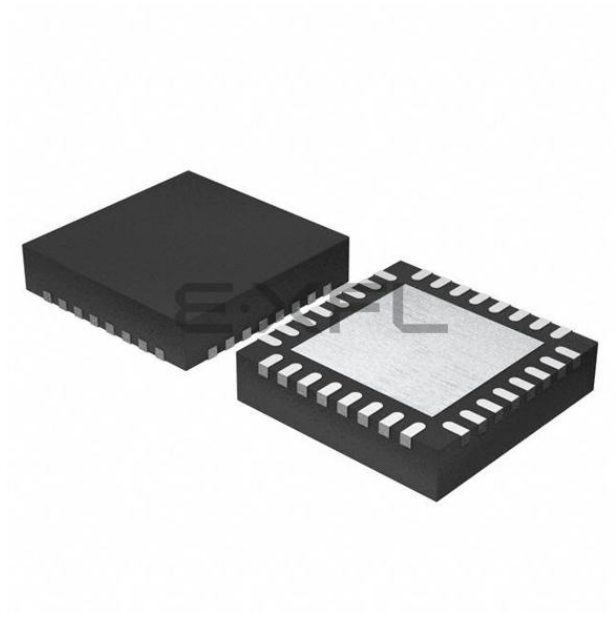


**Welcome to E-XFL.COM**

## 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 |

## REVISION HISTORY

**3/13—Rev.C to Rev. D**

Changed Fixed 10 MHz Frequency to Default 10 MHz Frequency (Throughout) ..... 1
Changes to Features Section and Figure 1 ..... 1
Changes to Table 1 ..... 5
Changes to Table 4 ..... 12
Changes to System Kernel Checksum Section ..... 24
Changes to Figure 9 ..... 25
Changes to ADCxOF Default Values ..... 41
Changes to Figure 20 ..... 49
Changes to Sequence Example Section ..... 50
Reorganized LIN Oscillator Calibration Section Layout ..... 56
Changes to Figure 22 and Table 44 ..... 59
Changes to Table 59 ..... 80
Changes to High Voltage Status Register Function Description .. 81
Changes to Figure 31 ..... 83
Changes to Part Identification Section and Added Table 64 ..... 88
Updated Outline Dimensions ..... 91
Changes to Ordering Guide ..... 91
Added Automotive Products Section ..... 91

**1/12—Rev.B to Rev. C**

Changes to LVF Level Test Conditions/Comments ..... 5
Changed Pin 25 from GPIO_4 to GPIO_4/IRQ1 ..... 10
Changes to Flash/EE Memory Security Section ..... 22
Changes to Figure 9 ..... 25
Changes to Figure 22 and Table 44 ..... 59
Changes to GPIO_4 Port Signal and Functionality (Table 49) ..... 69
Changes to Figure 33 ..... 90
Updated Outline Dimensions ..... 91

**8/10—Rev.A to Rev. B**

Changes to Fast Temperature Conversion Mode Section ..... 46
Changes to Bit 4 Description (LINCON[6] Reference) in Table 62 ..... 86
Changes to Part Identification Section ..... 87

**6/10—Rev. 0 to Rev. A**

Added SAEJ2602 to Features Section ..... 1
Changes to Table 2 ..... 9
Changes to Bit 2 and Bit 1 Descriptions in Table 34 ..... 39
Changes to System Clocks Section ..... 49
Added Syncronization of Timers across Asynchronous Clock Domains Section, Figure 23, and Figure 24 ..... 60
Added Programming the Timers Section ..... 61
Changes to Recommended Schematic Text ..... 89

**3/10—Revision 0: Initial Version**

| Parameter | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| ON-CHIP OSCILLATORS | | | | | |
| Low Power Oscillator | | | 128 | | kHz |
| Accuracy | After user calibration at nominal supply and room temperature; includes drift data from 1000 hr life-test | −3 | | +3 | % |
| Precision Oscillator | | | 128 | | kHz |
| Accuracy | After run time calibration | −1 | | +1 | % |
| MCU CLOCK RATE[1] | Default setting | | 10.24 | | MHz |
| MCU START-UP TIME[1] | | | | | |
| At Power-On | Includes kernel power-on execution time | | 25 | | ms |
| After Reset Event | Includes kernel power-on execution time | | 5 | | ms |
| From MCU Power-Down | Oscillator running | | 2 | | ms |
| Internal PLL Lock Time | | | 1 | | ms |
| LIN I/O GENERAL | | | | | |
| Baud Rate | | 1000 | | 20,000 | Bits/sec |
| $V_{DD}$ | Supply voltage range for which the LIN interface is functional | 7 | | 18 | V |
| Input Capacitance[1] | | | 5.5 | | pF |
| LIN Comparator Response Time[1] | Using 22 Ω resistor | | 38 | 90 | µs |
| LIN DC PARAMETERS | | | | | |
| $I_{LIN\ DOM\ MAX}$ | Current limit for driver when LIN bus is in dominant state; VBAT = VBAT (maximum) | 40 | | 200 | mA |
| $I_{LIN\_PAS\_REC}$[1] | Driver off; 7.0 V < $V_{BUS}$ < 18 V; $V_{DD}$ = $V_{LIN}$ − 0.7 V | | | 20 | µA |
| $I_{LIN\_PAS\_DOM}$[1] | Input leakage, $V_{LIN}$ = 0 V | −1 | | | mA |
| $I_{LIN\_NO\_GND}$[1, 20] | Control unit disconnected from ground, GND = $V_{DD}$; 0 V < $V_{LIN}$ < 18 V; VBAT = 12 V | −1 | | +1 | mA |
| $I_{BUS\_NO\_BAT}$[1] | VBAT disconnected, $V_{DD}$ = GND, 0 V < $V_{BUS}$ < 18 V | | | 100 | µA |
| $V_{LIN\_DOM}$[1] | LIN receiver dominant state, $V_{DD}$ > 7.0 V | | | 0.4 $V_{DD}$ | V |
| $V_{LIN\_REC}$[1] | LIN receiver recessive state, $V_{DD}$ > 7.0 V | 0.6 $V_{DD}$ | | | V |
| $V_{LIN\_CNT}$[1] | LIN receiver center voltage, $V_{DD}$ > 7.0 V | 0.475 $V_{DD}$ | 0.5 $V_{DD}$ | 0.525 $V_{DD}$ | V |
| $V_{HYS}$[1] | LIN receiver hysteresis voltage | | | 0.175 $V_{DD}$ | V |
| $V_{LIN\_DOM\_DRV\_LOSUP}$[1] | LIN dominant output voltage; $V_{DD}$ = 7.0 V | | | | |
| $R_L$ 500 Ω | | | | 1.2 | V |
| $R_L$ 1000 Ω | | 0.6 | | | V |
| $V_{LIN\_DOM\_DRV\_HISUP}$[1] | LIN dominant output voltage; $V_{DD}$ = 18 V | | | | |
| $R_L$ 500 Ω | | | | 2 | V |
| $R_L$ 1000 Ω | | 0.8 | | | V |
| $V_{LIN\_RECESSIVE}$[1] | LIN recessive output voltage | 0.8 $V_{DD}$ | | | V |
| VBAT Shift[20] | | 0 | | 0.115 $V_{DD}$ | V |
| GND Shift[20] | | 0 | | 0.115 $V_{DD}$ | V |
| $R_{SLAVE}$ | Slave termination resistance | 20 | 30 | 47 | kΩ |
| $V_{SERIAL\ DIODE}$[20] | Voltage drop at the serial diode, $D_{Ser\_Int}$ | 0.4 | 0.7 | 1 | V |
| LIN AC PARAMETERS[1] | Bus load conditions (CBUS\|\|RBUS): 1 nF\|\|1 kΩ; 6.8 nF\|\|660 Ω; 10 nF\|\|500 Ω | | | | |
| D1 | Duty Cycle 1 | 0.396 | | | |
| | $TH_{REC(MAX)}$ = 0.744 × VBAT | | | | |
| | $TH_{DOM(MAX)}$ = 0.581 × VBAT | | | | |
| | $V_{SUP}$ = 7.0 V … 18 V; $t_{BIT}$ = 50 µs | | | | |
| | D1 = $t_{BUS\_REC(MIN)}/(2 \times t_{BIT})$ | | | | |
| D2 | Duty Cycle 2 | | | 0.581 | |
| | $TH_{REC(MIN)}$ = 0.284 × VBAT | | | | |
| | $TH_{DOM(MIN)}$ = 0.422 × VBAT | | | | |
| | $V_{SUP}$ = 7.0 V … 18 V; $t_{BIT}$ = 50 µs | | | | |
| | D2 = $t_{BUS\_REC(MAX)}/(2 \times t_{BIT})$ | | | | |
| D3[1, 20] | $TH_{REC(MAX)}$ = 0.778 × VBAT | 0.417 | | | |
| | $TH_{DOM(MAX)}$ = 0.616 × VBAT | | | | |
| | $V_{DD}$ = 7.0 V ... 18 V | | | | |
| | $t_{BIT}$ = 96 µs | | | | |
| | D3 = $t_{BUS\_REC(MIN)}/(2 \times t_{BIT})$ | | | | |
| D4[1, 20] | $TH_{REC(min)}$ = 0.389 × VBAT | | | 0.590 | |
| | $TH_{DOM(min)}$ = 0.251 × VBAT | | | | |

# PIN CONFIGURATION AND FUNCTION DESCRIPTIONS

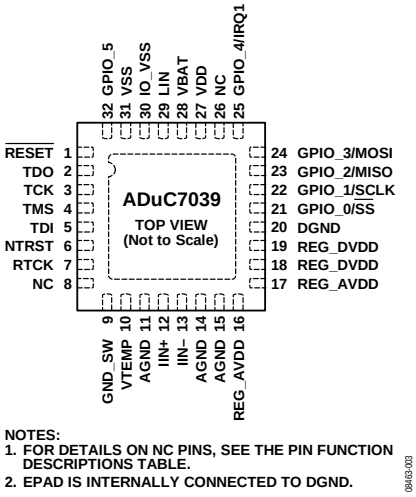



*Figure 3. Pin Configuration*

**Table 3. Pin Function Descriptions**

| Pin No. | Mnemonic | Type[1] | Description |
|---|---|---|---|
| 1 | $\overline{\text{RESET}}$ | I | Reset Input Pin. Active low. This pin has an internal, weak, pull-up resistor to REG_DVDD. When not in use, this pin can be left unconnected. For added security and robustness, it is recommended that this pin be strapped via a resistor to REG_DVDD. |
| 2 | TDO | O | JTAG Test Data Output. This data output pin is one of the standard 6-pin JTAG debug ports on the part. TDO is an output pin only. At power-on, this output is disabled and pulled high via an internal, weak, pull-up resistor. This pin can be left unconnected when not in use. |
| 3 | TCK | I | JTAG Test Clock. This clock input pin is one of the standard 6-pin JTAG debug ports on the part. TCK is an input pin only and has an internal, weak, pull-up resistor. This pin can be left unconnected when not in use. |
| 4 | TMS | I | JTAG Test Mode Select. This mode select input pin is one of the standard 6-pin JTAG debug ports on the part. TMS is an input pin only and has an internal, weak, pull-up resistor. This pin can be left unconnected when not in use. |
| 5 | TDI | I | JTAG Test Data Input. This data input pin is one of the standard 6-pin JTAG debug ports on the part. TDI is an input pin only and has an internal, weak, pull-up resistor. This pin can be left unconnected when not in use. |
| 6 | NTRST | I | JTAG Test Reset. This reset input pin is one of the standard 6-pin JTAG debug ports on the part. NTRST is an input pin only and has an internal, weak, pull-down resistor. This pin can be left unconnected when not in use. NTRST is also monitored by the on-chip kernel to enable LIN boot load mode. |
| 7 | RTCK | O | JTAG Return Test Clock. This output pin is used to adjust the JTAG clock speed to the highest possible rate of the ADuC7039. |
| 8 | NC | | No Connect. This pin is internally connected; therefore, do not externally connect this pin. |
| 9 | GND_SW | I | Switch to Internal Analog Ground Reference. This pin is the negative input for the external temperature channel and external reference. If this input is not used, connect it directly to the AGND system ground. |
| 10 | VTEMP | I | External Pin for NTC/PTC Temperature Measurement. |
| 11, 14, 15 | AGND | S | Ground Reference for On-Chip Precision Analog Circuits. |
| 12 | IIN+ | I | Positive Differential Input for Current Channel. |
| 13 | IIN– | I | Negative Differential Input for Current Channel. |
| 16, 17 | REG_AVDD | S | Nominal 2.6 V analog Output from On-Chip Regulator. Pin 16 and Pin 17 must be connected together to a capacitor to ground. |
| 18, 19 | REG_DVDD | S | Nominal 2.6 V digital Output from On-Chip Regulator. Pin 18 and Pin 19 must be connected together to capacitors to ground. |
| 20 | DGND | S | Ground Reference for On-Chip Digital Circuits. |

## 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.

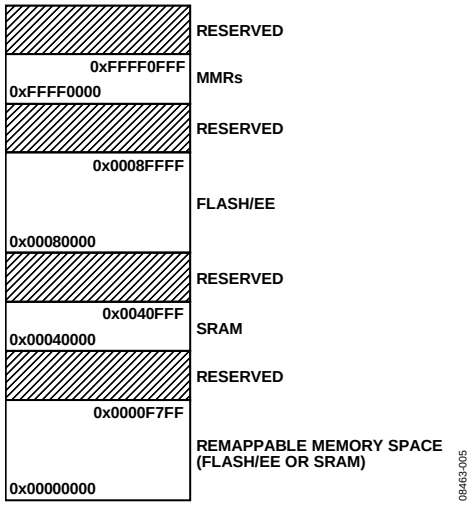



*Figure 5. ADuC7039 Memory Map, 64 kB Flash Option*

- 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.

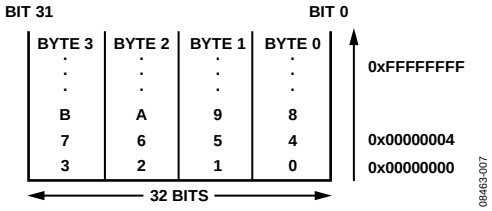



*Figure 6. Little Endian Format*

### *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.

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.<br>This bit is set by the user to remap the SRAM to 0x00000000.<br>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.<br>This bit is set by hardware when an external reset occurs.<br>This bit is cleared by setting the corresponding bit in RSTCLR. |
| 2 | Software reset.<br>This bit is set by user code to generate a software reset.<br>This bit is cleared by setting the corresponding bit in RSTCLR.[1] |
| 1 | Watchdog timeout.<br>This bit is set by hardware when a watchdog timeout occurs.<br>This bit is cleared by setting the corresponding bit in RSTCLR. |
| 0 | Power-on reset.<br>This bit is set by hardware when a power-on-reset occurs.<br>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.

# 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.

### *FEEMOD Register*

Name: FEEMOD

Address: 0xFFFF0E04

Default Value: 0x0000

Access: Read/write

Function: This register is written by user code to configure the mode of operation of the Flash/EE memory controller.

**Table 12. FEEMOD MMR Bit Designation**

| Bit | Description |
|---|---|
| 15 to 7 | Not used. These bits are reserved for future functionality and should be written as 0 by user code. |
| 6 to 5 | Flash/EE security lock bits. These bits must be written as [6:5] = 1, 0 to complete the Flash/EE security protect sequence. |
| 4 | Flash/EE controller command complete interrupt enable.<br>This bit is set by user code to enable the Flash/EE controller to generate an interrupt upon completion of a Flash/EE command.<br>This bit is cleared by user code to disable the generation of a Flash/EE interrupt upon completion of a Flash/EE command. |
| 3 | Flash/EE erase/write enable.<br>This bit is set by user code to enable the Flash/EE erase and write access via FEECON.<br>This bit is cleared by user code to disable the Flash/EE erase and write access via FEECON. |
| 2 | Reserved. |
| 1 | Flash/EE controller abort enable.<br>This bit is set by user code to enable the Flash/EE controller abort functionality.<br>This bit is cleared by user code to disable the Flash/EE controller abort functionality. |
| 0 | Reserved. |

### *FEEADR Registers*

Name: FEEADR

Address: 0xFFFF0E10

Default Value: Updated by kernel

Access: Read/write

Function: This 16-bit register dictates the address upon which any Flash/EE command executed via FEECON acts.

### *FEEDAT Registers*

Name: FEEDAT

Address: 0xFFFF0E0C

Default Value: 0x0000

Access: Read/write

Function: This 16-bit register contains the data either read from, or to be written to, the Flash/EE memory.

## FLASH/EE MEMORY SECURITY

The 62 kB of Flash/EE memory available to the user can be read- and write-protected using the FEEHID register.

The MSB of FEEHID (Bit 31) protects the entire Flash/EE from being read through JTAG.

Bits[30:0] of FEEHID protect Page 123 to Page 0 from writing. Each bit protects four pages, that is, 2 kB.

The FEEPRO register mirrors the bit definitions of the FEEHID MMR. The FEEPRO MMR allows user code to lock the protection or security configuration of the Flash/EE memory so that the protection configuration is automatically loaded on subsequent power-on or reset events. This flexibility allows the user to temporarily set and test protection settings using the FEEHID MMR and, subsequently, lock the required protection configuration (using FEEPRO) when shipping protection systems into the field.

### *Flash/EE Memory Protection Registers*

Name: FEEHID and FEEPRO

Address: 0xFFFF0E20 (for FEEHID) and 0xFFFF0E1C (for FEEPRO)

Default Value: 0xFFFFFFFF (for FEEHID) and 0x00000000 (for FEEPRO)

Access: Read/write

Function: These registers are written by user code to configure the protection of the Flash/EE memory.

**Table 13. FEEHID and FEEPRO MMR Bit Designations**

| Bit | Description |
|---|---|
| 31 | Read protection.<br>This bit is cleared by user code to read protect the 62 kB Flash/EE block code.<br>This bit is set by user code to allow read access to the 62 kB Flash/EE block via JTAG. |
| 30 to 0 | Write protection bits.<br>When set by user code, these bits unprotect Page 0 to Page 123 of the 62 kB Flash/EE code memory. Each bit write protects four pages, and each page consists of 512 bytes.<br>When cleared by user code, these bits write protect Page 0 to Page 123 of the 62 kB Flash/EE code memory. Each bit write protects four pages, and each page consists of 512 bytes. |

### *Voltage/Temperature Channel ADC Control Register*

Name: ADC1CON

Address: 0xFFFF0510

Default Value: 0x0000

Access: Read/write

Function: The voltage/temperature channel ADC control MMR is a 16-bit register that is used to configure the V/T-ADC. If both ADCs are being reconfigured, ADC1CON should be written before ADC0CON to ensure both ADCs start synchronously. If ADC0 is already on and converting and ADC1 is off, then, first turn on ADC1 and second disable, and re-enable ADC0 so that the two ADCs start simultaneously (this accounts for ADC start-up time).

**Table 30. ADC1CON MMR Bit Designations**

| Bit | Description |
|---|---|
| 15 | Voltage/temperature channel ADC enable.<br>This bit is set to 1 by user code to enable the V/T-ADC.<br>Clearing this bit to 0 powers down the V/T-ADC. |
| 14 to 13 | VTEMP current source enable.<br>0, 0 = current sources off.<br>0, 1 = enables 50 µA current source on VTEMP.<br>1, 0 = enables 50 µA current source on GND_SW.<br>1, 1 = enables 50 µA current source on both VTEMP and GND_SW. |
| 12 to 10 | Not used. These bits are reserved for future functionality and should not be modified by user code. |
| 9 | Voltage/temperature channel ADC output coding.<br>This bit is set to 1 by user code to configure V/T-ADC output coding as unipolar.<br>This bit is cleared to 0 by user code to configure V/T-ADC output coding as twos complement. |
| 8 | Not used. This bit is reserved for future functionality and should be written as 0 by user code. |
| 7 to 6 | Voltage/temperature channel ADC input select.<br>0, 0 = VBAT/24, AGND. VBAT attenuator selected.<br>0, 1 = VTEMP, GND_SW. External temperature input selected, conversion result written to ADC1DAT.<br>1, 0 = internal sensor. Internal temperature sensor input selected, conversion result written to ADC1DAT. The temperature gradient is 0.33 mV/°C; this is only applicable to the internal temperature sensor.<br>1, 1 = internal short. Shorted input. |
| 5 | Not used. This bit is reserved for future functionality and should be written as 0 by user code. |
| 4 | Voltage/temperature channel ADC reference select.<br>0 = internal, 1.2 V precision reference selected.<br>1 = (REG_AVDD, GND_SW)/2 selected. |
| 3 to 0 | Not used. These bits are reserved for future functionality and should not be written as 0 by user code. |

### *Current Channel ADC Result Counter Limit Register*

Name: ADC0RCL

Address: 0xFFFF0548

Default Value: 0x0001

Access: Read/write

Function: This 16-bit MMR sets the number of conversions required before an ADC interrupt is generated. By default, this register is set to 0x01. The ADC counter function must be enabled via the ADC result counter enable bit in the ADCCFG MMR.

### *Current Channel ADC Result Count Register*

Name: ADC0RCV

Address: 0xFFFF054C

Default Value: 0x0000

Access: Read only

Function: This 16-bit, read-only MMR holds the current number of I-ADC conversion results. It is used in conjunction with ADC0RCL to mask I-ADC interrupts, generating a lower interrupt rate. When ADC0RCV = ADC0RCL, the value in ADC0RCV resets to 0 and recommences counting. It can also be used in conjunction with the accumulator (ADC0ACC) to allow an average current calculation to be undertaken. The result counter is enabled via ADCCFG[0]. This MMR is also reset to 0 when the I-ADC is reconfigured, that is, when the ADC0CON or ADCMDE are written.

### *Current Channel ADC Threshold Register*

Name: ADC0TH

Address: 0xFFFF0550

Default Value: 0x0000

Access: Read/write

Function: This 16-bit MMR sets the threshold against which the absolute value of the I-ADC conversion result is compared. In unipolar mode, ADC0TH[15:0] are compared and in twos complement mode, ADC0TH[14:0] are compared.

### *Current Channel ADC Accumulator Register*

Name: ADC0ACC

Address: 0xFFFF055C

Default Value: 0x00000000

Access: Read only

Function: This 32-bit MMR holds the current accumulator value. The I-ADC ready bit in the ADCSTA MMR should be used to determine when it is safe to read this MMR. The MMR value is reset to 0 by disabling the accumulator in the ADCCFG MMR or reconfiguring the current channel ADC.

### ADC Calibration

As shown in detail in the top level diagrams (Figure 11 and Figure 12), the signal flow through all ADC channels can be described in the following steps:

1. An input voltage is applied through an input buffer (and PGA in the case of the I-ADC) to the Σ-Δ modulator.
2. The modulator output is applied to a programmable digital decimation filter.
3. The filter output result is then averaged if chopping is used.
4. An offset value (ADCxOF) is subtracted from the result.
5. This result is scaled by a gain value (ADCxGN).
6. Finally, the result is formatted as twos complement/ unipolar, rounded to 16 bits, or clamped to ± full scale.

Each ADC channel (current, voltage, and temperature) has a specific offset and gain correction or calibration coefficient associated with it that are stored in MMR-based offset and gain registers (ADCxOF and ADCxGN). The offset and gain registers can be used to remove system level offset and gain errors external to the part.

These registers are configured at power-on with a factory programmed calibration value. These factory calibration values vary from part to part reflecting the manufacturing variability of internal ADC circuits. These registers can also be overwritten by user code after a calibration.

On the current channel when a system calibration is initiated, the ADC generates its calibration coefficient based on an externally generated zero-scale voltage and full-scale voltage, which are applied to the external ADC input for the duration of the calibration cycle. The coefficients are written in the ADC0DAT MMR of the ADC channels; they are not automatically written in the ADC0OF or ADC0GN MMR. User code must copy these values to their appropriate registers.

The duration of an offset calibration is a full ADC filter settling time before returning the ADC to idle mode. When a calibration cycle is initiated, any ongoing ADC conversion is immediately halted, the calibration is automatically carried out at an ADC update rate programmed into ADCFLT, and the ADC is always returned to idle after any calibration cycle. It is strongly recommended that ADC calibration is initiated at as low an ADC update rate as possible (high SF value in ADCFLT) to minimize the impact of ADC noise during calibration.

On the voltage channel, a two-point calibration must be performed as the minimum voltage specified on the input is 4 V.

The temperature channel is factory calibrated for the internal temperature sensor.

### Calibrating the Voltage Channel

To calibrate the offset and gain of the voltage channel a two-point calibration method must be used. This method consists of converting two known voltages (for example, 8 V and 16 V) to determine lope and offset of the transfer function. The gain coefficient can be divided by the calculated slope to improve the gain error.

The offset error can be reduced by writing ⅜ of the calculated offset (in unipolar codes) into the ADC1OF MMR.

### Calibrating the Current Channel

If the chop bit (ADCFLT[15]) is enabled, then internal ADC offset errors are minimized and an offset calibration may not be required. If chopping is disabled, however, an initial offset calibration is required and may need to be repeated, particularly after a large change in temperature.

A gain calibration, particularly in the context of the I-ADC (with internal PGA), may need to be carried out at all relevant system gain ranges depending on system accuracy requirements. If it is not possible to apply an external full-scale current on all gain ranges, then it is possible to apply a lower current and scale the result produced by the calibration. For example, apply a 50% current and then divide the ADC0DAT value produced-by-two and write this value back into ADC0GN. Note that there is a lower limit to the input signal that can be applied for a system calibration because ADC0GN is only a 16-bit register. The input span (difference between the system zero-scale value and system full-scale value) should be greater than 40% of the nominal full-scale-input range, that is, >40% of $V_{REF}$/gain.

The on-chip Flash/EE memory can be used to store multiple calibration coefficients. These can be copied by user code directly into the relevant calibration registers, as appropriate, based on the system configuration.

A factory, or end-of-line calibration, for the I-ADC is a two-step procedure.

1. Apply the 0 A current. Configure the ADC in the required PGA setting, and so on, and write to ADCMDE[2:0] to perform a system zero-scale calibration. This writes a new offset calibration value into ADC0DAT. User code must store this value into ADC0OF or into Flash/EE memory.
2. Apply a full-scale current for the selected PGA setting. Write to ADCMDE to perform a system full-scale calibration. This writes a new gain calibration value into ADC0DAT. This value must be copied by user software to the ADC0GN MMR or into Flash/EE memory.

## OSCILLATORS CALIBRATION

The ADuC7039 features two oscillators and two calibration schemes:

- The low power oscillator can be calibrated from the precision oscillator or from the LIN communication. The trim value can also be modified by user code.
- The precision oscillator can be calibrated from the LIN communication. The trim value can also be modified by user code.

Each oscillator has dedicated calibration MMRs:

- LOCUSR0 is the low power oscillator user trim register. It is a 8-bit wide register. Increasing the value in LOCUSR0 decreases the frequency of the low power oscillator; decreasing the value increases the frequency. Based on a nominal frequency of 128 kHz, the typical trim range is between 103 kHz to 156 kHz. This MMR can be written directly by user code or changed automatically by the hardware relative to the LIN baud rate.
- LOCUSR1 is the precision oscillator user trim register. This is a 10-bit wide MMR. Increasing the value in LOCUSR1 decreases the frequency of the precision oscillator; decreasing the value increases the frequency. Based on a nominal frequency of 128 kHz, the typical trim range is between 94 kHz to 178 kHz. This MMR can be written directly by user code, or changed automatically by the hardware relative to the LIN baud rate.
- LOCVAL0 is an 8-bit, read-only MMR and displays the current trim value of the low power oscillator.
- LOCVAL1 is a 10-bit, read-only MMR and displays the current trim value of the precision oscillator. Note that 11 bits can be read from this register but only 10 are used for calibration.

### *Initial Low Power Oscillator Calibration*

After reset, the low power oscillator is running at a frequency of 128 kHz with a maximum error of −10% to +3% from the center frequency of 128 kHz. An end-of-line calibration at the customer production line must be run within a given temperature range of 25°C ± 5°C to center the low power oscillator on the precision oscillator. Once calibrated, the low power oscillator stays within ±3% of the center frequency.

This initial calibration only needs to be run once, at end-of-line. Further calibration can be performed in user code to compensate for temperature drift of the low power oscillator.

### *Low Power Oscillator Calibration Sequence*

The low power 128 kHz oscillator can be calibrated using the precision 128 kHz oscillator. Two dedicated calibration counters are used to implement this feature.

One counter, 9-bits wide, is clocked by the precision oscillator. The second counter, 10-bits wide, is clocked by the low power oscillator. The clock calibration mode is configured and controlled by the following MMRs:

- OSCCON—control bits for calibration.
- OSCSTA—calibration status register.
- OSCVAL0—9-bit counter, Counter 0.
- OSCVAL1—10-bit counter, Counter 1.

An example calibration routine is shown in Figure 21. User code configures and enables the calibration sequence using OSCCON. When the precision oscillator calibration counter, OSCVAL0, reaches 0x1FF, both counters are disabled.

User code then reads back the value of the low power oscillator calibration counter. There are three possible scenarios:

- OSCVAL0 = OSCVAL1. No further action is required.
- OSCVAL0 > OSCVAL1. The low power oscillator is running slow. LOCUSR0 must be decreased.
- OSCVAL0 < OSCVAL1. The low power oscillator is running fast. LOCUSR0 must be increased.

When the LOCUSR0 has been changed, the routine should be run again and the new frequency checked. Note that the LOCUSR0 MMR is key protected. The value 0x1324 must be written in LOCKEY prior to writing LOCUSR0.

Using the internal, precision oscillator, it takes approximately 4 ms to execute the calibration routine.

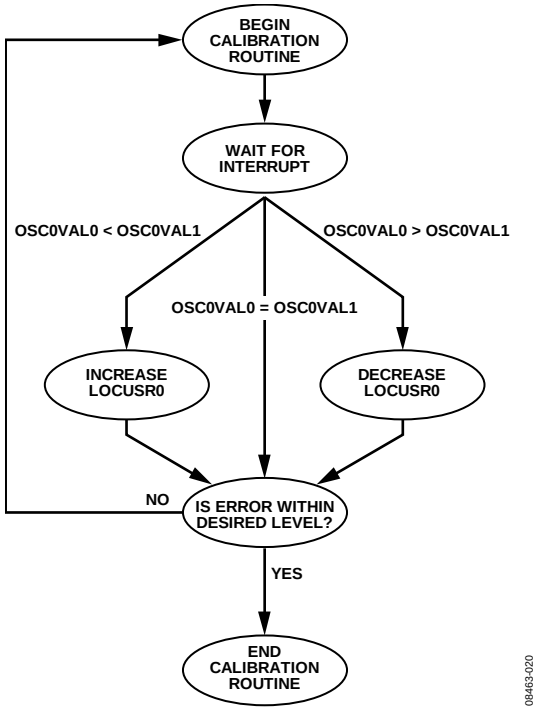



*Figure 21. OSCTRM Calibration Routine*

Prior to the clock calibration routine being started, it is required that the user switch to the precision oscillator to serve as the PLL clock source, otherwise, the PLL can lose lock each time LOCUSR0 is modified. This increases the length of time it takes to calibrate the low power oscillator.

### *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.<br>Setting/clearing this bit corresponds to setting/clearing Bit 1 of FIQSTA and FIQSIG. |
| 1 | Programmed interrupt IRQ.<br>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.

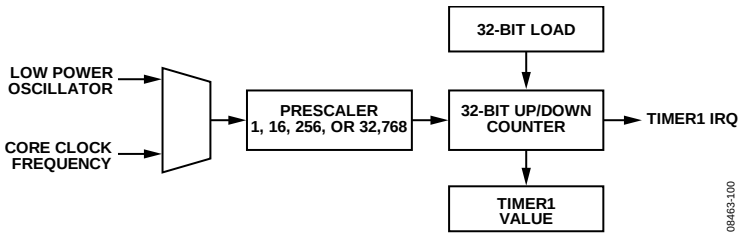



*Figure 26. Timer1 Block Diagram*

**Table 47. T1CON MMR Bit Designations**

| Bit | Description |
|---|---|
| 15 to 6 | Reserved. These bits should be written as 0. |
| 5 | Timer1 mode.<br>This bit is set by user code to operate in periodic mode.<br>This bit is cleared by user code to operate in free running mode (default). |
| 4 | Count up.<br>This bit is set by user code for Timer1 to count up.<br>This bit is cleared by user code for Timer1 to count down (default). |
| 3 | Timer1 enable bit.<br>This bit is set by user code to enable Timer1.<br>This bit is cleared by user code to disable Timer1 (default). |
| 2 | Clock select.<br>0 = core clock (default).<br>1 = low power (32.768 kHz) oscillator. |
| 1 to 0 | Prescaler.<br>00 = source clock/1 (default).<br>01 = source clock/16.<br>10 = source clock/256.<br>11 = source clock/32,768. |

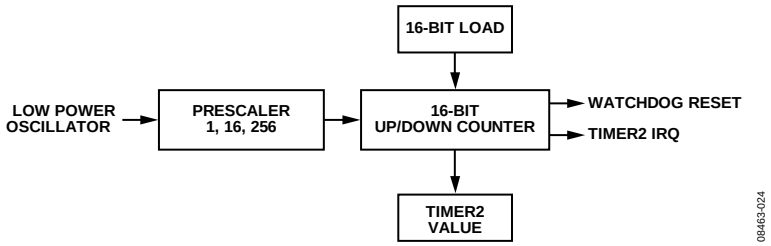



*Figure 27. Timer2 Block Diagram*

**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.<br>This bit is set by user code to configure Timer2 to count up.<br>This bit is cleared by user code to configure Timer2 to count down. |
| 7 | Timer2 enable.<br>This bit is set by user code to enable Timer2.<br>This bit is cleared by user code to disable Timer2. |
| 6 | Timer2 operating mode.<br>This bit is set by user code to configure Timer2 to operate in periodic mode.<br>This bit is cleared by user to configure Timer2 to operate in free running mode. |
| 5 | Watchdog timer mode enable.<br>This bit is set by user code to enable watchdog mode.<br>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.<br>00 = source clock/1 (default).<br>01 = source clock/16.<br>10 = source clock/256.<br>11 = reserved. |
| 1 | Watchdog timer IRQ enable.<br>This bit is set by user code to produce an IRQ instead of a reset when the watchdog reaches 0.<br>This bit is cleared by user code to disable the IRQ option. |
| 0 | PD_OFF.<br>This bit is set by the user code to stop Timer2 when the peripherals are powered down using Bit 4 in the POWCON MMR.<br>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 Control Register*

Name: SPICON

Address: 0xFFFF0A10

Default Value: 0x0000

Access: Read/write

Function: This 16-bit MMR configures the SPI peripheral in both master and slave modes.

**Table 55. SPICON MMR Bit Designations**

| Bit | Description |
|---|---|
| 15 to 14 | SPI IRQ mode bits. These bits configure when the Tx/Rx interrupts occur in a transfer.<br>00 = Tx interrupt occurs when 1 byte has been transferred. Rx interrupt occurs when 1 or more bytes have been received into the FIFO.<br>01 = Tx interrupt occurs when 2 bytes have been transferred. Rx interrupt occurs when 1 or more bytes have been received into the FIFO.<br>10 = Tx interrupt occurs when 3 bytes have been transferred. Rx interrupt occurs when 3 or more bytes have been received into the FIFO.<br>11 = Tx interrupt occurs when 4 bytes have been transferred. Rx interrupt occurs when the Rx FIFO is full, or 4 bytes are present. |
| 13 | SPI Tx FIFO flush enable bit.<br>Set this bit to flush the Tx FIFO. This bit does not clear itself and should be toggled if a single flush is required. If this bit is left high, then either the last transmitted value or 0x00 is transmitted depending on SPICON[7]. Any writes to the Tx FIFO are ignored while this bit is set.<br>Clear this bit to disable Tx FIFO flushing. |
| 12 | SPI Rx FIFO flush enable bit.<br>Set this bit to flush the Rx FIFO. This bit does not clear itself and should be toggled if a single flush is required. If this bit is set, all incoming data is ignored and no interrupts are generated. If set and SPICON[6] = 0, a read of the Rx FIFO initiates a transfer.<br>Clear this bit to disable Rx FIFO flushing. |
| 11 | Continuous transfer enable.<br>This bit is set by the user to enable continuous transfer. In master mode, the transfer continues until no valid data is available in the SPITX register. $\overline{SS}$ is asserted and remains asserted for the duration of each 8-bit serial transfer until TX is empty.<br>This bit is cleared by the user to disable continuous transfer. Each transfer consists of a single 8-bit serial transfer. If valid data exists in the SPITX register, then a new transfer is initiated after a stall period of 1 serial clock cycle. |
| 10 | Loop back enable bit.<br>This bit is set by the user to connect MISO to MOSI and test software.<br>This bit is cleared by the user to be in normal mode. |
| 9 | Slave MISO output enable bit.<br>Set this bit to disable the output driver on the MISO pin. The MISO pin becomes open drain when this bit is set.<br>Clear this bit for MISO to operate as normal. |
| 8 | SPIRX overflow overwrite enable.<br>This bit is set by the user; the valid data in the SPIRX register is overwritten by the new serial byte received.<br>This bit is cleared by the user; the new serial byte received is discarded. |
| 7 | SPI transmit zeros when Tx FIFO enable bit.<br>Set this bit to transmit 0x00 when there is no valid data in the Tx FIFO.<br>Clear this bit to transmit the last transmitted value when there is no valid data in the Tx FIFO. |
| 6 | SPI transfer and interrupt mode.<br>This bit is set by the user to initiate a transfer with a write to the SPITX register. Interrupt only occurs when SPITX is empty.<br>This bit is cleared by the user to initiate a transfer with a read of the SPIRX register. Interrupt only occurs when SPIRX is full. |
| 5 | LSB first transfer enable bit.<br>This bit is set by the user; the LSB is transmitted first.<br>This bit is cleared by the user; the MSB is transmitted first. |

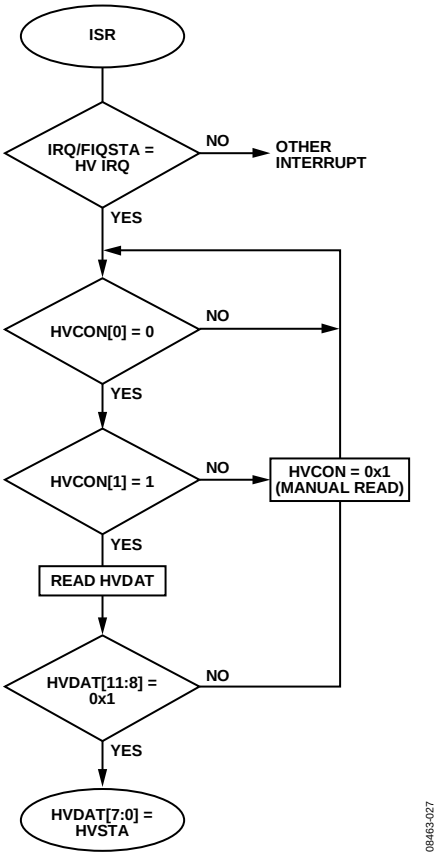



*Figure 30. High Voltage Interface Interrupt Flow Chart*

### *HV Configuration*

Following is a code example to enable LIN.

```
char HVstatus;
      do{                              // Enable LIN
            HVDAT = 0x01;              // Enable LIN physical layer mode
            HVCON = 0x08;              // Write to HVCFG
            do{
                  HVstatus = HVCON;
            }
            while(HVstatus & 0x1);     // Wait until command is finished
      }
      while (!(HVstatus & 0x4));       // Transmit command is correct
```

It is best practice to implement the high voltage communication routine in a function and call this function throughout the code.

### *LINCON Register*

Name: LINCON

Address: 0xFFFF0700

Default Value: 0x0000

Access: Read/write

Function: This 16-bit MMR controls the LIN peripheral.

### *LINCS Register*

Name: LINCS

Address: 0xFFFF0704

Default Value: 0xFF

Access: Read/write

Function: 8-bit checksum register.

### *LINBR Register*

Name: LINBR

Address: 0xFFFF0708

Default Value: 0x00FA0

Access: Read/write

Function: 19-bit baud rate register.

### *LINBK Register*

Name: LINBK

Address: 0xFFFF070C

Default Value: 0x0000157C

Access: Read/write

Function: 19-bit break timer register.

### *LINSTA Register*

Name: LINSTA

Address: 0xFFFF0710

Default Value: 0x0100

Access: Read only

Function: 16-bit status register.

### *LINDAT Register*

Name: LINDAT

Address: 0xFFFF0714

Default Value: 0x00

Access: Read/write

Function: 8-bit data register.

### *LINLOW Register*

Name: LINLOW

Address: 0xFFFF0718

Default Value: 0x00000

Access: Read/write

Function: 19-bit register; generates a LIN break to wake up other LIN peripherals on the bus.

### *LINWU Register*

Name: LINWU

Address: 0xFFFF071C

Default Value: 0x00013

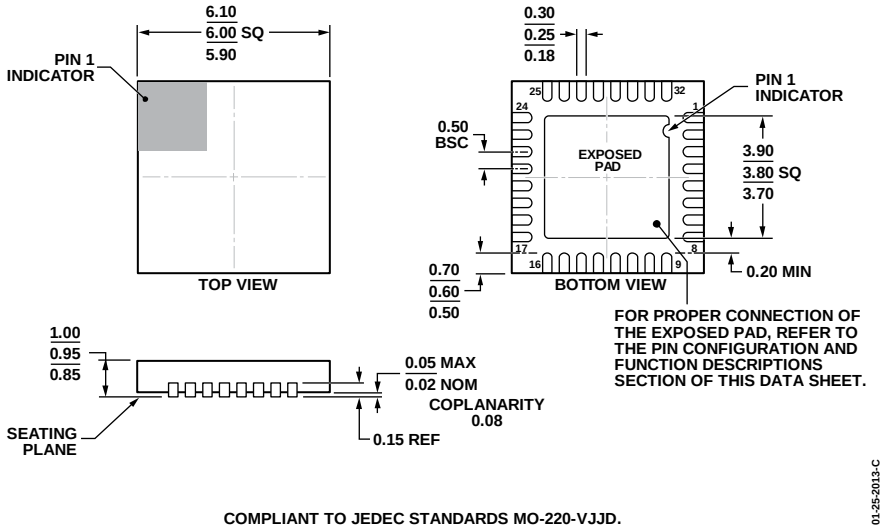
Access: Read/write

Function: 19-bit register; specify the length of a break waking up the device.

There are seven sources of interrupt; five of them are maskable in the LINCON MMR:

- LIN wake-up
- Data received
- Transmit ready—maskable
- Transmit complete—maskable
- Collision detected—maskable
- Break symbol—maskable
- Maximum negative edges within a frame error—maskable

# 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.

# NOTES


D08463-0-3/13(D)