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	Obsolete
Core Processor	ARM7®
Core Size	16/32-Bit
Speed	20.48MHz
Connectivity	LINbus, SPI, UART/USART
Peripherals	POR, 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 ~ 105°C
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/aduc7032bstz-8v-rl

Parameter	Test Conditions/Comments	Min	Typ	Max	Unit
LIN V2.0 Specification	Bus Load Conditions (C_{BUS} || R_{BUS}) : 1nF||1kΩ ; 6.8nF|| 660 Ω ; 10nF || 500Ω				
D1	Duty Cycle 1 $TH_{REC(MAX)}$ = 0.744 * V_{BAT} $TH_{DOM(MAX)}$ = 0.581 * V_{BAT} V_{SUP} = 7.0V…18V; t_{BIT} = 50µs D1 = $t_{BUS\_REC(MIN)}$ / (2 * t_{BIT})	0.396			
D2	Duty Cycle 2 $TH_{REC(MIN)}$ = 0..284 * V_{BAT} $TH_{DOM(MIN)}$ = 0.422 * V_{BAT} V_{SUP} = 7.0V…18V; t_{BIT} = 50µs D2 = $t_{BUS\_REC(MAX)}$ / (2 * t_{BIT})			0.581	
Wake	R_L = 1kOhm, C_{BUS} = 91nF, R_{LIMIT} = 39Ohms				
VDD[1]	Supply Voltage Range for which the Wake Pin is functional	7		18	V
V_{OH}[32]	Output High Level	5			V
V_{OL}[32]	Output Low Level			2	V
V_{IH}	Input High Level	4.6			V
V_{IL}	Input Low Level			1.2	V
Monoflop Timeout	Timeout Period		1.3		sec
Package Thermal Specifications					
Thermal Shutdown[33]		140	150	160	°C
Thermal Impedance (θ_{ja})[34]	48 LQFP, Stacked Die				
	Top Die		50		°C/W
	Bottom Die		25		°C/W
POWER REQUIREMENTS					
Power Supply Voltages					
V_{DD} (Battery Supply)		3.5		18	V
REG_DV_{DD}, REG_AV_{DD}[35]		2.5	2.6	2.7	V
Power Consumption					
I_{DD} – MCU Normal Mode[36]	MCU Clock Rate = 10.24MHz, ADC Off		10	20	mA
I_{DD} – MCU Normal Mode[36]	MCU Clock Rate = 20.48MHz, ADC Off		20		mA
I_{DD} – MCU Powered Down[1]	ADC Low Power Mode, measured over an ambient temperature range of -10°C to +40°C (Continuous ADC Conversion)		300	400	µA
I_{DD}–MCU Powered Down[1]	ADC Low Power Mode, measured over an ambient temperature range of -40°C to +85°C (Continuous ADC Conversion)		300	500	µA

Table 5 : SPI Slave Mode Timing (PHASE Mode = 0)

Parameter	Description	Min	Typ	Max	Unit
t_{CS}	CS to SCLOCK edge				ns
t_{SL}	SCLOCK low pulsewidth		(SPIDIV + 1) × t_{HCLK}		ns
t_{SH}	SCLOCK high pulsewidth		(SPIDIV + 1) × t_{HCLK}		ns
t_{DAV}	Data output valid after SCLOCK edge				ns
t_{DSU}	Data input setup time before SCLOCK edge				ns
t_{DHD}	Data input hold time after SCLOCK edge				ns
t_{DF}	Data output fall time				ns
t_{DR}	Data output rise time				ns
t_{SR}	SCLOCK rise time				ns
t_{SF}	SCLOCK fall time				ns
t_{DOCS}	Data output valid after CS edge				ns
t_{SFS}	CS high after SCLOCK edge				ns

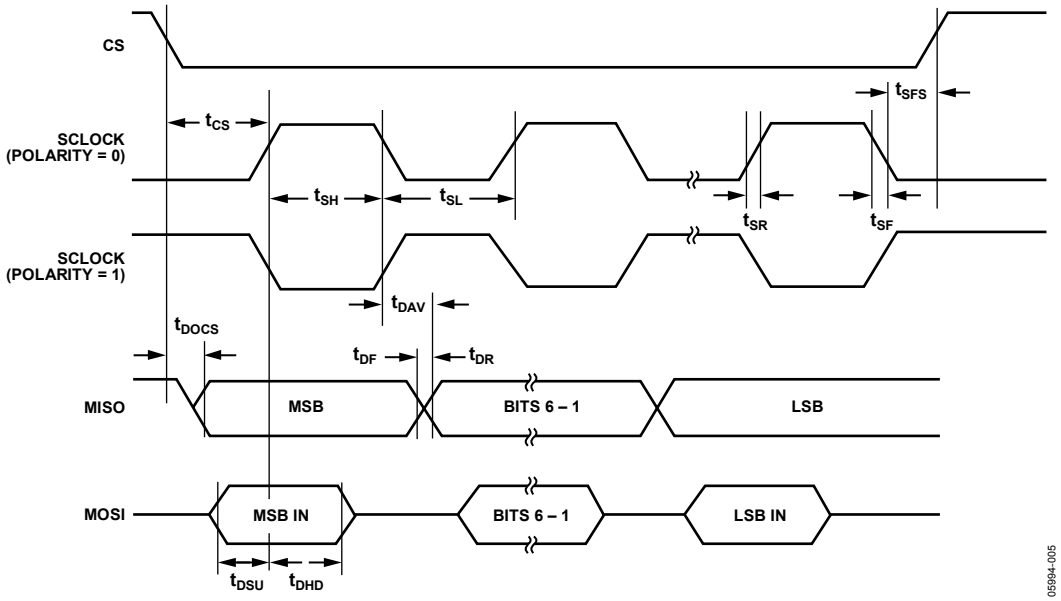

Figure 5 : SPI Slave Mode Timing (PHASE Mode = 0)

PIN FUNCTION DESCRIPTIONS

Table 7: Pin Function Descriptions

Pin#	Mnemonic	Type*	Function
1	$\overline{\text{RESET}}$	I	Reset Input Pin, Active Low. This pin has an internal, weak pull-up resistor to REG_DVDD. If this pin is not being used it can be left not connected. For added security and robustness, it is recommended that this pin be strapped via a resistor to REG_DVDD.
2	GPIO_5/IRQ1/RxD	I/O	General Purpose Digital I/O 5 is a Multi-Function Pin. By default and after Power-On-Reset, this pin is configured as an input. The pin has an internal weak pull-up resistor and if not being used it can be left unconnected. This multi-function pin can be configured in one of 3 states, namely: General Purpose Digital I/O 5 External Interrupt Request 1, Active High Receive Data for UART Serial Port This Pin may also be used as a clock input to Timer1.
3	GPIO_6/TxD	I/O	General Purpose Digital I/O 6 is a Multi-Function Pin. By default and after Power-On-Reset, this pin is configured as an input. The pin has an internal weak pull-up resistor and if not being used it can be left unconnected. This multi-function pin can be configured in one of 2 states, namely: General Purpose Digital I/O 6 Transmit Data for UART Serial Port
4	GPIO_7/IRQ4	I/O	General Purpose Digital I/O 7 is a multi-function pin. By default and after Power-On-Reset, this pin is configured as an input. The pin has an internal weak pull-up resistor and if not being used it can be left unconnected. This multi-function pin can be configured in one of 2 states, namely: General Purpose Digital I/O 7 External Interrupt Request 4, Active High
5	GPIO_8/ IRQ5	I/O	General Purpose Digital I/O 8 is a multi-function pin. By default and after Power-On-Reset, this pin is configured as an input. The pin has an internal weak pull-up resistor and if not being used it can be left unconnected. This multi-function pin can be configured in one of 2 states, namely: General Purpose Digital I/O 8 External Interrupt Request 5, Active High This Pin may also be used as a clock input to Timer1.
6	TCK	I	JTAG Test Clock. This clock input pin is one of the standard 5 pin JTAG debug port on the part. TCK is an input pin only and has an internal weak pull-up resistor. If not being used this pin can be left unconnected
7	TDI	I	JTAG Test Data Input. This data input pin is one of the standard 5 pin JTAG debug port on the part. TDI is an input pin only and has an internal weak pull-up resistor. If not being used this pin can be left unconnected
8	DGND	S	Ground Reference for On-Chip Digital Circuits
9	NC		No Connect, this pin is not connected internally but is reserved for possible future use, this pin should therefore not be connected externally
10	TDO	O	JTAG Test Data Output. This data output pin is one of the standard 5 pin JTAG debug port on the part. TDO is an output pin only. On power-on this output is disabled and pulled high via an internal weak pull-up resistor. If not being used this pin can be left unconnected
11	NTRST	I	JTAG Test Reset. This Reset input pin is one of the standard 5 pin JTAG debug port on the part. NTRST is an input pin only and has an internal weak pull-down resistor. If not being used this pin can be left unconnected. NTRST is also monitored by the on-chip kernel to enable LIN boot-load mode.
12	TMS	I	JTAG Test Mode Select. This Mode Select input pin is one of the standard 5 pin JTAG debug port on the part. TMS is an input pin only and has an internal weak pull-up resistor. If not being used this pin can be left unconnected

FLASH/EE MEMORY AND THE ADUC7032

The ADuC7032 incorporates Flash/EE memory technology on-chip to provide the user with non-volatile, in-circuit reprogrammable memory space.

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

Overall, Flash/EE memory represents a step closer to the ideal memory device that includes non-volatility, in-circuit programmability, high density, and low cost. Incorporated in the ADuC7032 Flash/EE memory technology allows the user to update program code space in-circuit, without the need to replace one time programmable (OTP) devices at remote operating nodes.

Flash/EE Memory

The total 96kBytes of Flash/EE memory are organized as 48k X 16 bits. 94kBytes are user space and 2kBytes are reserved for boot loader/kernel space. The page size of this Flash/EE memory is 512Bytes. Typically, it takes the Flash/EE memory controller 20msec to erase a page, and 50µsec to write a 16-Bit word. These Flash/EE memory timings are independent of MCU core clock.

94kBytes of Flash/EE memory are available to the user as code and non-volatile data memory. There is no distinction between data and program, as ARM code shares the same space. The real width of the Flash/EE memory is 16 bits, which means that in ARM mode (32-bit instruction), two accesses to the Flash/EE memory are necessary for each instruction fetch. When operating at speeds less than 20.48MHz the Flash/EE memory controller can transparently fetch the second 16-bit half word (part of the 32-bit ARM op-code) within a single core clock period. It is therefore recommend that for speeds less than 20.48MHz, i.e. CD > 0, that ARM mode is used. For 20.48MHz operation, i.e. CD = 0 , it is recommended to operate in THUMB mode.

The Flash/EE memory is physically located at 0x80000. Upon a hard reset it is logically mapped to 0x00000000. The factory default contents of all Flash/EE memory locations is 0xFF. Flash/EE memory may be read in 8/16/32 bit segments, and written in segments of 16 bits. The Flash/EE memory is rated for 10K endurance cycles. This rating is based on the number of times that each individual half word (16 bit location) is cycled i.e. erased and programmed. A redundancy scheme may be implemented in software to ensure greater than 10K cycles endurance.

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

It is possible to write to a single 16 bit location only twice between erases, i.e. It is possible to walk bytes, not bits. If a location is written to more than twice, then it is possible that the contents of the Flash/EE page may be corrupted.

The 94kBytes of Flash/EE memory can be programmed in-circuit, using a serial download mode via the LIN interface or the integrated JTAG port.

(1) Serial Downloading (In-Circuit Programming)

The ADuC7032 facilitates code download via the LIN pin.

(2) JTAG access

The ADuC7032 features an on-chip JTAG Debug Port to facilitate code download and debug.

FLASH/EE MEMORY CONTROL INTERFACE

The access to and control of the Flash/EE memory on the ADuC7032 is managed by an on-chip memory controller. The controller manages the Flash/EE memory as two separate blocks (0 and 1).

Block 0 consists of the 32KB Flash/EE memory mapped from 0x0009 0000 to 0x0009 7FFF (including the 2KB kernel space which is reserved at the top of this block).

Block 1 consists of the 64KB Flash/EE memory mapped from from 0x0008 0000 to 0x0008 FFFF.

It should be noted that MCU core can continue to execute code from one memory block while an active erase or program cycle is being carried out on the other block. If a command operates on the same block as the code currently executing, the core is halted until the command is completed, this also applies to code execution.

User Code, LIN and JTAG programming use the Flash/EE memory Control Interface, which consists of the following MMRs :

- **FEExSTA (x= 0 or 1):** read only register, reflects the status of the Flash Control Interface
- **FEExMOD (x= 0 or 1):** sets the operating mode of the Flash Control Interface
- **FEExCON (x= 0 or 1):** 8-bit command register. The commands are interpreted as described in Table 11.
- **FEExDAT (x= 0 or 1):** 16-bit data register.
- **FEExADR (x= 0 or 1):** 16-bit address register.
- **FEExSIGN (x= 0 or 1):** Holds the 24-bit code signature as a result of the signature command being initiated.
- **FEExHIDE (x= 0 or 1):** Protection MMR. Controls read and write protection of the Flash memory code space. If previously configured via the FEEPRO register, FEEHIDE may require a software key to enable access.

- **FEExPRO (x= 0 or 1):** A buffer of the FEEHIDE register, which is used to store the FEEHIDE value, so it is automatically downloaded to the FEEHIDE registers on subsequent reset and power-on events.

NOTE: User Software must ensure that the Flash/EE memory controller has completed any Erase or Write cycle ***before*** the PLL is powered down. If the PLL is powered down before an Erase or Write cycle is completed, the Flash/EE page or byte may be corrupted.

The following sections describe in detail the bit designations of each of Flash/EE control MMRs

FEE0CON and FEE1CON Registers :

Name : FEE0CON and FEE1CON
Address : 0xFFFF0E08 and 0xFFFF0E88
Default Value (both registers) : 0x07
Access : Read/Write Access

Function : These 8-bit registers are written by user code to control the operating modes of the Flash/EE memory controllers for Block0 (32KB) and Block1 (64KB).

Table 11: Command Codes in FEE0CON and FEE1CON

Code	Command	Description (note x is 0 or 1 to designate Flash/EE Block 0 or 1)
0x00*	Reserved	*Reserved, this command should not be written by user code*
0x01*	Single Read	Load FEExDAT with the 16-bit data indexed by FEExADR
0x02*	Single Write	Write FEExDAT at the address pointed by FEExADR. This operation takes 50µs.
0x03*	Erase-Write	Erase the page indexed by FEExADR and write FEExDAT at the location pointed by FEExADR. This operation takes 20ms
0x04*	Single Verify	Compare the contents of the location pointed by FEExADR to the data in FEExDAT. The result of the comparison is returned in FEExSTA bit 1
0x05*	Single Erase	Erase the page indexed by FEExADR
0x06*	Mass erase	Erase Block0(30kByte) or Block1(64kByte) of user space. The 2kByte Kernel is protected. This operation takes 1.2s To prevent accidental execution, a command sequence is required to execute this instruction, this is described below.
0x07		Default command.
0x08	Reserved	*Reserved, this command should not be written by user code*
0x09	Reserved	*Reserved, this command should not be written by user code*
0x0A	Reserved	*Reserved, this command should not be written by user code*
0x0B	Signature	FEE0CON: This command will result in a 24-bit LFSR based signature been generated and loaded into FEE0SIG. If FEE0ADR is less than 0x97800, this command will result in a 24 bit LFSR based signature of the user code space from the page specified in FEE0ADR upwards, including the Kernel, security bits and Flash/EE key. If FEE0ADR is greater than 0x97800, the Kernel and manufacturing data is signed FEE1CON: This command will result in a 24-bit LFSR based signature been generated, beginning at FEE1ADR and ending at the end of the 63.5k Block, and loaded into FEE1SIG. The last page of this block is not included in the Sign generation.
0x0C	Protect	This command can be run only once. The value of FEExPRO is saved and can be removed only with a mass erase (0x06) or with the key
0x0D	*Reserved*	*Reserved, this command should not be written by user code*
0x0E	*Reserved*	*Reserved, this command should not be written by user code*
0x0F	Ping	No operation, interrupt generated

* The FEExCON will always read 0x07 immediately after execution of any of these commands.

FEE0MOD and FEE1MOD Registers :

Name : FEE0MOD and FEE1MOD
Address : 0xFFFF0E04 and 0xFFFF0E84
Default Value (both registers) : 0x00
Access : Read/Write

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

Table 13: FEE0MOD and FEE1MOD MMR bit designations

Bit	Description (note: x is 0 or 1 to designate Flash/EE Block 0 or 1)
15-7	Not Used
	These bits are reserved for future functionality and should be written as 0 by user code
6, 5	*Flash/EE Security Lock Bits* *These bits must be written as [6,5] = 1,0 to complete the Flash security protect sequence*
4	Flash/EE Controller Command Complete Interrupt Enable This bit is set to 1 by user code to enable the Flash/EE controller to generate an interrupt upon completion of a Flash/EE command. This bit is cleared to disable the generation of a Flash/EE interrupt upon completion of a Flash/EE command.
3	Flash/EE Erase/Write Enable *Set* by user code to enable the Flash/EE erase and write access via FEExCON *Cleared* by user code to disable the Flash/EE erase and write access via FEExCON
2	*Reserved and should be written as zero*
1	Flash/EE Controller Abort Enable This bit is set to 1 by user code to enable the Flash/EE controller abort functionality.
0	*Reserved and should be written as zero*

FLASH/EE MEMORY SECURITY

The 94kByte 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 30kBytes. Bits 0-28 of this register protect pages 0-57 from writing. Each bit protects 2 pages, i.e. 1kBytes. Bits 29-30 protect pages 58 and 59 respectively, i.e. each bit write protects a single page of 512 bytes. The MSB of this register (Bit31) 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 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 (64K), the FEE1HID MMR protects the 64kBytes. Bits 0-29 of this register protect pages 0-119 from writing. Each bit protects 4 pages, i.e. 2kBytes. Bit30 protect pages 120-127, i.e. bit 30 write protects eight pages of 512 bytes. The MSB of this register (Bit31) 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 is allows user code to lock the protection or security configuration of the Flash memory so that the protection configuration is automatically loaded on subsequent power-on or reset events.

0x0D 38	GP1CLR[3]	4	W	0x000000XX	95	GPIO Port 1 Data Clear MMR
0x0D 40	GP2DAT[3]	4	W	0x000000XX	92	GPIO Port 2 Data Control MMR
0x0D 44	GP2SET[3]	4	W	0x000000XX	94	GPIO Port 2 Data Set MMR
0x0D 48	GP2CLR[3]	4	W	0x000000XX	96	GPIO Port 2 Data Clear MMR
Flash/EE base address = 0xFFFF0E00						
0x0E00	FEE0STA	1	R	0x00	33	Flash/EE Status MMR
0x0E04	FEE0MOD	2	RW	0x00	34	Flash/EE Control MMR
0x0E08	FEE0CON	1	RW	0x07	32	Flash/EE Control MMR
0x0E0C	FEE0DAT	2	RW		33	Flash/EE Data MMR
0x0E10	FEE0ADR	2	RW		33	Flash/EE Address MMR
0x0E18	FEE0SIG	3	R	0xFFFFFF		Flash/EE LFSR MMR
0x0E1C	FEE0PRO	4	RW	0x00000000	35	Flash/EE Protection MMR
0x0E20	FEE0HID	4	RW	0xFFFFFFFF	35	Flash/EE Protection MMR
0x0E80	FEE1STA	1	R	0x00	33	Flash/EE Status MMR
0x0E84	FEE1MOD	2	RW	0x00	34	Flash/EE Control MMR
0x0E88	FEE1CON	1	RW	0x07	32	Flash/EE Control MMR
0x0E8C	FEE1DAT	2	RW		33	Flash/EE Data MMR
0x0E90	FEE1ADR	2	RW		33	Flash/EE Address MMR
0x0E98	FEE1SIG	3	R	0x0000		Flash/EE LFSR MMR
0x0E9C	FEE1PRO	4	RW	0x00000000	35	Flash/EE Protection MMR
0x0EA0	FEE1HID	4	RW	0xFFFFFFFF	35	Flash/EE Protection MMR

[1] Depends on the level on the external interrupt pins GP0, GP5, GP7 and GP8
[2] Updated by Kernel
[3] Depends on the level on the external GPIO pins

Table 28 : ADC Conversion Rates and Settling Times

Chop Enabled	Running Average	Averaging Factor	F_{ADC}	$*T_{Settling}$
No	No	No	$\frac{512000}{[SF+1]*64}$	$\frac{3}{F_{ADC}}$
No	No	Yes	$\frac{512000}{[SF+1]*64*[3+AF]}$	$\frac{1}{F_{ADC}}$
No	Yes	No	$\frac{512000}{[SF+1]*64}$	$\frac{4}{F_{ADC}}$
No	Yes	Yes	$\frac{512000}{[SF+1]*64*[3+AF]}$	$\frac{2}{F_{ADC}}$
Yes	N/A	N/A	$\frac{512000}{(SF+1) \times 64 \times (AF+3) + 3}$	$\frac{2}{F_{ADC}}$

**An additional time of 60us per enabled ADC is required before the first ADC result is available.*

Table 29 : Allowable Combinations of SF and AF

SF \ AF Range	0	1 to 7	8 to63
0-31	✓	✓	✓
32-63	✓	✓	×
64-127	✓	×	×

Current Channel ADC Data Register :

Name : ADC0DAT
Address : 0xFFFF0520
Default Value : 0x0000
Access : Read Only

Function : This ADC Data MMR holds the 16-bit conversion result from the I-ADC. The ADC will not update this MMR if the ADC0 Conversion Result READY bit (ADCSTA[0]) is set. A read of this MMR by the MCU clears all asserted READY flags (ADCSTA[2:0]).

Voltage Channel Data Register:

Name : ADC1DAT
Address : 0xFFFF0524
Default Value : 0x0000
Access : Read Only

Function : This ADC Data MMR holds the 16-bit conversion result from the V-ADC. The ADC will not update this MMR if the Voltage Conversion Result READY bit (ADCSTA[1]) is set. If I-ADC is not active, a read of this MMR by the MCU clears all asserted READY flags (ADCSTA[2:1]).

Temperature Channel ADC Data Register :

Name : ADC2DAT
Address : 0xFFFF0528
Default Value : 0x0000
Access : Read Only

Function : This ADC Data MMR holds the 16-bit conversion result from the T-ADC. The ADC will not update this MMR if the Temperature Conversion Result READY bit (ADCSTA[2]) is set.

ADC FIFO Register :

Name : ADCFIFO
Address : 0xFFFF052C
Default Value : 0x0000
Access : Read Only

Function : This 32-bit, read-only register returns the value of I-ADC and V-ADC conversion result held in the FIFO location currently pointed to by the FIFO read pointer. The low 16 bits [15-0] of this 32-bit word are the I-ADC result and the high 16-bits [31-16] are the V-ADC result. The FIFO function is enabled via the ADCCFG[1] bit and 3 flags available in the ADCSTA register allow user code monitor and read the FIFO contents.

Current Channel ADC Offset Calibration Register :

Name : ADC0OF
Address : 0xFFFF0530
Default Value : Part Specific, factory programmed
Access : Read/Write

Function : This ADC Offset MMR holds a 16-bit offset calibration coefficient for the I-ADC. The register is configured at power-on with a factory default value. However, this register will be automatically overwritten if an offset calibration of the I-ADC is initiated by the user via bits in the ADCMDE MMR. User code can only write to this calibration register if the ADC is in idle mode. An ADC must be enabled and in idle mode before written to any Offset or Gain Register. A delay of 23us should be included before ADCMDE is modified.

Voltage Channel Offset Calibration Register :

Name : ADC1OF
Address : 0xFFFF0534
Default Value : Part Specific, factory programmed
Access : Read/Write

Function : This Offset MMR holds a 16-bit offset calibration coefficient for the voltage channel. The register is configured at power-on with a factory default value. However, this register will be automatically overwritten if an offset calibration of the voltage channel is initiated by the user via bits in the ADCMDE MMR. User code can only write to this calibration register if the ADC is in idle mode. An ADC must be enabled and in idle mode before written to any Offset or Gain Register. A delay of 23us should be included before ADCMDE is modified.

Temperature Channel Offset Calibration Register:

Name : ADC2OF
Address : 0xFFFF0538
Default Value : Part Specific, factory programmed
Access : Read/Write

Function : This ADC Offset MMR holds a 16-bit offset calibration coefficient for the temperature channel. The register is configured at power-on with a factory default value. However, this register will be automatically overwritten if an offset calibration of the temperature channel is initiated by the user via bits in the ADCMDE MMR. User code can only write to this calibration register if the ADC is in idle mode. An ADC must be enabled and in idle mode before written to any Offset or Gain Register. A delay of 23us should be included before ADCMDE is modified.

ADC Calibration

As described in detail in the top level diagrams at the start of this section, the signal flow through all ADC Channels can be described in simple terms as:

- An Input-voltage is applied through an input buffer (and PGA in the case of the I-ADC) to the Sigma-Delta Modulator.
- The Modulator-output is applied to a programmable Digital Decimation Filter
- The filter output result is then averaged if chopping is used.
- An Offset value (ADCxOF) is subtracted from the result.
- This result is scaled by a Gain value (ADCxGN).
- Finally, the result is formatted as
  - 2's Complement. / Offset-Binary,
  - Rounded to 16-Bit
  - Clamped to +/-Full-Scale

Each ADC has a specific Offset and Gain correction or Calibration coefficient associated with it that are stored in MMR based Offset and Gain registers(ADCxOF and ADCxGN). The offset and gain registers can be used to remove offsets and gain errors arising within the part as well as System-level offset and gain errors external to the part.

These registers are configured at power-on with a factory programmed calibration value. These factory calibration values will vary from part to part reflecting the manufacturing variability of internal ADC circuits . However, these registers can also be overwritten by user code (only if the ADC is in idle mode) and will be automatically overwritten if an offset or gain calibration cycle is initiated by user via the mode bits in the ADCMDE[2:0] MMR. 2 types of automatic calibration are available to the user, namely :

- ***Self (Offset or Gain) Calibration***, where the ADC generates its calibration coefficient based on an internally generated 0V in the case of Self-Offset calibration and full-scale voltage in the case of Self-Gain calibration. It should be emphasized that ADC Self-Calibrations correct for offset and gain errors within the ADC. Self calibrations cannot compensate for other external errors in the system, e.g. Shunt-Resistor tolerance/drift, external offset voltages etc.

- ***System (Offset or Gain) Calibration***, where the ADC generates its calibration coefficient based on an externally generated zero-scale voltage in the case of System-Offset calibration and Full-scale voltage in the case of System-Gain calibration which are applied to the external ADC input for the duration of the calibration cycle.

The duration of an Offset calibration is 1 single conversion cycle ($3/F_{ADC}$ Chop off, $2/F_{ADC}$ Chop on) before returning the ADC to idle mode. A Gain calibration is a 2 stage process and subsequently takes twice as long as an offset calibration cycle. Once a calibration cycle is initiated, any ongoing ADC conversion is immediately halted, the calibration is carried out automatically at an ADC update rate programmed into ADCFLT and the ADC is always returned to idle after any calibration cycle. It is strongly recommended that ADC calibration is initiated at as low an ADC update rate as possible (high SF value in ADCFLT) in order to minimize the impact of ADC noise during calibration.

NOTE: ADC0OF and ADC0GN must first contain the values for PGA = 1 before a calibration scheme is started

Using the Offset and Gain Calibration Registers

If the Chop bit (ADCFLT[15]) is enabled, then internal ADC offset errors will be minimized and an Offset calibration may not be required. If chopping is disabled however, an initial Offset calibration will be required and may need to be repeated.

A Gain calibration, particularly in the context of the I-ADC (with internal PGA) may need to be carried out at all relevant system gain ranges depending on system accuracy requirements. If it is not possible to apply an external full-scale current on all gain ranges then it is possible to apply a lower current, and scale the result produced by the calibration. e.g Apply a 50% current and then divide the ADC0GN value produced by 2 and write this value back into ADC0GN. It should be noted that there is a lower limit to the input signal that can be applied for a System-Calibration because the ADC0GN register is only 16-Bit. The input span (difference between the System Zero-Scale value and System Full-Scale value) should be greater than 40% of the nominal Full-Scale-Input range, ie > 40% of Vref/Gain.

The on-chip Flash/EE memory can be used to store multiple calibration coefficients which can be copied by user code directly into the relevant calibration registers as appropriate based on system configuration. In general, the simplest way to use the calibration registers is to let the ADC calculate the values required as part of the ADC automatic calibration modes.

The operating mode, clocking mode and programmable clock divider are controlled via two MMRs, PLLCON and POWCON, and the status of the PLL is indicated by PLLSTA. PLLCON controls the operating mode of the clock system while POWCON controls the core clock frequency and the power-down mode. PLLSTA indicates the presence of an oscillator on the XTAL1 pin, the PLL Lock status, and the PLL Interrupt.

It is recommended that before the ADuC7032 is powered down, that the clock source for the PLL is switched to the Low Power 131kHz oscillator to reduce wake up time. The Low Power, Oscillator is always active.

When the ADuC7032 wakes up from power down, the MCU core will begin executing code once the PLL begins oscillating. This occurs before the PLL has locked to a frequency of 20.48MHz. To ensure the Flash memory controller is executing with a valid clock, the controller is driven with a PLL-Output/8 clock source while the PLL is locking. Once the PLL locks, the PLL's output is switched from the PLL-Output/8 to the locked PLL-Output.

If user code requires an accurate PLL output, user code must poll the Lock bit (PLLSTA[1]) after wake-up before resuming normal code execution.

The PLL will be locked and executing user code within 2ms, if the PLL is clocked from an active clock source, e.g. Low Power 131kHz oscillator after waking up.

PLLCON is a protected MMR with two 32 bit keys PLLKEY0, a pre write key, and PLLKEY1, a post write key.

- PLLKEY0 = 0x000000AA
- PLLKEY1 = 0x00000055

POWCON is a protected MMR with two 32 bit keys POWKEY0, a pre write key, and POWKEY1, a post write key.

- POWKEY0 = 0x00000001
- POWKEY1 = 0x000000F4

An example of writing to both MMRs is shown below:

```
POWKEY0 = 0x01      //POWCON KEY
POWCON  = 0x00      //Full Power-down
POWKEY1 = 0xF4      //POWCON KEY
iA1*iA2             //dummy cycle

PLLKEY0 =  0xAA     //PLLCON KEY
PLLCON  =  0x0      //Switch to Low
                    //Power Osc.
PLLKEY1 = 0x55      //PLLCON KEY
iA1*iA2             //dummy cycle
```

PLLSTA Register :

Name : PLLSTA
Address : 0xFFFF0400
Default Value : 0x02
Access : Read/Write

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

Table 35 : PLLSTA MMR Bit Description

Bit	Description
31-3	*Reserved and should be written as zeros*
2	XTAL Clock, Read Only This is a live representation of the current logic level on XTAL1. This allows the user to check to see if an external clock source is present. If present this bit will alternate high and low at a frequency of 32.768kHz.
1	PLL Lock Status Bit, Read Only *Set* when the PLL is locked and outputting 20.48MHz. *Clear* when the PLL is not locked and outputting a Fcore/8 clock source
0	PLL Interrupt: *Set* if the PLL Lock status bit signal goes low. *Cleared* by writing 1 to this bit

POWCON Register :

Name : POWCON
Address : 0xFFFF0408
Default Value : 0x079
Access : Read/Write
Function : This 8-bit register allows user code dynamically enter various Low Power modes and modify the CD divider which controls the speed of the ARM7TDMI Core.

Table 37 : POWCON MMR bit designations

Bit	Description		
31-8	Reserved		
7	Precision 131kHz Input Enable: Cleared by the user to Power down the Precision 131kHz Input Enable. Set by the user to enable the Precision 131kHz Input Enable. The Precision 131kHz oscillator must also be enabled via HVCFG0[6]. Setting this bit increases current consumption by approximately 50uA and should be disabled when not in use.		
6	XTAL Power Down: Cleared by the user to Power down the external crystal circuitry. Set by the user to enable the external crystal circuitry.		
5	PLL Power Down[1]: This bit is cleared to 0 to power down the PLL. The PLL can not be powered down if either the core or peripherals are enabled: Bits 3, 4 and 5 must be cleared simultaneously. Set by default, and set by hardware on a wake up event		
4	Peripherals[2,3,4]Power Down: Cleared to power down the peripherals. The peripherals cannot be powered down if the core is enabled: bits 3 and 4 must be cleared simultaneously. Set by default, or and by hardware on a wake up event		
3	Core Power Down:[5] Cleared to power down the ARM Core Set by default, and set by hardware on a wake up event		
2-0	CD Core clock divider bits:		
	000	20.48 MHz	48.83ns
	001	10.24 MHz	97.66ns
	010	5.12 MHz	195.31ns
	011	2.56 MHz	390.63ns
	100	1.28 MHz	781.25ns
	101	640 kHz	1.56µs
	110	320 kHz	3.125µs
	111	160 kHz	6.25µs

[1] Timer peripherals will be powered down if driven from the PLL Output clock. Timers driven from an active clock source will stay in normal power mode.
[2] The peripherals that are powered down by this bit are as follows:
SRAM, Flash/EE Memory and GPIO Interfaces
SPI and UART Serial Ports
[3] LIN can still respond to wake-up events even if this bit is cleared.
[4] Wake-Up Timer (Timer2) can still be active if driven from low power oscillator even if this bit is set.
[5] If user code powers down the MCU, a dummy MCU cycle should be included after the power-down command is written to POWCON.

TIMER0 – LIFE-TIME TIMER

Timer0 is a general purpose 48-bit count-up, or a 16-bit count up/down timer with a programmable prescalar. Timer0 may be clocked from either the Core clock, the Low Power 32.768kHz Oscillator, the Precision 32.768kHz Oscillator or an external 32.768kHz crystal, with a prescalar of 1,16, 256 or 32768. This gives a minimum resolution of 48.83ns when the core is operating at 20.48MHz, and with a prescalar of 1.

In 48-bit mode, Timer0 counts up from zero. The current counter value may be read from T0VAL0 and T0VAL1.

In 16-Bit mode,Timer0 may count up or count down. A 16-bit value may be written to T0LD which will be loaded into the counter. The current counter value may be read from T0VAL0. Timer0 has a capture register (T0CAP), which may be triggered by a selected IRQ's source initial assertion. Once triggered, the current timer value is copied to T0CAP, and the timer keeps running. This feature can be used to determine the assertion of an event with more accuracy than by servicing an interrupt alone.

Timer0 reloads the value from T0LD either when TIMER0 overflows, or immediately when T0CLRI is written.

Timer0 interface consists of six MMRS:

- **T0LD** is a 16-bit register which holds the 16 bit value that is loaded into the counter. Only available in 16-bit mode.
- **T0CAP** is a 16-bit register which holds the 16-bit value captured by an enabled IRQ event. Only available in 16-bit mode.
- **T0VAL0/T0VAL1** are 16-bit and 32-bit registers which hold the 16 least significant bits and 32 most significant bits respectively. T0VAL0 and T0VAL1 is read-only. In 16-bit mode 16-bit T0VAL0 is used. In 48-bit mode both 16-bit T0VAL0 and 32-bit T0VAL1 are used.
- **T0CLRI** is an 8-bit register. Writing any value to this register will clear the interrupt. Only available in 16-bit mode.
- **T0CON** is the configuration MMR described in Table 44.

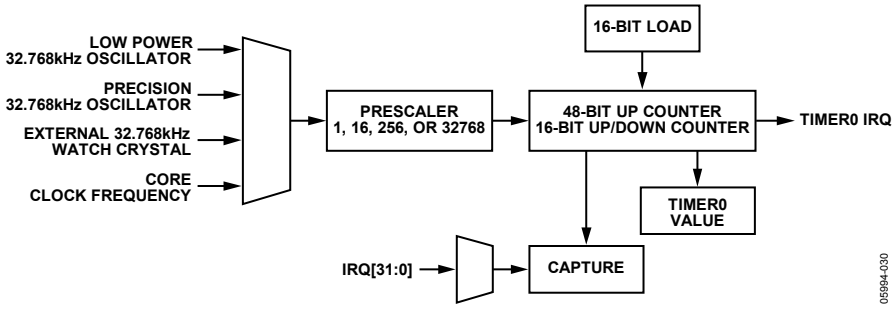

Figure 30 : Timer 0 block diagram

Timer0 Value Register :

Name : T0VAL0/T0VAL1
Address : 0xFFFF0304, 0xFFFF0308
Default Value : 0x00, 0x00
Access : Read Only
Function : T0VAL0 and T0VAL1 are 16-bit and 32-bit registers which hold the 16 least significant bits and 32 most significant bits respectively. T0VAL0 and T0VAL1 is read-only. In 16-bit mode 16-bit T0VAL0 is used. In 48-bit mode both 16-bit T0VAL0 and 32-bit T0VAL1 are used.

Timer0 Capture Register :

Name : T0CAP
Address : 0xFFFF0314
Default Value : 0x00
Access : Read Only
Function : This is a 16-bit register which holds the 16-bit value captured by an enabled IRQ event. Only available in 16-bit mode.

GENERAL PURPOSE I/O

The ADuC7032 features 9 General Purpose bi-directional I/O pins (GPIO). In general, many of the GPIO pins have multiple functions which can be configured by user code. By default, the GPIO pins are configured in GPIO mode. All GPIO pins have an internal pull up resistor and their sink capability is 0.8mA and they can source 0.1mA.

The 9 GPIO are grouped into three ports, Port0, Port1 and Port2. Port0 is 5 bits wide. Port1 and Port2 are both 2 bits wide. The GPIO assignment within each port is detailed in Table 48.

A typical GPIO structure is shown Figure 34.

External Interrupts are present on GP0, GP5, GP7 and GP8. This interrupts are level triggered and are active high. These interrupts are not latched, therefore the interrupts source must be present until either IRQSTA or FIQSTA are interrogated. The Interrupt source must be active for at least 1 CD divided core clock to guarantee recognition.

All port pins are configured and controlled by 4 sets (1 set for each port) of four port specific MMRs:

GPxCON: Port x Control Register
GPxDAT: Port x Configuration and Data Register
GPxSET: Data set port x
GPxCLR: Data clear port x

where x corresponds to the port number 0,1 or 2

During normal operation, user code can control the function and state of the external GPIO pins via these general purpose registers. All GPIO pins will retain their external (high or low) during power-down (POWCON) mode.

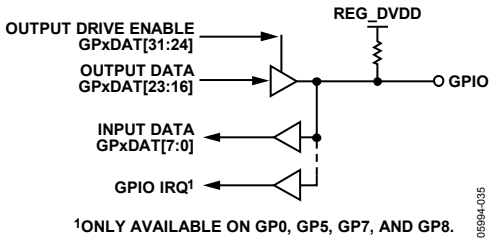

Figure 34 : ADuC7032 GPIO

GPIO Port0 Control Register :

Name : GP0CON
Address : 0xFFFF0D00
Default Value : 0x00000000
Access : Read/Write

Function : The 32-bit MMR selects the pin function for each Port0 pin.

Table 49 : GP0CON MMR Bit Designations

Bit	Description
31-29	Reserved These bits are reserved and should be written as 0 by user code
28	Reserved This bit is reserved and should be written as 1 by user code
27-25	Reserved These bits are reserved and should be written as 0 by user code
24	Internal P0.6 Enable Bit This bit must be set to 1 by user software to enable the High Voltage Serial Interface before using the HVCON and HVDAT registered high voltage interface
23-21	Reserved These bits are reserved and should be written as 0 by user code
20	Internal P0.5 Enable Bit This bit must be set to 1 by user software to enable the High Voltage Serial Interface before using the HVCON and HVDAT registered high voltage interface
19-17	Reserved These bits are reserved and should be written as 0 by user code
16	GPIO4 Function Select Bit This bit is cleared by user code to 0 to configure the GPIO4 pin as a General Purpose I/O (GPIO) pin This bit is set to 1 by user code to configure the GPIO4 pin as ECLK enabling a 2.56MHz clock output on this pin
15-13	Reserved These bits are reserved and should be written as 0 by user code
12	GPIO3 Function Select Bit This bit is cleared by user code to 0 to configure the GPIO3 pin as a General Purpose I/O (GPIO) pin This bit is set to 1 by user code to configure the GPIO2 pin as MOSI, Master Output, Slave Input Data for the SPI Port
11-9	Reserved These bits are reserved and should be written as 0 by user code
8	GPIO2 Function Select Bit This bit is cleared to 0 by user code to configure the GPIO2 pin as a General Purpose I/O (GPIO) pin This bit is set to 1 by user code to configure the GPIO3 pin as MISO, Master Input, Slave Output Data for the SPI Port
7-5	Reserved These bits are reserved and should be written as 0 by user code
4	GPIO1 Function Select Bit This bit is cleared to 0 by user code to configure the GPIO1 pin as a General Purpose I/O (GPIO) pin This bit is set to 1 by user code to configure the GPIO1 pin as SCLK, Serial Clock I/O for the SPI Port
3-1	Reserved These bits are reserved and should be written as 0 by user code
0	GPIO0 Function Select Bit This bit is cleared to 0 by user code to configure the GPIO0 pin as a General Purpose I/O (GPIO) pin This bit is set to 1 by user code to configure the GPIO0 pin as $\overline{SS}$, Slave Select I/O for the SPI Port

UART Control Register 1:

Name : COMCON1
Address : 0xFFFF0710
Default Value : 0x00
Access : Read/Write

Function : This 8-bit register controls the operation of the UART in conjunction with COMCON0

Table 72 : COMCON1 MMR Bit Descriptions

Bit	Name	Description
7-6		UART Input Mux 00 RxD driven by LIN Input Required for LIN Communications via LIN pin 01 *Reserved* 10 RxD driven by GP5 Required for Serial communications via GP5 pin (RxD) 11 *Reserved*
5		*Reserved*
4	LOOPBACK	Loop back Set by user to enable loop back mode. In loop back mode the TxD is forced high.
3-0		*Reserved*

UART Status Register 0:

Name : COMSTA0
Address : 0xFFFF0714
Default Value : 0x60
Access : Read Only

Function : This 8-bit read only register reflects the current status on the UART.

Table 73: COMSTA0 MMR Bit Descriptions

Bit	Name	Description
7		*Reserved*
6	TEMT	COMTX empty status bit Set automatically if COMTX is empty Cleared automatically when writing to COMTX
5	THRE	COMTX and COMRX empty Set automatically if COMTX and COMRX are empty Cleared automatically when one of the register receives data
4	BI	Break Indicator Set when SIN is held low for more than the maximum word length Cleared automatically
3	FE	Framing error Set when invalid stop bit Cleared automatically
2	PE	Parity error Set when a parity error occurs Cleared automatically
1	OE	Overrun error Set automatically if data are overwrite before been read Cleared automatically
0	DR	Data ready Set automatically when COMRX is full Cleared by reading COMRX

LIN Frame Break Symbol

As shown in Figure 40, the LIN "break" symbol is used to signal the start of a new frame. It lasts at least 13 bit periods and a slave must be able to detect a "break" symbol, even if it expects data or is in the process of receiving data. The ADuC7032 accomplishes this by using the LHSVAL1 Break Condition and Break Error detect functionality as described earlier. The "break" period does not have to be accurately measured, but if a bus fault condition (bus held low) occurs it must be flagged.

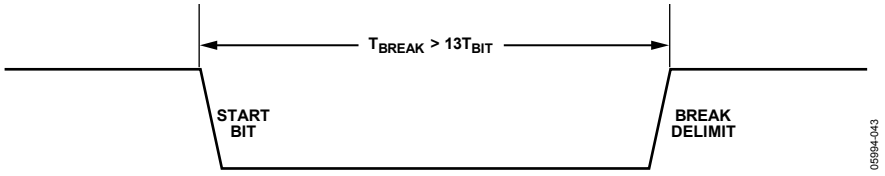

Figure 40 : LIN Break Field

LIN Frame Synchronization Byte

The baud rate of the communication via LIN is calculated from the SYNC Byte. This can be seen in Figure 41. The time between the first falling edge of the Sync Field and the fifth falling edge of the Sync Field is measured. This is then divided by eight to give the baud rate of the data that will be transmitted. The ADuC7032 implements the timing of this Sync byte in hardware. For more information on this feature, please refer to LIN Hardware Synchronization.

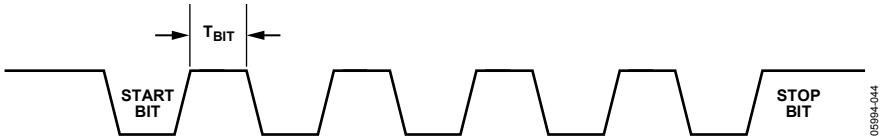

Figure 41 : LIN Synch byte Field

LIN Frame Protected Identifier

After receiving the LIN synch field, the required baud rate for the UART is calculated. The UART is then configured, which allows the ADuC7032 to receive the Protected Identifier, as shown in Figure 42. The Protected Identifier consists of two sub-fields, the identifier and the identifier parity. The six bit identifier contains the identifier of the target for the frame. The identifier signifies the number of data bytes to be either received or transmitted. The number of bytes is user configurable at system level design. The parity is calculated on the identifier, and is dependent on the revision of LIN the system is designed for.

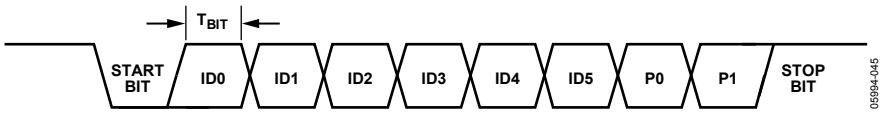

Figure 42 : LIN Identifier Byte Field

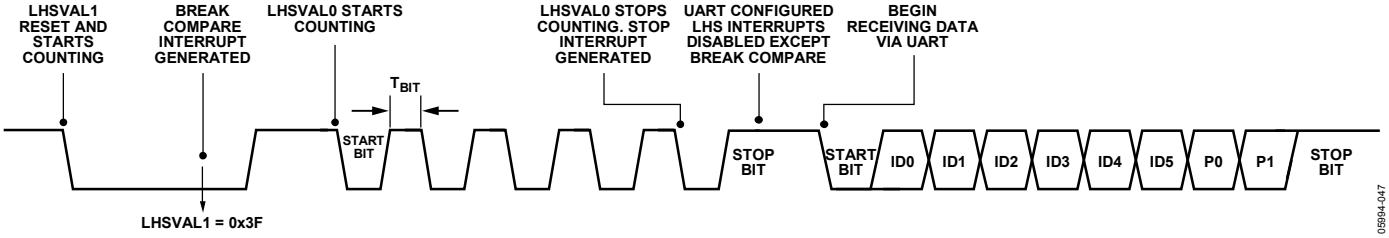

Figure 44 : Example LIN Configuration

LIN Diagnostics

The ADuC7032 features the capability to non-intrusively monitor the current state of the LIN pin. This read back functionality is implemented via GPIO11. The current state of the LIN pin is contained in GP2DAT[4]

It is also possible to drive the LIN pin high and low via user software, allowing the user to detect open circuit conditions. This functionality is implemented via GPIO12. To enable this functionality GPIO12 must be configured as a GPIO via GP2CON[20]. Once configured, the LIN pin may be pulled high or low via GP2DAT.

The ADuC7032 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 via HVCFG1[3]. It is possible to disable this feature via HVCFG1[2].

ADUC7032 ON-CHIP DIAGNOSTICS

The ADuC7032 integrates multiple diagnostic support circuits on-chip. These circuits allow the device to test core digital functionality, analog front-end and high-voltage I/O ports in-circuit.

ADC Diagnostics

Internal Test Voltage

The current channel can be configured to convert on an internal 8.3mV test voltage . On any gain range the result should be within ±0.5% of the expected result.

Internal Short Mode

The current and voltage input channels can also be shorted internally. Converting on the internal short will allow an assessment of the internal ADC noise to be determined.

Internal Current Sources

Internal current sources can also be enabled on both current and temperature channels. These current sources can be used to determine external short or open circuit conditions in both external shunt or temperature sensor configurations.

High Voltage I/O Diagnostics

High Voltage I/O Read back

All high voltage I/O pins are supported with read back capability. This allows the detection of external short conditions.

High Voltage Current Detection

All high voltage I/O pins also have a high current detection capability allowing high side connections to VBAT to be detected and controlled.

System Serial ID Register 1:

Name : **SYSSER1**
Address : 0xFFFF023C
Default Value : 0x00000000(Updated by kernel at power-on)
Access : Read/Write

Function : At power-on, this 32-bit register will hold the values of the part ID number, silicon mask revision number and kernel revision number (bottom die only) as detailed below.

Table 85: SYSSER1 MMR Bit Descriptions

Bit	Description
31-28	Silicon Mask Revision ID The 4 bits read from this nibble reflect the silicon mask ID number. Specifically, the hex value in this nibble should be decoded as the lower hex nibble in the hex numbers reflecting the ASCII characters in the range 'A' to 'O'. Examples: Bits 19-16 = 0001 = 1hex, therefore this value should be interpreted as 4***1*** which is ASCII character A corresponding to silicon mask revision A Bits 19-16 = 1011 = Bhex, therefore the number is interpreted as 4***B*** which is ASCII character K corresponding to silicon mask revision K The allowable range for this value is 1 to 15 which is interpreted as 41 to 4F or A to O)
27-20	Kernel Revision ID This byte contains the hex number which should be interpreted as an ASCII character indicating the revision of the kernel firmware embedded in the on-chip Flash/EE memory. Example: Reading 0x41 from this byte should be interpreted as A indicating a revision A kernel is on-chip.
19-16	Kernel Minor Revision number For PreProduction Release, these bits refer to the Device's Kernel Minor Revision Number
15-0	Part ID These 16 LSBs will hold a 16-bit number which should be interpreted as the part ID number. When used in conjunction with the value in SYSSER0 i.e. the manufacturing lot ID, this number is a unique identifier for the part.

System Kernel Checksum:

Name : **SYSCHK**
Address : 0xFFFF0240
Default Value : 0x00000000(Updated by kernel at power-on)
Access : Read/Write

Function : At power-on, this 32-bit register will hold the kernel checksum