input. The pin has an internal, weak, pull-up resistor and, when not in use, can be left unconnected. This multifunction pin can be configured in one of two states, namely General-Purpose Digital I/O 0. SPI interface, slave select input.
22	GPIO_1/SCLK	I/O	General-Purpose Digital I/O 1 or SPI Interface. By default, this pin is configured as an input. The pin has an internal, weak, pull-up resistor and, when not in use, can be left unconnected. This multifunction pin can be configured in one of two states, namely General-Purpose Digital I/O 1. SPI interface, serial clock input.
23	GPIO_2/MISO	I/O	General-Purpose Digital I/O 2 or SPI Interface. By default, this pin is configured as an input. The pin has an internal, weak, pull-up resistor and, when not in use, can be left unconnected. This multifunction pin can be configured in one of two states, namely General-Purpose Digital I/O 2. SPI interface, master input/slave output pin.
24	GPIO_3/MOSI	I/O	General-Purpose Digital I/O 3 or SPI Interface. By default, this pin is configured as an input. The pin has an internal, weak, pull-up resistor and, when not in use, can be left unconnected. This multifunction pin can be configured in one of two states, namely General-Purpose Digital I/O 3. SPI interface, master output/slave input pin.
25	GPIO_4/IRQ1	I/O	General-Purpose I/O 4/External Interrupt Request 1. By default, this pin is configured as an input. The pin has an internal, weak, pull-up resistor and when not in use, can be left unconnected.
26	NC		No Connect. This pin is not internally connected, but is reserved for possible future use. Therefore, do not externally connect this pin.
27	VDD	S	Battery Power Supply to On-Chip Regulator.
28	VBAT	I	Battery Voltage Input to Resistor Divider.
29	LIN	I/O	LIN Serial Interface Input/Output Pin.
30	IO_VSS	S	Ground Reference for LIN Pin.
31	VSS	S	Ground Reference. This is the ground reference for the internal voltage regulators.
32	GPIO_5	I/O	General-Purpose I/O 5. By default, this pin is configured as an input. The pin has an internal, weak, pull-up resistor and when not in use, can be left unconnected.
EPAD	EPAD		The exposed pad is internally connected to DGND.

[1] I = input, O = output, S = supply.

THEORY OF OPERATION

The ADuC7039 is a complete system solution for battery monitoring in 12 V automotive applications. This device integrates all of the required features to precisely and intelligently monitor, process, and diagnose 12 V battery parameters including battery current, voltage, and temperature over a wide range of operating conditions.

Minimizing external system components, the device is powered directly from the 12 V battery. An on-chip, low dropout regulator generates the supply voltage for two integrated, 16-bit, Σ-Δ ADCs. The ADCs precisely measure battery current, voltage, and temperature to characterize the state of health and charge of the car battery.

A Flash/EE memory-based ARM7™ microcontroller (MCU) is also integrated on-chip. It is used to both preprocess the acquired battery variables and to manage communications from the ADuC7039 to the main electronic control unit (ECU) via a local interconnect network (LIN) interface that is integrated on-chip.

The MCU can be configured to operate in normal or flexible power saving modes of operation.

In its normal operating mode, the MCU is clocked indirectly from an on-chip oscillator via the phase-locked loop (PLL) at a maximum clock rate of 10.24 MHz. In its power saving operating modes, the MCU can be totally powered down, waking up only in response to the wake-up timer, a POR, or a LIN communication event.

The ADC can be configured to operate in a normal (full power) mode of operation, interrupting the MCU after various sample conversion events.

On-chip factory firmware supports in-circuit Flash/EE reprogramming via the LIN or JTAG serial interface ports, and nonintrusive emulation is also supported via the JTAG interface. These features are incorporated into a low cost QuickStart™ development system supporting the ADuC7039.

The ADuC7039 operates directly from the 12 V battery supply and is fully specified over a temperature range of −40°C to +115°C. The ADuC7039 is functional, but with degraded performance, at temperatures from 115°C to 125°C.

OVERVIEW OF THE ARM7TDMI-S CORE

The ARM7 core is a 32-bit, reduced instruction set computer (RISC), developed by ARM® Ltd. The ARM7TDMI-S is a von Neumann-based architecture, meaning that it uses a single 32-bit bus for instruction and data. The length of the data can be 8, 16, or 32 bits, and the length of the instruction word is either 16 bits or 32 bits, depending on the mode in which the core is operating.

The ARM7TDMI-S is an ARM7 core with four additional features, as listed in Table 5.

Table 5. ARM7TDMI-S

Feature	Description
T	Support for the Thumb® (16-bit) instruction set
D	Support for debug
M	Enhanced multiplier
I	Includes the EmbeddedICE™ module to support embedded system debugging

Thumb Mode (T)

An ARM instruction is 32 bits long. The ARM7TDMI-S processor supports a second instruction set compressed into 16 bits, the Thumb instruction set. Faster code execution from 16-bit memory and greater code density can be achieved by using the Thumb instruction set, making the ARM7TDMI-S core particularly suited for embedded applications.

However, the Thumb mode has three limitations.

- Relative to ARM, the Thumb code usually requires more instructions to perform that same task. Therefore, ARM code is best for maximizing the performance of time-critical code in most applications.
- The Thumb instruction set does not include some instructions that are needed for exception handling, so ARM code can be required for exception handling.
- When an interrupt occurs, the core vectors to the interrupt location in memory and executes the code present at that address. The first command is required to be in ARM code.

Multiplier (M)

The ARM7TDMI-S instruction set includes an enhanced multiplier, with four extra instructions to perform 32-bit by 32-bit multiplication with a 64-bit result, and 32-bit by 32-bit multiplication-accumulation (MAC) with a 64-bit result.

EmbeddedICE (I)

The EmbeddedICE module provides integrated on-chip debug support for the ARM7TDMI-S. The EmbeddedICE module contains the breakpoint and watchpoint registers that allow nonintrusive user code debugging. These registers are controlled through the JTAG test port. When a breakpoint or watchpoint is encountered, the processor halts and enters the debug state. Once in a debug state, the processor registers can be interrogated, as can the Flash/EE, SRAM, and memory mapped registers.

The remap command must be executed from the absolute Flash/EE address, and not from the mirrored, remapped segment of memory, because this may be replaced by SRAM. If a remap operation is executed while operating code from the mirrored location, prefetch/data aborts can occur, or the user can observe abnormal program operation.

Any kind of reset logically remaps the Flash/EE memory to the bottom of the memory array.

SYSMAP Register

Name: SYSMAP

Address: 0xFFFF0220

Default Value: Updated by the kernel

Access: Read/write

Function: This 8-bit register allows user code to remap either RAM or Flash/EE space into the bottom of the ARM memory space starting at Address 0x00000000.

Table 7. SYSMAP MMR Bit Designations

Bit	Description
7 to 1	Reserved. These bits are reserved and should be written as 0 by user code.
0	Remap bit. This bit is set by the user to remap the SRAM to 0x00000000. This bit is cleared automatically after reset to remap the Flash/EE memory to 0x00000000.

RESET

There are four kinds of reset: external reset, power-on-reset, watchdog reset, and software reset. The RSTSTA register indicates the source of the last reset and can also be written by user code to initiate a software reset event. The bits in this register can be cleared to 0 by writing to the RSTCLR MMR at 0xFFFF0234. The bit designations in RSTCLR mirror those of RSTSTA. These registers can be used during a reset exception service routine to identify the source of the reset. The implications of all four kinds of reset event are tabulated in Table 9.

RSTSTA Register

Name: RSTSTA

Address: 0xFFFF0230

Default Value: N/A

Access: Read/write

Function: This 8-bit register indicates the source of the last reset event and can also be written by user code to initiate a software reset.

RSTCLR Register

Name: RSTCLR

Address: 0xFFFF0234

Access: Write only

Function: This 8-bit write-only register clears the corresponding bit in RSTSTA.

Table 8. RSTSTA/RSTCLR MMR Bit Designations

Bit	Description
7 to 4	Not used. These bits are not used and always read as 0.
3	External reset. This bit is set by hardware when an external reset occurs. This bit is cleared by setting the corresponding bit in RSTCLR.
2	Software reset. This bit is set by user code to generate a software reset. This bit is cleared by setting the corresponding bit in RSTCLR.[1]
1	Watchdog timeout. This bit is set by hardware when a watchdog timeout occurs. This bit is cleared by setting the corresponding bit in RSTCLR.
0	Power-on reset. This bit is set by hardware when a power-on-reset occurs. This bit is cleared by setting the corresponding bit in RSTCLR.

[1] If the software reset bit in RSTSTA is set, any write to RSTCLR that does not clear this bit generates a software reset.

Table 9. Device Reset Implications

	Impact							
RESET	Reset External Pins to Default State	Kernel Executed	Reset All External MMRs (Excluding RSTSTA)	Reset All HV Indirect Registers	Peripherals Reset	Watchdog Timer Reset	RAM Valid[1]	RSTSTA (Status After Reset Event)
POR	Yes	Yes	Yes	Yes	Yes	Yes	Yes/No[2]	RSTSTA[0] = 1
Watchdog	Yes	Yes	Yes	Yes	Yes	No	Yes	RSTSTA[1] = 1
Software	Yes	Yes	Yes	Yes	Yes	No	Yes	RSTSTA[2] = 1
External Pin	Yes	Yes	Yes	Yes	Yes	No	Yes	RSTSTA[3] = 1

[1] RAM is not valid in the case of a reset following LIN download.
[2] The impact on RAM is dependent on the HVSTA[2] contents if LVF is enabled. When LVF is enabled using HVCFG[4], RAM has not been corrupted by the POR reset mechanism if the LVF Status Bit HVSTA[2] is 1. See the Low Voltage Flag (LVF) section for more information.

Command Sequence for Executing a Mass Erase

Given the significance of the mass erase command, a specific code sequence must be executed to initiate this operation.

1. Ensure FEESTA is cleared.
2. Set Bit 3 in FEEMOD.
3. Write 0xFFC3 in FEEADR.
4. Write 0x3CFF in FEEDAT.
5. Run the mass erase command (0x06) in FEECON.

Command Sequence Example

The command sequence for excecuting a mass erase is illustrated in the following example:

```
int a = FEESTA;           // Ensure FEESTA is cleared
FEEMOD = 0x08
FEEADR = 0xFFC3
FEEDAT = 0x3CFF
FEECON = 0x06;            //Mass erase command
while (FEESTA & 0x04){}   //Wait for command to finish
```

FEESTA Register

Name: FEESTA

Address: 0xFFFF0E00

Default Value: 0xXXX0

Access: Read only

Function: This 16-bit, read-only register can be read by user code and reflects the current status of the Flash/EE memory controller.

Table 11. FEESTA MMR Bit Designation

Bit	Description
15 to 4	Reserved.
3	Flash/EE interrupt status bit. This bit is set automatically when an interrupt occurs, that is, when a command is complete and the Flash/EE interrupt enable bit in the FEEMOD register is set. This bit is cleared automatically when the FEESTA register is read by user code.
2	Flash/EE controller busy. This bit is set automatically when the Flash/EE controller is busy. This bit is cleared automatically when the controller is not busy.
1	Command fail. This bit is set automatically when a command written to FEECON completes unsuccessfully. This bit is cleared automatically when the FEESTA register is read by user code.
0	Command successful. This bit is set automatically by MCU when a command is completed successfully. This bit is cleared automatically when the FEESTA register is read by user code.

FLASH/EE MEMORY SIGNATURE

The entire 62 kB or the part of Flash/EE memory available to the user can be signed using the FEESIG register and signature command.

This feature automatically reads the code in that section of the memory specified by the FEEADR and FEEDAT MMRS:

- FEEADR contains an address situated in the first half page of the section to be signed.
- FEEDAT contains an address situated in the first half page above the last page of the section to be signed. See Figure 7 in this example, Page 0 and Page 1 are signed.

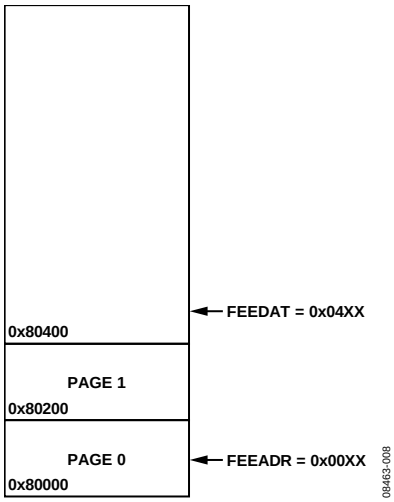

Figure 7. Signature Command Indexing

- If the 8 MSB of FEEADR and FEEDAT are identical, that is, the MMRS point to the same page, nothing is signed.
- The last two 16-bit locations are not included in the signature; they are reserved for the user-programmed signature.
- It is possible to sign half pages, by specifying a half page address in FEEADR and FEEDAT. For example, to sign the second half of Page 0 and the first half of Page 1, FEEADR = 0x0100 and FEEDAT = 0x0300.

This feature is also used by the on-chip kernel at power-up to check the validity of Page 0 before jumping to user code. Store the signature of Page 0 at Address 0x801FC when programming the device. See the ADuC7039 Kernel section for more details.

Flash/EE Memory Signature Registers

Name:	FEESIG
Address:	0xFFFF0E18
Default Value:	Updated by kernel
Access:	Read only
Function:	This MMR contains a 24-bit signature of the Flash/EE memory.

Example of User Code Signature

```
Int a = FEESTA;          // Ensure FEESTA is cleared
FEEADR = 0x0000;         // Start page address
FEEDAT = 0x0600;         // Stop (page + 1) address
FEECON = 0x0B;           // Signs Page 0 to Page 2 excluding
                         // Address 0x805FC
while (FEESTA & 0x04){}  // Wait for command to finish
```

User code can compare the content of FEESIG with the content of Address 0x805FC.

Polynomial

A software routine is provided by Analog Devices, Inc., to calculate the unique 24-bit signature.

In summary, there are three levels of protection as follows.

Temporary Protection

Temporary protection can be set and removed by writing directly into FEEHID MMR. This register is volatile and, therefore, protection is only in place for as long as the part remains powered on. This protection is not reloaded after a power cycle.

Keyed Permanent Protection

Keyed permanent protection can be set via FEEPRO to lock the protection configuration. The software key used at the start of the required FEEPRO write sequence is saved one time only and must be used for any subsequent access of the FEEHID or FEEPRO MMRs. A mass erase sets the software protection key back to 0xFFFF but also erases the entire user code space.

Permanent Protection

Permanent protection can be set via FEEPRO, similarly to keyed permanent protection, the only difference being that the software key used is 0xDEADDEAD. When the FEEPRO write sequence is saved, only a mass erase sets the software protection key back to 0xFFFFFFFF. This also erases the entire user code space.

Sequence to Write the Software Protection Key and Set Permanent Protection

1. Write in FEEPRO corresponding to the pages to be protected.
2. Write the new (user-defined) 32-bit software protection key in FEEADR (Bits[31:16]) and FEEDAT (Bits[15:0]).
3. Write 1, 0 in FEEMOD (Bits[6:5]) and set FEEMOD (Bit 3).
4. Run the Write Key Command 0x0C in FEECON.

To remove or modify the protection, the same sequence can be used with a modified value of FEEPRO.

The previous sequence for writing the key and setting permanent protection is illustrated in the following example, this protects writing Page 8 to Page 11 of the Flash/EE.

Sequence Example

```
Int a = FEESTA;       // Ensure FEESTA is cleared
FEEPRO = 0xFFFFFFFB;  // Protect Page 8 to Page 11
FEEADR = 0x66BB;      // 32-bit key value (Bits[31:16])
FEEDAT = 0xAA55;      // 32-bit key value (Bits 15:0])
FEEMOD = 0x0048       // Lock security sequence
FEECON = 0x0C;        // Write key command
while (FEESTA & 0x04){}
                      // Wait for command to finish
```

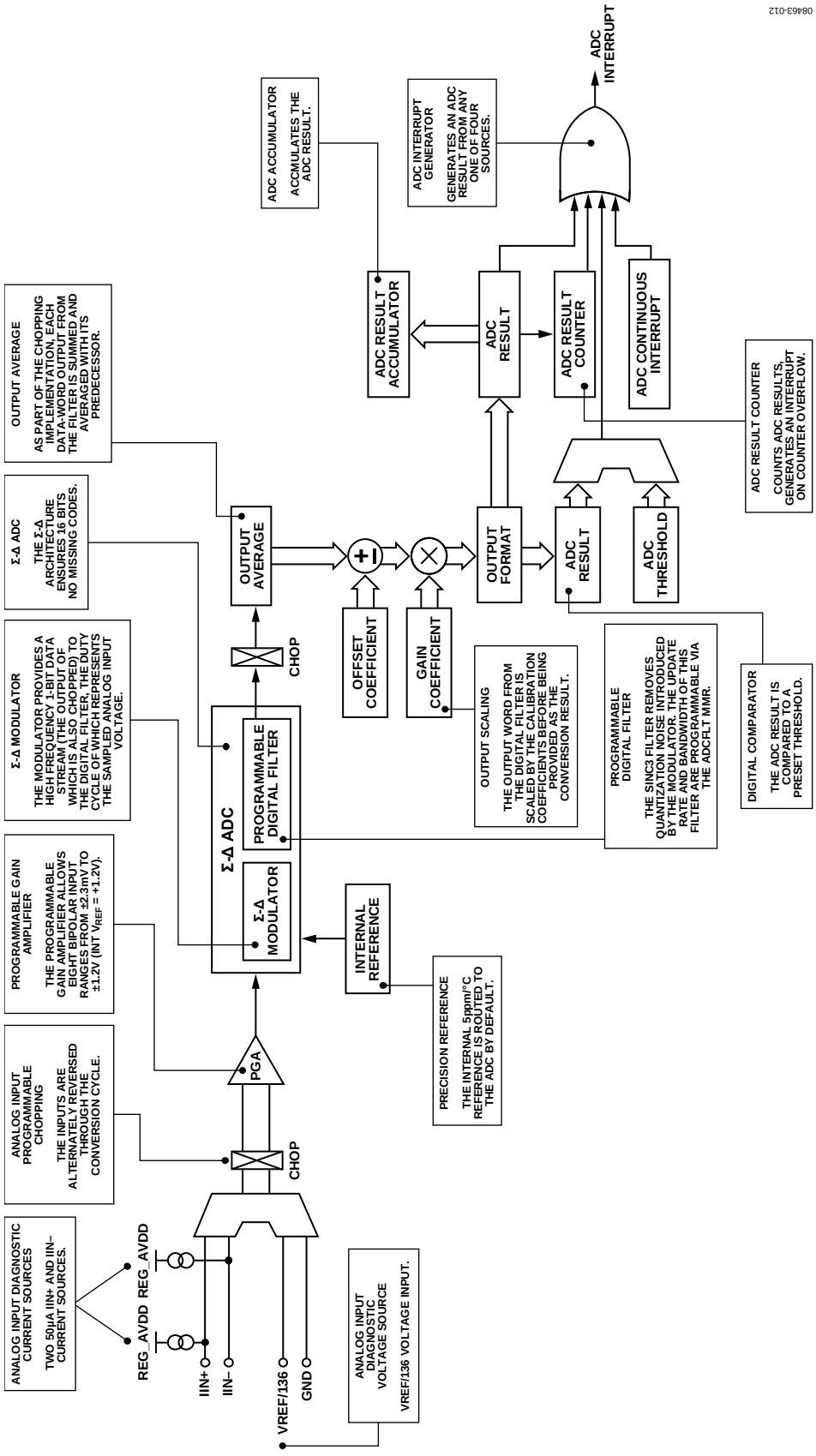

Figure 11. Current ADC, Top Level Overview

Voltage/Temperature Channel ADC (V/T-ADC)

The voltage/temperature channel ADC (V/T-ADC) converts additional battery parameters such as voltage and temperature. The input to this channel can be multiplexed from an external voltage and an on-chip temperature sensor.

As with the current channel ADC described previously, the V/T-ADC employs an identical Σ-Δ conversion technique, including a modified sinc3 low-pass filter to give a valid 16-bit data conversion result at programmable output rates from 10 Hz to 1 kHz. An external RC filter network is not required because this is internally implemented in the voltage channel.

The external battery voltage (VBAT) is routed to the ADC input via an on-chip, high voltage (divide-by-24), resistive attenuator. This ADC channel, unlike the current channel, has a fixed input range of 0 V to 28.8 V on VBAT.

The battery temperature can be derived through the on-chip temperature sensor.

By default, the time to a first valid (fully settled) result after an input channel switch on the voltage/temperature channel is three ADC conversion cycles with chop mode turned off.

A top level overview of the ADC signal chain is shown in Figure 12.

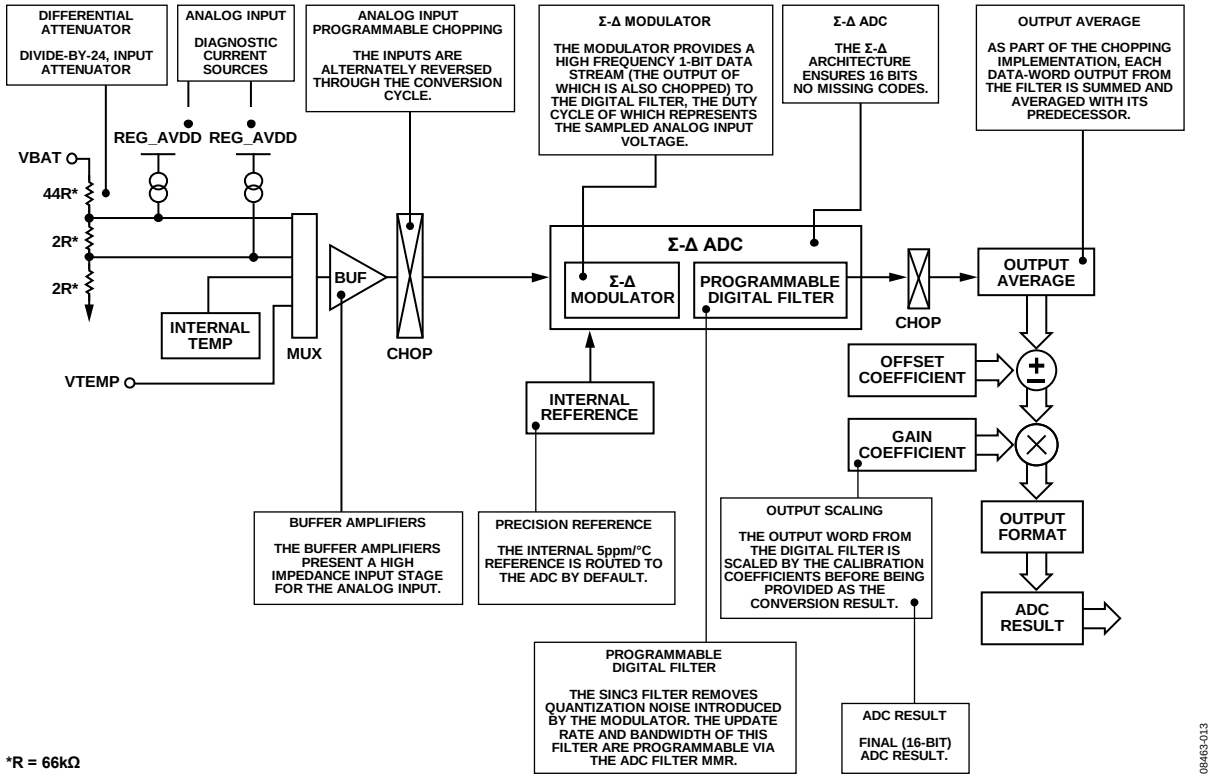

Figure 12. Voltage/Temperature ADC, Top Level Overview

ADC GROUND SWITCH

The ADuC7039 features an integrated ground switch pin, GND_SW, Pin 9. This switch allows the user to dynamically disconnect ground from external devices and allows a connection to ground using a 20 kΩ resistor, reducing the number of external components required for an NTC circuit as shown in Figure 13. The ground switch feature can be used for reducing power consumption on application specific boards.

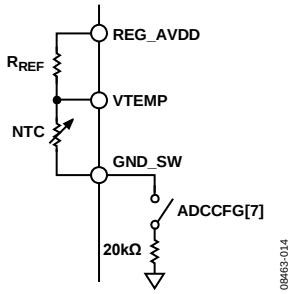

Figure 13. Internal Ground Switch Configuration

ADC NOISE PERFORMANCE TABLES

Table 24, Table 25, and Table 26 list the output rms noise in microvolts for some typical output update rates on the I- and V/T-ADCs. The numbers are typical and are generated at a differential input voltage of 0 V (I-ADC), 4V (V-ADC) and 0.1 V (T-ADC). The output rms noise is specified as the standard deviation (or 1 Σ) of the distribution of ADC output codes collected when the ADC input voltage is at a dc voltage. It is expressed as μV rms.

Table 24. Current Channel ADC, Normal Power Mode, Typical Output RMS Noise

		ADC Input Range		
ADCFLT	**Data Update Rate**	**±2.3 mV (512)**	**±37.5 mV (32)**	**±300 mV (4)[1]**
0x961F	10 Hz	0.065 μV	0.087 μV	0.7 μV
0x007F	50 Hz	0.144 μV	0.170 μV	0.7 μV
0x0007	1 kHz	0.663 μV	0.780 μV	2.6 μV

[1] The maximum absolute input voltage allowed is −200 mV to +300 mV relative to ground.

Table 25. Voltage Channel ADC, Typical Output RMS Noise (Referred to ADC Voltage Attenuator Input)

ADCFLT	Data Update Rate	28.8 V ADC Input Range
0x961F	10 Hz	65 μV
0x007F	50 Hz	65 μV
0x0007	1 kHz	180 μV

Table 26. Temperature Channel ADC, Typical Output RMS Noise

ADCFLT	Data Update Rate	0 V to 1.2 V ADC Input Range
0x961F	10 Hz	2.8 μV
0x007F	50 Hz	2.8 μV
0x0007	1 kHz	7.5 μV

PLLSTA Register

Name: PLLSTA

Address: 0xFFFF0400

Default Value: 0xXX

Access: Read only

Function: This 8-bit register allows user code to monitor the lock state of the PLL.

Table 37. PLLSTA MMR Bit Designations

Bit	Description
7 to 2	Reserved.
1	PLL lock status bit, read only. This bit is set automatically when the PLL is locked and outputting 20.48 MHz. This bit is cleared automatically when the PLL is not locked and outputting an f_{CORE} divide-by-8 clock source.
0	PLL interrupt. Set this bit if the PLL lock status bit signal goes low. This bit is cleared by user code when writing 1 to this bit.

PLLCON Prewrite Key PLLKEY0

Name: PLLKEY0

Address: 0xFFFF0410

Access: Write only

Key: 0x000000AA

Function: PLLCON is a keyed register that requires a 32-bit key value to be written before and after PLLCON. PLLKEY0 is the prewrite key.

PLLCON Postwrite Key PLLKEY1

Name: PLLKEY1

Address: 0xFFFF0418

Access: Write only

Key: 0x00000055

Function: PLLCON is a keyed register that requires a 32-bit key value to be written before and after PLLCON. PLLKEY1 is the postwrite key.

PLLCON Register

Name: PLLCON

Address: 0xFFFF0414

Default Value: 0x00

Access: Read/write

Function: This 8-bit register allows user code to dynamically select the PLL source clock from two different oscillator sources.

A typical sequence for starting the LIN calibration is as follows.

```
LOCMIN = EXPECTED_LINBR_VALUE-0x20;              //Define tolerance
LOCMAX = EXPECTED_LINBR_VALUE+0x20;              //Define tolerance
LOCKEY = 0x1324;                                 //Unlock key protection
LOCCON = 0x1;                                    //Enable calibration with step size = 1
```

Sequence Example

An example to calibrate the low power oscillator using LIN communication:

```
LINCON = 0x800;                                  //Enable LIN
expected_baudrate = 0x10AB;                      //Correspond to 19200 bps
LOCMIN = EXPECTED_LINBR_VALUE-0x20;              //Define tolerance
LOCMAX = EXPECTED_LINBR_VALUE+0x20;              //Define tolerance
LOCKEY = 0x1324;                                 //Unlock key protection
LOCCON = 0x1;                                    //Enable calibration with step size = 1
while ((LOCSTA & 0x05)!= 0x05){}                 //Wait for the trim value to be modified
while ((LINBR<LOCMIN)||(LINBR>LOCMAX)){}         //Wait for the correct baud rate
temp_trim = LOCVAL0;                             //Store the trim value given
LOCKEY = 0x1324;                                 //Unlock key protection
LOCUSR0 = temp_trim;                             //Write the trim value into the user MMR
LOCKEY = 0x1324;                                 //Unlock key protection
LOCCON = 0;                                      //Turn off the LIN calibration block
```

TIMER0—GENERAL-PURPOSE TIMER

Timer0 is a general-purpose 16-bit count-up/count-down timer. Timer0 is clocked from the core clock with a prescalar of either 1 or 16,384. This gives a minimum resolution of 1.6 ms with a prescalar of 16,384, and the timer can count for more than 1 minute.

Timer0 can count up or count down. A 16-bit value can be written to T0LD that is loaded into the counter. The current counter value can be read from T0VAL. Timer0 reloads the value from T0LD either when Timer0 overflows.

The Timer0 interface consists of four MMRs.

- T0LD is a 16-bit register that holds the 16-bit value that is loaded into the counter.
- T0VAL is a 16-bit register that holds the 16-bit current value of Timer0.
- T0CLRI is an 8-bit register. Writing any value to this register clears the Timer0 interrupt.
- T0CON is a 16-bit configuration register described in Table 46.

Timer0 Load Registers

Name: T0LD

Address: 0xFFFF0300

Default Value: 0x0000

Access: Read/write

Function: T0LD is the 16-bit register holding the 16-bit value that is loaded into the counter.

Timer0 Value Registers

Name: T0VAL

Address: 0xFFFF0304

Default Value: 0x0000

Access: Read only

Function: T0VAL is a 16-bit register that holds the current value of Timer0.

Timer0 Control Register

Name: T0CON

Address: 0xFFFF0308

Default Value: 0x0000

Access: Read/write

Function: This 16-bit MMR configures the mode of operation for Timer0.

Timer0 Clear Register

Name: T0CLRI

Address: 0xFFFF030C

Access: Write only

Function: This 8-bit, write-only MMR is written (with any value) by user code to clear the interrupt.

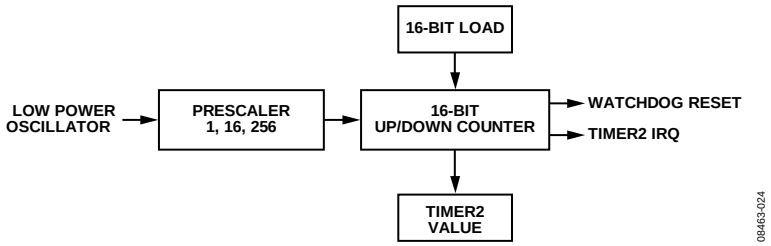

Figure 27. Timer2 Block Diagram

Table 48. T2CON MMR Bit Designations

Bit	Description
15 to 9	Reserved. These bits are reserved and should be written as 0 by user code.
8	Count up/count down enable. This bit is set by user code to configure Timer2 to count up. This bit is cleared by user code to configure Timer2 to count down.
7	Timer2 enable. This bit is set by user code to enable Timer2. This bit is cleared by user code to disable Timer2.
6	Timer2 operating mode. This bit is set by user code to configure Timer2 to operate in periodic mode. This bit is cleared by user to configure Timer2 to operate in free running mode.
5	Watchdog timer mode enable. This bit is set by user code to enable watchdog mode. This bit is cleared by user code to disable watchdog mode.
4	Reserved. This bit is reserved and should be written as 0 by user code.
3 to 2	Timer2 clock prescaler. 00 = source clock/1 (default). 01 = source clock/16. 10 = source clock/256. 11 = reserved.
1	Watchdog timer IRQ enable. This bit is set by user code to produce an IRQ instead of a reset when the watchdog reaches 0. This bit is cleared by user code to disable the IRQ option.
0	PD_OFF. This bit is set by the user code to stop Timer2 when the peripherals are powered down using Bit 4 in the POWCON MMR. This bit is cleared by the user code to enable Timer2 when the peripherals are powered down using Bit 4 in the POWCON MMR.

SPI Status Register

Name: SPISTA

Address: 0xFFFF0A00

Default Value: 0x0000

Access: Read only

Function: This 16-bit MMR contains the status of the SPI interface in both master and slave modes.

Table 54. SPISTA MMR Bit Designations

Bit	Description
15 to 12	Reserved bits.
11	SPI Rx FIFO excess bytes present. This bit is set when there are more bytes in the Rx FIFO than indicated in the SPIRXMDE bits in SPICON. This bit is cleared when the number of bytes in the FIFO is equal or less than the number in SPIRXMDE.
10 to 8	SPI Rx FIFO status bits. [000] = Rx FIFO is empty. [001] = 1 valid byte in the FIFO. [010] = 2 valid byte in the FIFO. [011] = 3 valid byte in the FIFO. [100] = 4 valid byte in the FIFO.
7	SPI Rx FIFO overflow status bit. This bit is set when the Rx FIFO was already full when new data was loaded to the FIFO. This bit generates an interrupt except when SPICON[12] is set. This bit is cleared when the SPISTA register is read.
6	SPI Rx IRQ status bit. This bit is set when a receive interrupt occurs. This bit is set when SPICON[6] is cleared and the required number of bytes have been received. This bit is cleared when the SPISTA register is read.
5	SPI Tx IRQ status bit. This bit is set when a transmit interrupt occurs. This bit is set when SPICON[6] is set and the required number of bytes have been transmitted. This bit is cleared when the SPISTA register is read.
4	SPI Tx FIFO underflow. This bit is set when a transmit is initiated without any valid data in the Tx FIFO. This bit generates an interrupt except when SPICON[13] is set. This bit is cleared when the SPISTA register is read.
3 to 1	SPI Tx FIFO status bits. 000 = Tx FIFO is empty. 001 = 1 valid byte in the FIFO. 010 = 2 valid byte in the FIFO. 011 = 3 valid byte in the FIFO. 100 = 4 valid byte in the FIFO.
0	SPI interrupt status bit. This bit is set to 1 when an SPI-based interrupt occurs. This bit is cleared after reading SPISTA.

HIGH VOLTAGE PERIPHERAL CONTROL INTERFACE

The ADuC7039 integrates a number of high voltage circuit functions that are controlled and monitored through a registered interface consisting of two MMRs, namely, HVCON and HVDAT. The HVCON register acts as a command byte interpreter allowing the microcontroller to indirectly read or write 8-bit data (the value in HVDAT) from or to one of two high voltage status/configuration registers. These high voltage registers are not MMRs but registers commonly referred to as indirect registers; that is, they can only be accessed indirectly via the HVCON and HVDAT MMRs.

The physical interface between the HVCON register and the indirect high voltage registers is a 2-wire (data and clock) serial interface based on a 2.56 MHz serial clock. Therefore, there is a finite, 10 µs (maximum) latency between the MCU core writing a command into HVCON and that command or data reaching the indirect high voltage registers. There is also a finite 10 µs latency between the MCU core writing a command into HVCON and indirect register data being read back into the HVDAT register. A busy bit (Bit 0 of the HVCON when read by MCU) can be polled by the MCU to confirm when a read/write command is complete.

Figure 29 describes the top-level architecture of the high voltage interface and related circuits. The LIN physical interface is controlled and monitored via this interface.

The high voltage interface consists of two MMRs and two indirect registers:

- HVCON is an 8-bit register acting as a command byte interpreter for the high voltage control interface. Bytes written to this register are interpreted as read or write commands to a set of two indirect registers related to the high voltage circuits. These commands are described in Table 56. The success of that operation can be monitored by reading back the HVCON MMR.
- HVDAT is a 12-bit register that is used to hold data to be written indirectly to and read indirectly from the high voltage interface registers.
- HVCFG is an 8-bit register controlling the function of high voltage circuits, accessed using HVCON and HVDAT.
- HVSTA is an 8-bit read-only register reflecting the state of the high voltage circuits. This register is not an MMR and does not appear in the MMR memory map. It is accessed through the HVCON registered interface, and data is read back from this register via HVDAT. In response to a high voltage interrupt event, the high voltage interrupt controller simultaneously and automatically loads the current value of the high voltage status register (HVSTA) into the HVDAT register.

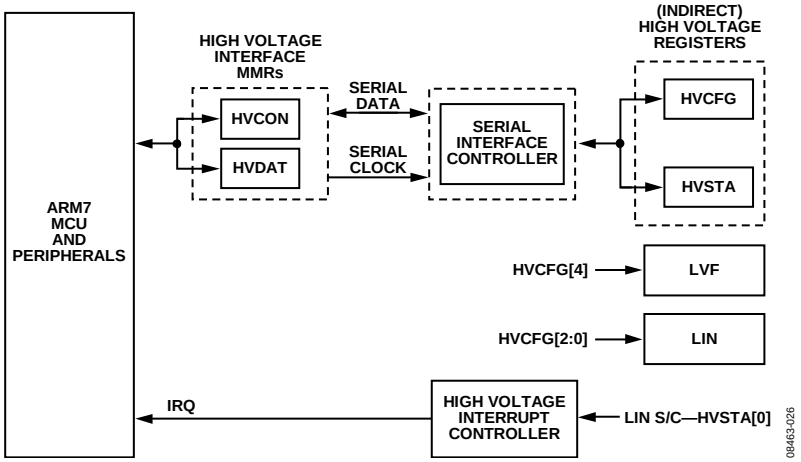

Figure 29. High Voltage Interface, Top-Level Block Diagram

High Voltage Interface Control Register

Name: HVCON

Address: 0xFFFF0804

Default Value: Updated by kernel

Access: Read/write

Function: This 8-bit register acts as a command byte interpreter for the high voltage control interface. Bytes written to this register are interpreted as read or write commands to a set of two indirect registers related to the high voltage circuits. The HVDAT register is used to store data to be written to, or read back from, the indirect registers.

Table 56. HVCON MMR Write Bit Designations

Bit	Description
7 to 0	Command byte. Interpreted as 0x00 = read back high voltage register, HVCFG, into HVDAT. 0x02 = read back high voltage status register, HVSTA, into HVDAT. 0x08 = write the value in HVDAT to the high voltage register, HVCFG. Other = reserved.

Table 57. HVCON MMR Read Bit Designations

Bit	Description
7 to 3	Reserved.
2	Transmit command to high voltage die status. 1 = command completed successfully. 0 = command failed.
1	Read command from high voltage die status. 1 = command completed successfully. 0 = command failed.
0	Bit 0 (read-only) busy bit. When user code reads this register, Bit 0 should be interpreted as the busy signal for the high voltage interface. This bit can be used to determine if a read request has completed. High voltage (read/write) commands as described in this table should not be written to HVCON unless busy = 0. Busy = 1, high voltage interface is busy and has not completed the previous command written to HVCON. Bit 1 and Bit 2 are not valid. Busy = 0, high voltage interface is not busy and has completed the command written to HVCON. Bit 1 and Bit 2 are valid.

High Voltage Data Register

Name: HVDAT

Address: 0xFFFF080C

Default Value: Updated by kernel

Access: Read/write

Function: HVDAT is a 12-bit register that is used to hold data to be written indirectly to, and read indirectly from, the HVDAT, HVSTA, HVCFG high voltage interface registers.

Table 58. HVDAT MMR Bit Designations

Bit	Description
11 to 8	Command with which the high voltage data, HVDAT[7:0], is associated. These bits are read-only and should be written as 0s. 0x00 = read back the high voltage register, HVCFG, into HVDAT. 0x02 = read back the high voltage status register, HVSTA, into HVDAT. 0x08 = write the value in HVDAT to the high voltage register, HVCFG.
7 to 0	High voltage data to read/write.

High Voltage Status Register

Name: HVSTA

Address: Indirectly addressed via the HVCON high voltage interface

Default Value: 0x00

Access: Read only, this register should only be read on a high voltage interrupt

Function: This 8-bit, read-only register reflects the state of the low voltage flag and the LIN short circuit interrupt status. This register is not an MMR and does not appear in the MMR memory map. It is accessed through the HVCON registered interface and data is read back from this register via HVDAT. In response to a high voltage interrupt event, the high voltage interrupt controller simultaneously and automatically loads the current value of the high voltage status register (HVSTA) into the HVDAT register.

Table 60. HVSTA Bit Designations

Bit	Description
7 to 3	Reserved. These bits should not be used and are reserved for future use.
2	Low voltage flag status bit. Valid only if enabled via HVCFG[4]. This bit is 0 on power-on if REG_DVDD has dropped below 2.1 V. In this state, RAM contents can be deemed corrupt. This bit is 1 on power-on if REG_DVDD has not dropped below 2.1 V. In this state, RAM contents can be deemed valid. It is only cleared by re-enabling the low voltage flag in HVCFG[4].
1	Reserved. This bit should not be used and is reserved for future use.
0	LIN short-circuit status interrupt. This bit is 0 during normal LIN operation and is cleared automatically by reading the HVSTA register. This bit is 1 if a LIN short circuit is detected. In this condition, the LIN driver is automatically disabled.

LOW VOLTAGE FLAG (LVF)

The ADuC7039 features a low voltage flag (LVF) that, when enabled, allows the user to monitor REG_DVDD (see the Low Voltage Flag (LVF) section). When enabled via HVCFG[4], the LVF can be monitored through HVSTA[2]. If REG_DVDD drops below 2.1 V, then HVSTA[2] is cleared and the RAM contents are corrupted. After the LVF is enabled, it is only reset by REG_DVDD dropping below 2.1 V or by disabling the LVF functionality using HVCFG[4].

HANDLING HV INTERFACE INTERRUPT AND HV COMMUNICATION

HV Interrupt

An interrupt controller is also integrated with the high voltage circuits. If enabled through IRQEN[10], a LIN short circuit event can assert the high voltage interrupt signal and interrupt the MCU core.

Although the normal MCU response to this interrupt event is to vector to the IRQ or FIQ interrupt vector address, the high voltage interrupt controller simultaneously and automatically loads the current value of the high voltage status register (HVSTA) into the HVDAT register. During this time, the busy bit in HVCON[0] is set to indicate the transfer is in progress and clears after 10 μs to indicate the HVSTA contents are available in HVDAT.

The interface should be interrupt driven. The interrupt handler can, therefore, poll the busy bit in HVCON until it deasserts. Once the busy bit is cleared, HVCON[1] must be checked to ensure the data was read correctly. Then the HVDAT register can be read. At this time, HVDAT holds the value of the HVSTA register.

Reading the HVSTA register clears the interrupt; therefore, it is not recommended to read HVSTA at any time.

RECOMMENDED SCHEMATIC

This schematic contains external components that are recommended for proper operation of the ADuC7039.

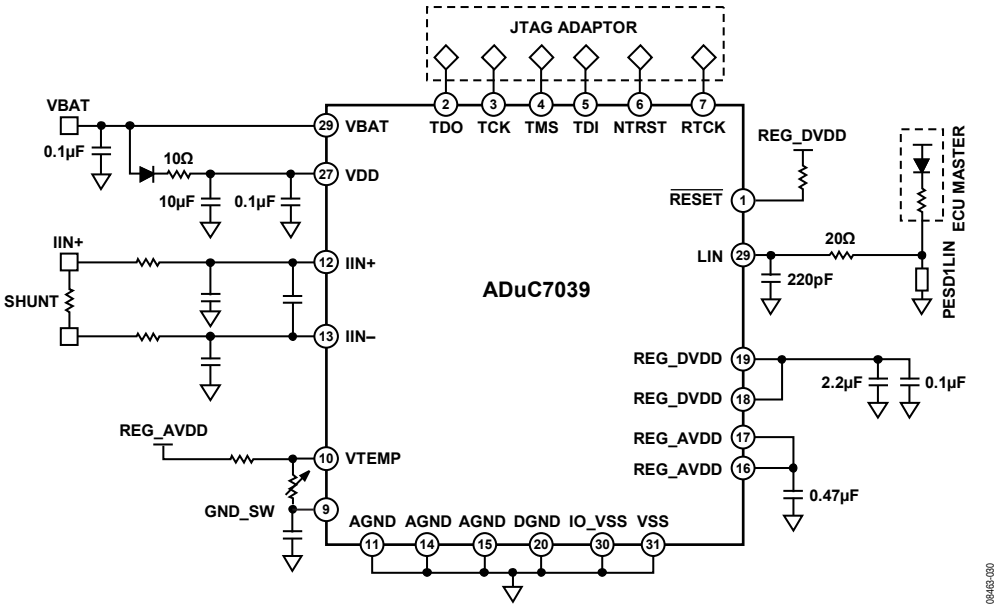

Figure 33. Recommended Schematic