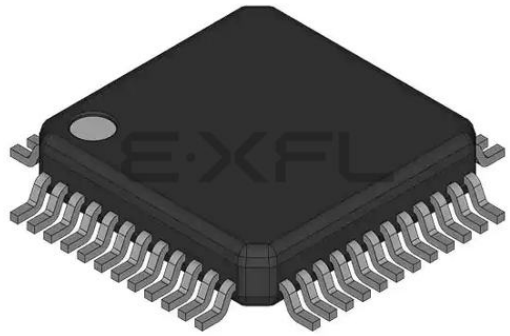

Welcome to E-XFL.COM

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	-
Core Size	-
Speed	-
Connectivity	-
Peripherals	-
Number of I/O	-
Program Memory Size	-
Program Memory Type	-
EEPROM Size	-
RAM Size	-
Voltage - Supply (Vcc/Vdd)	-
Data Converters	-
Oscillator Type	-
Operating Temperature	-
Mounting Type	-
Package / Case	-
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/aduc7033bstz-88-rl

Table 3. SPI Master Mode Timing (PHASE Mode = 0)

Parameter	Description	Min	Typ	Max	Unit
t_{SL}	SCLK low pulse width[1]		(SPIDIV + 1) × t_{HCLK}		ns
t_{SH}	SCLK high pulse width[1]		(SPIDIV + 1) × t_{HCLK}		ns
t_{DAV}	Data output valid after SCLK edge[2]			(2 × t_{UCLK}) + (2 × t_{HCLK})	ns
t_{DOSU}	Data output setup before SCLK edge		½ t_{SL}		ns
t_{DSU}	Data input setup time before SCLK edge	0			ns
t_{DHD}	Data input hold time after SCLK edge[2]	3 × t_{UCLK}			ns
t_{DF}	Data output fall time		3.5		ns
t_{DR}	Data output rise time		3.5		ns
t_{SR}	SCLK rise time		3.5		ns
t_{SF}	SCLK fall time		3.5		ns

[1] t_{HCLK} depends on the clock divider (CD) bits in POWCON MMR. $t_{HCLK} = t_{UCLK}/2^{CD}$.
[2] t_{UCLK} = 48.8 ns and corresponds to the 20.48 MHz internal clock from the PLL before the clock divider.

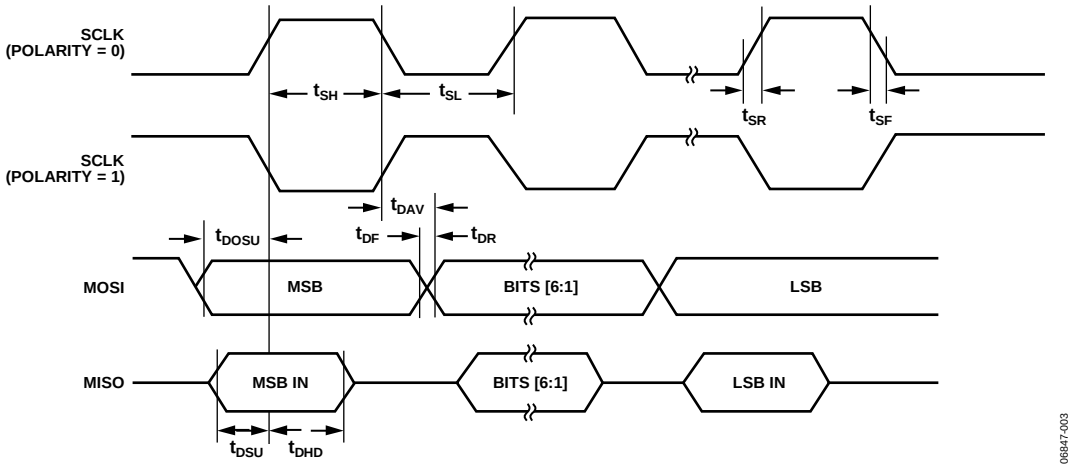

Figure 3. SPI Master Mode Timing (PHASE Mode = 0)

ABSOLUTE MAXIMUM RATINGS

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

Table 6.

Parameter	Rating
AGND to DGND to VSS to IO_VSS	−0.3 V to +0.3 V
VBAT to AGND	−22 V to +40 V
VDD to VSS	−0.3 V to +33 V
VDD to VSS for 1 sec	−0.3 V to +40 V
LIN to IO_VSS	−16 V to +40 V
STI/WU to IO_VSS	−3 V to +33 V
WU Continuous Current	50 mA
High Voltage I/O Pins Short-Circuit Current	100 mA
Digital I/O Voltage to DGND	−0.3 V to REG_DVDD + 0.3 V
VREF to AGND	−0.3 V to REG_AVDD + 0.3 V
ADC Inputs to AGND	−0.3 V to REG_AVDD + 0.3 V
Storage Temperature	125°C
Junction Temperature	
Transient	150°C
Continuous	130°C
Lead Temperature	
Soldering Reflow (15 sec)	260°C

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

ESD CAUTION

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

TERMINOLOGY

Conversion Rate
The conversion rate specifies the rate at which an output result is available from the ADC, after the ADC settles.

The sigma-delta (Σ-Δ) conversion techniques used on this part mean that while the ADC front-end signal is oversampled at a relatively high sample rate, a subsequent digital filter is used to decimate the output giving a valid 16-bit data conversion result at output rates from 1 Hz to 8 kHz.

Note that when software switches from one input to another on the same ADC, the digital filter must first be cleared and then allowed to average a new result. Depending on the configuration of the ADC and the type of filter, this may require multiple conversion cycles.

Integral Nonlinearity (INL)
INL is the maximum deviation of any code from a straight line passing through the endpoints of the transfer function. The endpoints of the transfer function are zero scale, a point ½ LSB below the first code transition, and full scale, a point ½ LSB above the last code transition (111 . . . 110 to 111 . . . 111). The error is expressed as a percentage of full scale.

No Missing Codes
No missing codes is a measure of the differential nonlinearity of the ADC. The error is expressed in bits and specifies the number of codes (ADC results) as 2^N bits, where N = no missing codes, guaranteed to occur through the full ADC input range.

Offset Error
Offset error is the deviation of the first code transition ADC input voltage from the ideal first code transition.

Offset Error Drift
Offset error drift is the variation in absolute offset error with respect to temperature. This error is expressed as LSBs per °C.

Gain Error
Gain error is a measure of the span error of the ADC. It is a measure of the difference between the measured and the ideal span between any two points in the transfer function.

Output Noise
The output noise is specified as the standard deviation (or 1 × Σ) of ADC output codes distribution collected when the ADC input voltage is at a dc voltage. It is expressed as μ rms. The output, or rms noise, can be used to calculate the effective resolution of the ADC as defined by the following equation:

$$\textit{Effective Resolution} = \log_2(\textit{Full-Scale Range/rms Noise})$$

where *Effective Resolution* is expressed in bits.

The peak-to-peak noise is defined as the deviation of codes that fall within 6.6 × Σ of the distribution of ADC output codes collected when the ADC input voltage is at dc. The peak-to-peak noise is therefore calculated as 6.6 × the rms noise.

The peak-to-peak noise can be used to calculate the ADC (noise-free code) resolution for which there is no code flicker within a 6.6-Σ limit as defined by the following equation:

$$\textit{Noise-Free Code Resolution} = \log_2(\textit{Full-Scale Range}/\textit{Peak-to-Peak Noise})$$

where *Noise-Free Code Resolution* is expressed in bits.

Data Sheet Acronyms

ADC	analog-to-digital converter
ARM	advanced RISC machine
JTAG	joint test action group
LIN	local interconnect network
LSB	least significant byte/bit
LVF	low voltage flag
MCU	microcontroller
MMR	memory mapped register
MSB	most significant byte/bit
OTP	one time programmable
PID	protected identifier
POR	power-on reset
PSM	power supply monitor
rms	root mean square
STI	serial test interface

THEORY OF OPERATION

The ADuC7033 is a complete system solution for battery monitoring in 12 V automotive applications. The 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 ADuC7033 to the main electronic control unit (ECU) via a local interconnect network (LIN) interface that is integrated on-chip.

Both the MCU and the ADC subsystem can be individually 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 20.48 MHz. In its power-saving operating modes, the MCU can be totally powered down, waking up only in response to an ADC conversion result ready, digital comparators, the wake-up timer, a POR, or an external serial 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. The current channel features two low power modes, low power and low power plus, generating conversion results to a lower performance specification.

On-chip factory firmware supports in-circuit Flash/EE memory 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 ADuC7033.

The ADuC7033 operates directly from the 12 V battery supply and is fully specified over a temperature range of −40°C to +115°C. The ADuC7033 is functional, but with degraded performance, at temperatures from 115°C to 125°C.

OVERVIEW OF THE ARM7TDMI CORE

The ARM7 core is a 32-bit, reduced instruction set computer (RISC) developed by ARM® Ltd. The ARM7TDMI® 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 is an ARM7 core with four additional features, as listed in Table 8.

Table 8. ARM7TDMI

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 processor supports a second instruction set compressed into 16 bits, called 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 core particularly suited for embedded applications.

However, the Thumb mode has three limitations as follows:

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

EmbeddedICE (I)

The EmbeddedICE module provides integrated on-chip debug support for the ARM7TDMI. 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 memory, SRAM, and memory mapped registers.

ARM7 Exceptions

The ARM7 supports five types of exceptions, with a privileged processing mode associated with each type. The five types of exceptions are as follows:

Remap Operation

When a reset occurs on the ADuC7033, 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 ADuC7033 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 memory is mirrored at the bottom of the memory array at reset, the reset routine must always be written in Flash/EE memory.

The remap command must be executed from the absolute Flash/EE memory 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.

SYSMAP0 Register

Name: SYSMAP0

Address: 0xFFFF0220

Default Value: Updated by the kernel

Access: Read/write access

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

Table 10. SYSMAP0 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. Set by the user to remap the SRAM to 0x00000000. Cleared automatically after reset to remap the Flash/EE memory to 0x00000000.

FLASH/EE MEMORY SECURITY

The 94 kB of Flash/EE memory available to the user can be read and write protected using the FFE0HID and FEE1HID registers.

In Block0, the FEE0HID MMR protects the 30 kB of Flash/EE memory. Bit 0 to Bit 28 of this register protect Page 0 to Page 57 from writing. Each bit protects two pages, that is, 1 kB. Bit 29 to Bit 30 protect Page 58 and Page 59 respectively, that is, each bit write protects a single page of 512 bytes. The MSB of this register (Bit 31) protects Block0 from been read via JTAG.

The FEE0PRO register mirrors the bit definitions of the FEE0HID MMR. The FEE0PRO 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 set and test protection settings temporarily using the FEE0HID MMR and subsequently lock the required protection configuration (using FEE0PRO) when shipping protection systems into the field.

In Block1 (64 kB), the FEE1HID MMR protects the 64 kB of Flash/EE memory. Bit 0 to Bit 29 of this register protect Page 0 to Page 119 from writing. Each bit protects four pages, that is, 2 kB. Bit 30 protects Page 120 to Page 127, that is, Bit 30 write protects eight pages of 512 bytes. The MSB of this register (Bit 31) protects Flash/EE Block1, from been read via JTAG.

As with Block0, FEE1PRO register mirrors the bit definitions of the FEE1HID MMR. The FEE1PRO 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.

Block0, Flash/EE Memory Protection Registers

Name: FEE0HID and FEE0PRO

Address: 0xFFFF0E20 (for FEE0HID) and 0xFFFF0E1C (for FEE0PRO)

Default Value: 0xFFFFFFFF (for FEE0HID) and 0x00000000 (for FEE0PRO)

Access: Read/write access

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

Table 16. FEE0HID and FEE0PRO MMR Bit Designations

Bit	Description
31	Read Protection. Cleared by user to protect the 32 kB Flash/EE block code through JTAG read access. Set by user to allow reading of the 32 kB Flash/EE block code through JTAG read access.
30	Write Protection Bit. Set by user code to unprotect Page 59. Cleared by user code to write protect Page 59.
29	Write Protection Bit. Set by user code to unprotect Page 58. Cleared by user code to write protect Page 58.
28 to 0	Write Protection Bits. When set by user code, these bits unprotect Page 0 to Page 57 of the 30 kB Flash/EE code memory. Each bit write protects two pages and each page consists of 512 bytes. When cleared by user code, these bits write protect Page 0 to Page 57 of the 30 kB Flash/EE code memory. Each bit write protects two pages and each page consists of 512 bytes.

In summary, there are three levels of protection.

Temporary Protection

Set and remove temporary protection by writing directly into the FEExHID MMR. This register is volatile and therefore protection is only in place while the part remains powered on. This protection is not reloaded after a power cycle.

Keyed Permanent Protection

Set keyed permanent protection via FEExPRO to lock the protection configuration. The software key used at the start of the required FEExPRO write sequence is saved one time only and thereafter must be used for any subsequent access of the FEExHID or FEExPRO MMRs. A mass erase resets the key to 0xFFFF but also erases the entire user code space.

Permanent Protection

Set permanent protection via FEExPRO, similarly to keyed permanent protection, the only difference is that the Software Key 0xDEADDEAD is used. Once the FEExPRO write sequence is saved, only a mass erase resets the key to 0xFFFFFFFF. This also erases the entire user code space.

Sequence to Write the Key and Set Permanent Protection

1. Write in FEExPRO corresponding to the pages to be protected.
2. Write the new (user defined) 32-bit key in FEExADR (Bits[31:16]) and FEExDAT (Bits[15:0]).
3. Write 1, 0 in FEExMOD[6:5] and set FEExMOD[3].
4. Run the Write Key Command 0x0C in FEExCON.

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

Sequence Example

The sequence to write the key and set permanent protection is illustrated in the following example; this protects writing Page 4 and Page 5 of the Flash/EE memory:

```
Int a = FEExSTA;          //Ensure FEExSTA is cleared
FEExPRO = 0xFFFFFFFB;     //Protect Pages[4:5]
FEExADR = 0x66BB;         //32 bit key value (Bits[31:16])
FEExDAT = 0xAA55;         //32 bit key value (Bits[15:0])
FEExMOD = 0x0048;         //Lock security sequence
FEExCON = 0x0C;           //Write key command
while (FEExSTA & 0x04){}  //Wait for command to finish
```

ADC GROUND SWITCH

The ADuC7033 features an integrated ground switch pin, GND_SW, Pin15. This switch allows the user to dynamically disconnect ground from external devices. It allows either a direct connection to ground, or a connection to ground using a 20 kΩ resistor. This additional resistor can be used to reduce the number of external components required for an NTC circuit. The ground switch feature can be used for reducing power consumption on application specific boards.

An example application is shown in Figure 19.

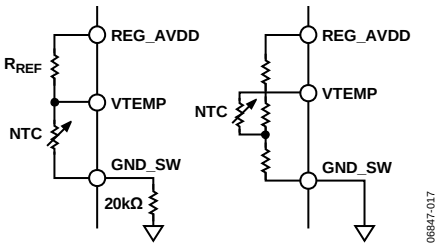

Figure 19. Example External Temperature Sensor Circuits

This diagram shows an external NTC used in two modes, one using the internal 20 kΩ resistor, and the second showing a direct connection to ground, via the GND_SW.

ADCCFG[7] controls the connection of the ground switch to ground and $\overline{\text{ADCMDE}}$[6] controls the GND_SW resistance as shown in Figure 20.

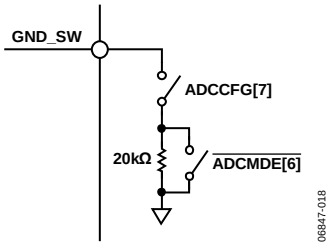

Figure 20. Internal Ground Switch Configuration

The possible combinations are shown in Table 31.

Table 31. GND_SW Configuration

ADCCFG[7]	$\overline{\text{ADCMDE}}$[6]	GND_SW
0	0	Floating
0	1	Floating
1	0	Direct connection to ground
1	1	Connected to ground via 20 kΩ resistor

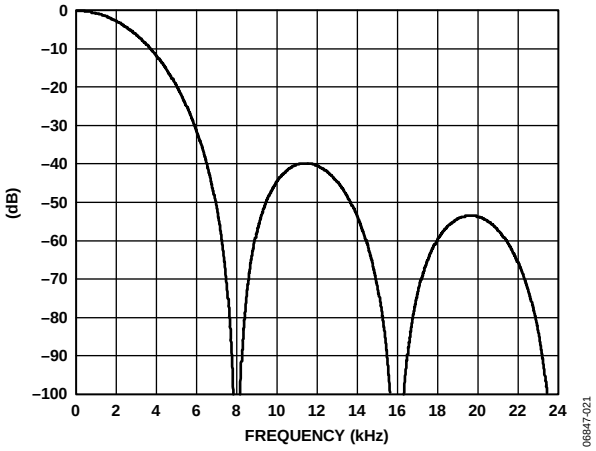

Figure 23. Typical Digital Filter Response at f_{ADC} = 8 kHz (ADCFLT = 0x0000)

A modified version of the 8 kHz filter response can be configured by setting the running average bit (ADCFLT[14]). This has the effect of introducing an additional running-average-by-two filter on all ADC output samples. This further reduces the ADC output noise and by maintaining an 8 kHz ADC throughput rate, the ADC settling time is increased by one full conversion period. The modified frequency response for this configuration is shown in Figure 24.

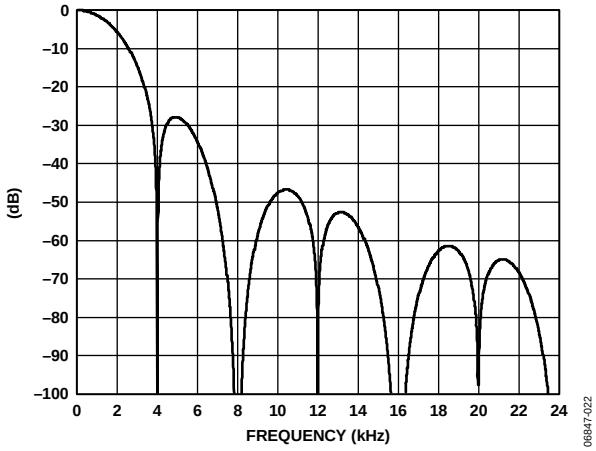

Figure 24. Typical Digital Filter Response at f_{ADC} = 8 kHz (ADCFLT = 0x4000)

At very low throughput rates, the chop bit in the ADCFLT register can be enabled to minimize offset errors and, more importantly, temperature drift in the ADC offset error. With chop enabled, there are two primary variables (Sinc3 decimation factor and averaging factor) available to allow the user to select an optimum filter response, trading off filter bandwidth against ADC noise.

For example, with the Chop Enable Bit ADCFLT[15] set to 1, increasing the SF value (ADCFLT[6:0]) to 0x1F (31 decimal) and selecting an AF value (ADCFLT[13:8]) of 0x16 (22 decimal) results in an ADC throughput of 10 Hz. The frequency response in this case is shown in Figure 25.

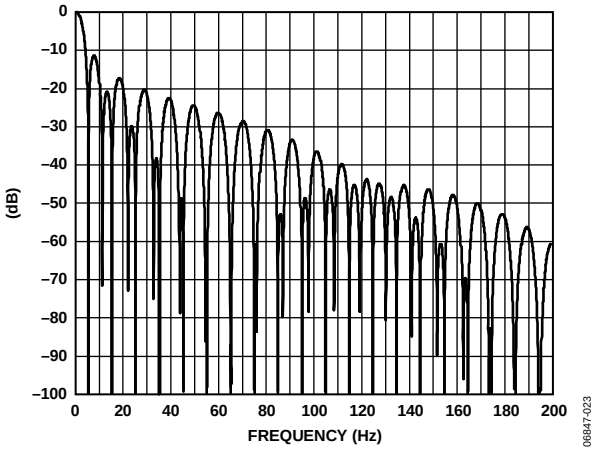

Figure 25. Typical Digital Filter Response at f_{ADC} = 10 Hz (ADCFLT = 0x961F)

Changing SF to 0x1D and setting AF to 0x3F with the chop bit enabled configures the ADC into its minimum throughput rate in normal mode of 4 Hz. The digital filter frequency response with this configuration is shown in Figure 26.

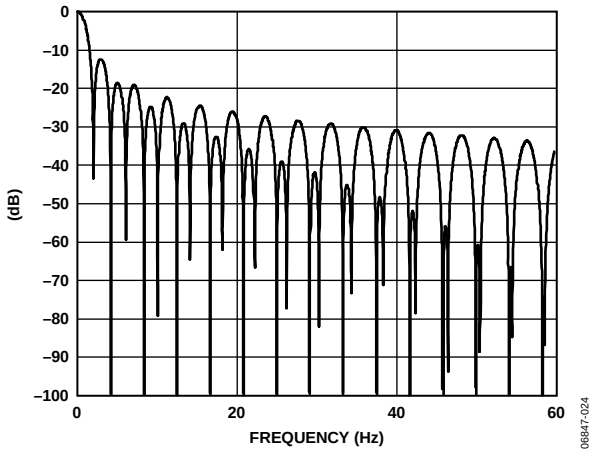

Figure 26. Typical Digital Filter Response at f_{ADC} = 4 Hz (ADCFLT = 0xBF1D)

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 (131 kHz) oscillator. Subsequently, for the same ADCFLT configurations in normal mode, all filter values should be scaled by a factor of approximately four. This means that it is possible to configure the ADC for 1 Hz throughput in low power mode. The filter frequency response for this configuration is shown in Figure 27.

In summary, the simplified ADC transfer function can be described as

$$ADC_{OUT} = \left[\frac{V_{IN} \times PGA}{V_{REF}} - ADCOF \right] \times \frac{ADCGN}{ADCGN_{NOM}}$$

This equation is valid for the voltage/temperature channel ADC.

For the current channel ADC,

$$ADC_{OUT} = \left[\frac{V_{IN} \times PGA}{V_{REF}} - K \times ADCOF \right] \times \frac{ADCGN}{ADCGN_{NOM}}$$

where *K* is dependent on the *PGA* gain setting and *ADC* mode.

Normal Mode

For PGA gains of 1, 4, 8, 16, 32, and 64, the K factor is 1. For PGA gains of 2 and 128, the K factor is 2. For a PGA gain of 256, the K factor is 4. For a PGA gain of 512, the K factor is 8.

Low Power Mode

The PGA gain is set to 128 and the K factor is 32.

Low Power Plus Mode

The PGA gain is set to 512 and the K factor is 8.

In low power and low power plus modes, the K factor doubles if (REG_AVDD)/2 is used as the reference.

ADC DIAGNOSTICS

The ADuC7033 features diagnostic capability on both ADCs.

Current ADC Diagnostics

The ADuC7033 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%. A PGA gain ≥ 2 (ADC0CON[3:0] ≥ 0001) must be used when current sources are enabled.

Temperature ADC Diagnostics

The ADuC7033 features the capability to detect open-circuit conditions on the temperature channel inputs. This is accomplished using the two current sources on VTEMP and GND_SW, controlled via ADC1CON[14:13].

OSC0STA Register

Name: OSC0STA

Address: 0xFFFF0444

Default Value: 0x00

Access: Read access only

Function: This 8-bit register gives the status of the low power oscillator calibration routine.

Table 49. OSC0STA MMR Bit Designations

Bit	Description
7 to 2	Reserved.
1	Calibration Complete. Set by hardware on full completion of a calibration cycle. Cleared by a read of OSC0VAL1.
0	Set if calibration is in progress. Cleared if calibration is completed.

OSC0VAL0 Register

Name: OSC0VAL0

Address: 0xFFFF0448

Default Value: 0x0000

Access: Read access only

Function: This 9-bit counter is clocked from either the 131 kHz precision oscillator or the 32.768 kHz external crystal.

OSC0VAL1 Register

Name: OSC0VAL1

Address: 0xFFFF044C

Default Value: 0x0000

Access: Read access only

Function: This 10-bit counter is clocked from the low power, 131 kHz oscillator.

IRQ

The IRQ is the exception signal to enter the IRQ mode of the processor. It is used to service 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 core. The four 32-bit registers dedicated to IRQ follow.

IRQSIG

IRQSIG 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. IRQSIG is read only.

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 clearing the IRQEN register 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 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 51. This MMR allows the control of a programmed source interrupt.

Table 51. 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 to be detected by the interrupt controller and to be detected by the user in the IRQSTA and FIQSTA registers, any interrupt signal must be active for at least the minimum interrupt latency time.

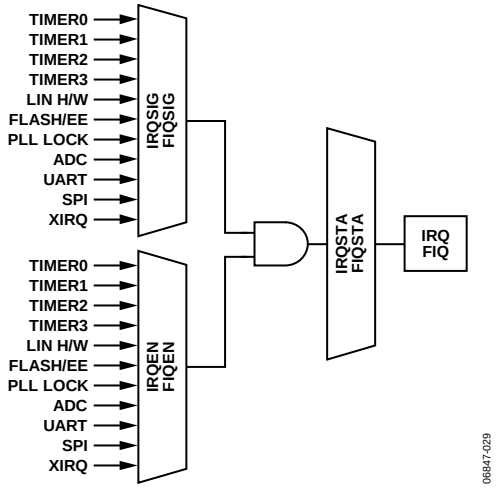

Figure 31. Interrupt Structure

TIMER3 OR WATCHDOG TIMER

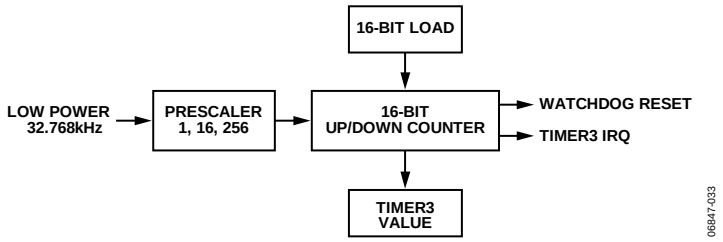

Figure 37. Timer3 Block Diagram

Timer3 has two modes of operation, normal mode and watchdog mode. The watchdog timer is used to recover from an illegal software state. When enabled, it requires periodic servicing to prevent it from forcing a reset of the processor.

Normal Mode

The Timer3 in normal mode is identical to Timer0 in 16-bit mode of operation, except for the clock source. The clock source is the low power, 32.768 kHz oscillator scalable by a factor of 1, 16, or 256.

Timer3 reloads the value from T3LD when Timer3 overflows.

Watchdog Mode

Watchdog mode is entered by setting T3CON[5]. Timer3 decrements from the timeout value present in the T3LD register until zero. The maximum timeout is 512 seconds, using a maximum prescaler/256 and full scale in T3LD.

User software should not configure a timeout period of less than 30 ms. This is to avoid any conflict with Flash/EE memory page erase cycles that require 20 ms to complete a single page erase cycle and kernel execution.

If T3VAL reaches 0, a reset or an interrupt occurs, depending on T3CON[1]. To avoid a reset or an interrupt event, any value must be written to T3CLRI before T3VAL reaches zero. This reloads the counter with T3LD and begins a new timeout period.

When watchdog mode is entered, T3LD and T3CON are write protected. These two registers cannot be modified until a power-on reset event resets the watchdog timer. After any other reset event, the watchdog timer continues to count. The watchdog timer should be configured in the initial lines of user code to avoid an infinite loop of watchdog resets. User software should only configure a minimum timeout period of 30 ms.

Timer3 is automatically halted during JTAG debug access and only recommences counting after JTAG has relinquished control of the ARM7 core. By default, Timer3 continues to count during power-down. This can be disabled by setting Bit 0 in T3CON. It is recommended to use the default value, that is, that the watchdog timer continues to count during power-down.

Timer3 Interface

The Timer3 interface consists of four MMRs.

T3CON is the configuration MMR described in Table 56.

T3LD and T3VAL are 16-bit registers (Bit 0 to Bit 15) and hold 16-bit unsigned integers. T3VAL is read only.

T3CLRI is an 8-bit register. Writing any value to this register clears the Timer3 interrupt in normal mode or resets a new timeout period in watchdog mode.

Timer3 Load Register

Name: T3LD

Address: 0xFFFF0360

Default Value: 0x0040

Access: Read/write

Function: This 16-bit MMR holds the Timer3 reload value.

Timer3 Value Register

Name: T3VAL

Address: 0xFFFF0364

Default Value: 0x0040

Access: Read only

Function: This 16-bit, read only MMR holds the current Timer3 count value.

Timer3 Clear Register

Name: T3CLRI

Address: 0xFFFF036C

Access: Write only

Function: This 16-bit, write-only MMR is written (with any value) by user code to refresh (reload) Timer3 in watchdog mode to prevent a watchdog timer reset event.

GPIO Port2 Data Register

Name: GP2DAT

Address: 0xFFFF0D40

Default Value: 0x000000XX

Access: Read/write

Function: This 32-bit MMR configures the direction of the GPIO pins assigned to Port2 (see Table 58). This register also sets the output value for GPIO pins configured as outputs and reads the status of GPIO pins configured as inputs.

Table 64. GP2DAT MMR Bit Designations

Bit	Description
31	Reserved. This bit is reserved and should be written as 0 by user code.
30	Port2.6 Direction Select Bit. Cleared to 0 by user code to configure the GPIO pin assigned to Port2.6 as an input. Set to 1 by user code to configure the GPIO pin assigned to Port2.6 as an output.
29	Port2.5 Direction Select Bit. Cleared to 0 by user code to configure the GPIO pin assigned to Port2.5 as an input. Set to 1 by user code to configure the GPIO pin assigned to Port2.5 as an output. This configuration is used to support diagnostic write capability to the high voltage I/O pins.
28	Port2.4 Direction Select Bit. Cleared to 0 by user code to configure the GPIO pin assigned to Port2.4 as an input. This configuration is used to support diagnostic readback capability from the high voltage I/O pins (see HVCFG1[2:0]). Set to 1 by user code to configure the GPIO pin assigned to Port2.4 as an output.
27 to 26	Reserved. These bits are reserved and should be written as 0 by user code.
25	Port2.1 Direction Select Bit. Cleared to 0 by user code to configure the GPIO pin assigned to Port2.1 as an input. Set to 1 by user code to configure the GPIO pin assigned to Port2.1 as an output.
24	Port2.0 Direction Select Bit. Cleared to 0 by user code to configure the GPIO pin assigned to Port2.0 as an input. Set to 1 by user code to configure the GPIO pin assigned to Port2.0 as an output.
23	Reserved. This bit is reserved and should be written as 0 by user code.
22	Port2.6 Data Output. The value written to this bit appears directly on the GPIO pin assigned to Port2.6.
21	Port2.5 Data Output. The value written to this bit appears directly on the GPIO pin assigned to Port2.5.
20 to 18	Reserved. These bits are reserved and should be written as 0 by user code.
17	Port2.1 Data Output. The value written to this bit appears directly on the GPIO pin assigned to Port2.1.
16	Port2.0 Data Output. The value written to this bit appears directly on the GPIO pin assigned to Port2.0.
15 to 7	Reserved. These bits are reserved and should be written as 0 by user code.
6	Port2.6 Data Input. This bit is a read-only bit that reflects the current status of the GPIO pin assigned to Port2.6. User code should write 0 to this bit.
5	Port2.5 Data Input. This bit is a read-only bit that reflects the current status of the GPIO pin assigned to Port2.5. User code should write 0 to this bit.
4	Port2.4 Data Input. This bit is a read-only bit that reflects the current status of the GPIO pin assigned to Port2.4. User code should write 0 to this bit.
3 to 2	Reserved. These bits are reserved and should be written as 0 by user code.
1	Port2.1 Data Input. This bit is a read-only bit that reflects the current status of the GPIO pin assigned to Port2.1. User code should write 0 to this bit.
0	Port2.0 Data Input. This bit is a read-only bit that reflects the current status of the GPIO pin assigned to Port2.0. User code should write 0 to this bit.

High Voltage Configuration1 Register

Name: HVCFG1

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 ADuC7033. 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 through HVDAT and data is read back from this register using HVDAT.

Table 75. HVCFG1 Bit Designations

Bit	Description
7	Attenuator Enable Bit. Cleared to 0 to disable the internal voltage attenuator and attenuator buffer. Set to 1 to enable the internal voltage attenuator and attenuator buffer.
6	High Voltage Temperature Monitor. The high voltage temperature monitor is an uncalibrated temperature monitor located on-chip close to the high voltage circuits. This monitor is completely separate to the on-chip, precision temperature sensor (controlled via ADC1CON[7:6]) and allows user code to monitor die temperature change close to the hottest part of the ADuC7033 die. The monitor generates a typical output voltage of 600 mV at 25°C and has a negative temperature coefficient of typically −2.1 mV/°C. Set to 1 to enable the on-chip, high voltage temperature monitor. When enabled, this voltage output temperature monitor is routed directly to the voltage channel ADC. Cleared to 0 to disable the on-chip, high voltage temperature monitor.
5	Voltage Channel Short Enable Bit. Set to 1 to enable an internal short (at the attenuator, before the ADC input buffer) on the voltage channel ADC and allows noise be measured as a self-diagnostic test. Cleared to 0 to disable an internal short on the voltage channel.
4	WU and STI Readback Enable Bit. Cleared to 0 to disable input capability on the external WU/STI pins. Set to 1 to enable input capability on the external WU/STI pins. In this mode, a rising or falling edge transition on the WU/STI pins generates a high voltage interrupt. When this bit is set, the state of the WU/STI pins can be monitored via the HVMON register (HVMON[7] and HVMON[5]).
3	High Voltage I/O Driver Enable Bit. Set to 1 to re-enable any high voltage I/O pins (LIN/BSD, STI, and WU) that have been disabled as a result of a short-circuit current event (the event must last longer than 20 µs for LIN/BSD and STI pins and 400 µs for the WU pin). This bit must also be set to 1 to re-enable the WU and STI pins if they were disabled by a thermal event. Note that this bit must be set to clear any pending interrupt generated by the short-circuit event (even if the event has passed) as well as re-enabling the high voltage I/O pins. Cleared to 0 automatically.
2	Enable/Disable Short-Circuit Protection (LIN/BSD and STI). Set to 1 to enable passive short-circuit protection on the LIN pin. In this mode, a short-circuit event on the LIN/BSD pin generates a high voltage interrupt, IRQ3 (if enabled in IRQEN[16]), and asserts the appropriate status bit in HVSTA, but does not disable the short-circuiting pin. Cleared to 0 to enable active short-circuit protection on the LIN/BSD pin. In this mode, during a short-circuit event, the LIN/BSD pin generates a high voltage interrupt (IRQ3), asserts HVSTA[16], and automatically disables the short-circuiting pin. When disabled, the I/O pin can only be re-enabled by writing to HVCFG1[3].
1	WU Pin Timeout (Monoflop) Counter Enable/Disable. Set to disable the WU I/O timeout counter. Cleared to enable a timeout counter that automatically deasserts the WU pin 1.3 seconds after user code has asserted the WU pin via HVCFG0[4].
0	WU Open-Circuit Diagnostic Enable. Set to enable an internal WU I/O diagnostic pull-up resistor to the VDD pin, thus allowing detection of an open-circuit condition on the WU pin. Cleared to disable an internal WU I/O diagnostic pull-up resistor.

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 a change of state for all the corresponding bits in the HVMON register. 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 77. HVSTA Bit Designations

Bit	Description
7 to 6	Reserved. These bits should not be used and are reserved for future use.
5	PSM Status Bit. Valid only if enabled via HVCFG0[3]. This bit is not latched and the IRQ needs to be enabled to detect it. This bit is 0 if the voltage at the VDD pin stays above 6.0 V. This bit is 1 if the voltage at the VDD pin drops below 6.0 V.
4	WU Request Status Bit. Valid only if enabled via HVCFG1[4]. When enabled via HVCFG1[4], this bit is set to 1 to indicate that a rising or falling edge transition on the WU pin generated a high voltage interrupt.
3	Overtemperature. This bit is always enabled. This bit is 0 if a thermal shutdown event has not occurred. This bit is 1 if a thermal shutdown event has occurred. All high voltage (LIN/BSD, WU, and STI) pin drivers are automatically disabled once a thermal shutdown has occurred.
2	LIN/BSD Short-Circuit Status Flag. This bit is 0 during normal LIN/BSD operation and is cleared automatically by reading the HVSTA register. This bit is 1 if a LIN/BSD short circuit is detected. In this condition, the LIN driver is automatically disabled.
1	STI Short-Circuit Status Flag. This bit is 0 if the STI driver is operating normally and is cleared automatically by reading the HVSTA register. This bit is 1 if the STI driver has experienced a short-circuit condition.
0	WU Short-Circuit Status Flag. This bit is 0 during normal wake operation. This bit is 1 if a wake-up short circuit is detected.

WAKE-UP (WU) PIN

The wake-up (WU) pin is a high voltage GPIO controlled through HVCON and HVDAT.

WU Pin Circuit Description

The WU pin is configured by default as an output with an internal 10 kΩ pull-down resistor and high-side FET driver. The WU pin, in its default mode of operation, is specified to generate an active high system wake-up request by forcing the external system WU bus high. User code can assert the WU output by writing directly to HVCFG0[4].

Note that the output responds only after the 10 µs latency through the (serial communication based) high voltage interface.

The internal FET is capable of sourcing significant current and, therefore, substantial on-chip self-heating can occur if this driver is asserted for a long time period. For this reason, a monoflop (a 1.3-second timeout timer) is included.

By default, the monoflop is enabled and disables the wake-up driver after 1.3 seconds. It is possible to disable the monoflop through HVCFG1[1]. If the wake-up monoflop is disabled, then the wake-up driver should be disabled after 1.3 seconds.

The WU pin also features a short-circuit detection feature. When the wake-up pin sources more than 100 mA typically for 400 µs, a high voltage interrupt is generated with HVMON[0] set.

A thermal shutdown event disables the WU driver. The WU driver must be re-enabled manually after a thermal event using HVCFG1[3]. It is possible to disable the automatic shutdown during a thermal event via HVCFG0[7].

The WU pin can be configured in I/O mode by writing a 1 to HVCFG1[4]. In this mode, a rising or falling edge immediately generates a high voltage interrupt. HVMON[7] directly reflects the state of the external WU pin. This comparator has a trip level of 3 V_{TYP}.

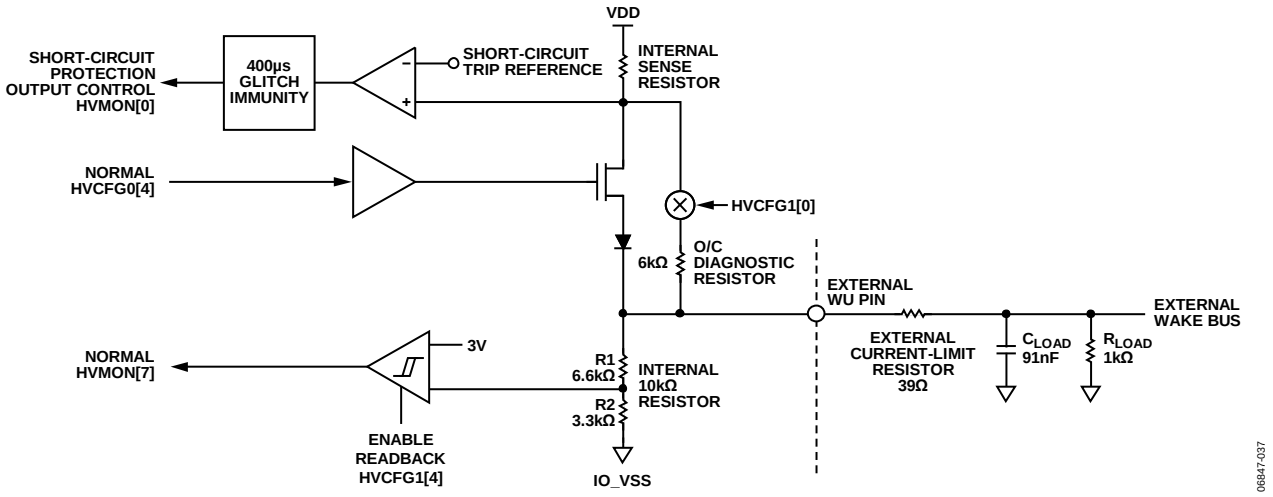

Figure 41. WU Circuit, Block Diagram

Serial Test Interface Data 2 Register

Name: STIDAT1

Address: 0xFFFF0894

Default Value: 0x0000

Access: Read/write

Function: The STIDAT2 MMR is a 16-bit register that holds the fifth and sixth data bytes that are to be transmitted on the STI pin when the STI port is enabled. The fifth byte to be transmitted occupies Bits[7:0] and the sixth byte occupies Bits[15:8].

Serial Test Interface Control Register

Name: STICON

Address: 0xFFFF0884

Default Value: 0x0000

Access: Read/write access, write protected by two key registers (STIKEY0 and STIKEY1). A write access to STICON is completed correctly only when the following triple write sequence is followed:

1. STIKEY0 MMR is written with 0x0007.
2. STICON is written.
3. The sequence is completed by writing 0x00B9 to STIKEY1.

Function: The STI control MMR is a 16-bit register that configures the mode of operation of the serial test interface.

Note: GPIO_13 must be configured for STI operation in GP2CON for STI communications.

```
while((GP2DAT & 0x10 ) == 0 )
{}                              // Wait until LIN Bus returns high
LHSCON0 = 0x4;                  // Enable LHS to detect Break Condition Ungate RX Line
                                // Disable all Interrupts except Break Compare Interrupt
IRQEN = 0x800;                  // Enable UART Interrupt
                                // The UART is now configured and ready to be used for LIN
```

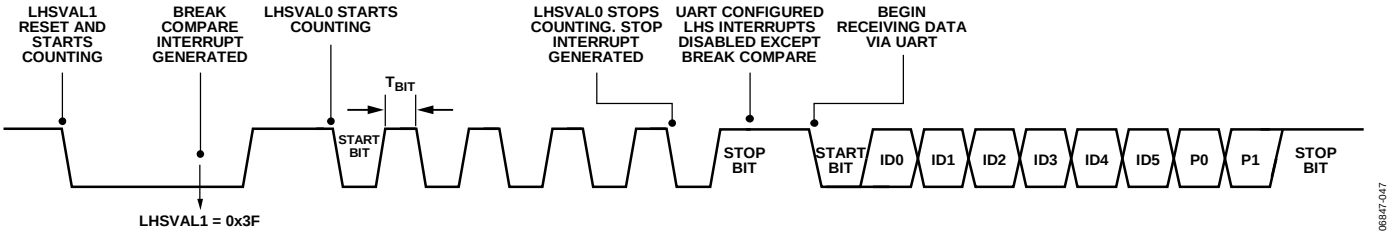

Figure 51. Example LIN Configuration

LIN Diagnostics

The ADuC7033 features the capability to unintrusively monitor the current state of the LIN pin. This readback functionality is implemented using GPIO_11. The current state of the LIN pin is contained in GP2DAT[4].

It is also possible to drive the LIN pin high and low through user software, allowing the user to detect open-circuit conditions. This functionality is implemented via GPIO_12. To enable this functionality, GPIO_12 must be configured as a GPIO through GP2CON[20]. After it is configured, the LIN pin can be pulled high or low using GP2DAT.

The ADuC7033 also features short-circuit protection on the LIN pin. If a short-circuit condition is detected on the LIN pin, HVSTA[2] is set. This bit is cleared by re-enabling the LIN driver using HVCFG1[3]. It is possible to disable this feature through HVCFG1[2].

LIN Operation During Thermal Shutdown

When a thermal event occurs, that is, HVSTA[3] is set, LIN communications continue uninterrupted.

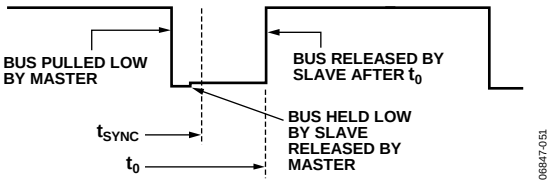

Figure 55. BSD Slave Transmitting Zero

Typical BSD Program Flow

Because BSD is a PWM communications protocol controlled by software, it is necessary for the user to construct the required data from each bit. For example, in constructing the slave address, the slave node receives the three bits and the user constructs the relevant address.

When the master initiates BSD, data is transmitted and received by the slave node. A flow diagram showing this process is shown in Figure 54.

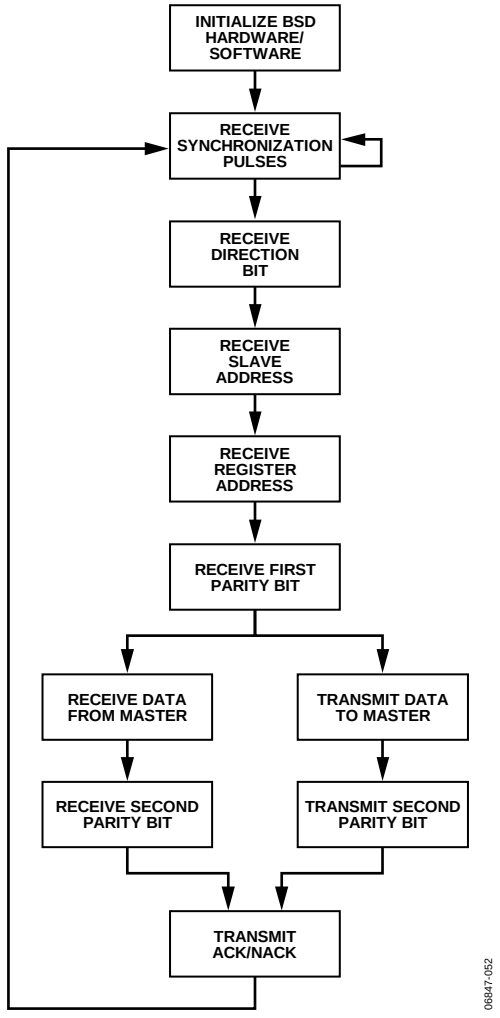

Figure 56. BSD Slave Node State Machine

BSD DATA RECEPTION

To receive data, the LIN/BSD peripheral must first be configured in BSD mode where LHSCON[6] = 1. In this mode, LHSCON0[8] should be set to ensure the LHS break timer (see LHSVAL1 in the LIN Hardware Break Timer1 Register section) generates an interrupt on the rising edge of the BSD bus.

The LHS break timer is cleared and starts counting on the falling edge of the BSD bus and is subsequently stopped and generates an interrupt on the rising edge of the BSD bus. Given that the LHS break timer is clocked by the low power (131 kHz) oscillator, the value in LHSVAL1 can be interpreted by user code to determine if the received data bit is a BSD sync pulse, 0, or 1.

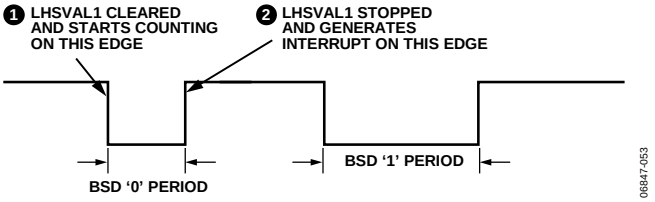

Figure 57. Master Transmit, Slave Read

BSD DATA TRANSMISSION

User code forces a GPIO signal (GPIO_12) low for a specified time to transmit data in BSD mode. In addition, user code uses the sync timer (LHSVAL0), LHS sync capture register (LHSCAP), and the LHS sync compare register (LHSCMP) to time how long the BSD bus should be held low for 0 or 1 bit transmissions.

As described previously, even when the slave is transmitting, the master always starts the bit transmission period by pulling the BSD bus low. If BSD mode is selected (LHSCON0[6] = 1), the LIN sync timer value is captured in LHSCAP on every falling edge of the BSD bus. The LIN sync timer runs continuously in BSD mode.

User code can then immediately force GPIO_12 low and reads the captured timer value from LHSCAP. A calculation of how many (5 MHz) clock periods should elapse before the GPIO_12 should be driven high for a 0 or 1 pulse width can be made. This number is added to the LHSCAP value and written into the LHSCMP register. If LHSCON0[5] is set, the sync timer, which continues to count (being clocked by a 5 MHz clock), eventually equals the LHSCMP value and generates an LHS compare interrupt (LHSSTA[3]).

The response to this interrupt should be to force the GPIO_12 signal (and, therefore, the BSD bus) high. The software control of the GPIO_12 signal along with the correct use of the LIN synchronization timers ensures that valid 0 and 1 pulse widths can be transmitted from the ADuC7033, as shown in Figure 56. Again, care needs to be taken when switching from BSD write mode to BSD read mode, as described in LHSCON0[8].