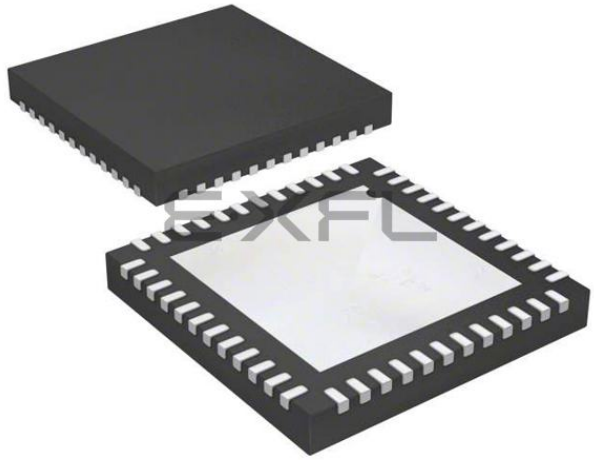

Welcome to E-XFL.COM

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, UART/USART
Peripherals	PSM, Temp Sensor, WDT
Number of I/O	9
Program Memory Size	96KB (48K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1.5K 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	48-VFQFN Exposed Pad, CSP
Supplier Device Package	48-LFCSP-VQ (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/aduc7036dcpz-rl

SPECIFICATIONS

ELECTRICAL SPECIFICATIONS

VDD = 3.5 V to 18 V, VREF = 1.2 V internal reference, f_{CORE} = 20.48 MHz (unless otherwise noted) driven from external 32.768 kHz watch crystal or on-chip precision oscillator. All specifications T_A = −40°C to +115°C, unless otherwise noted.

Table 1.

Parameter	Test Conditions/Comments	Min	Typ	Max	Unit
ADC SPECIFICATIONS					
Conversion Rate[1]	Chop off, ADC normal operating mode	4		8000	Hz
	Chop on, ADC normal operating mode	4		2600	Hz
	Chop on, ADC low power mode	1		650	Hz
Current Channel					
No Missing Codes[1]	Valid for all ADC update rates and ADC modes	16			Bits
Integral Nonlinearity[1, 2]			±10	±60	ppm of FSR
Offset Error[2, 3, 4, 5]	Chop off, 1 LSB = (36.6/gain) μV	−10	±3	+10	LSB
Offset Error[1, 3, 6]	Chop on	−2	±0.5	+2	μV
Offset Error[1, 3]	Chop on, low power or low power plus mode, MCU powered down	100	−50	−300	nV
Offset Error[1, 3]	Chop on, normal mode	+0.5	−1.25	−3	μV
Offset Error Drift[6]	Chop off, valid for ADC gains of 4 to 64, normal mode		0.03		LSB/°C
Offset Error Drift[6]	Chop off, valid for ADC gains of 128 to 512, normal mode		30		nV/°C
Offset Error Drift[6]	Chop on		10		nV/°C
Total Gain Error[1, 3, 7, 8, 9, 10]	Normal mode	−0.5	±0.1	+0.5	%
Total Gain Error[1, 3, 7, 9]	Low power mode, using ADCREF MMR	−4	±0.2	+4	%
Total Gain Error[1, 3, 7, 9, 11]	Low power plus mode, using precision VREF	−1	±0.2	+1	%
Gain Drift			3		ppm/°C
PGA Gain Mismatch Error			±0.1		%
Output Noise[1, 12]	4 Hz update rate, gain = 512, ADCFLT = 0xBF1D		60	90	nV rms
	4 Hz update rate, gain = 512, ADCFLT = 0x3F1D		75	115	nV rms
	10 Hz update rate, gain = 512, ADCFLT = 0x961F		100	150	nV rms
	10 Hz update rate, gain = 512, ADCFLT = 0x161F		120	180	nV rms
	1 kHz update rate, gain ≥ 64, ADCFLT = 0x8101		0.8	1.2	μV rms
	1 kHz update rate, gain ≥ 64, ADCFLT = 0x0101		1	1.5	μV rms
	1 kHz update rate, gain = 512, ADCFLT = 0x0007		0.6	0.9	μV rms
	1 kHz update rate, gain = 32, ADCFLT = 0x0007		0.8	1.2	μV rms
	1 kHz update rate, gain = 8, ADCFLT = 0x8101		2.1	4.1	μV rms
	1 kHz update rate, gain = 8, ADCFLT = 0x0007		1.6	2.4	μV rms
	1 kHz update rate, gain = 8, ADCFLT = 0x0101		2.6	3.9	μV rms
	1 kHz update rate, gain = 4, ADCFLT = 0x0007		2.0	2.8	μV rms
	8 kHz update rate, gain = 32, ADCFLT = 0x0000		2.5	3.5	μV rms
	8 kHz update rate, gain = 4, ADCFLT = 0x0000		14	21	μV rms
	ADC low power mode, f_{ADC} = 10 Hz, gain = 128		1.25	1.9	μV rms
	ADC low power mode, f_{ADC} = 1 Hz, gain = 128		0.35	0.5	μV rms
	ADC low power plus mode, f_{ADC} = 1 Hz, gain = 512		0.1	0.15	μV rms
	ADC low power plus mode, f_{ADC} = 250 Hz, gain = 512		0.6	0.9	μV rms

LIN Timing Specifications

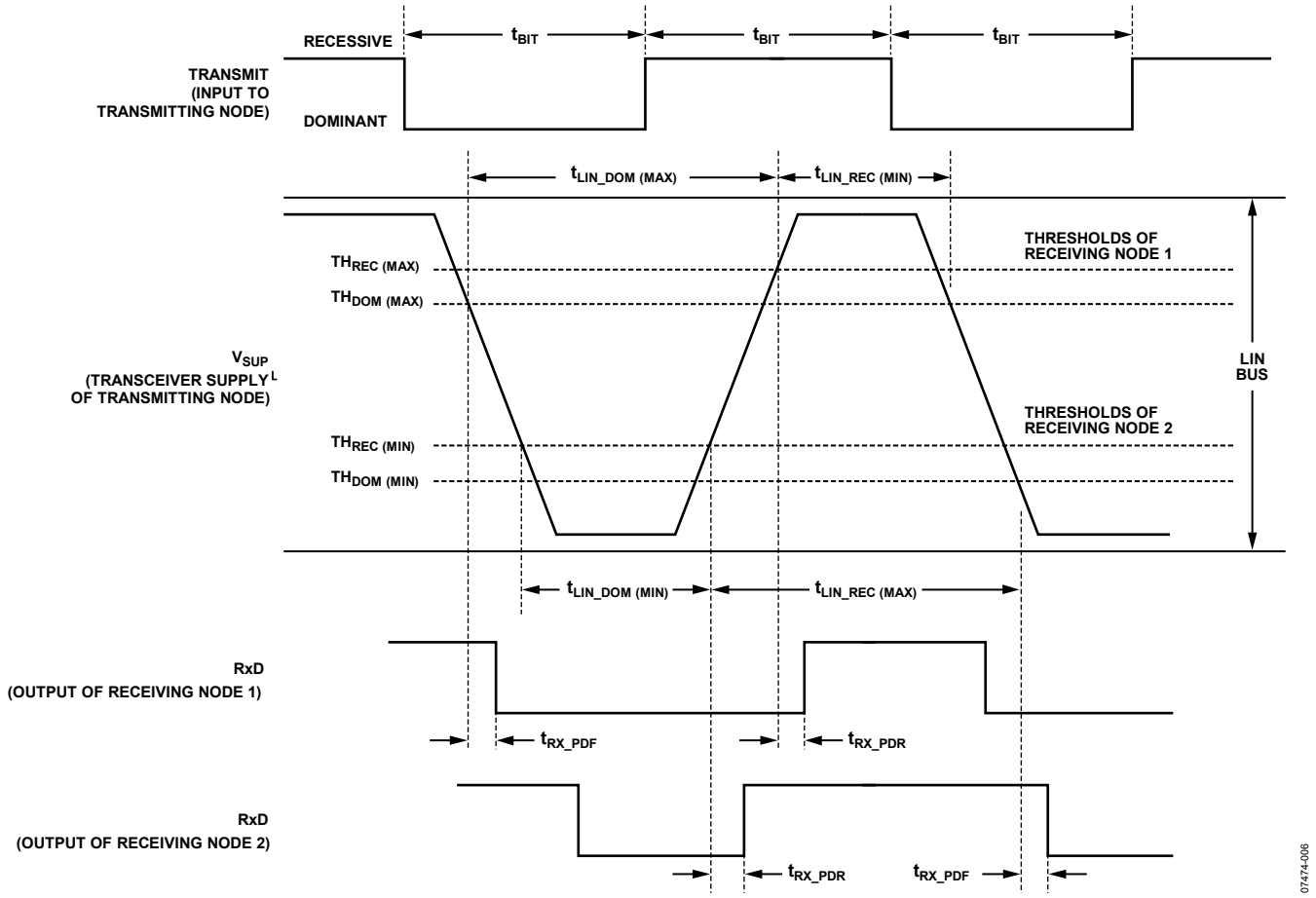

Figure 6. LIN 2.0 Timing Specification

THEORY OF OPERATION

The ADuC7036 is a complete system solution for battery monitoring in 12 V automotive applications. These devices integrate 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 the 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 ADuC7036 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 event, a digital comparator event, a wake-up timer event, a POR event, 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 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 ADuC7036.

The ADuC7036 operates directly from the 12 V battery supply and is fully specified over a temperature range of −40°C to +115°C. The ADuC7036 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 eight, 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, 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 well-suited for embedded applications.

However, the Thumb mode has three limitations.

- Relative to ARM, the Thumb code usually requires more instructions to perform a task. Therefore, ARM code is best for maximizing the performance of time-critical code in most applications.
- The Thumb instruction set does not include some instructions that are needed for exception handling, so ARM code may 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 by 32-bit multiplication with a 64-bit result, or 32-bit by 32-bit multiplication-accumulation (MAC) with a 64-bit result.

EmbeddedICE (I)

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

RESET

There are four kinds of resets: external reset, power-on reset, watchdog reset, and software reset. The RSTSTA register indicates the source of the last reset and can be written to 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 events are shown in Table 12.

RSTSTA Register

Name: RSTSTA

Address: 0xFFFF0230

Default Value: Varies according to type of reset (see Table 11)

Access: Read/write access

Function: This 8-bit register indicates the source of the last reset event and can be written to 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 11. 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. Set automatically to 1 when an external reset occurs. Cleared by setting the corresponding bit in RSTCLR.
2	Software reset. Set to 1 by user code to generate a sofware reset. Cleared by setting the corresponding bit in RSTCLR.[1]
1	Watchdog timeout. Set automatically to 1 when a watchdog timeout occurs. Cleared by setting the corresponding bit in RSTCLR.
0	Power-on reset. Set automatically when a power-on reset occurs. 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 12. Device Reset Implications

	Impact							
Reset	Reset External Pins to Default State	Execute Kernel	Reset All External MMRs (Excluding RSTSTA)	Reset All HV Indirect Registers	Reset Peripherals	Reset Watchdog Timer	Valid RAM[1]	RSTSTA Status (After a 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 a LIN download.
[2] The impact on RAM is dependent on the HVMON[3] contents if LVF is enabled. When LVF is enabled using HVCFG0[2], RAM has not been corrupted by the POR mechanism if the LVF status bit, HVMON[3], is 1. See the Low Voltage Flag (LVF) section for more information.

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 Block 0, the FEE0HID MMR protects the 30 kB. Bits[0:28] of this register protect Page 0 to Page 57 from writing. Each bit protects two pages, that is, 1 kB. Bits[29:30] 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 Block 0 from being 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 Block 1 (64 kB), the FEE1HID MMR protects the 64 kB. Bits[0:29] 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 Block 1 from being read via JTAG.

As with Block 0, the 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.

There are three levels of protection: temporary protection, keyed permanent protection, and permanent protection.

Temporary Protection

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

Keyed Permanent Protection

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

Permanent Protection

Permanent protection can be set via FEExPRO, in a manner similar to the way keyed permanent protection is set, with the only difference being that the software key used is 0xDEADDEAD. When the FEExPRO write sequence is saved, only a mass erase sets the key back to 0xFFFFFFFF. The mass erase also erases the entire user code space.

Sequence to Write the Key and Set Permanent Protection

1. Write 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 Bits[6:5] = 0x10 in FEExMOD.
4. Run the write key command (Code 0x0C) in FEExCON.

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

The previous sequence for writing the key and setting permanent protection is illustrated in the following example sequence, which protects writing Page 4 and Page 5 of the Flash/EE.

```
Int a = FEExSTA;           //Ensure FEExSTA is cleared
FEExPRO =0 xFFFFFFFB;      //Protect Page 4 and Page 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
```

Block 0, 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[1]
31	Read protection bit. Set by user code to allow reading the 32 kB Flash/EE block code via JTAG read access. Cleared by user code to protect the 32 kB Flash/EE block code via JTAG read access.
30	Write protection bit. Set by user code to allow writes to Page 59. Cleared by user code to write protect Page 59.
29	Write protection bit. Set by user code to allow writes to Page 58. Cleared by user code to write protect Page 58.
28 to 0	Write protection bits. Set by user code to allow writes to 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. Cleared by user code to 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.

[1] The x represents 0 or 1, designating Flash/EE Block 0 or Flash/EE Block 1.

Block 1, Flash/EE Memory Protection Registers

Name: FEE1HID and FEE1PRO
Address: 0xFFFF0EA0 (for FEE1HID) and 0xFFFF0E9C (for FEE1PRO)
Default Value: 0xFFFFFFFF (for FEE1HID) and 0x00000000 (for FEE1PRO)
Access: Read/write access
Function: These registers are written by user code to configure the protection of the Flash/EE memory.

Table 17. FEE1HID and FEE1PRO MMR Bit Designations

Bit	Description
31	Read protection bit. Set by user code to allow reading of the 64 kB Flash/EE block code via JTAG read access. Cleared by user code to read protect the 64 kB Flash/EE block code via JTAG read access.
30	Write protection bit. Write protects eight pages. Each page consists of 512 bytes. Set by user code to allow writes to Page 120 to Page 127 of the 64 kB Flash/EE code memory. Cleared by user code to write protect Page 120 to Page 127 of the 64 kB Flash/EE code memory.
29 to 0	Write protection bits. Set by user code to allow writes to Page 0 to Page 119 of the 64 kB Flash/EE code memory. Each bit write protects four pages, and each page consists of 512 bytes. Cleared by user code to write protect Page 0 to Page 119 of the 64 kB Flash/EE code memory. Each bit write protects two pages, and each page consists of 512 bytes.

ON-CHIP KERNEL

The ADuC7036 features an on-chip kernel resident in the top 2 kB of the Flash/EE code space. After any reset event, this kernel copies the factory-calibrated data from the manufacturing data space into the various on-chip peripherals. The peripherals calibrated by the kernel are as follows:

- Power supply monitor (PSM)
- Precision oscillator
- Low power oscillator
- REG_AVDD/REG_DVDD
- Low power voltage reference
- Normal mode voltage reference
- Current ADC (offset and gain)
- Voltage/temperature ADC (offset and gain)

User MMRs that can be modified by the kernel and differ from their POR default values are as follows:

- R0 to R15
- GP0CON/GP2CON
- SYSCHK
- ADCMDE/ADC0CON
- FEE0ADR/FEE0CON/FEE0SIG
- HVDAT/HVCON
- HVCFG0/HVCFG1
- T3LD

The ADuC7036 also features an on-chip LIN downloader. The derivatives ADuC7036BCPZ and ADuC7036CCPZ use Protocol 4 for programming Flash/EE memory via LIN, where Protocol 6 is used on derivative ADuC7036DCPZ. The protocols are described in Application Note AN-881 (Protocol 4) and Application Note AN-946 (Protocol 6).

Flowcharts of the execution of the kernel are shown in Figure 15 and Figure 16. The current revision of the kernel can be derived from SYSSER1, as described in Table 99.

After a POR, the watchdog timer is disabled once the kernel code is exited. For the duration of the kernel execution, the watchdog timer is active with a timeout period of 500 ms. This ensures that when an error occurs in the kernel, the ADuC7036 automatically resets. After any other reset, the watchdog timer maintains user code configuration for the period of the kernel and is refreshed just prior to kernel exit. A minimum watchdog period of 30 ms is required to allow correct LIN downloader operation. If LIN download mode is entered, the watchdog is periodically refreshed.

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

SRAM is not modified during normal kernel execution; rather, SRAM is modified during a LIN download kernel execution.

Note that even with NTRST = 0, user code is not executed unless Address 0x14 contains either 0x27011970 or the checksum of Page 0, excluding Address 0x14. If Address 0x14 does not contain this information, user code is not executed and LIN download mode is entered. During kernel execution, JTAG access is disabled.

With NTRST = 1, user code is always executed.

The ADuC7036DCPZ allows for user-defined bootloader functionality. The bootloader can be of any size up to 30 kB but must be located at the top of user flash. The top-most three words must be the following:

- Address 0x977FC must contain the checksum of the bootloader.
- Address 0x977F8 must contain the lowest address of the bootloader block.
- Address 0x977F4 must contain the entry point of the bootloader code.

The kernel uses the values at these addresses in determining if the bootloader is valid.

Note that this bootloader checksum is the sum of all half words from the value pointed to by 0x977F8 up to the half word at 0x977F6.

Table 27. STI Base Address = 0xFFFF0880

Address	Name	Byte	Access Type	Default Value	Description
0x0880	STIKEY0	4	W	N/A	STICON prewrite key. See the Serial Test Interface Key0 Register section.
0x0884	STICON	2	RW	0x0000	Serial test interface control MMR. See the Serial Test Interface Control Register section and Table 91.
0x0888	STIKEY1	4	W	N/A	STICON postwrite key. See the Serial Test Interface Key1 Register section.
0x088C	STIDAT0	2	RW	0x0000	STI Data MMR 0. See the Serial Test Interface Data0 Register section.
0x0890	STIDAT1	2	RW	0x0000	STI Data MMR 1. See the Serial Test Interface Data1 Register section.
0x0894	STIDAT2	2	RW	0x0000	STI Data MMR 2. See the Serial Test Interface Data2 Register section.

Table 28. SPI Base Address = 0xFFFF0A00

Address	Name	Byte	Access Type	Default Value	Description
0x0A00	SPISTA	1	R	0x00	SPI status MMR. See the SPI Status Register section and Table 90.
0x0A04	SPIRX	1	R	0x00	SPI receive MMR. See the SPI Receive Register section.
0x0A08	SPITX	1	W	N/A	SPI transmit MMR. See the SPI Transmit Register section.
0x0A0C	SPIDIV	1	RW	0x1B	SPI baud rate select MMR. See the SPI Divider Register section.
0x0A10	SPICON	2	RW	0x0000	SPI control MMR. See the SPI Control Register section and Table 89.

Table 29. GPIO Base Address = 0xFFFF0D00

Address	Name	Byte	Access Type	Default Value	Description
0x0D00	GP0CON	4	RW	0x11100000	GPIO Port0 control MMR. See the GPIO Port0 Control Register section and Table 59.
0x0D04	GP1CON	4	RW	0x10000000	GPIO Port1 control MMR. See the GPIO Port1 Control Register section and Table 60.
0x0D08	GP2CON	4	RW	0x01000000	GPIO Port2 control MMR. See the GPIO Port2 Control Register section and Table 61.
0x0D20	GP0DAT[1]	4	RW	0x000000XX	GPIO Port0 data control MMR. See the GPIO Port0 Data Register section and Table 62.
0x0D24	GP0SET	4	W	N/A	GPIO Port0 data set MMR. See the GPIO Port0 Set Register section and Table 65.
0x0D28	GP0CLR	4	W	N/A	GPIO Port0 data clear MMR. See the GPIO Port0 Clear Register section and Table 68.
0x0D30	GP1DAT[1]	4	RW	0x000000XX	GPIO Port1 data control MMR. See the GPIO Port1 Data Register section and Table 63.
0x0D34	GP1SET	4	W	N/A	GPIO Port1 data set MMR. See the GPIO Port1 Set Register section and Table 66.
0x0D38	GP1CLR	4	W	N/A	GPIO Port1 data clear MMR. See the GPIO Port1 Clear Register section and Table 69.
0x0D40	GP2DAT[1]	4	RW	0x000000XX	GPIO Port2 data control MMR. See the GPIO Port2 Data Register section and Table 64.
0x0D44	GP2SET	4	W	N/A	GPIO Port2 data set MMR. See the GPIO Port2 Set Register section and Table 67.
0x0D48	GP2CLR	4	W	N/A	GPIO Port2 data clear MMR. See the GPIO Port2 Clear Register section and Table 70.

[1] Depends on the level on the external GPIO pins.

Voltage/Temperature Channel ADC Control Register

Name: ADC1CON

Address: 0xFFFF0510

Default Value: 0x0000

Access: Read/write

Function: This 16-bit register is used to configure the V-/T-ADC.

Note that when selecting the VBAT attenuator input, the voltage attenuator buffers are automatically enabled.

Table 38. ADC1CON MMR Bit Designations

Bit	Description
15	Voltage/temperature channel ADC enable. Set to 1 by user code to enable the V-/T-ADC. Cleared to 0 to power down the V-/T-ADC.
14, 13	VTEMP current source enable. 00 = current sources off. 01 = enables 50 µA current source on VTEMP. 10 = enables 50 µA current source on GND_SW. 11 = 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. Set to 1 by user code to configure V-/T-ADC output coding as unipolar. 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, 6	Voltage/temperature channel ADC input select. 00 = VBAT/24, AGND. VBAT attenuator selected. The high voltage buffers are enabled automatically in this configuration. 01 = VTEMP, GND_SW. External temperature input selected, conversion result written to ADC2DAT. 10 = internal sensor. Internal temperature sensor input selected, conversion result written to ADC2DAT. The temperature gradient is 0.33 mV/°C; this is only applicable to the internal temperature sensor. 11 = internal short. Shorted input.
5, 4	Voltage/temperature channel ADC reference select. 00 = internal, 1.2 V precision reference selected. 01 = external reference inputs (VREF, GND_SW) selected. 10 = external reference inputs divided-by-2 (VREF, GND_SW)/2 selected. This allows an external reference up to REG_AVDD. 11 = (REG_AVDD, AGND)/2 selected for the voltage channel. (REG_AVDD, GND_SW)/2 selected for the temperature channel.
3 to 0	Not used. These bits are reserved for future functionality and should not be written as 0 by user code.

ADC Filter Register

Name: ADCFLT

Address: 0xFFFF0518

Default Value: 0x0007

Access: Read/write

Function: This 16-bit register controls the speed and resolution of the on-chip ADCs.

Note that if ADCFLT is modified, the current and voltage/temperature ADCs are reset.

Table 39. ADCFLT MMR Bit Designations

Bit	Description
15	Chop enable. Set by the user to enable system chopping of all active ADCs. When this bit is set, the ADC has very low offset errors and drift, but the ADC output rate is reduced by a factor of three if AF = 0 (see Sinc3 decimation factor, Bits[6:0], in this table). If AF > 0, then the ADC output update rate is the same with chop on or off. When chop is enabled, the settling time is two output periods.
14	Running average. Set by the user to enable a running-average-by-two function reducing ADC noise. This function is automatically enabled when chopping is active. It is an optional feature when chopping is inactive, and if enabled (when chopping is inactive), does not reduce the ADC output rate but does increase the settling time by one conversion period. Cleared by the user to disable the running average function.
13 to 8	Averaging factor (AF). The values written to these bits are used to implement a programmable first-order Sinc3 postfilter. The averaging factor can further reduce ADC noise at the expense of output rate, as described in Bits[6:0], Sinc3 decimation factor, in this table.
7	Sinc3 modify. Set by the user to modify the standard Sinc3 frequency response to increase the filter stop-band rejection by approximately 5 dB. This is achieved by inserting a second notch (NOTCH2) a $f_{NOTCH2} = 1.333 \times f_{NOTCH}$, where f_{NOTCH} is the location of the first notch in the response.
6 to 0	Sinc3 decimation factor (SF).[1] The value (SF) written in these bits controls the oversampling (decimation factor) of the Sinc3 filter. The output rate from the Sinc3 filter is given by $f_{ADC} = (512{,}000/([SF + 1] \times 64))$ Hz[2], when the chop bit (Bit 15, chop enable) = 0 and the averaging factor (AF) = 0. This is valid for all SF values ≤ 125. For SF = 126, f_{ADC} is forced to 60 Hz. For SF = 127, f_{ADC} is forced to 50 Hz. For information on calculating the f_{ADC} for SF (other than 126 and 127) and AF values, refer to Table 40.

[1] Due to limitations on the digital filter internal data path, there are some limitations on the combinations of the Sinc3 decimation factor (SF) and averaging factor (AF) that can be used to generate a required ADC output rate. This restriction limits the minimum ADC update in normal power mode to 4 Hz or 1 Hz in lower power mode.
[2] In low power mode and low power plus mode, the ADC is driven directly by the low power oscillator (131 kHz) and not 512 kHz. All f_{ADC} calculations should be divided by 4 (approximately).

Table 40. ADC Conversion Rates and Settling Times

Chop Enabled	Averaging Factor	Running Average	f_{ADC}	$t_{SETTLING}$[1]
No	No	No	$\frac{512{,}000}{[SF+1]\times 64}$	$\frac{3}{f_{ADC}}$
No	No	Yes	$\frac{512{,}000}{[SF+1]\times 64}$	$\frac{4}{f_{ADC}}$
No	Yes	No	$\frac{512{,}000}{[SF+1]\times 64\times[3+AF]}$	$\frac{1}{f_{ADC}}$
No	Yes	Yes	$\frac{512{,}000}{[SF+1]\times 64\times[3+AF]}$	$\frac{2}{f_{ADC}}$
Yes	N/A	N/A	$\frac{512{,}000}{[SF+1]\times 64\times[3+AF]+3}$	$\frac{2}{f_{ADC}}$

[1] An additional time of approximately 60 µs per ADC is required before the first ADC is available.

Table 41. Allowable Combinations of SF and AF

	AF Range		
SF	**0**	**1 to 7**	**8 to 63**
0 to 31	Yes	Yes	Yes
32 to 63	Yes	Yes	No
64 to 127	Yes	No	No

In ADC normal power mode, the maximum ADC throughput rate is 8 kHz. This is configured by setting the SF and AF bits in the ADCFLT MMR to 0, with all other filtering options disabled. As a result, 0x0000 is written to ADCFLT. Figure 24 shows a typical 8 kHz filter response based on these settings.

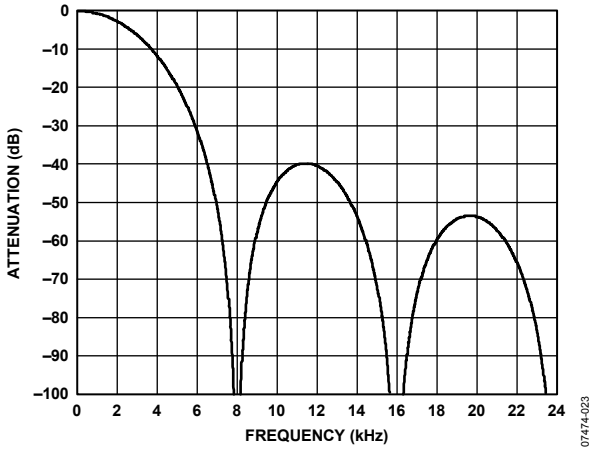

Figure 24. Typical Digital Filter Response at f_{ADC} = 8 kHz (ADCFLT = 0x0000)

A modified version of the 8 kHz filter response can be configured by setting the running average bit (ADCFLT[14]). As a result, an additional running-average-by-two filter is introduced on all ADC output samples, which further reduces the ADC output noise. In addition, 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 25.

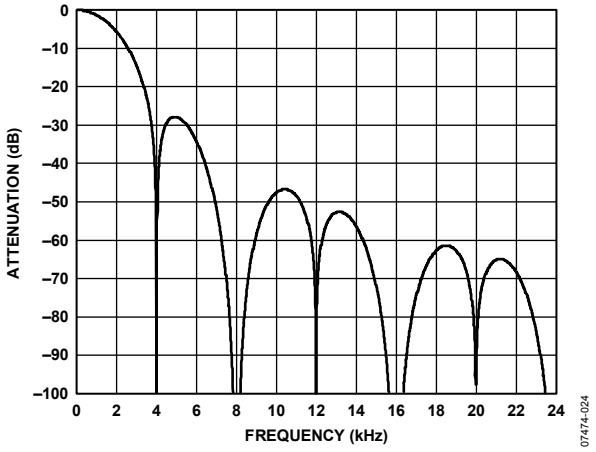

Figure 25. Typical Digital Filter Response at f_{ADC} = 8 kHz (ADCFLT = 0x4000)

At very low throughput rates, the chop enable 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, but there is a trade-off between filter bandwidth and ADC noise.

For example, with the chop enable bit (ADCFLT[15]) set to 1, the SF value (ADCFLT[6:0]) increases to 0x1F (31 decimal) and an AF value (ADCFLT[13:8]) of 0x16 (22 decimal) is selected, resulting in an ADC throughput of 10 Hz. The frequency response in this case is shown in Figure 26.

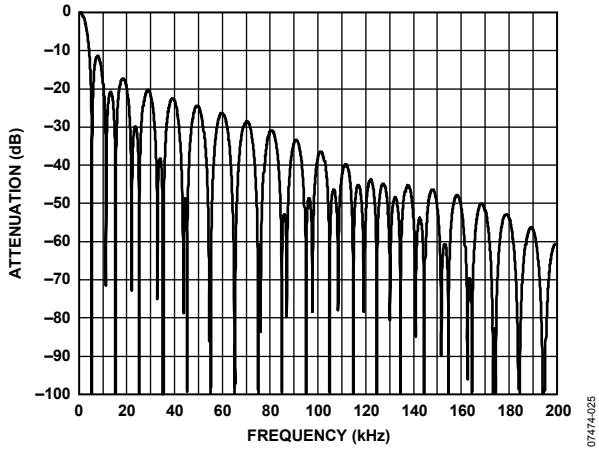

Figure 26. Typical Digital Filter Response at f_{ADC} = 10 Hz (ADCFLT = 0x961F)

Changing SF to 0x1D and setting AF to 0x3F with the chop enable bit still enabled configures the ADC with its minimum throughput rate of 4 Hz in normal mode. The digital filter frequency response with this configuration is shown in Figure 27.

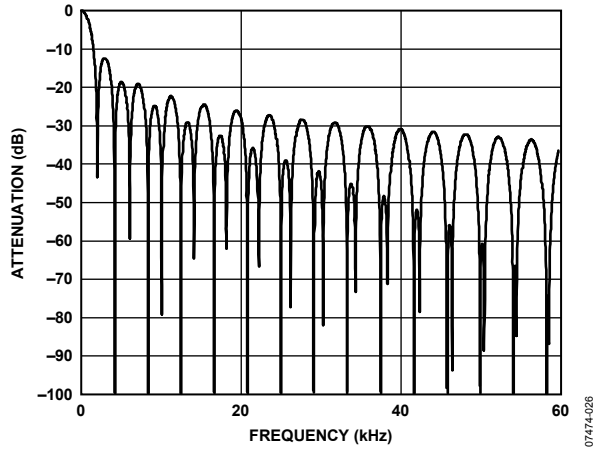

Figure 27. Typical Digital Filter Response at f_{ADC} = 4 Hz (ADCFLT = 0xBF1D)

In ADC low power mode, the Σ-Δ modulator clock of the ADC is no longer driven at 512 kHz, but is driven directly from the on-chip, low power, 131 kHz oscillator. Subsequently, if normal mode is used for the same ADCFLT configuration, all filter values should be scaled by a factor of approximately 4. Therefore, 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 28.

The offset coefficient is read from the ADC0OF calibration register and is a 16-bit, twos complement number. The range of this number, in terms of the signal chain, is effectively ±1. Therefore, 1 LSB of the ADC0OF register is not the same as 1 LSB of the ADC0DAT register.

A positive value of ADC0OF indicates that when offset is subtracted from the output of the filter, a negative value is added. The nominal value of this register is 0x0000, indicating zero offset is to be removed. The actual offset of the ADC can vary slightly from part to part and at different PGA gains. The offset within the ADC is minimized if the chopping mode is enabled (that is, ADCFLT[15] = 1).

The gain coefficient is read from the ADC0GN register and is a unitless scaling factor. The 16-bit value in this register is divided by 16,384 and then multiplied by the offset-corrected value. The nominal value of this register equals 0x5555, corresponding to a multiplication factor of 1.3333, and scales the nominal ±0.75 signal to produce a full-scale output signal of ±1. The resulting output signal is checked for overflow/underflow and converted to twos complement or unipolar mode before being output to the data register.

The actual gain and the required scaling coefficient for zero gain error vary slightly from part to part at different PGA settings in normal and low power modes. The value downloaded into ADC0GN during a power-on reset represents the scaling factor for a PGA gain of 1. If a different PGA setting is used, however, some gain error may be present. To correct this error, overwrite the calibration coefficients via user code or perform an ADC calibration.

The simplified ADC transfer function can be described as

$$ADC_{OUT} = \left[\frac{V_{IN} \times PGA}{V_{REF}} - ADCxOF\right] \times \frac{ADCxGN}{ADCxGN_{NOM}}$$

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

For the current channel ADC,

$$ADC_{OUT} = \left[\frac{V_{IN} \times PGA}{V_{REF}} - K \times ADCxOF\right] \times \frac{ADCxGN}{ADCxGN_{NOM}}$$

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

Normal Mode

In normal mode, K = 1 for PGA gains of 1, 4, 8, 16, 32, and 64; K = 2 for PGA gains of 2 and 128; K = 4 for a PGA gain of 256; and K = 8 for a PGA gain of 512.

Low Power Mode

In low power mode, K = 32 for a PGA gain of 128. In addition, if the REG_AVDD/2 reference is used, the K factor doubles.

Low Power Plus Mode

In low power plus mode, K = 8 for a PGA gain of 512. In addition, if the REG_AVDD/2 reference is used, the K factor doubles.

ADC DIAGNOSTICS

The ADuC7036 features a diagnostic capability and open-circuit detection on both ADCs.

Current ADC Diagnostics

The ADuC7036 features the capability to detect open-circuit conditions on the current channel inputs. This is accomplished using the two current sources on IIN+ and IIN−, which are controlled via ADC0CON[14:13].

Note that the IIN+ and IIN− current sources have a tolerance of ±30%. Therefore, a PGA gain ≥ 2 (ADC0CON[3:0] ≥ 0001) must be used when current sources are enabled.

Temperature ADC Diagnostics

The ADuC7036 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, which are controlled via ADC1CON[14:13].

Voltage ADC Diagnostics

The ADuC7036 features the capability to detect open-circuit conditions on the voltage channel input. This is accomplished using the current source on the voltage attenuator, controlled by the high voltage register HVCFG1[7].

POWER SUPPLY SUPPORT CIRCUITS

The ADuC7036 incorporates two on-chip low dropout (LDO) regulators that are driven directly from the battery voltage to generate a 2.6 V internal supply. This 2.6 V supply is then used as the supply voltage for the ARM7 MCU and the peripherals, including the on-chip precision analog circuits.

The digital LDO functions with two output capacitors (2.2 μF and 0.1 μF) in parallel on REG_DVDD, whereas the analog LDO functions with an output capacitor (0.47 μF) on REG_AVDD.

The ESR of the output capacitor affects stability of the LDO control loop. An ESR of 5 Ω or less for frequencies greater than 32 kHz is recommended to ensure the stability of the regulators.

In addition, the power-on reset (POR), power supply monitor (PSM), and low voltage flag (LVF) functions are integrated to ensure safe operation of the MCU, as well as continuous monitoring of the battery power supply.

The POR circuit is designed to operate with a VDD (0 V to 12 V) power-on time of greater than 100 μs. It is, therefore, recommended that the external power supply decoupling components be carefully selected to ensure that the VDD supply power-on time can always be guaranteed to be greater than 100 μs, regardless of the VBAT power-on conditions. The series resistor and decoupling capacitor combination on VDD should be chosen to result in an RC time constant of at least 100 μs (for example, 10 Ω and 10 μF, as shown on Figure 60).

As shown in Figure 29, when the supply voltage on VDD reaches a typical operating voltage of 3 V, a POR signal keeps the ARM core in reset state for 20 ms. This ensures that the regulated power supply voltage (REG_DVDD) applied to the ARM core and associated peripherals is greater than the minimum operational voltage, thereby guaranteeing full functionality. A POR flag is set in the RSTSTA MMR to indicate a POR event has occurred.

The ADuC7036 also features a PSM function. When enabled through HVCFG0[3], the PSM continuously monitors the voltage at the VDD pin. If this voltage drops below 6 V typical, the PSM flag is automatically asserted and can generate a system interrupt if the high voltage IRQ is enabled via IRQEN[16] or FIQEN[16]. An example of this operation is shown in Figure 29.

At voltages below the POR level, an additional low voltage flag can be enabled (HVCFG0[2]). This flag can be used to indicate that the contents of the SRAM remain valid after a reset event. The operation of the low voltage flag is shown in Figure 29. When HVCFG0[2] is enabled, the status of this bit can be monitored via HVMON[3]. If the HVCFG0[2] bit is set, the SRAM contents are valid. If this bit is cleared, the SRAM contents may become corrupted.

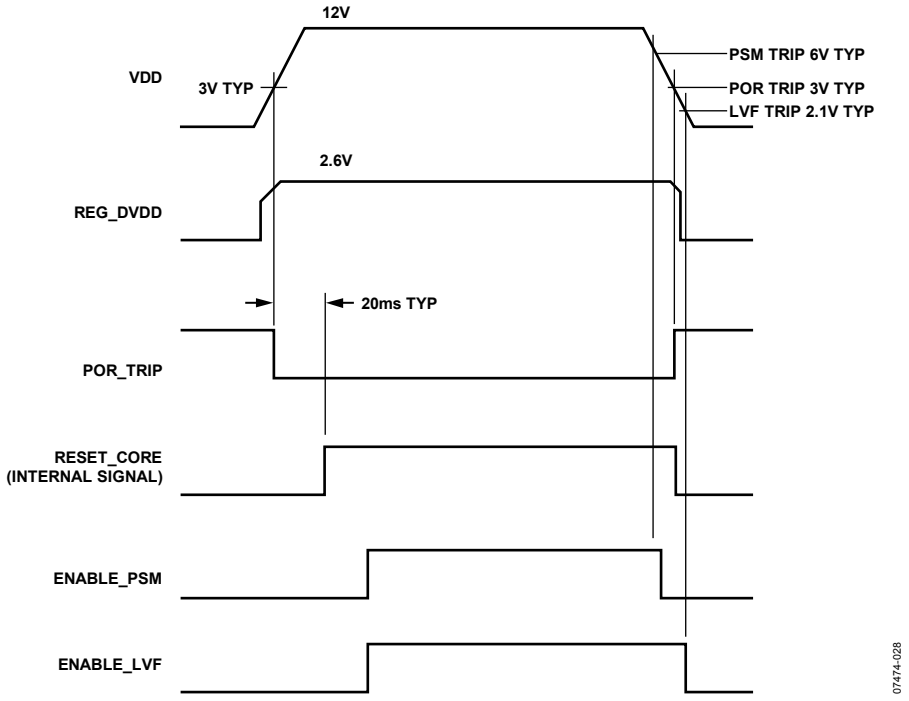

Figure 29. Typical Power-On Cycle

PLLCON Prewrite Key

Name: PLLKEY0

Address: 0xFFFF0410

Access: Write only

Key: 0x000000AA

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

PLLCON Postwrite Key

Name: PLLKEY1

Address: 0xFFFF0418

Access: Write only

Key: 0x00000055

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

PLLCON Register

Name: PLLCON

Address: 0xFFFF0414

Default Value: 0x00

Access: Read/write

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

Table 45. PLLCON MMR Bit Designations

Bit	Description
7 to 2	Reserved. These bits should be written as 0 by user code.
1 to 0	PLL clock source.[1] 00 = lower power, 131 kHz oscillator. 01 = precision 131 kHz oscillator. 10 = external 32.768 kHz crystal. 11 = reserved.

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

POWCON Prewrite Key

Name: POWKEY0

Address: 0xFFFF0404

Access: Write only

Key: 0x00000001

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

POWCON Postwrite Key

Name: POWKEY1

Address: 0xFFFF040C

Access: Write only

Key: 0x000000F4

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

GPIO Port1 Control Register

Name: GP1CON

Address: 0xFFFF0D04

Default Value: 0x10000000

Access: Read/write

Function: This 32-bit MMR selects the pin function for each Port1 pin.

Table 60. GP1CON MMR Bit Designations

Bit	Description
31 to 5	Reserved. These bits are reserved and should be written as 0 by user code.
4	GPIO_6 function select bit. Set to 1 by user code to configure the GPIO_6 pin as TxD, transmit data for UART serial port. Cleared by user code to 0 to configure the GPIO_6 pin as a general-purpose I/O (GPIO) pin.
3 to 1	Reserved. These bits are reserved and should be written as 0 by user code.
0	GPIO_5 function select bit. Set by user code to 1 to configure the GPIO_5 pin as RxD. Receive data for UART serial port. Cleared by user code to 0 to configure the GPIO_5 pin as a general-purpose I/O (GPIO) pin.

GPIO Port0 Data Register

Name: GP0DAT

Address: 0xFFFF0D20

Default Value: 0x000000XX

Access: Read/write

Function: This 32-bit MMR configures the direction of the GPIO pins assigned to Port0 (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 62. GP0DAT MMR Bit Designations

Bit	Description
31 to 29	Reserved. These bits are reserved and should be written as 0 by user code.
28	Port 0.4 direction select bit. Set to 1 by user code to configure the GPIO pin assigned to Port 0.4 as an output. Cleared to 0 by user code to configure the GPIO pin assigned to Port 0.4 as an input.
27	Port 0.3 direction select bit. Set to 1 by user code to configure the GPIO pin assigned to Port 0.3 as an output. Cleared to 0 by user code to configure the GPIO pin assigned to Port 0.3 as an input.
26	Port 0.2 direction select bit. Set to 1 by user code to configure the GPIO pin assigned to Port 0.2 as an output. Cleared to 0 by user code to configure the GPIO pin assigned to Port 0.2 as an input.
25	Port 0.1 direction select bit. Set to 1 by user code to configure the GPIO pin assigned to Port 0.1 as an output. Cleared to 0 by user code to configure the GPIO pin assigned to Port 0.1 as an input.
24	Port 0.0 direction select bit. Set to 1 by user code to configure the GPIO pin assigned to Port 0.0 as an output. Cleared to 0 by user code to configure the GPIO pin assigned to Port 0.0 as an input.
23 to 21	Reserved. These bits are reserved and should be written as 0 by user code.
20	Port 0.4 data output. The value written to this bit appears directly on the GPIO pin assigned to Port 0.4.
19	Port 0.3 data output. The value written to this bit appears directly on the GPIO pin assigned to Port 0.3.
18	Port 0.2 data output. The value written to this bit appears directly on the GPIO pin assigned to Port 0.2.
17	Port 0.1 data output. The value written to this bit appears directly on the GPIO pin assigned to Port 0.1.
16	Port 0.0 data output. The value written to this bit appears directly on the GPIO pin assigned to Port 0.0.
15 to 5	Reserved. These bits are reserved and should be written as 0 by user code.
4	Port 0.4 data input. This bit is a read only bit that reflects the current status of the GPIO pin assigned to Port 0.4. User code should write 0 to this bit.
3	Port 0.3 data input. This bit is a read only bit that reflects the current status of the GPIO pin assigned to Port 0.3. User code should write 0 to this bit.
2	Port 0.2 data input. This bit is a read only bit that reflects the current status of the GPIO pin assigned to Port 0.2. User code should write 0 to this bit.
1	Port 0.1 data input. This bit is a read only bit that reflects the current status of the GPIO pin assigned to Port 0.1. User code should write 0 to this bit.
0	Port 0.0 data input. This bit is a read only bit that reflects the current status of the GPIO pin assigned to Port 0.0. User code should write 0 to this bit.

WAKE-UP (WU) PIN

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

Wake-Up (WU) Pin Circuit Description

By default, the WU pin is configured as an output with an internal 10 kΩ pull-down resistor and high-side FET driver. In its default mode of operation, the WU pin 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 a 10 µs latency has elapsed; this latency is inherent in a serial communication between the HVCON or HVDAT MMR and the high voltage interface (see the High Voltage Peripheral Control Interface section).

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

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

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, and HVMON[0] is set.

A thermal shutdown event disables the WU driver. The WU driver must be reenabled manually after using HVCFG1[3] after a thermal event.

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 and indicates if the external wake-up bus (including R_{LOAD} = 1 kΩ, C_{LOAD} = 91 nF, and R_{LIMIT} = 39 Ω) is above or below a typical voltage of 3 V.

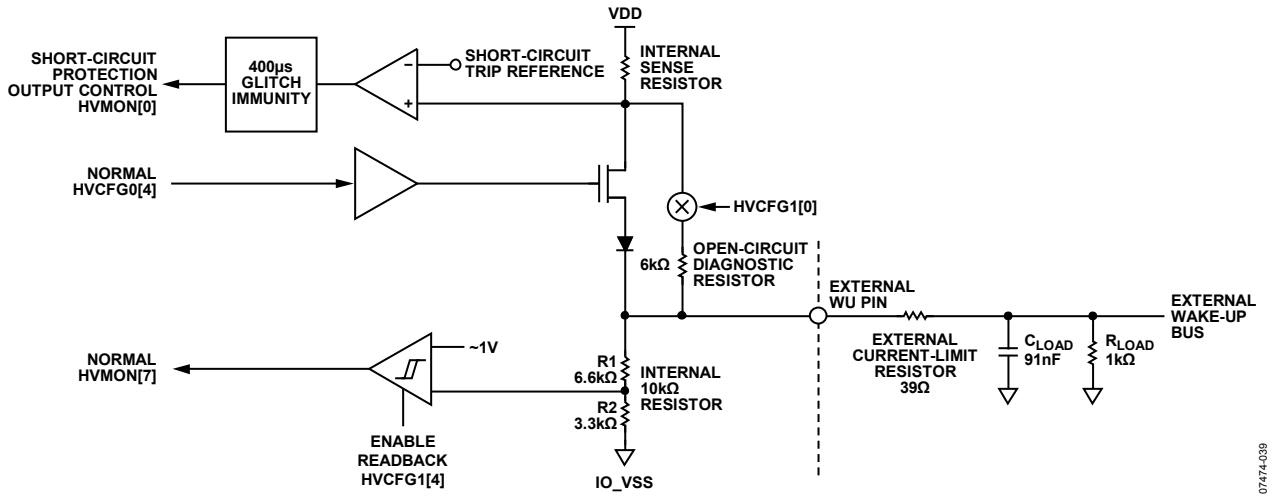

Figure 42. WU Circuit, Block Diagram

LIN Diagnostics

The ADuC7036 features the capability to nonintrusively monitor the current state of the LIN/BSD pin. This readback functionality is implemented using GPIO_11. The current state of the LIN/BSD pin is contained in GP2DAT[4].

It is also possible to drive the LIN/BSD 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/BSD pin can be pulled high or low using GP2DAT.

The ADuC7036 also features short-circuit protection on the LIN/BSD pin. If a short-circuit condition is detected on the LIN/BSD pin, HVSTA[2] is set. This bit is cleared by reenabling 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, when HVSTA[3] is set, LIN communications continue uninterrupted.

BSD RELATED MMRS

The ADuC7036 emulates the BSD communication protocol using a software (bit bang) interface with some hardware assistance form LIN hardware synchronization logic. In effect, the ADuC7036 BSD interface uses the following protocols:

- An internal GPIO signal (GPIO_12) that is routed to the external LIN/BSD pin and is controlled directly by software to generate 0s and 1s.
- When reading bits, the LIN synchronization hardware uses LHSVAL1 to count the width of the incoming pulses so that user code can interpret the bits as sync, 0, or 1.
- When writing bits, user code toggles a GPIO pin and uses the LHSCAP and LHSCMP registers to time pulse widths and generate an interrupt when the BSD output pulse width has reached its required width.

The ADuC7036 MMRs required for BSD communication are as follows:

- LHSSTA: LIN hardware synchronization status register
- LHSCON0: LIN hardware synchronization control register
- LHSVAL0: LIN hardware synchronization Timer0 (16-bit timer)
- LHSCON1: LIN hardware synchronization edge setup register
- LHSVAL1: LIN hardware synchronization break timer
- LHSCAP: LIN hardware synchronization capture register
- LHSCMP: LIN hardware synchronization compare register
- IRQEN/IRQCLR: enable interrupt register
- FIQEN/FIQCLR: enable fast interrupt register
- GP2DAT: GPIO Port 2 data register
- GP2SET: GPIO Port 2 set register
- GP2CLR: GPIO Port 2 clear register

Detailed bit definitions for most of these MMRs have been listed previously. In addition to the registers described in the LIN MMR Description section, LHSCAP and LHSCMP are registers that are required for the operation of the BSD interface. Details of these registers follow.

LIN Hardware Synchronization Capture Register

Name: LHSCAP

Address: 0xFFFF0794

Default Value: 0x0000

Access: Read only

Function: This 16-bit, read only register holds the last captured value of the internal LIN synchronization timer (LHSVAL0). In BSD mode, LHSVAL0 is clocked directly from an internal 5 MHz clock, and its value is loaded into the capture register on every falling edge of the BSD bus.

LIN Hardware Synchronization Compare Register

Name: LHSCMP

Address: 0xFFFF0798

Default Value: 0x0000

Access: Read/write

Function: This register is used to time BSD output pulse widths. When enabled through LHSCON0[5], a LIN interrupt is generated when the value in LHSCAP equals the value written in LHSCMP. This functionality allows user code to determine how long a BSD transmission bit (sync, 0, or 1) should be asserted on the bus.