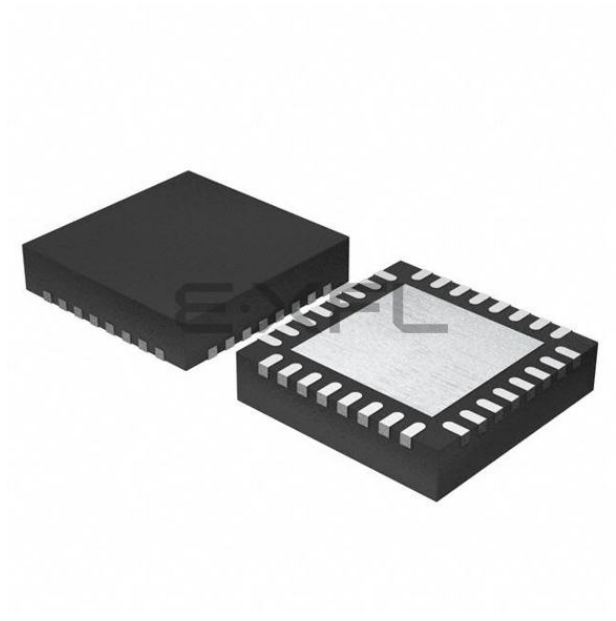


Welcome to E-XFL.COM

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/aduc7039bcp6z-rl

TABLE OF CONTENTS

Parameter	Test Conditions/Comments	Min	Typ	Max	Unit
ADC SPECIFICATIONS, ANALOG INPUT					
Current Channel					
Absolute Input Voltage Range[1]	Applies to both IIN+ and IIN−	−200		+300	mV
Input Voltage Range[11, 12]					
	Gain = 4		±300		mV
	Gain = 8		±150		mV
	Gain = 32		±37.5		mV
	Gain = 512		±2.3		mV
Input Leakage Current[1]		−3		+3	nA
Input Offset Current[1, 13]			0.5	1.5	nA
Voltage Channel					
Absolute Input Voltage Range[1]	Voltage ADC specifications are valid in this range	4		18	V
Input Voltage Range[1]			0 to 28.8		V
VBAT Input Current	VBAT = 18 V	3	5.5	8	µA
Temperature Channel	V_{REF} = (REG_AVDD, GND_SW)/2				
Absolute Input Voltage Range[14]		100		1300	mV
Input Voltage Range[1]			0 to V_{REF}		V
VTEMP Input Current[1]			2.5	100	nA
VOLTAGE REFERENCE					
Internal VREF			1.2		V
Power-Up Time[1]			0.5		ms
Initial Accuracy[1]	Measured at T_A = 25°C	−0.15		+0.15	%
Temperature Coefficient[1, 15]		−20	±5	+20	ppm/°C
Long-Term Stability[16]			100		ppm/1000 hr
ADC DIAGNOSTICS					
VREF/136 Accuracy[1]	At any gain setting, after self gaincalibration at chosen gain	8.4		9.4	mV
Voltage Attenuator Current	Differential voltage increase on the attenuator when current is on	2.3		3.6	V
Source Accuracy[1]					
RESISTIVE ATTENUATOR[1]					
Divider Ratio			24		
Resistor Mismatch Drift	Included in the voltage channel total gain error		±3		ppm/°C
ADC GROUND SWITCH					
Resistor to Ground		15	20	30	kΩ
TEMPERATURE SENSOR[17]	Uncalibrated				
Accuracy	MCU in power down or standby mode; T_A = −40°C to +85°C	−10	±3	+10	°C
	MCU in power down or standby mode; T_A = −20°C to +60°C	−8	±2	+8	°C
POWER-ON RESET (POR)[1]					
POR Trip Level	Refers to voltage at the VDD pin	2.85	3.0	3.15	V
POR Hysteresis			300		mV
Reset Timeout from POR			20		ms
LOW VOLTAGE FLAG(LVF)					
LVF Level	Refers to voltage at the REG_DVDD pin	1.9	2.1	2.3	V
WATCHDOG TIMER (WDT)					
Timeout Period[1]	32 kHz clock, 256 prescale	0.008		524	Seconds
Timeout Step Size			7.8		ms
FLASH/EE MEMORY[1]					
Endurance[18]		10,000			Cycles
Data Retention[19]		20			Years
DIGITAL INPUTS[1]	All digital inputs except NTRST				
Input Leakage Current	Input (high) = REG_DVDD		±1	±10	µA
Input Pull-Up Current[1]	Input (low) = 0 V	10	20	80	µA
Input Capacitance[1]			10		pF
Input Leakage Current	NTRST only; input (low) = 0 V		±1	±10	µA
Input Pull-Down Current	NTRST only; input (high) = REG_DVDD	30	55	100	µA
LOGIC INPUTS[1]	All logic inputs				
Input Low Voltage (VINL)				0.4	V
Input High Voltage (VINH)		2.0			V

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.

MEMORY ORGANIZATION

The ARM7, a von Neumann architecture, MCU core sees memory as a linear array of 2^{32} byte locations. As shown in Figure 5, the ADuC7039 maps this into four distinct user areas, namely: a memory area that can be remapped, an SRAM area, a Flash/EE area, and a memory mapped register (MMR) area.

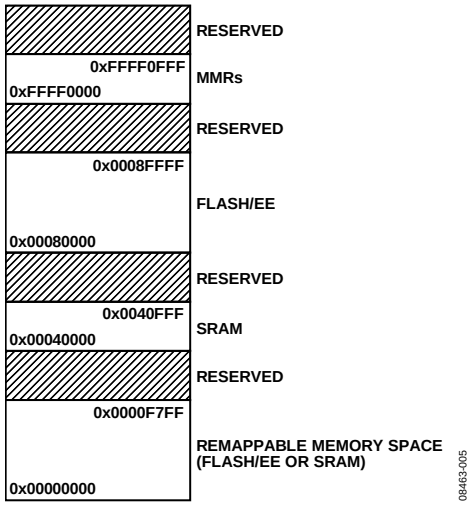


Figure 5. ADuC7039 Memory Map, 64 kB Flash Option

- The first 64 kB of this memory space is used as an area into which the on-chip Flash/EE or SRAM can be remapped.
- The ADuC7039 features a second 4 kB area at the top of the memory map used to locate the MMRs, through which all on-chip peripherals are configured and monitored.
- The ADuC7039 features an SRAM size of 4 kB.
- The ADuC7039 features 64 kB of on-chip Flash/EE memory. However, 62 kB of on-chip Flash/EE memory are available to the user. In addition, 2 kB are reserved for the on-chip kernel.

Any access, either reading or writing, to an area not defined in the memory map results in a data abort exception.

Memory Format

The ADuC7039 memory organization is configured in little endian format: the least significant byte is located in the lowest byte address, and the most significant byte in the highest byte address.

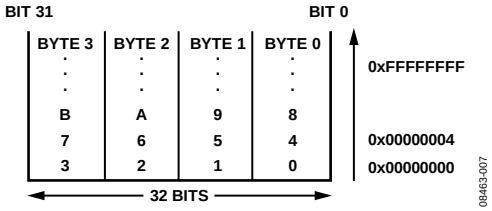


Figure 6. Little Endian Format

SRAM

The ADuC7039 features 4 kB of SRAM, organized as 1024 × 32 bits, that is, 1024 words, which is located at 0x40000.

The RAM space can be used as data memory and also as a volatile program space.

ARM code can run directly from SRAM at full clock speed given that the SRAM array is configured as a 32-bit wide memory array. SRAM is read/writeable in 8-, 16-, and 32-bit segments.

Remap

The ARM exception vectors are all situated at the bottom of the memory array, from Address 0x00000000 to Address 0x00000020.

By default, after a reset, the Flash/EE memory is logically mapped to Address 0x00000000. It is possible to logically remap the SRAM to Address 0x00000000. This is accomplished by setting Bit 0 of the SYSMAP MMR located at 0xFFFF0220. To revert Flash/EE to 0x00000000, Bit 0 of SYSMAP is cleared.

It is sometimes desirable to remap RAM to 0x00000000 to execute code from SRAM while erasing a page of Flash/EE memory.

Remap Operation

When a reset occurs on the ADuC7039, execution starts automatically in the factory programmed internal configuration code. This so-called kernel is hidden and cannot be accessed by user code. If the ADuC7039 is in normal mode, it executes the power-on configuration routine of the kernel and then jumps to the reset vector, Address 0x00000000, to execute the user's reset exception routine. Because the Flash/EE is mirrored at the bottom of the memory array at reset, the reset routine must always be written in Flash/EE.

FLASH/EE MEMORY

The ADuC7039 incorporates Flash/EE memory technology on chip to provide the user with nonvolatile, in-circuit reprogrammable memory space.

Like EEPROM, flash memory can be programmed in-system at a byte level, although it must first be erased, the erase being performed in page blocks. Thus, flash memory is often and more correctly referred to as Flash/EE memory.

Overall, Flash/EE memory represents a step closer to the ideal memory device that includes nonvolatility, in-circuit programmability, high density, and low cost. Incorporated within the ADuC7039, Flash/EE memory technology allows the user to update program code space in-circuit, without the need to replace one-time programmable (OTP) devices at remote operating nodes.

The Flash/EE memory is physically located at 0x80000. Upon a hard reset, it logically maps to 0x00000000. The factory default contents of all Flash/EE memory locations is 0xFFFF. Flash/EE can be read in 8-/16-/32-bit segments, and written in segments of 16 bits. The Flash/EE is rated for 10,000 endurance cycles. This rating is based on the number of times that each individual byte is cycled, that is, erased and programmed. Implementing a redundancy scheme in the software ensures a greater than 10,000-cycle endurance.

The user can also write data variables to the Flash/EE memory during run-time code execution, for example, for storing diagnostic battery parameter data.

The entire Flash/EE is available to the user as code and nonvolatile data memory. There is no distinction between data and program, because ARM code shares the same space. The real width of the Flash/EE memory is 16 bits, meaning that in ARM mode (32-bit instruction), two accesses to the Flash/EE are necessary for each instruction fetch. The ARM7TDMI-S operates at a default 10.24 MHz clock frequency, but the Flash/EE memory controller is operating at 20.48 MHz. This means that the Flash/EE memory controller can transparently fetch the second 16-bit half-word (part of the 32-bit ARM operation code) within a single core clock period.

The page size of this Flash/EE memory is 512 bytes. Typically, it takes the Flash/EE controller 20 ms to erase a page. To write a 16-bit word requires 50 μs.

It is possible to write to a single, 16-bit location at most twice between erases; that is, it is possible to walk bytes, not bits. If a location is written to more than twice, then it is possible to corrupt the contents of the Flash/EE page.

The Flash/EE memory can be programmed in-circuit, using a serial download mode via the LIN interface or the integrated JTAG port.

Serial Downloading (In-Circuit Programming)

The ADuC7039 facilitates code download via the LIN pin. The protocol is documented in the AN-946 Application Note, *Flash/EE Memory Programming via LIN (Protocol 6)*.

JTAG Access

The ADuC7039 features an on-chip JTAG debug port to facilitate code download and debug.

FLASH/EE MMR INTERFACE

Access to, and control of, the Flash/EE memory on the ADuC7039 is managed by an on-chip memory controller. The controller manages the Flash/EE memory as a single block of 64 kB.

Note that if executing from Flash/EE memory, the MCU core is halted until the command is completed. User software must ensure that the Flash/EE controller has completed any erase or write cycle before the PLL is powered down. If the PLL is powered down before an erase or write cycle is completed, the Flash/EE page can be corrupted. User code, LIN, and JTAG programming use the Flash/EE control interface, consisting of the following MMRs:

- FEESTA: read-only register, reflects the status of the Flash/EE control interface.
- FEEMOD: sets the operating mode of the Flash/EE control interface.
- FEECON: 8-bit command register. The commands are interpreted as described in Table 10.
- FEEDAT: 16-bit data register.
- FEEADR: 16-bit address register.
- FEESIG: holds the 24-bit code signature as a result of the signature command being initiated.
- FEEHID: protection MMR. Controls read and write protection of the Flash/EE memory code space. If previously configured via the FEEPRO register, FEEHID can require a software key to enable access.
- FEEPRO: a buffer of the FEEHID register that stores the FEEHID value, thus, it automatically downloads to the FEEHID registers on subsequent reset and power-on events.

The following sections provide detailed descriptions of the bit designations for each of the Flash/EE control MMRs.

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.

Normal kernel execution time, excluding LIN download, is less than 5 ms. It is only possible to enter and leave LIN download mode through a reset.

SRAM Address 0 to Address 0x2B are modified during normal kernel execution, SRAM Address 0xFF to Address 0x110 are also modified during a LIN download.

Note that even with NTRST = 0, user code is not executed unless Address 0x801FC contains either 0x16400000 or the CRC of Page 0. If Address 0x801FC does not contain this information, user code is not executed and LIN download mode is entered. During kernel execution, JTAG access is disabled.

The ADuC7039 is delivered with flash user space fully erased and if NTRST = 0 at first power-up, the LIN download mode is entered.

With NTRST = 1, user code is always executed and JTAG is enabled.

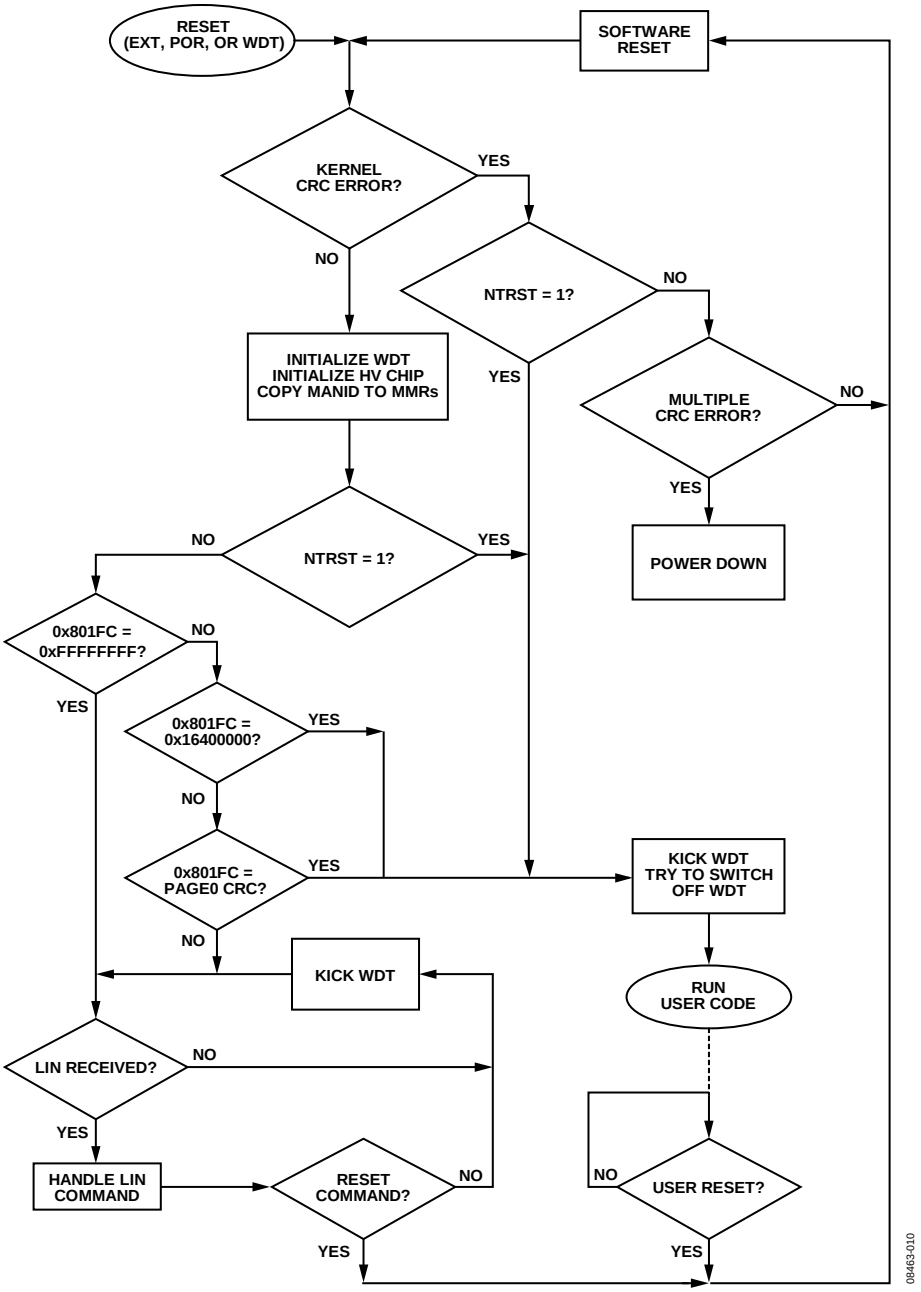


Figure 9. ADuC7039 Kernel Flowchart

ADC Configuration Register

Name: ADCCFG

Address: 0xFFFF051C

Default Value: 0x00

Access: Read/write

Function: The 8-bit ADC configuration MMR controls extended functionality related to the on-chip ADCs.

Table 34. ADCCFG MMR Bit Designations

Bit	Description
7	Analog ground switch enable. This bit is set to 1 by user software to connect the external GND_SW pin (Pin 9) to an internal 20 kΩ resistor to ground. This bit can be used to connect and disconnect external circuits and components to ground under program control and thereby minimize dc current consumption when the external circuit or component is not being used.
6 to 5	Not used. These bits are reserved for future functionality and should not be modified by user code.
4	Current channel ADC accumulator enable. 0 = accumulator disabled and reset to 0. The accumulator must be disabled for a full ADC conversion (ADCSTA[0] set twice) before the accumulator can be re-enabled to ensure the accumulator is reset. 1 = accumulator enabled.
3	Current channel ADC comparator enable. 0 = comparator disabled. 1 = comparator active, interrupt asserted if absolute value of I-ADC conversion result |I| ≥ ADC0TH.
2	Not used. This bit is reserved for future functionality and should be written as 0 by user code.
1	Fast temperature sensor mode. This bit is set by the user to enable the feature described in the Fast Temperature Conversion Mode section. When this bit is set, ADC1CON[7:6] is temporarily ignored to produce a single fully settled result of the internal temperature sensor. This fast conversion mode is applicable to the internal temperature sensor only. This bit should be cleared by the user when the fast conversion is complete.
0	Current channel ADC, result counter enable. This bit is set by the user to enable the result count mode. In this mode, an I-ADC interrupt is generated only when ADC0RCV = ADC0RCL. This allows the I-ADC to continuously monitor current but only interrupt the MCU core after a defined number of conversions. The voltage/temperature ADC also continues to convert if enabled, but again, only the last conversion result is available (intermediate V/T-ADC conversion results are not stored) when the ADC counter interrupt occurs.

Current Channel ADC Data Register

Name: ADC0DAT

Address: 0xFFFF0520

Default Value: 0x0000

Access: Read only

Function: This ADC data MMR holds the 16-bit conversion result from the I-ADC. The ADC does not update this MMR if the ADC0 conversion result ready bit (ADCSTA[0]) is set. A read of this MMR by the MCU clears all asserted ready flags (ADCSTA[2:0] and ADCSTA[5]).

Voltage/Temperature Channel ADC Data Register

Name: ADC1DAT

Address: 0xFFFF0524

Default Value: 0x0000

Access: Read only

Function: This ADC data MMR holds the 16-bit voltage (or temperature) conversion result from the V/T-ADC. The ADC does not update this MMR if the voltage (or temperature) conversion result ready bit (ADCSTA[1] or ADCSTA[2]) is set. If I-ADC is not active, a read of this MMR by the MCU clears all asserted ready flags (ADCSTA[2:1] and ADCSTA[5]).

In ADC low power mode, the ADC, Σ-Δ modulator clock is no longer driven at 512 kHz but is driven directly from the on-chip low power (128 kHz) oscillator. Subsequently, for the same ADCFLT configurations in normal mode, all filter values should be scaled by a factor of approximately four.

In general, it is possible to program different values of SF and AF in the ADCFLT register and achieve the same ADC update rate. In practical terms, the trade off with any value of ADCFLT is frequency response vs. ADC noise. For optimum filter response and ADC noise when using combinations of SF and AF, best practice suggests choosing an SF in the range of 16 decimal to 40 decimal, or 0x10 to 0x28, and then increasing the AF value to achieve the required ADC throughput. Table 35 shows some common ADCFLT configurations.

ADC MODES OF OPERATION

The ADCs can be configured into reduced (low power) or full power (normal) mode of operation by configuring ADCMDE[3] as appropriate. The ARM7 MCU can also be configured in low power modes of operation (POWCON[5:3]). The core power modes are independently controlled and are not related to the ADC power modes described in this section.

ADC Normal Power Mode

In normal mode, the current and voltage/temperature channels are fully enabled. The ADC modulator clock is 512 kHz and enables the ADCs to provide regular conversion results at a rate of between 10 Hz and 1 kHz (see the ADC Filter Register section). Both channels are under full control of the MCU and can be reconfigured at any time. The default ADC update rate for all channels in this mode is 1.0 kHz.

It is worth emphasizing that I-ADC and V/T-ADC channels can be configured to initiate periodic, normal power mode, high accuracy, single conversion cycles before returning to ADC full power-down mode. This flexibility is facilitated under full MCU control via the ADCMDE MMR; it ensures that continuous periodic monitoring of battery current, voltage, and temperature settings is feasible while ensuring the average dc current consumption is minimized.

In ADC normal mode, the PLL must not be powered down.

ADC Low Power Mode

In ADC low power mode, the I-ADC is enabled in a reduced power and reduced accuracy configuration. The ADC modulator clock is now driven directly from the on-chip 128 kHz low power oscillator. The gain of the ADC in this mode is fixed at 512.

All of the ADC peripheral functions (result counter, digital comparator and accumulator) can be enabled in low power mode.

Typically, in low power mode, the I-ADC only, is configured to run at a low update rate, continuously monitoring battery current. The MCU is in power-down mode and wakes up when the I-ADC interrupts the MCU. This happens after the I-ADC detects a current conversion beyond a preprogrammed threshold, setpoint, or a set number of conversions.

ADC Comparator and Accumulator

Every I-ADC result can also be compared to a preset threshold level (ADC0TH) as configured via ADCCFG[3]. An MCU interrupt is generated if the absolute (sign independent) value of the ADC result is greater than the preprogrammed comparator threshold level.

Finally, a 32-bit accumulator (ADC0ACC) function can be configured (ADCCFG[5]) allowing the I-ADC to add (or subtract) multiple I-ADC sample results. User code can read the accumulated value directly (ADC0ACC) without any further software processing.

ADC Continuous Interrupt Mode

Setting ADCMDE[5] allows the user to generate an ADC interrupt after each ADC conversion period, even if the ADC filter is not fully settled. The corresponding ADC interrupt bit in the ADCMSKI must also be set to allow the continuous interrupt. In this mode of operation, ADCSTA[2:0] are used as valid flags and should not be used to generate interrupts.

Table 35. Common ADCFLT Configurations

ADC Mode	SF	AF	Other Config.	ADCFLT	f_{ADC}	t_{SETTLE}
Normal	0x1F	0x16	Chop on	0x961F	10 Hz	0.2 sec
Normal	0x07	0x00	None	0x0007	1 kHz	3 ms
Normal	0x07	0x00	Sinc3 modify	0x0087	1 kHz	3 ms
Low Power	0x10	0x03	Chop on	0x8310	20 Hz	100 ms
Low Power	0x10	0x09	Chop on	0x8910	10 Hz	200 ms

The fast temperature option cannot be used on the first conversion after ADC power-on. It can only be set after at least the first ADC interrupt. Waiting for a valid ADC result is not necessary. Also, a restriction when using the fast temperature mode is to ensure that SF >=1. In addition, a conversion rate of 1 ms is recommended in this mode of operation. This is to ensure the fast result occurs simultaneously with the current channel result.

When changing the ADCs configuration, by writing ADCMDE, ADC0CON or ADCFLT, the fast temperature bit must also be cleared to ensure correct operation. This condition is similar to a first conversion after ADC power-on.

I-ADC DIAGNOSTICS

The ADuC7039 features the capability to detect open-circuit conditions on the application board. This is accomplished using the two current sources on IIN+ and IIN−; these are controlled via ADC0CON[14:13].

Note that these current sources have a tolerance of ±30%.

Table 38. PLLCON MMR Bit Designations

Bit	Description
7 to 1	Reserved. These bits should be written as 0 by user code.
0	PLL clock source.[1] 0 = lower power oscillator. 1 = precision oscillator.

[1] If the user code switches MCU clock sources, a dummy MCU cycle should be included after the clock switch is written to PLLCON.

POWCON Prewrite Key POWKEY0

Name: POWKEY0

Address: 0xFFFF0404

Access: Write only

Key: 0x00000001

Function: POWCON is a keyed register that requires a 32-bit key value to be written before and after POWCON. POWKEY0 is the prewrite key.

POWCON Postwrite Key POWKEY1

Name: POWKEY1

Address: 0xFFFF040C

Access: Write only

Key: 0x000000F4

Function: POWCON is a keyed register that requires a 32-bit key value to be written before and after POWCON. POWKEY1 is the postwrite key.

POWCON Register

Name: POWCON

Address: 0xFFFF0408

Default Value: 0x079

Access: Read/write

Function: This 12-bit register allows user code to dynamically enter various low power modes.

OSCVAL0 Register

Name: OSCVAL0

Address: 0xFFFF0448

Default Value: 0x00

Access: Read only

Function: This 9-bit counter is clocked from the 128 kHz precision oscillator.

OSCVAL1 Register

Name: OSCVAL1

Address: 0xFFFF044C

Default Value: 0x0000

Access: Read only

Function: This 10-bit counter is clocked from the low power, 128 kHz oscillator. Note that 11 bits can be read, but only 10 bits are used.

LIN Oscillator Calibration

A second calibration mechanism is provided on the ADuC7039 to calibrate the oscillators. This new feature allows for the calibration of the clocks relative to a synchronization element of a LIN packet. It is based on a fixed and predefined LIN baud rate. The new trim value is derived from a LIN communication. This method requires sufficient LIN transactions, approximately 1 for every 10 degrees change in temperature.

If the baud rate measured by the LIN peripheral in the LINBR MMR is outside the limits defined by user code in LOCMIN and LOCMAX MMR, the selected oscillator trim bit is modified accordingly to the options selected in the LIN oscillator calibration control register, that is, by the number of steps defined in LOCCON.

Two read-only trim registers indicate the trim currently used for each of the oscillators. The LIN oscillator calibration block modifies these two read-only registers and does not modify the user registers LOCUSRx. When a calibration is complete the LIN oscillator calibration can be disabled, but before being disabled, the value of the LOCVALx must be copied into the corresponding LOCUSRx register. An example is given in the next section. The status register indicates if the trim register of the selected oscillator has been altered.

There are 9 MMRs available:

LOCCON Register

Name: LOCCON

Address: 0xFFFF0480

Default Value: 0x00

Access: Read/write (key protected)

Function: Oscillator calibration via LIN control register.

Table 42. LOCCON MMR Bit Designations

Bit	Description
7 to 3	Reserved. Read 0.
2 to 1	Oscillator calibration step size. 0x00 = default value, step of 1. 0x01 = step of 2. 0x10 = step of 3. 0x11 = step of 4.
0	Oscillator calibration via LIN enabled. This bit is set by the user to enable automatic calibration of the selected oscillator, based on the LIN baud rate. This bit is cleared by the user to disable this automatic calibration.

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
```

IRQ

The IRQ is the exception signal to enter the IRQ mode of the processor. It is used to service the general-purpose interrupt handling of internal and external events.

All 32 bits are logically OR'ed to create a single IRQ signal to the ARM7TDMI-S core. The four 32-bit registers dedicated to IRQ follow.

IRQSIG

IRQSIG is a read-only register that reflects the status of the different IRQ sources. If a peripheral generates an IRQ signal, the corresponding bit in the IRQSIG is set; otherwise, it is cleared. The IRQSIG bits are cleared when the interrupt in the particular peripheral is cleared. All IRQ sources can be masked in the IRQEN MMR.

IRQEN

IRQEN provides the value of the current enable mask. When a bit is set to 1, the corresponding source request is enabled to create an IRQ exception. When a bit is set to 0, the corresponding source request is disabled or masked which does not create an IRQ exception. The IRQEN register cannot be used to disable an interrupt.

IRQCLR

IRQCLR is a write-only register that allows the IRQEN register to clear to mask an interrupt source. Each bit set to 1 clears the corresponding bit in the IRQEN register without affecting the remaining bits. The pair of registers, IRQEN and IRQCLR, allow independent manipulation of the enable mask without requiring an atomic read-modify-write.

IRQSTA

IRQSTA is a read-only register that provides the current enabled IRQ source status (effectively a logic AND of the IRQSIG and IRQEN bits). When set to 1, that source generates an active IRQ request to the ARM7TDMI-S core. There is no priority encoder or interrupt vector generation. This function is implemented in software in a common interrupt handler routine.

Fast Interrupt Request (FIQ)

The fast interrupt request (FIQ) is the exception signal to enter the FIQ mode of the processor. It is provided to service data transfer or communication channel tasks with low latency. The FIQ interface is identical to the IRQ interface and provides the second level interrupt (highest priority). Four 32-bit registers are dedicated to FIQ: FIQSIG, FIQEN, FIQCLR, and FIQSTA.

Bit 31 to Bit 1 of FIQSTA are logically OR'ed to create the FIQ signal to the core and to Bit 0 of both the FIQ and IRQ registers (FIQ source).

The logic for FIQEN and FIQCLR does not allow an interrupt source to be enabled in both IRQ and FIQ masks. A bit set to 1 in FIQEN clears, as a side effect, the same bit in IRQEN. Likewise, a bit set to 1 in IRQEN clears, as a side effect, the same bit in FIQEN. An interrupt source can be disabled in both IRQEN and FIQEN masks.

Programmed Interrupts

Because the programmed interrupts are not maskable, they are controlled by another register, SWICFG that writes into both IRQSTA and IRQSIG registers and/or the FIQSTA and FIQSIG registers at the same time.

The 32-bit register dedicated to software interrupt is SWICFG described in Table 45. This MMR allows the control of a programmed source interrupt.

Table 45. SWICFG MMR Bit Designations

Bit	Description
31 to 3	Reserved.
2	Programmed interrupt FIQ. Setting/clearing this bit corresponds to setting/clearing Bit 1 of FIQSTA and FIQSIG.
1	Programmed interrupt IRQ. Setting/clearing this bit corresponds to setting/clearing Bit 1 of IRQSTA and IRQSIG.
0	Reserved.

Note that any interrupt signal must be active for at least the minimum interrupt latency time, to be detected by the interrupt controller and to be detected by the user in the IRQSTA/FIQSTA register.

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.

GPIO Port Clear Register

Name: GPCLR

Address: 0xFFFF0D18

Default Value: N/A

Access: Write only

Function: This 32-bit MMR allows user code to individually bit address external GPIO pins to clear them low only. User code can accomplish this using the GPCLR MMR without having to modify or maintain the status of any other GPIO pins (as user code requires when using GPDAT).

Table 53. GPCLR MMR Bit Designations

Bit	Description
31 to 22	Reserved. These bits are reserved and should be written as 0 by user code.
21	Port 0.5 clear bit. This bit is set to 1 by user code to clear the external GPIO_5 pin low. If user software clears this bit to 0, it has no effect on the external GPIO_5 pin.
20	Port 0.4 clear bit. This bit is set to 1 by user code to clear the external GPIO_4 pin low. If user software clears this bit to 0, it has no effect on the external GPIO_4 pin.
19	Port 0.3 clear bit. This bit is set to 1 by user code to clear the external GPIO_3 pin low. If user software clears this bit to 0, it has no effect on the external GPIO_3 pin.
18	Port 0.2 clear bit. This bit is set to 1 by user code to clear the external GPIO_2 pin low. If user software clears this bit to 0, it has no effect on the external GPIO_2 pin.
17	Port 0.1 clear bit. This bit is set to 1 by user code to clear the external GPIO_1 pin low. If user software clears this bit to 0, it has no effect on the external GPIO_1 pin.
16	Port 0.0 clear bit. This bit is set to 1 by user code to clear the external GPIO_0 pin low. If user software clears this bit to 0, it has no effect on the external GPIO_0 pin.
15 to 0	Reserved. These bits are reserved and should be written as 0 by user code.

High Voltage Configuration Register

Name: HVCFG

Address: Indirectly addressed via the HVCON high voltage interface

Default Value: 0x00

Access: Read/write

Function: This 8-bit register controls the function of high voltage circuits on the ADuC7039. This register is not an MMR and does not appear in the MMR memory map. It is accessed via the HVCON registered interface. Data to be written to this register is loaded via the HVDAT MMR, and data is read back from this register via the HVDAT MMR.

Table 59. HVCFG Bit Designations

Bit	Description
7 to 5	Reserved.
4	Low voltage flag (LVF) enable bit. This bit is cleared to 0 to disable the LVF function. This bit is set to 1 to enable the LVF function. The low voltage flag can be interrogated via HVSTA[2]after power-up to determine if the REG_DVDD voltage previously dropped below 2.1 V.
3	Voltage attenuator diagnostic enable bit. This bit is set to 1 to turn on a current source, which adds a differential voltage to the voltage channel measurement. This bit is cleared to 0 to disable the voltage attenuator diagnostic.
2	LIN driver re-enable bit. This bit is set to 1 to re-enable the LIN driver that would have been disabled as a result of a short-circuit current event. This bit is cleared to 0 automatically.
1	Enable/disable LIN short-circuit protection. This bit is set to 1 to enable passive short-circuit protection on the LIN pin. In this mode, a short-circuit event on the LIN pin generates a high voltage interrupt (if enabled in IRQEN[10]) and asserts the appropriate status bit in HVSTA but does not disable the short-circuiting pin. This bit is cleared to 0 to enable active short-circuit protection on the LIN pin. In this mode, during a short-circuit event, the LIN pin generates a high voltage interrupt (IRQSTA[10]), asserts HVSTA[0], and automatically disables the short-circuiting pin. When disabled, the I/O pin can only be re-enabled by writing to HVCFG[2]. A LIN short circuit event must last longer than 20 µs to be detected.
0	LIN operating mode. 0 = LIN disabled. 1 = LIN enabled.

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.

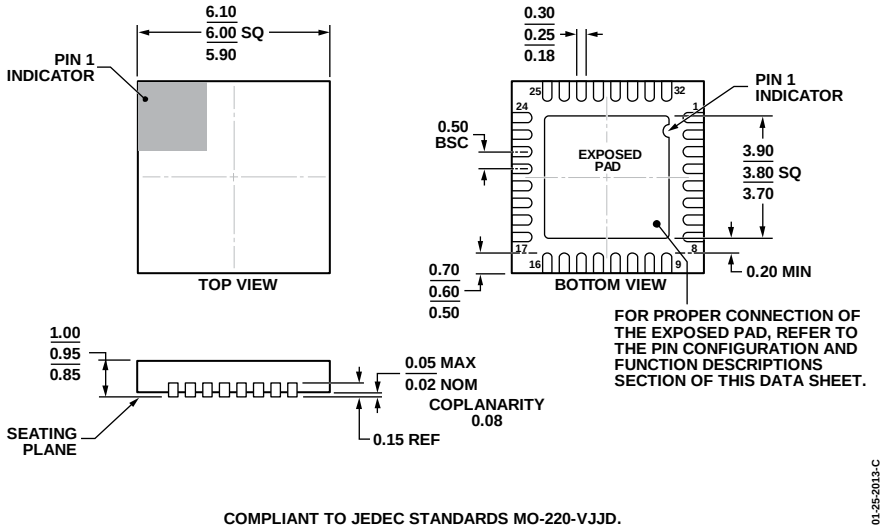
The interface should be interrupt driven. The interrupt handler can, therefore, poll the busy bit in HVCON until it deasserts. Once the busy bit is cleared, HVCON[1] must be checked to ensure the data was read correctly. Then the HVDAT register can be read. At this time, HVDAT holds the value of the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

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

Table 62. LINSTA MMR Bit Designations

Bits	Description
15 to 11	Reserved.
10	LIN wake-up interrupt. This bit is set if LIN woke up the ADuC7039. The wake-up functionality (LINWU MMR) is only used when POWCON[3] = 0. This bit is cleared automatically by a read of the LINSTA MMR.
9	Break time maximum. This bit is 0 if the first break symbol after enabling the LIN interface has ended before maximum count reached. This bit is 1 if the first break symbol after enabling the LIN interface has ended after maximum count reached. This bit is cleared automatically by hardware when reading LINSTA MMR. This bit is only valid on the first break symbol after enabling the LIN peripheral via LINCON[11].
8	PID parity error. This bit is set automatically by hardware if the current byte in the LINDAT register does not correctly match the parity scheme for a PID as described in the LIN 2.1 specifications. This bit is cleared by hardware if the current byte in the LINDAT register correctly matches the parity scheme for a PID. It is left to the user to determine when a PID is actually in the data register. The parity check is done whether a PID or data byte is in the LINDAT MMR.
7	Checksum match. This bit is only valid when LINSTA[0] = 1. This bit is set automatically if the value in LINCS does not match the received data in LINDAT. This bit is cleared on a read of the LINSTA MMR or when LINDAT and LINCS match while LINSTA[0] = 1.
6	Frame error. This bit is 0 if there is no frame error. It is cleared automatically by a read of LINSTA. This bit is 1 if a frame error has occurred during reception of a data byte, that is, a valid stop was not detected.
5	Negative edge maximum error interrupt (maskable). This bit is 0 if the number of negative edges allowed in a frame is not surpassed. This bit is 1 if the number of negative edges is 57 or more. This bit is cleared automatically by hardware when reading LINSTA MMR.
4	LIN collision detect interrupt (maskable). This bit is set automatically by hardware if the device has stopped transmission due to a collision on the bus. This bit is not set if the collision detect and transmit complete interrupt is enabled (LINCON[6] = 0) and the transmit complete interrupt bit is set (LINSTA[2] = 1). This bit is cleared automatically by hardware when reading LINSTA MMR.
3	Break receive interrupt (maskable). This bit is 0 if no valid break symbol has been received. This bit is 1 if a low time of 11 nominal bits is detected on the LIN bus. This bit is cleared automatically on reading the LINSTA MMR.
2	Transmit complete interrupt (maskable). This bit is 0 if data is still in the LINDAT register. This bit is 1 when all data is transmitted. It remains set to 1 until the LIN receives a break symbol. It is cleared automatically in receive mode.
1	Transmit ready interrupt (maskable). This bit is 0 if the previous data written in the LINDAT MMR is still in the LINDAT and has not being shifted to the transmit register. Writing data to the LINDAT MMR while the transmit ready bit is 0 overwrites the previous byte to be transmitted. This bit is 1 if the previous data written in the LINDAT MMR is now in the transmit register.
0	Receive ready interrupt. This bit is 0 if there is no new data to read in the LINDAT MMR. This bit is 1 if there is new data to read in the LINDAT MMR. Reading the LINDAT MMR clears this bit.

OUTLINE DIMENSIONS



COMPLIANT TO JEDEC STANDARDS MO-220-VJJD.

Figure 34 .32-Lead Lead Frame Chip Scale Package [LFCSP_VQ]
6 mm × 6 mm Body, Very Thin Quad
(CP-32-15)
Dimensions shown in millimeters

ORDERING GUIDE

Model[1, 2]	Notes	Temperature Range	Flash/Ram	Package Description	Package Option
ADuC7039BCP6Z		−40°C to +115°C	64K/4K	32-Lead Frame Chip Scale Package [LFCSP_VQ]	CP-32-15
ADuC7039BCP6Z-RL		−40°C to +115°C	64K/4K	32-Lead Frame Chip Scale Package [LFCSP_VQ]	CP-32-15
ADuC7039WBCPZ	[3]	−40°C to +115°C	64K/4K	32-Lead Frame Chip Scale Package [LFCSP_VQ]	CP-32-15
ADuC7039WBCPZ-RL	[3]	−40°C to +115°C	64K/4K	32-Lead Frame Chip Scale Package [LFCSP_VQ]	CP-32-15
EVAL-ADUC7039QSPZ				Evaluation Board	

[1] Z = RoHS Compliant Part.
[2] Qualified for automotive applications.
[3] Recommended for new designs.

AUTOMOTIVE PRODUCTS

The ADuC7039W models are available with controlled manufacturing to support the quality and reliability requirements of automotive applications. Note that these automotive models may have specifications that differ from the commercial models; therefore, designers should review the Specifications section of this data sheet carefully. Only the automotive grade products shown are available for use in automotive applications. Contact your local Analog Devices account representative for specific product ordering information and to obtain the specific Automotive Reliability reports for these models.