iND86201 “Ernie”

indie’s highly integrated, microcontroller-based glucose measurement subsystem

Preliminary Datasheet

Table of Contents

1.0 SYSTEM OVERVIEW 7

1.1 Introduction 7

1.2 Key Features 7

1.3 Package Variants 8

1.4 Application Context 8

2.0 MEMORY 9

2.1 Register Description Convention 9

2.2 Top Level Memory Map 10

3.0 SYSTEM AND PERIPHERAL IMPLEMENTATION 11

3.1 Digital Functions 11
3.1.1 Processor 13
3.1.2 Flash Memory 13
3.1.3 Flash Memory Controller 14
3.1.4 Data Memory 22
3.1.5 Clock Sources 22
3.1.6 Timer0,1,2 28
3.1.7 Timer3 30
3.1.8 Wake-Up Timer 34
3.1.9 Watch Dog Timer 35
3.1.10 Nested Vector Interrupt Controller 36
3.1.11 Crystal Oscillator and Real-time Clock 37
3.1.12 LCD 39
3.1.13 GPIO 49
3.1.14 SPI Interface 59
3.1.15 UART Interface 63
3.1.16 I²C Interface 68
3.1.17 USB Interface 79
3.1.18 Pulse Width Modulator 84
3.1.19 Internal Interface Bus 87

3.2 Analog Functions 89
3.2.1 Sensor Configurations 89
3.2.2 Analog Sensor Interface and Switching 90
3.2.3 Voltage Reference 98
3.2.4 SG Reference 98
3.2.5 Source DACs 99
3.2.6 Transimpedance Amplifiers 102
3.2.7 Sigma Delta ADC 103

3.2.8 Auxiliary 8-bit ADC 108
3.2.9 Power Management Unit 113

3.3 Design for Test Features 122

4.0 ELECTRICAL CHARACTERISTICS 123

4.1 Electrical Conditions 123
4.1.1 Absolute Maximum ratings 123
4.1.2 Operating conditions 123
4.1.3 Electrical Characteristics over Operating Range 124

5.0 PACKAGE 129

5.1 Ball Assignment 129

5.2 Signal Descriptions 130

5.3 PACKAGE DRAWING BGA PACKAGE 133

5.4 Lead Assignment (QFN 10x10, 88 Lead) 134

5.5 Signal Description – QFN Package 135

5.6 Package Drawing – QFN Package 137

6.0 REVISION HISTORY 138

7.0 REFERENCES 139

8.0 CONTACTS 140

Table of Figures

Figure 1: Digital Engine Block Diagram 12
Figure 2: Flash Memory State Machine Diagram 16
Figure 3: Clock Management System 25
Figure 4: RTC alarm timing example 37
Figure 5: RTC overflow timing example 38
Figure 6: Quadplexed LCD Electrode Connections 41
Figure 7: Quadplexed LCD Drive Waveform Example 43
Figure 8: Quadplexed LCD Drive Waveform Example 44
Figure 9: Connecting Pins for LCD Charge Pump and Divider 45
Figure 10: GPIO Functional Diagram 49
Figure 11: SPI Timing Diagram 60
Figure 12: Slave Mode Timing w/CLK_ST_ENB=1 (Reception, 7-bit Address) 70
Figure 13: Slave Mode Timing w/CLK_ST_ENB=0 (Reception, 7-bit Address) 70
Figure 14: Slave Mode Timing (Transmission, 7-bit Address) 71
Figure 15: Slave Mode Timing (Reception, 10-bit Address) 71
Figure 16: Slave Mode Timing (Transmission, 10-bit Address) 71
Figure 17: Master Timing Waveform (Transmission) 74
Figure 18: Master Timing Waveform (Reception) 75
Figure 19: Switch matrix shown with external components for ground-referenced sensor 92
Figure 20: Sigma Delta ADC Input Channel Multiplexing 98
Figure 21: SG Trim DAC Circuit 99
Figure 22: Sub-ranging (Coarse + Fine) DAC Architecture 100
Figure 23: External Voltage Reference for Amplifier G 102
Figure 24: External Voltage Reference for Amplifier A 103
Figure 25: 8-bit ADC Input Settling Time 109
Figure 26: 8-bit ADC Reference Voltage Equations 110
Figure 27: Power Management System Diagram 116
Figure 28: BGA Package Ball Assignment (Top View) 130
Figure 29: BGA Package Outline Drawing 133

Figure 30: QFN Package Ball Assignment (Top View) 134
Figure 31: QFN Package Outline Drawing 137

Table of Tables

Table 2.1 System Memory Map 10
Table 2.2 Peripheral Fast Access Memory Map (7-bit address space) 10
Table 2.3 Peripheral Slow Access Memory Map (16-bit address space) 11
Table 4.4 Test Signals 13
Table 3.5 Flash Memory Map 14
Table 3.6 Flash Controller Memory Map 14
Table 3.7 Interrupt Vectors 36
Table 3.8 Oscillator Pins 37
Table 3.9 LCD signals 40
Table 3.10 LCD 1/3 Bias Drive Voltage Levels 41
Table 3.11 LCD Select and Deselect Voltages (COM0 through COM3) 42
Table 3.12 LCD 1/3 Bias Drive Voltage Levels 45
Table 3.13 LCD Registers 46
Table 3.14 GPIO Port Map 50
Table 3.15 GPIO Registers 51
Table 3.16 SPI (Master) Signals 59
Table 3.17 SPI Registers 60
Table 3.18 UART Signals 63
Table 3.19 UART Registers 64
Table 3.20 I2C Signals 68
Table 3.21 I2C Registers 75
Table 3.22 USB Signals 79
Table 3.23 USB Registers 80
Table 3.24 PWM Signal 85
Table 3.25 PWM Prescaler Divider Values 85
Table 3.26 PWM Registers 86
Table 3.27 Sensor interface 92
Table 3.28 Gain Setting Pins 93

Table 3.29 Reference resistor and Thermistor Interface 93
Table 3.30 Analog Voltage Reference Pins 94
Table 3.31 Auxiliary 8-bit ADC registers 111
Table 3.32 Power Management Signals 113
Table 3.33 Analog Boost Regulator Signals 117
Table 4.34 Absolute maximum ratings 123
Table 4.35 Operating Conditions 123
Table 4.36 Supply current 124
Table 4.37 Supply Voltage monitor 124
Table 4.38 General Purpose I/O 125
Table 4.39 Buzzer Driver 125
Table 4.40 Source DAC Voltage 126
Table 4.41 Transimpedance Amplifier 127
Table 4.42 Switch Matrix 127
Table 4.43 Sigma Delta ADC 128
Table 4.44 LCD 128
Table 5.45 Package Signal Descriptions 130
Table 5.46 QFN Package Signal Descriptions 135
Table 6.47 Revision History 138

1.0 System Overview

1.1 Introduction

The Ernie System-in-Package (SiP) platform for glucose meter applications provides industry-leading front-end accuracy, power consumption, display and connectivity features at a not-seen-before integration and cost level. This is achieved by tight integration with the application requirement and special-purpose hardware designed to reduce common measurement errors.

Traditional glucose meters often use a separate micro-controller (MCU) and analog front-end (AFE) products, making the end-user of this approach pay for unnecessary additional packaging, increased PCB space, and significant hardware and embedded software design effort to test the communications and data transfer between the MCU and AFE. The Ernie device overcomes these limitations by providing a single small system-in-package (SiP) that not only combines a high performance low power ARM Cortex-M0 MCU and glucose measurement AFE dice into a single package, but also functionally maps the AFE functions within the MCU address space for easy error-free software development.

1.2 Key Features

- ARM Cortex M0 32-bit micro-controller
- 160 kByte eFlash memory (automotive grade reliability)
- 8 kByte SRAM
- One 17-bit signed sigma-delta ADC + one 8-bit auxiliary ADC
- Two 14-bit DACs
- Integrated sensor interface switch matrix supporting up to 9 sensor electrodes
- USB 1.1 interface including all associated power and clock management
- 32.768 kHz crystal oscillator
- 12 MHz RC oscillator
- Integer-N PLL using crystal reference
- RTC function with wake-up capability
- Integrated 4x30 LCD controller
- UART, SPI and I2C Hardware Interfaces
- 27 digital GPIO pins
- 2 kV HBM ESD protection on all pins (8 kV HBM ESD on select sensor interface pins)
- Temperature Range -40°C to +85°C
- Multiple package options

1.3 PACKAGE VARIANTS

Ernie comes in two package options as described later in this document. The difference in package options is limited to the number of sensor contacts in the analog front end and the total number of GPIOs available to the end application. All other features remain the same in both packages.

For a detailed description of these differences please see the analog front end sensor description in section 3.2 and the package pin assignments in Section 5. A brief summary of the high level difference is shown here:

Package	Pins	Size (mm)	Pitch (mm)	GPIOs	Sensor I/F
BGA	120	8x8	0.65	27	22
QFN-EP	88	10x10	0.4	20	9

1.4 APPLICATION CONTEXT

This document defines the functionality of Ernie, a hybrid chip consisting of a microcontroller with peripherals, referred to as the digital functions, and a dedicated data acquisition circuitry, referred to as the analog functions, to be used in a portable meter for measurement of blood glucose or other analytes. The meter is designed to be compact, low cost, battery powered and intended for in vitro diagnostic use. It can be used in conjunction with electrochemical glucose test strips. The device can support different sensor types using software-controlled pin reconfiguration in the switch matrix part of the analog front-end.

Under digital engine software control, the analog engine can apply voltages to a sensor and measure resulting currents in a potentiostat configuration. The analog engine can also be used to measure and compensate for the effects of ambient temperature using measurements of an external thermistor. The digital engine interfaces with a liquid crystal display (LCD) that can display results or operator prompting. The digital engine can transfer stored results to an external computer via a USB interface for analysis or printing. The digital engine supports a real-time clock that is used in a firmware-based calendar for time-stamping glucose measurement results and/or any scheduling activity.

2.0 MEMORY

2.1 REGISTER DESCRIPTION CONVENTION

The following tabular format is used to describe device registers throughout this document. The first row of a table enumerates the register name, the starting hexadecimal address, and the reset or default hexadecimal value. The second row indicates whether associated bits are read only, write only, or read/write (R/W). The third row enumerates bit names for the lower order byte, with most-significant bit to the left and least-significant to the right. For a 16-bit register, the next byte contains bit names for the next most-significant eight bits. Should the read/write status of any bits differ from that for the lower order byte, this row may be preceded by a read/write status row. If no difference, no additional read/write status row is included. For a 32-bit register, two additional bytes are used to enumerate bit names for the most-significant sixteen bits. Either may be preceded by a read/write status row, but not if there is no difference from the next lower byte. The next row of the table indicates that bit ordering is MSB to LSB. The last row summarizes the naming convention for the register and provides a brief functional description.

The format for an 8-bit register definition is below:

Descriptive register title							
Register Name		Starting Address (Hex)			Reset Value (Hex)		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit_Name	Bit_Name	Bit_Name	Bit_Name	Bit_Name	Bit_Name	Bit_Name	Bit_Name
MSB							LSB
Bit usage description							

The format for a 32-bit register definition is below:

Descriptive register title							
TMRREG		0x50020000			0x00000000		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit_Name	Bit_Name	Bit_Name	Bit_Name	Bit_Name	Bit_Name	Bit_Name	Bit_Name
Bit_Name	Bit_Name	Bit_Name	Bit_Name	Bit_Name	Bit_Name	Bit_Name	Bit_Name
Bit_Name	Bit_Name	Bit_Name	Bit_Name	Bit_Name	Bit_Name	Bit_Name	Bit_Name
Bit_Name	Bit_Name	Bit_Name	Bit_Name	Bit_Name	Bit_Name	Bit_Name	Bit_Name
MSB							LSB
Bit usage description							

2.2 TOP LEVEL MEMORY MAP

All processor and peripheral memory is part of a unified memory map, as shown below:

Table 2.1 System Memory Map

Address	Memory	Description	Reference
0x00000000 – 0x00027FFF	Flash	160 Kbyte Flash Memory	
0x00028000 – 0x0003FFFF	N/A	*Reserved*	
0x00040000 – 0x000400FF	Flash	256 Byte 1st NVR Sector	
0x00040100 – 0x000401FF	Flash	256 Byte 2nd NVR Sector	
0x00040200 – 0x1FFFFFFF	N/A	*Reserved*	
0x20000000 – 0x20001FFF	SRAM	8 Kbyte SRAM	
0x20002000 – 0x4FFFFFFF	N/A	*Reserved*	
0x50000000 – 0x5000007F	Peripheral	128 Byte peripheral fast access	Table 2.2
0x50000080 – 0x50000085	Peripheral	Block Transfer control	
0x50000086 – 0x5000FFFF	N/A	*Reserved*	
0x50010000 – 0x5001FFFF	Peripheral	64 Kbyte peripheral slow access	Table 2.3
0x50020000 – 0x5002001F	Peripheral	32 Byte timer control	
0x50020020 – 0x50020047	Flash	Flash program/erase control	
0x50020048 – 0xDFFFFFFF	N/A	*Reserved*	
0xE0000000 – 0xE00FFFFF	Private peripheral bus	ARM peripherals	
0xE0100000 – 0xEFFFFFFF	N/A	*Reserved*	
0xF0000000 – 0xF0001FFF	System ROM tables	ARM core IDs	
0xF0002000 – 0xFFFFFFFF	N/A	*Reserved*	

Table 2.2 Peripheral Fast Access Memory Map (7-bit address space)

Address	Peripheral	Description	Max. Size	Reference
0x50000000 – 0x50000007	PMU	Power management control	8 bytes	
0x50000008 – 0x5000000F	I2C	I2C bus control	8 bytes	[1]
0x50000010 – 0x50000017	UART	UART control	8 bytes	[1]
0x50000018 – 0x5000001B		Reserved	4 bytes	
0x5000001C – 0x5000001F	SPI	SPI bus control	4 bytes	[1]
0x50000020 – 0x5000002F	ADC16	16-bit Sigma-delta ADC	16 bytes	
0x50000030 – 0x50000037	ADC8	8-bit Charge-sharing ADC	8 bytes	
0x50000038 – 0x5000003F		Reserved	8 bytes	
0x50000040 – 0x5000005F	AFE	Analog Front-End Control	32 bytes	
0x50000060 – 0x5000006F	RTC	Real time clock control	16 bytes	
0x50000070 – 0x50000077	Timer3	Timer 3 control	8 bytes	
0x50000078 – 0x5000007F	GPIO	GPIO Byte control	16 bytes	Table 3.15

Table 2.3 Peripheral Slow Access Memory Map (16-bit address space)				
Address	**Peripheral**	**Description**	**Max. Size**	**Reference**
0x50010000 – 0x5001033F	GPIO	GPIO bit control	1K bytes	
0x50010340 – 0x5001133F		Reserved	4K bytes	
0x50011340 – 0x5001173F	USB	USB end-point buffer memory	1K bytes	
0x50011740 – 0x500117FF	USB	USB Special Function Registers	192 bytes	
0x50011800	USB	USB IP Top-Level Register	1 byte	
0x50018001 – 0x5001807F		Reserved		
0x50018080 – 0x50018087	GPIO	PortB_Config	8 bytes	
0x50018088 – 0x5001808F	GPIO	PortC_Config	8 bytes	
0x50018090 – 0x50018097	GPIO	PortD_Config	8 bytes	
0x50018098 – 0x5001809F	GPIO	PortE_Config	8 bytes	
0x500180A0 – 0x500180BF	LCD	LCD driver control	32 bytes	Table 3.13
0x500180C0 – 0x500180CF	PWM	Pulse Width Modulator Control	16 bytes	Table 3.26
0x500180D0 – 0x500180D7	LDO	LDO regulator control	8 bytes	
0x500180D8 – 0x500180DF		Reserved		
0x500180E0 – 0x500180E7	CLK	Clock source trim settings	8 bytes	
0x500180E8 – 0x500180F7		Reserved		
0x500180F8 – 0x500180FF	TEST	Debug test bus addressing	8 bytes	

3.0 SYSTEM AND PERIPHERAL IMPLEMENTATION

3.1 DIGITAL FUNCTIONS

Ernie digital controller uses an ARM M0 Core microcontroller with embedded Flash and SRAM memory as well as 3 general purpose Timers (also referred to as “Clough” or “CM0160K8” hereinafter). This core CPU handles control, user interface, data processing, I/O, and timing functions. Ernie SiP device contains the Clough, with an analog ASIC die with oscillators, SPI, UART and I2C I/O logic, USB interface, real-time clock, interval timers, general purpose I/O pins, and interrupt vectoring logic, LCD controller and the glucose meter analog front-end. Button or switch inputs can be handled using general purpose I/O pins.

A description of digital engine functions follows after the block diagram shown below:

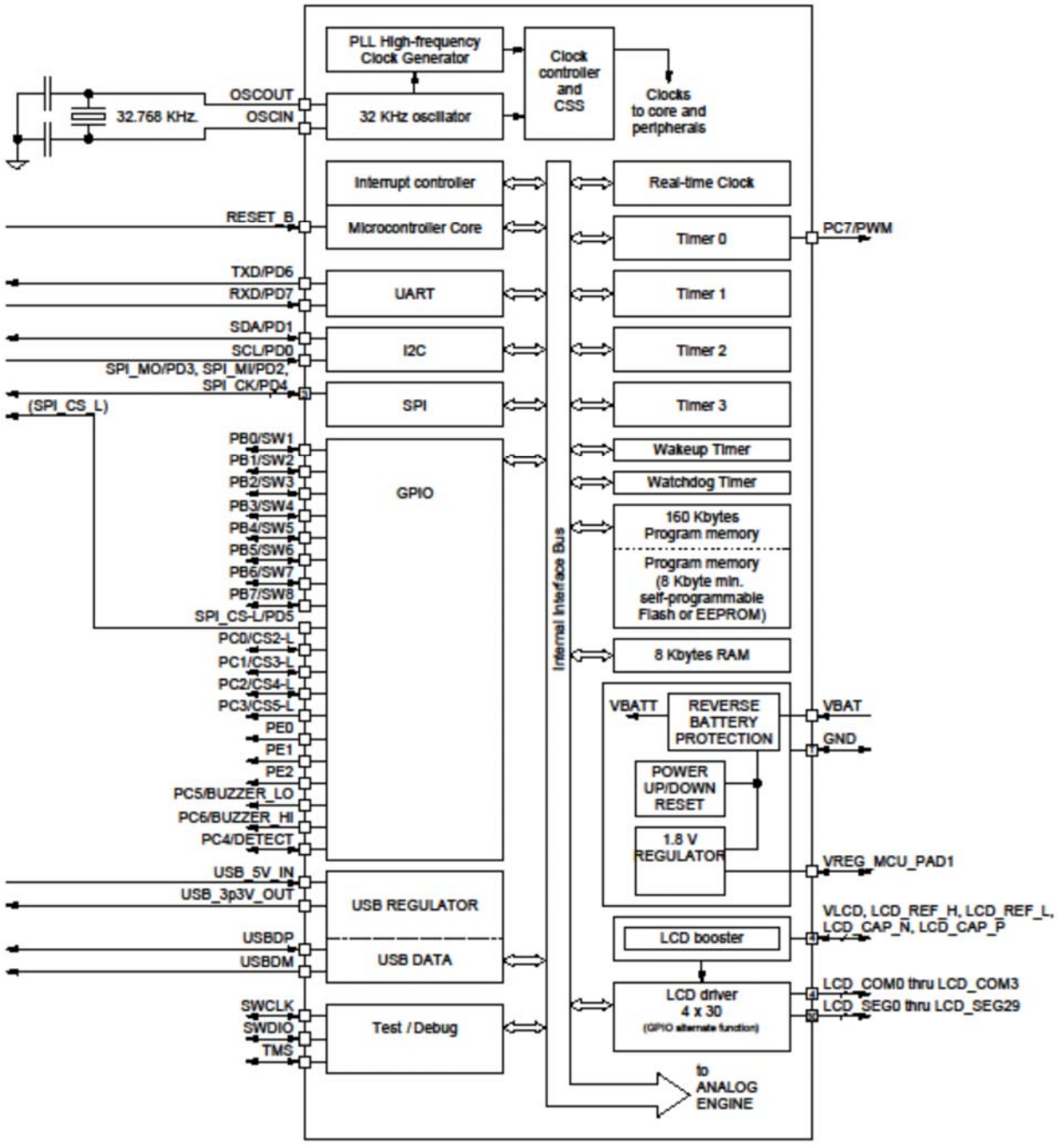


Figure 1: Digital Engine Block Diagram

3.1.1 Processor

The ASIC includes an embedded microcontroller subsystem based upon the ARM Cortex M0 core. It also includes a 160 Kbyte program flash memory and 8 Kbyte SRAM. It includes three 32-bit timers, a 16-bit timer, a dedicated watchdog timer and a wakeup timer. Additionally, it includes a Nested Vector Interrupt Controller (NVIC) to schedule hardware interrupts, and a Wakeup Interrupt Controller (WIC), which enables the control of the various power modes. The ARM Cortex M0 implementation supports 8-bit, 16-bit, and 32-bit reads/writes to peripherals.

More detailed information regarding the Ernie internal MCU "Clough", with M0 registers and included peripherals can be found in reference [1], though all relevant information is excerpted in appropriate sections in this document.

3.1.1.1 Debug Interface

An SWD-compliant 2-wire debug interface is implemented to access the processor and all memory mapped peripherals, as described below. Breakpoints and watch-points are supported. Trace capability is not supported on the Cortex M0 processor.

Table 4.4 Test Signals

Pin Name	Description
SWCLK	SerialWire Clock – SerialWire clock input connection. To minimize power dissipation, this pin should be connected to an external 10 KΩ pull-down resistor to GND.
SWDIO	SerialWire Data – SerialWire data I/O connection. To minimize power dissipation, this pin should be connected to an external 10 KΩ pull-down resistor to GND.
TMS	Test Mode Select input connection. Should be connected to GND in application PCB.

3.1.2 Flash Memory

The device contains 160 Kbyte of embedded Flash based non-volatile program memory. Endurance is greater than 100,000 write cycle endurance, so that part of this memory may also be used for storing clinical results.

The address space of this memory is from `0x0000000` to `0x00027FFF`. Since this space includes the boot vector, the microcontroller boots from an address configured in this flash memory, and user code will typically execute from this memory as well.

The flash memory is arranged in 2kB blocks, each of which is further comprised of 256B sectors. Memory can be erased either one sector or one block at a time, and can be written one byte at a time. These addresses cannot be written to directly. The writing and erasing of memory is handled by a memory-mapped Flash controller peripheral described in Section 3.1.3 Flash Memory Controller.

Note: The sector starting at `0x00027F00` is reserved for Indie production test use and cannot be written or erased by the user code. Consequentially, the block starting at 0x00027000 cannot be erased using the block erase command. All other sectors in this block can still be erased and bytes in these sectors can be written to by the user code.

Table 3.5 Flash Memory Map		
Starting Address	**Ending Address**	**Description**
0x00000000	0x000277FF	Read only access FLASH. Each block and sector can be erased and bytes can be written using flash controller peripheral.
0x00027800	0x00027EFF	Read only access FLASH. Each sector can be erased and bytes can be written using flash controller peripheral. This block cannot be erased using block erase.
0x00027F00	0x00027FFF	Read only access FLASH. This sector cannot be modified and typically contains Indie-specific information written during production test.

3.1.3 Flash Memory Controller

The flash memory can be written and erased using a memory-mapped flash control peripheral. To avoid unintentional modification to the flash memory, an unlocking scheme is implemented which requires multiple sequential operations in a fixed order in order to erase or write the flash contents.

Flash byte writes complete within 20µs, while erase operations can take up to 10ms to complete.

The flash cannot be read during a write or erase operation. Since program code typically resides in flash, this will result in the program stalling for the duration of the write / erase operation unless counter-measures are taken. If continued program execution during flash modification is required, then any code which must continue to run should execute from SRAM during the flash write/erase to avoid the processor stalling from attempting to fetch its code from the flash memory.

Furthermore, the interrupt vector table is located in flash memory. If an interrupt occurs and it is enabled, then the MCU will attempt to retrieve the interrupt service routine’s address from the flash memory. If the flash is busy performing a write or erase when this occurs, the processor will halt until the interrupt vector can be read from the flash. The interrupts numbered 0, 1, 2 and 16 are cached so that the vector fetch will not be stalled by the flash unavailability. So for these interrupts, if the service routine is mapped to a function in SRAM then the interrupt servicing will not stall the MCU during flash modification operations. Other interrupts should be disabled prior to initiating the flash operation if a MCU stall caused by their servicing must be avoided. Disabled interrupts will still pend, so re-enabling them after completion of the flash modification operation will cause any of the disabled interrupts which had occurred to be serviced at that time.

Table 3.6 Flash Controller Memory Map			
Address	**Register Name**	**R/W**	**Description**
0x50020020	FLADDR	W	Target address for write/erase operation. In byte writes, this is the read address of the flash to be written to. In erase modes, it is a read address inside the block or sector to be erased. This register must be written in the correct sequence or the operation will fail. This register must be written with a 32 bit access.

0x50020024	FLWRDT	W	Content to be written into the target byte. Bits 31 to 8 are ignored. This register must be written in the correct sequence or the operation will fail.
0x50020028	UNLBWR	W	Control register to unlock byte write. A value of 0x55555555 must be written to this address at the correct point in the byte write sequence or the operation will fail. This register must be written with a 32 bit access.
0x5002002C	BWRSTRT	W	Control register to start a byte write. A value of 0xAAAAAAAA must be written to this address at the correct point in the byte write sequence or the operation will fail. This register must be written with a 32 bit access.
0x50020030	UNLSER	W	Control register to unlock a sector erase. A value of 0x66666666 must be written to this address at the correct point in the sector erase sequence or the operation will fail. This register must be written with a 32 bit access.
0x50020034	SERSTRT	W	Control register to commit a sector erase. A value of 0x99999999 must be written to this address at the correct point in the sector erase sequence or the operation will fail. This register must be written with a 32 bit access.
0x50020038	UNLBLKER	W	Control register to unlock a block erase. A value of 0x77777777 must be written to this address at the correct point in the sector erase sequence or the operation will fail. This register must be written with a 32 bit access.
0x5002003C	BKERSTRT	W	Control register to commit a block erase. A value of 0x88888888 must be written to this address at the correct point in the sector erase sequence or the operation will fail. This register must be written with a 32 bit access.
0x50020040	FLSCTRL	R/W	[2:0]: read wait cycle setting (to allow use of clock frequency faster than 16MHz)
		R/W	[5:3]: write/erase clock frequency wait cycle setting (to allow use of clock frequency faster than 16MHz)
		R	[6]: read value is 1 when byte write is in progress
		R	[7]: read value is 1 sector erase is in progress
		R	[8]: read value is 1 block erase is in progress
0x50020044	FLSCP	W	Code protection control register. Write a value of 0xF2E11047 to disable the SerialWire interface. Write 0x00000000 to enable it. This register **must** be written with a 32 bit access.

3.1.3.1 Flash Memory Control State Machine

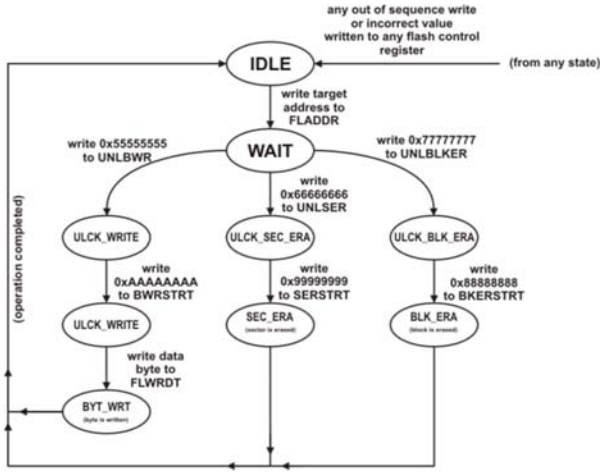


Figure 2: Flash Memory State Machine Diagram

3.1.3.2 Flash Control Registers Detailed Description

Flash Write/Erase Destination Address							
FLADDR		0x50020020			0xxxxxxxxx		
W	W	W	W	W	W	W	W
FLADD7	FLADD6	FLADD5	FLADD4	FLADD3	FLADD2	FLADD1	FLADD0
FLADD15	FLADD14	FLADD13	FLADD12	FLADD11	FLADD10	FLADD9	FLADD8
FLADD23	FLADD22	FLADD21	FLADD20	FLADD19	FLADD18	FLADD17	FLADD16
FLADD31	FLADD30	FLADD29	FLADD28	FLADD27	FLADD26	FLADD25	FLADD24
MSB							LSB
Bit31-0 FLADD[31:0] Target address for write/erase operation. In byte writes, this is the read address of the flash to be written to. In erase modes, it is a read address inside the block or sector to be erased. This register must be written in the correct sequence or the operation will fail. This register **must** be written with a 32 bit access.							
Flash Write Data Byte							
FLWRDT		0x50020024			0xxxxxxxxx		
W	W	W	W	W	W	W	W
FLDT7	FLDT6	FLDT5	FLDT4	FLDT3	FLDT2	FLDT1	FLDT0
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
MSB							LSB
Bit7 -0 FLDT[7:0] Content to be written into the target byte. Bits 31 to 8 are ignored. This register must be written in the correct sequence or the operation will fail. Control register to unlock byte write. A value of 0x55555555 must be written to this address at the correct point in the byte write sequence or the operation will fail. Bit31-8 Ignored.							

Flash Data Byte-Unlock Register							
UNLBWR		0x50020028			0xxxxxxxxx		
W	W	W	W	W	W	W	W
UBW7	UBW6	UBW5	UBW4	UBW3	UBW2	UBW1	UBW0
UBW15	UBW14	UBW13	UBW12	UBW11	UBW10	UBW9	UBW8
UBW23	UBW22	UBW21	UBW20	UBW19	UBW18	UBW17	UBW16
UBW31	UBW30	UBW29	UBW28	UBW27	UBW26	UBW25	UBW24
MSB							LSB
Bit31-0 UBW[31:0] Control register to unlock byte write. A value of 0x55555555 must be written to this address at the correct point in the byte write sequence or the operation will fail. This register **must** be written with a 32 bit access.							

Flash Byte-write Start Register							
BWRSTRT		0x5002002C			0xxxxxxxxx		
W	W	W	W	W	W	W	W
BRST7	BRST6	BRST5	BRST4	BRST3	BRST2	BRST1	BRST0
BRST15	BRST14	BRST13	BRST12	BRST11	BRST10	BRST9	BRST8
BRST23	BRST22	BRST21	BRST20	BRST19	BRST18	BRST17	BRST16
BRST31	BRST30	BRST29	BRST28	BRST27	BRST26	BRST25	BRST24
MSB							LSB
Bit31-0 BRST[31:0] Control register to start a byte write. A value of 0xAAAAAAAA must be written to this address at the correct point in the byte write sequence or the operation will fail. This register **must** be written with a 32 bit access.							

Flash Sector Erase Unlock Register							
UNLSER		0x50020030			0xxxxxxxxx		
W	W	W	W	W	W	W	W
USE7	USE6	USE5	USE4	USE3	USE2	USE1	USE0
USE15	USE14	USE13	USE12	USE11	USE10	USE9	USE8
USE23	USE22	USE21	USE20	USE19	USE18	USE17	USE16
USE31	USE30	USE29	USE28	USE27	USE26	USE25	USE24
MSB							LSB
Bit31-0 USE[31:0] Control register to unlock a sector erase. A value of 0x66666666 must be written to this address at the correct point in the sector erase sequence or the operation will fail. This register must be written with a 32 bit access.							

Flash Sector Erase Start Register							
SERSTRT		0x50020034			0xxxxxxxxx		
W	W	W	W	W	W	W	W
SES7	SES6	SES5	SES4	SES3	SES2	SES1	SES0
SES15	SES14	SES13	SES12	SES11	SES10	SES9	SES8
SES23	SES22	SES21	SES20	SES19	SES18	SES17	SES16
SES31	SES30	SES29	SES28	SES27	SES26	SES25	SES24
MSB							LSB
Bit31-0 SES[31:0] Control register to commit a sector erase. A value of 0x99999999 must be written to this address at the correct point in the sector erase sequence or the operation will fail. This register must be written with a 32 bit access.							

Flash Block Erase Unlock Register							
UNLBLKER		0x50020038			0xxxxxxxxx		
W	W	W	W	W	W	W	W
UBKE7	UBKE6	UBKE5	UBKE4	UBKE3	UBKE2	UBKE1	UBKE0
UBKE15	UBKE14	UBKE13	UBKE12	UBKE11	UBKE10	UBKE9	UBKE8
UBKE23	UBKE22	UBKE21	UBKE20	UBKE19	UBKE18	UBKE17	UBKE16
UBKE31	UBKE30	UBKE29	UBKE28	UBKE27	UBKE26	UBKE25	UBKE24
MSB							LSB
Bit31-0 UBKE[31:0] Control register to unlock a block erase. A value of 0x77777777 must be written to this address at the correct point in the sector erase sequence or the operation will fail. This register must be written with a 32 bit access.							

Flash Block Erase Start Register							
BKERSTRT		0x5002003C			0xxxxxxxxx		
W	W	W	W	W	W	W	W
BES7	BES6	BES5	BES4	BES3	BES2	BES1	BES0
BES15	BES14	BES13	BES12	BES11	BES10	BES9	BES8
BES23	BES22	BES21	BES20	BES19	BES18	BES17	BES16
BES31	BES30	BES29	BES28	BES27	BES26	BES25	BES24
MSB							LSB
Bit31-0 BES[31:0] Control register to commit a block erase. A value of 0x88888888 must be written to this address at the correct point in the sector erase sequence or the operation will fail.							

This register must be written with a 32 bit access.

Flash Control Register							
FLSCTRL		0x50020040			0x00000009		
R	R	R/W	R/W	R/W	R/W	R/W	R/W
SECER	PRGFLG	WECD2	WECD1	WECD0	RWC2	RWC1	RWC0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	R
-	-	-	-	-	-	-	BLKER
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
-	-	-	-	-	-	-	-
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
-	-	-	-	-	-	-	-
MSB							LSB

Bit31-9 Reserved
Bit8 BLKER: Block erase activity flag.
0 = Flash block erase not in progress / 1 = Flash block erase in progress
Bit7 SECER: Sector erase activity flag.
0 = Flash sector erase not in progress / 1 = Flash sector erase in progress
Bit6 PRGFLG: Program activity flag.
0 = Flash byte write not in progress / 1 = Flash byte write in progress
Bit5-3 WECD[2:0] Write/erase clock frequency divider setting. This register should be programmed with the same value as RWC. Attempting to write or erase the Flash memory with this value set too low for the MCU clock frequency may result in unpredictable behavior.
Bit2-0 RWC[2:0] Number of wait states used in the reading process. Each read from flash memory will take number of cycles equal to 1+RWC to complete. The number should be programmed according to the MCU clock frequency:
000 = 0 to 16MHz
001 = 16MHz to 32MHz
010 = 32MHz to 50MHz
Before changing to a clock frequency faster than the flash can support, this register should be updated to an appropriate value or risk flash accesses producing incorrect data.

Flash Code Protection Register							
FLSCP		0x50020044			0x00000000		
W	W	W	W	W	W	W	W
FCP7	FCP6	FCP5	FCP4	FCP3	FCP2	FCP1	FCP0
FCP15	FCP14	FCP13	FCP12	FCP11	FCP10	FCP9	FCP8
FCP23	FCP22	FCP21	FCP20	FCP19	FCP18	FCP17	FCP16
FCP31	FCP30	FCP29	FCP28	FCP27	FCP26	FCP25	FCP24
MSB							LSB
Bit31-0 FCP[31:0] Code Protection / SerialWire Lockout Control Code protection control register. Write a value of 0xF2E11047 to disable the SerialWire interface. Write 0x00000000 to enable it. This allows the user program to disable the SerialWire interface to prevent unauthorized debug access to the part. This register must be written with a 32 bit access. NOTE1: This register does not lock the Flash Memory against read/write/erase by the applications program. It instead disables all communications with the debug interface, therefore preventing any external attack. The application code is still able to modify the Flash content. NOTE2: Upon Power-On Reset or Normal Reset, CM0160K8 disables the communication for a small time interval (8192 clock cycles). If the application needs to be protected it is mandatory to set this register with the appropriate code in the beginning of the initialization process and before the internal hardware enables the debug communication.							

3.1.3.3 Flash Memory Operations & Examples

The following operations can be performed in the Flash Memory:

- Block Erase
- Sector Erase
- Byte Erase
- Code Protect

3.1.3.3.1 Block Erase

To erase a 2kB block the following sequence must be followed:

1. Write an address inside the block to be erased to the FLADDR register.
2. Unlock the block for erasure by writing the 0x77777777 pattern to the UNLBLKER register.
3. Start the block erase process by writing the 0x88888888 pattern into the BKERSTRT register.

Code Example:

```
*FLADDR   = 0x000050A0;          //Point to the block starting at 0x00005000
                                 //(any address 0x5000 to 0x57FF will work)
*UNLBLKER = 0x77777777;          //Unlock block erase
*BKERSTRT = 0x88888888;          //Start erase process
```

```
while ( *FLSCTRL & 0x100 );       //Check the Block Erase Flag until done
```

NOTE: If this code is executed from flash memory, the execution will stall even without the while loop due to the next opcode fetch being delayed until the erase is complete. This example is intended to illustrate the behavior of the FLSCTRL flags if the code is being executed from SRAM.

NOTE: The erase process of a block or sector can take up to 10msec.

3.1.3.3.2 Sector Erase

To erase a 256 byte sector the following sequence must be followed:

1. Write an address inside the sector to be erased to the FLADDR register.
2. Unlock the sector for erasure by writing the 0x66666666 pattern to the UNLSER register.
3. Start the sector erase process by writing the 0x99999999 pattern to the SERSTRT register.

Code Example:

```
*FLADDR = 0x000050A0;             //Point to the block starting at 0x00005000
                                  //(any address 0x5000 to 0x50FF will work)
*UNLSER = 0x66666666;             //Unlock sector erase
*SERSTRT = 0x99999999;            //Start erase process
while ( *FLSCTRL & 0x80 ); //Check the Sector Erase Flag, wait until done
```

NOTE: The erase process of a block or sector can take up to 10msec.

3.1.3.3.3 Write Byte

To write a byte the following sequence must be followed:

1. Write the flash address to be programmed to the FLADDR register.
2. Unlock the byte write by writing the 0x55555555 pattern to the UNLBWR register.
3. Start the writing process by writing the 0xAAAAAAAA pattern into the BWRSTRT register.
4. Write the value to be written into the FLWRDT register. (Bits 31 to 8 are ignored)

Code Example:

```
*FLADDR  = 0x000050BB;            //Point to the byte address
*UNLBWR  = 0x55555555;            //Unlock byte write
*BWRSTRT = 0xAAAAAAAA;            //Start write process
*FLWRDT  = 0x000000AB;            //Load 0xAB to be written
while ( *FLSCTRL & 0x40 ); //Check the Byte Write Flag, wait until done
```

NOTE: The byte write process can take up to 20µs.

3.1.3.3.4 Code Protect

CM0160K8 implements a protection scheme to prevent reverse engineering and tampering. It works as follows:

1. After any reset (Power-On or otherwise) the internal HW prevents any communication between the CPU and the debug interface during the first 8192 clock cycles.
2. The debug interface can also be disabled by loading the FLSCP register with the pattern 0xF2E11047. This lock can be released by loading FLSCP with the pattern 0x00000000.

By disabling the debug interface using the FLSCP lock during the first 8192 clock cycles, the program code can ensure that the debug interface is never active unless it specifically activates it by releasing the FLSCP lock.

Code Example:

```
*FLSCP = 0xF2E11047;              //Code protection activated
```

NOTE: If code protection is necessary it is important to consider the time it takes from the moment the part is reset until the execution reaches the main function. In some exceptional cases where the initialization of the system (stack, heap, global variables) take more than 8K clock cycles it may be necessary to add code directly in the initialization routine.

3.1.4 Data Memory

The device contains a single logically contiguous block of 8 Kbyte of data SRAM in the address range `0x20000000 – 0x20001FFF`

3.1.5 Clock Sources

The ASIC supports four clock sources as follows:

1. Internal low-frequency (10 kHz) RC oscillator, divided by 2 internally to make 5 kHz for use (this clock source is always enabled)
2. Crystal oscillator for driving external 32.768 kHz crystal
3. Internal high-frequency RC oscillator
4. Internal high-frequency PLL oscillator using undivided 32.768 kHz crystal clock as reference

Upon Reset or Power-On Reset, the system starts using the always-enabled internal 10 kHz low-frequency RC oscillator as the clock source. Depending upon application requirements, the user can:

- Enable or disable the external 32.768 kHz crystal
- Enable or disable the high-frequency PLL using crystal as reference
- Enable or disable the high-frequency RC oscillator
- Select the system clock source
- Enable the clock monitor interrupt to detect and process eventual failures
- Configure the hardware to automatically switch to high-frequency RC oscillator when waking up from sleep to reduce the wake-up time

The 10 KHz RC oscillator is always enabled by hardware. It is used for internal power management sequencing by the hardware.

The crystal oscillator is disabled upon exiting Power-On Reset to optimize pre-sale shelf life of the packaged meter including the battery before start of end user interaction. For several application scenarios, it is envisaged that after exiting the Power-On Reset event, the firmware will enable the 32.768 kHz crystal oscillator and leave it enabled during subsequent sleep events for proper RTC operation.

Hardware provides a high-frequency RC oscillator with fast start-up time of less than 20 µs normally used as the default system clock source for the MCU. This oscillator’s frequency is programmable between 12 MHz and 18 MHz.

There is also a high-frequency PLL oscillator using the 32.768 kHz crystal oscillator clock as a reference, with start-up time of 10 ms. This oscillator is required for USB 1.1 operation, and may be advantageous for any operation requiring higher frequency accuracy than achievable by the high-frequency RC oscillator. Such operations may include high baud rate UART or high sampling rate sigma delta ADC operation.

An integer division pre-scaler is available to scale the frequency used by the MCU if either the high-frequency RC oscillator or high-frequency PLL oscillator are selected as the clock source.

Figure 3 describes the clock management system

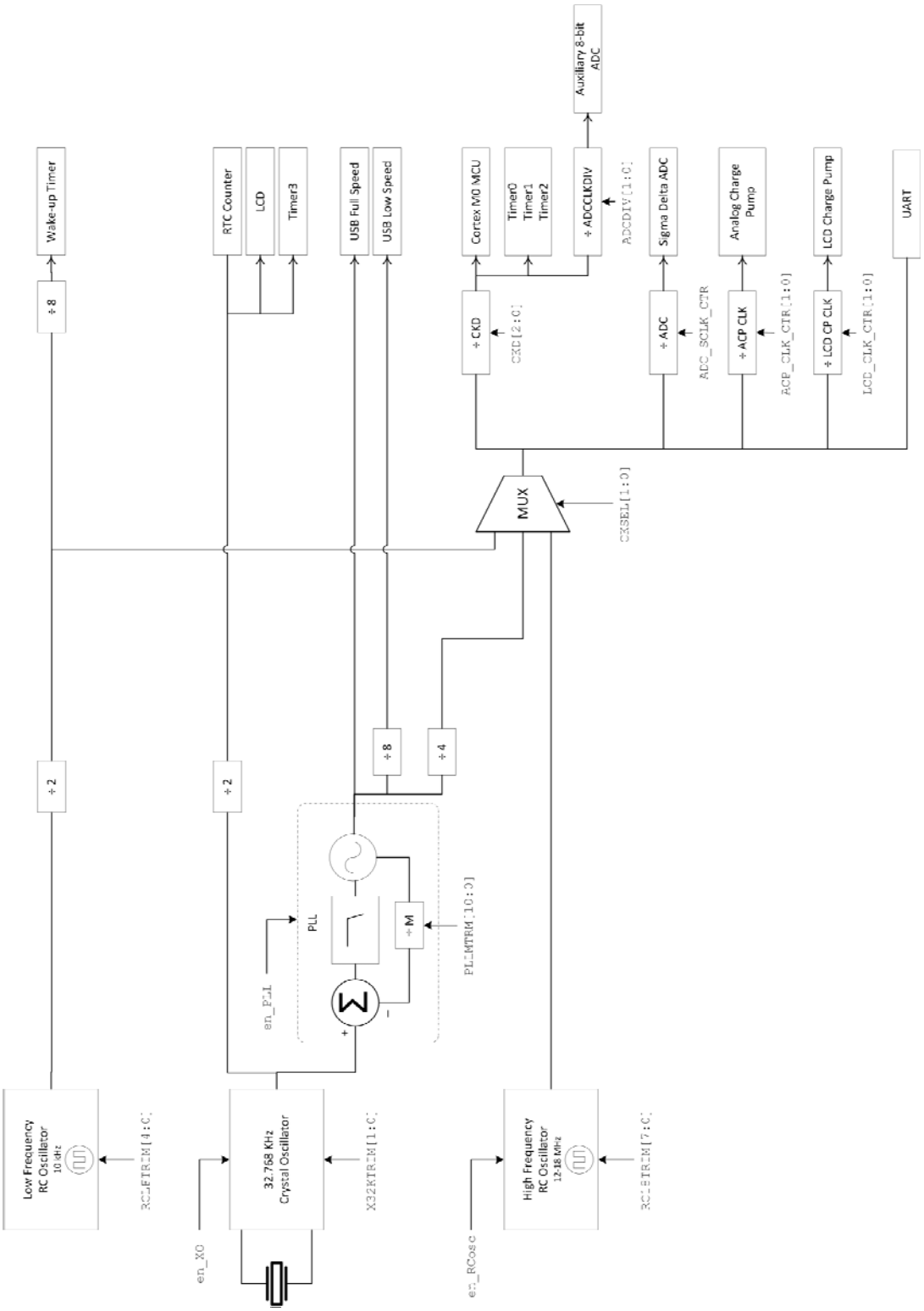


Figure 3: Clock Management System

3.1.5.1 Low-Frequency RC Oscillator Registers

Low frequency (10 kHz) RC oscillator frequency trim register							
RCLFTRIM		0x500180E4			0x10		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reserved	Reserved	Reserved	RCLFTRIM4	RCLFTRIM3	RCLFTRIM2	RCLFTRIM1	RCLFTRIM0
MSB							LSB

Bit7-5 *Reserved*

Bit4-0 **RCLFTRIM[4:0]**: Trim low frequency RC oscillator frequency

0x00 = Maximum RC oscillator output frequency

...

0x1F = Minimum RC oscillator output frequency

3.1.5.2 32.768 kHz Crystal Oscillator Registers

32.768 kHz crystal oscillator bias trim register							
X32KTRIM		0x500180E0			0x03		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	X32KTRIM1	X32KTRIM0
MSB							LSB

Bit7-2 *Reserved*

Bit1-0 **X32KTRIM[1:0]**: Trim crystal oscillator bias current (and transconductance)

0x00 = Maximum bias current (maximum transconductance gain)

...

0x03 = Minimum bias current (minimum transconductance gain)

3.1.5.3 High-Frequency RC Oscillator Registers

The post factory calibration accuracy is 5% over extended temperature and supply conditions for this oscillator. The high-frequency RC oscillator is factory calibrated with the resulting oscillator trim constant stored in a special sector of Flash memory that begins at the address FLASH_ADK_SECTOR and is mapped as defined in the `clough_core.h` header file.

The application software must transfer this value to register RC18MTRIM upon exiting Power-On Reset state. This value, once transferred, will be preserved during sleep events.

High frequency RC oscillator frequency trim register							
RC18TRIM		0x500180E5			0x80		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RC18TRIM7	RC18TRIM6	RC18TRIM5	RC18TRIM4	RC18TRIM3	RC18TRIM2	RC18TRIM1	RC18TRIM0
MSB							LSB

Bit7-0 **RC18TRIM[7:0]**: Trim RC oscillator frequency
0x00 = Maximum RC oscillator frequency
...
0xFF = Minimum RC oscillator frequency

3.1.5.4 PLL Oscillator Registers

Firmware can program the integer value of the feedback divider of this PLL to achieve any clock frequency of M*32.768 kHz, where M must be in the range of 1280 to 1536, inclusive. The 32.768 kHz crystal oscillator must be enabled and this divide ratio must be set by firmware before the PLL is enabled.

PLL feedback integer divider ratio register							
PLLFBDIV		0x500180D6			0x05B9		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reserved	Reserved	Reserved	Reserved	Reserved	PLLMTRM10	PLLMTRM9	PLLMTRM8
PLLMTRM7	PLLMTRM6	PLLMTRM5	PLLMTRM4	PLLMTRM3	PLLMTRM2	PLLMTRM1	PLLMTRM0
MSB							LSB

Bit15-11 *Reserved*
Bit10-0 **PLLMTRM[10:0]**: PLL Feedback Divider Ratio
PLL output frequency = PLLMTRM * 32768 Hz
Programmed value must be in the range 1280 to 1536, inclusive
Example values:
USB full speed operation: PLLMTRM = d1465
Sigma delta ADC operation: PLLMTRM = d1536
UART (115.2 Kbps) operation: PLLMTRM = d1350

3.1.5.5 Top Level Clock Management Registers

The following registers controls the behavior of the clock sources

Clock Control Register							
PMUCLK		0x50000000			0x45		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
CKD2	CKD1	CKD0	en_PLL	en_XO	en_RCosc	CKSEL1	CKSEL0
MSB							LSB

Bit7-5 **CKD[2:0]**: Clock Frequency divider.
000 = Clock divided by 1
001 = Clock divided by 2
010 = Clock divided by 3
011 = Clock divided by 4
100 = Clock divided by 6
101 = Clock divided by 8
110 = Clock divided by 12
111 = Clock divided by 24

Bit4 **en_PLL:** PLL control
0 = PLL disabled
1 = PLL enabled

Bit3 **en_XO:** Crystal Oscillator Control
0 = Crystal Oscillator disabled
1 = Crystal Oscillator enabled

Bit2 **en_RCosc:** Main RC Oscillator Control
0 = Main RC Oscillator disabled
1 = Main RC Oscillator enabled

Bit1-0 **CKSEL[1:0]:** Clock Select
00 = 10 KHz Auxiliary Clock Source (divided by 2 to make 5 KHz for actual use)
01 = 12 MHz Main RC Oscillator Clock Source
10 = PLL Clock Source divided by 4
11 = Clock fault indicator -> Auxiliary Clock active (5 KHz in actual use)

Function-specific Clock Control							
CLKCTR		0x50000003			0x08		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	adc_sclk_ctr	acp_clk_ctr1	acp_clk_ctr0	lcd_clk_ctr1	lcd_clk_ctr0
MSB							LSB

Bit7-5 *Reserved*

Bit4 **adc_sclk_ctr:** ADC sampling clock frequency divider
0 = Divide by 24, nominal ADC sampling clock frequency =~ 500 kHz (default)
1 = Divide by 12, nominal ADC sampling clock frequency =~ 1 MHz

Bit3-2 **acp_clk_ctr[1:0]:** Analog charge pump input clock
00 = 250 KHz
01 = 500 KHz
10 = 1 MHz (default)
11 = 2 MHz

Bit1-0 **lcd_clk_ctr[1:0]:** LCD charge pump clock select
00 = 16.384 kHz, derived as 32.768 kHz XTAL clock divided by 2 (default)
01 = 250 KHz
10 = 500 KHz
11 = 1 MHz

3.1.6 Timer0,1,2

The MCU implements three identical timers: Timer0, Timer1, and Timer2. These use the system clock as a clock source and, once activated, count up continuously. They start from the value initially loaded into the counting register (32-bit) and, if enabled, generate an interrupt upon rolling over (0xFFFFFFFF → 0x00000000).

3.1.6.1 Timer0,1,2 Registers

There is an initial value register and a control register associated with each of the three timers.

32-bit Timer0 initial value register							
TMR0REG		0x50020000			0x00000000		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
T7	T6	T5	T4	T3	T2	T1	T0
T15	T14	T13	T12	T11	T10	T9	T8
T23	T22	T21	T20	T19	T18	T17	T16
T31	T30	T29	T28	T27	T26	T25	T24
MSB							LSB

Bit31-0 **T[31:0]:** Timer Register initial value register

Timer0 Control register							
TMR0CTRL		0x50020004			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	TSTART
MSB							LSB
Bit0 **TSTART**: Timer enable bit. 0 = Timer not running 1 = Timer running							

32-bit Timer1 initial value register							
TMR1REG		0x50020008			0x00000000		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
T7	T6	T5	T4	T3	T2	T1	T0
T15	T14	T13	T12	T11	T10	T9	T8
T23	T22	T21	T20	T19	T18	T17	T16
T31	T30	T29	T28	T27	T26	T25	T24
MSB							LSB
Bit31-0 **T[31:0]**: Timer Register initial value register							

Timer1 Control register							
TMR1CTRL		0x5002000C			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	TSTART
MSB							LSB
Bit0 **TSTART**: Timer enable bit. 0 = Timer not running 1 = Timer running							

32-bit Timer2 initial value register							
TMR2REG		0x50020010			0x00000000		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
T7	T6	T5	T4	T3	T2	T1	T0
T15	T14	T13	T12	T11	T10	T9	T8
T23	T22	T21	T20	T19	T18	T17	T16
T31	T30	T29	T28	T27	T26	T25	T24
MSB							LSB
Bit31-0 **T[31:0]:** Timer Register initial value register							

Timer2 Control register							
TMR2CTRL		0x50020014			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	TSTART
MSB							LSB
Bit0 **TSTART:** Timer enable bit. 0 = Timer not running 1 = Timer running							

3.1.6.2 Timer0,1,2 Operation

Operation of each timer involves loading the initial counter register, enabling the timer, and either polling the current value of the counter register or enabling the interrupt processing it within the interrupt service routine. Note that the timer counting register must be reloaded by application code within the interrupt service routine.

3.1.7 Timer3

A special purpose timer with the following features is implemented on the ASIC:

- 16-bit resolution
- Loads an initial value and counts up at each rising edge of input clock
- Can generate interrupt on the cycle when the timer overflows to zero
- Provides ability to read current value at any time
- Separate register that stores the initial load value
  - Auto-reload of this value when overflow occurs (if this feature is enabled)
  - Over-writing current value register when timer is counting could result in errant condition
- Multiple input clock sources (see below) asynchronous with MCU clock
  - The accuracy of timers used for sensor measurement timing shall be within 0.1% by selecting appropriate clock source
  - Timing intervals of 0.1 s, 0.125 s, 0.25 s are supported
- Programmable pre-scaler (see below)

3.1.7.1 Timer3 Registers

16-bit Timer3 current value register							
TMR3REG		0x50000070			0x0000		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
T7	T6	T5	T4	T3	T2	T1	T0
T15	T14	T13	T12	T11	T10	T9	T8
MSB							LSB
Bit31-0 **T[15:0]**: Timer current value register							

16-bit Timer3 load value register							
TMR3LOAD		0x50000072			0x0000		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
T7	T6	T5	T4	T3	T2	T1	T0
T15	T14	T13	T12	T11	T10	T9	T8
MSB							LSB
Bit31-0 **T[15:0]**: Timer load value register (loaded automatically when overflow occurs if TM3CTRL[*AUTOLD] = 1)							

Timer3 Control register							
TMR3CTRL		0x50000074			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reserved	IRQEN	PRESCALE1	PRESCALE0	CLKSRC1	CLKSRC0	AUTOLD	TMR3EN
MSB							LSB

Bit0 TMR3EN: Timer enable bit.
0 = Timer not running (freezes current value register; counter starts from this value next time TMR3EN = 1)
1 = Timer running
Bit1 AUTOLD: Auto-load enable bit.
0 = Disable auto-reload of the value in load register into current value register when timer overflows to 0
1 = Enable auto reload of the value in load register into current value register when timer overflows to 0
Bit3-2 CLKSRC1:CLKSRC0 Clock source selection bits
00 = 5 KHz AUX clock (10 kHz RC clock divided by 2)
01 = 16.384 kHz clock (32.768 KHz XTAL clock divided by 2)
10 = RC clock (nominal 12 MHz)
11 = PLL output clock (48 MHz)
Bit5-4 PRESCALE1:PRESCALE0: Pre-scaler setting:
00 = Divide input clock by 1
01 = Divide input clock by 4
10 = Divide input clock by 16
11 = Divide input clock by 256
Bit6 IRQEN Interrupt enable bit
0 = Disable interrupt generation when overflow occurs (prevents MCU from waking up upon overflow)
1 = Enable interrupt generation when overflow occurs (wakes up MCU upon overflow even if MCU is in deep sleep)
Bit7 *Reserved*

3.1.7.2 Timer3 Usage

Proposed software instruction order:

- Program TMR3CTRL with appropriate settings keeping TMR3EN = 0 (till input clock is available)
- Program TMR3LOAD register
- Enable interrupt in NVIC (if desired)
- Program TMR3CTRL with TMR3EN = 1 to start counting

Calculation of interval time

- Consider load value to be a negative number that is truncated to 16-bits when programming
- **Interval time in seconds = (Pre-scaler Divider / Clock Frequency in Hz) * {abs(Load Value) + 1}**

Examples

1. Time 0.125 s intervals using 16.384 KHz XTAL-derived clock.

- Set Pre-scaler to divide by 1, Load Value = -2047 (0xF801)

2. Time 0.1 s intervals using 16.384 KHz XTAL-derived clock.

- Set Pre-scaler to divide by 1, Load Value = -1637.4 → -1637 (0xF99B)
- Fixed error in interval timing when using this setting: 24.4 us (244 ppm error << 1000 ppm spec)

3. Time 0.25 s intervals using 48 MHz clock.

- Set Pre-scaler to divide by 256, Load Value = -46874 (0x48E6)

4. Determine maximum interval time possible using each clock source

- Set Pre-scaler to divide by 256, Load value = -65535 (0x0001)
  - 5 KHz RC clock: ~3355.4 seconds (~56 minutes)
  - 16.384 KHz XTAL-derived clock: 1024 seconds (~17 minutes)
  - 12 MHz RC clock: 1.398 seconds
  - 48 MHz PLL clock: 0.3495 second

3.1.8 Wake-Up Timer

In addition to Timer 0/1/2, the processor includes a timer capable of waking-up the microcontroller from a deep sleep (halt) state.

The wake-up timer is a timer used to allow for recovery from deep sleep (halt), including when the microcontroller is disconnected from its power supply.

The following register controls the wake-up timer:

Wakeup Timer Control							
WKPTIME		0x50000004			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MANT3	MANT2	MANT1	MANT0	EXP3	EXP2	EXP1	EXP0
MSB							LSB

Bit7-4 **MANT[3:0]**: Mantissa of the wakeup timer

Bit7-4 **EXP[3:0]**: Exponent of the wakeup timer (range: 0…12)

$$Wakeup\ Period = 16 \times \frac{Mantissa \times 2^{Exponent}}{RCLFCLK}$$

Maximum wake-up period is 16*15*2^12/10e3 =~ 98.3 seconds

3.1.9 Watch Dog Timer

The device implements a Watch Dog Timer (WDT) that can operate in one of two basic ways. In Interrupt Mode, an interrupt will be generated in the event of a WDT rollover. In Reset Mode, the microcontroller will reset in the event of a WDT rollover.

Watch Dog Timer control register (32-bit)							
WDTCTRL		0x50020018			0x00000000		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	WDTPRES1	WDTPRES0	RSTFLAG	RESETEN	WDTEN
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
MSB							LSB

Bit4-3 **WDTPRES1:WDTPRES0:** WDT Prescaler:
00 : $t_{timeout}=2^{13}/f_{SystemClock}$
01 : $t_{timeout}=2^{19}/f_{SystemClock}$
10 : $t_{timeout}=2^{22}/f_{SystemClock}$
11 : $t_{timeout}=2^{32}/f_{SystemClock}$

Bit2 **RSTFLAG:** Reset Flag. This flag is set by the system at the initialization if the initialization was caused by a reset triggered by the WDT. The bit can be de-asserted by the application.

Bit1 **RESETEN:** Reset enable. If enabled, a WDT time-out will force the microcontroller to reset. This bit can be asserted, but it cannot be de-asserted.

Bit0 **WDTEN:** WDT enable. This bit can be asserted, but it cannot be de-asserted. Once the WDT is enabled, it cannot be turned off until a Reset or Power-On Reset occurs.

WDT Clear register (32-bit)							
WDTCLR		0x5002001C			0x00000000		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
WCLR7	WCLR6	WCLR5	WCLR4	WCLR3	WCLR2	WCLR1	WCLR0
WCLR15	WCLR14	WCLR13	WCLR12	WCLR11	WCLR10	WCLR9	WCLR8
WCLR23	WCLR22	WCLR21	WCLR20	WCLR19	WCLR18	WCLR17	WCLR16
WCLR31	WCLR30	WCLR29	WCLR28	WCLR27	WCLR26	WCLR25	WCLR24
MSB							LSB

Bit31-0 **WCLR[31:0]**:Clear Register. To clear the WDT counting, the following words must be written in this order and without any instruction between them:
0x3C570001
0x007F4AD6

Warning: Programming WDTCLR with other values or in the wrong order will cause the watchdog to generate an interrupt or reset the system.

3.1.10 Nested Vector Interrupt Controller

Interrupt vectors are implemented as defined in the following table:

Table 3.7 Interrupt Vectors			
Cortex M0 Specific Exceptions			
Name	**#**	**Description**	**Required Interrupt Handler (Function)**
HardFault_IRQn	-13	HardFault handler	HardFault_Handler (void)
SVCall_IRQn	-5	Supervisory call	
PendSV_IRQn	-2	Interrupt-driven request for system level service*	
SysTick_IRQn	-1	SysTick Timer interrupt	void SysTick_Handler(void)
ASIC Specific Exceptions			
Name	**#**	**Comments**	**Required Interrupt Handler (Function)**
BrownOut_IRQn	0	Brownout detection interrupt	void BrownOut_Handler (void)
ClkMon_IRQn	1	Clock monitor interrupt	void ClkMon_Handler (void)
PIN_IRQn	2	Pin change interrupt	void PIN_Handler (void)
RTC_Alarm_IRQn	3	RTC alarm interrupt	void RTC_Alarm_Handler (void)
RTC_Overflow_IRQn	4	RTC overflow interrupt	void RTC_Overflow_Handler (void)
I2C_Collision_IRQn	5	I2C Collision detection interrupt	void I2C_Collision_Handler (void)
I2C_IRQn	6	I2C event interrupt	void I2C_Handler (void)
UART_IRQn	7	UART event interrupt	void UART_Handler (void)
SPI_IRQn	8	SPI event interrupt	void SPI_Handler (void)
PLL_IRQn	9	PLL event interrupt	void PLL_Handler (void)
USB_Detect_IRQn	10	USB detection interrupt	void USB_Detect_Handler (void)
USB_Wakeup_IRQn	11	USB resume interrupt	void USB_Wakeup_Handler (void)
USB_Interrupt_IRQn	12	USB communications interrupt	void USB_Interrupt_Handler (void)
SDADC_IRQn	13	Sigma delta ADC conversion complete interrupt	void SDADC_Handler (void)
TIMER3_Interrupt	14	Special purpose interval timer interrupt	void Timer3_Handler (void)
IRQ15_IRQn	15	Reserved	void Default_IRQ_Handler(void)
TIMER0_IRQn	16	Timer0 interrupt	void Timer0_Handler (void)
TIMER1_IRQn	17	Timer1 interrupt	void Timer1_Handler (void)
TIMER2_IRQn	18	Timer2 interrupt	void Timer2_Handler (void)
WATCHDOG_IRQn	19	Watchdog timer interrupt	void Watchdog_Handler (void)

3.1.11 Crystal Oscillator and Real-time Clock

The digital engine supports a 32.768 KHz crystal oscillator for basic test timing and to support a Real Time Clock (RTC) counter for time and date maintenance. The crystal connections are to be compensated with shunt capacitors whose value must be chosen based on the selected crystal, taking into account PCB parasitic capacitance. The value of each shunt capacitor is expected to be in the 10-20 pF range. With factory calibration, the RTC is capable of maintaining ±2 min/month accuracy when operated within a 0°C to 50°C temperature range and within the specified operating supply voltage range.

The calendar function based on the 1 Hz tick provided by the RTC is maintained in firmware.

3.1.11.1 Crystal Oscillator Pins

Following pins describe the 32.768 KHz crystal oscillator external pins

Table 3.8 Oscillator Pins

Pin Name	Description
OSCOUT	32.768 KHz Oscillator Output – Driven connection to external 32.768 KHz crystal. To be compensated using a 18 pF shunt capacitor to ground (exact value dependent on precise crystal used and PCB layout)
OSCIN	32.768 KHz Oscillator Input – Input connection from external 32.768 KHz crystal. To be compensated using a 18 pF shunt capacitor to ground ((exact value dependent on precise crystal used and PCB layout)

3.1.11.2 Real Time Counter Operation

The Real-Time Counter is an independent counter running from a 16.384 kHz clock (32.768 kHz XTAL clock divided by 2). The RTC provides two interrupts: alarm and overflow. Either can wake-up the system from sleep mode.

When the counter is enabled, the current RTC register can always be read without any timing error.

However, writing to RTC registers is normally prevented by a locking mechanism. To enter configuration mode for the RTC, a special lock bit must be cleared to configure the RTC registers. To prevent cycle skip errors, firmware must complete all configuration operations for the RTC in less than ½ clock cycle of the 16.384 kHz clock (about < 30 µs). This is typically possible since the MCU clock can be chosen to be the high frequency RC oscillator clock during reconfiguration.

Please contact Indie Semiconductor for an example firmware driver implementing a firmware calendar function using this RTC.

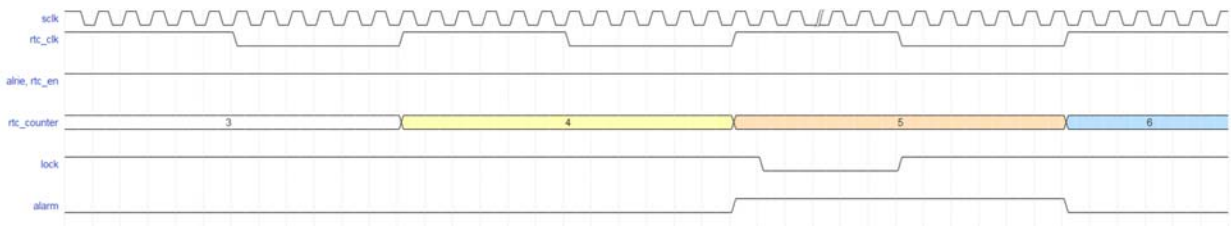


Figure 4: RTC alarm timing example

Counter starts at 'd0, alarm register set to 'd4

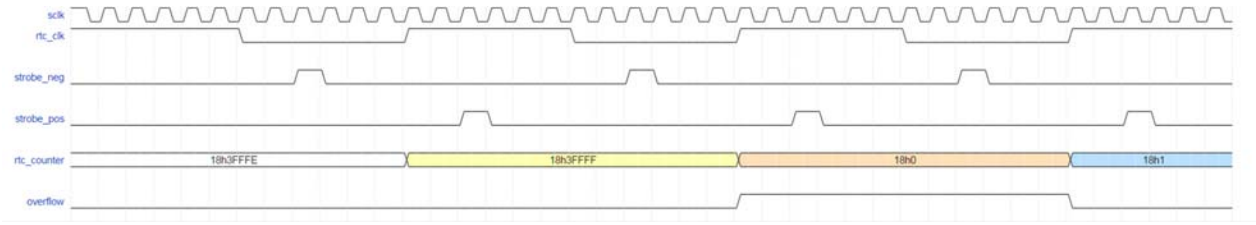


Figure 5: RTC overflow timing example

3.1.11.3 Real Time Counter Register Map

RTC control register							
RTC_CTRL		0x5000060			0x0D		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reserved	alrf0	alrf1	owf	alrie	owie	rtc_en	lock
MSB							LSB

Bit7 *Reserved*

Bit6 **alrf0**: alarm 0 flag
0 = alarm not detected
1 = alarm detected

Bit5 **alrf1**: alarm 1 flag
0 = alarm not detected
1 = alarm detected

Bit4 **owf**: overflow flag
0 = E overflow occurs
1 = E overflow not occurs

Bit3 **alrie**: alarm interrupt enable
0 = alarm interrupt is masked
1 = alarm interrupt is enable

Bit2 **owie**: overflow interrupt enable
0 = overflow interrupt is masked
1 = overflow interrupt is enable

Bit1 **lock**: lock
0 = configuration register is allowed to be written
1 = configuration register is not allowed to be written

Bit0 **rtc_en**: rtc enable
0 = rtc logic is not enable
1 = rtc logic is enable

RTC programmable current counter							
RTC_COUNT		0x5000064			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	cnt[17]	cnt[16]
cnt[15]	cnt[14]	cnt[13]	cnt[12]	cnt[11]	cnt[10]	cnt[9]	cnt[8]
cnt[7]	cnt[6]	cnt[5]	cnt[4]	cnt[3]	cnt[2]	cnt[1]	cnt[0]
MSB							LSB
Bit32-18	*Reserved*						
Bit17-0	**cnt[17-0]**: RTC counter current value						

RTC alarm 0 register							
RTC_ALARM0		0x5000068			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	alr0[17]	alr0[16]
alr0[15]	alr0[14]	alr0[13]	alr0[12]	alr0[11]	alr0[10]	alr0[9]	alr0[8]
alr0[7]	alr0[6]	alr0[5]	alr0[4]	alr0[3]	alr0[2]	alr0[1]	alr0[0]
MSB							LSB
Bit32-18	*Reserved*						
Bit17-0	**alr0[17-0]**: alarm 0 comparison value						

RTC alarm 1 register							
RTC_ALARM1		0x500006C			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	alr1[17]	alr1[16]
alr1[15]	alr1[14]	alr1[13]	alr1[12]	alr1[11]	alr1[10]	alr1[9]	alr1[8]
alr1[7]	alr1[6]	alr1[5]	alr1[4]	alr1[3]	alr1[2]	alr1[1]	alr1[0]
MSB							LSB
Bit32-18	*Reserved*						
Bit17-0	**alr1[17-0]**: alarm 1 comparison value						

3.1.12 LCD

The device includes an LCD interface whose main characteristics are:

- Support for quadplexed LCD drive; 1/4 duty (1/3 bias)
- Integrated LCD charge pump generating 4.5 V from battery voltage.
- Support for up to 120 display elements (4 commons by 30 segment lines).

The LCD interface requires up to 34 connections. For quadplexed drive there are four common signals (COM0, COM1, COM2 and COM3). Up to 30 segment signal connections (S0 through S29) can also be made, supporting a maximum of 120 display elements when operating with quadplexed drive.

An LCD charge pump generates a regulated 4.5 V supply for the LCD interface from the available battery or USB supply voltage. The charge pump and LCD ladder voltage network require the use of five external capacitors.

Table 3.9 LCD signals

Pin Name	Description
LCD_COM0 – LCD_COM3	LCD COM Driver Outputs (4) : Digital engine LCD COM driver outputs supporting quadplexed LCD drive.
LCD_SEG0 – LCD_SEG29	LCD SEG Driver Outputs (30) : Digital engine LCD SEG driver outputs supporting quadplexed LCD drive.
VLCD	LCD Charge Pump Output Voltage : Digital engine LCD charge pump external capacitor connection. Bypass with 4.7 µF capacitor.
LCD_REF_H	LCD Bias 1 : Digital engine LCD bias network external capacitor connection. Bypass with 0.47 µF capacitor.
LCD_REF_L	LCD Bias 2 : Digital engine LCD bias network external capacitor connection. Bypass with 0.47 µF capacitor.
LCD_CAP_N	LCD Charge Pump - : Digital engine LCD charge pump external 1 µF series charge pump capacitor connection.
LCD_CAP_P	LCD Charge Pump + : Digital engine LCD charge pump external 1 µF series charge pump capacitor connection.

3.1.12.1 LCD Drive Waveforms

The following describes the typical LCD drive waveform when operating with 1/4 duty cycle, 1/3 bias drive. This utilizes four common signals, COM0 through COM4, and a segment line for every four display elements. The ASIC supports up to 30 segment signals, S0 through S29, and can support an LCD with up to 120 display elements. The internal LCD charge pump regulator is designed to support 4.5 V display panels.

Each pixel of the LCD panel turns on when the potential difference between the corresponding common and segment signals becomes higher than a specific voltage (LCD drive voltage, V_{LCD}). The pixels turn off when the potential difference becomes lower than V_{LCD}.

Applying a DC voltage to the common and segment signals causes degradation of an LCD panel. To avoid this, the LCD panel is driven by AC voltage. When the display is disabled, no DC bias is applied between common and segment signals.

3.1.12.1.1 Common Signals

For quadplexed drive, each common signal is selected sequentially and allocated 1/4 of the cycle.

3.1.12.1.2 Segment Signals

The segment signals correspond to 30 bytes of the LCD display data memory (S0 through S29). Bits 0, 1, 2, and 3 of each byte are read in synchronization with COM0, COM1, COM2, and COM3, respectively. If a bit is 1, it is converted to the ‘select’ voltage, and if it is 0, it is converted to the deselect voltage. The conversion results are output to the segment pins (S0 through S29).

Determine the combination of front-surface electrodes (corresponding to the segment signal) and rear-surface electrodes (corresponding to the common signals) forms display patterns in the LCD data memory, and write the bit data that corresponds to the desired display pattern on a one-to-one basis.

LCD display data bits are not used and should be set to 0. They are reserved for potential future use supporting a blink function.

3.1.12.1.3 Output Waveforms for Common and Segment Signals

Table 3.10 lists the voltages output as common and segment signals for the 1/3 bias method.

Table 3.10 LCD 1/3 Bias Drive Voltage Levels

Segment Signal / Common Signal		Select Signal Level	Deselect Signal Level
		GND/VLCD	VLCD_REF_H/VLCD_REF_L
Select signal level	VLCD/GND	-VLCD/+VLCD	-(1/3)VLCD/+(1/3)VLCD
Deselect signal level	VLCD_REF_L/VLCD_REF_H	-(1/3)VLCD/+(1/3)VLCD	-(1/3)VLCD/+(1/3)VLCD

3.1.12.1.4 Quadplexed display example

Figure 6 shows how segment signals (S0 through S29) and common signals (COM0 through COM3) might be connected to a displayed digit of a multi-digit LCD panel. For this example, assume multiple digits are displayed by the panel. The contents of the display memory correspond to this display.

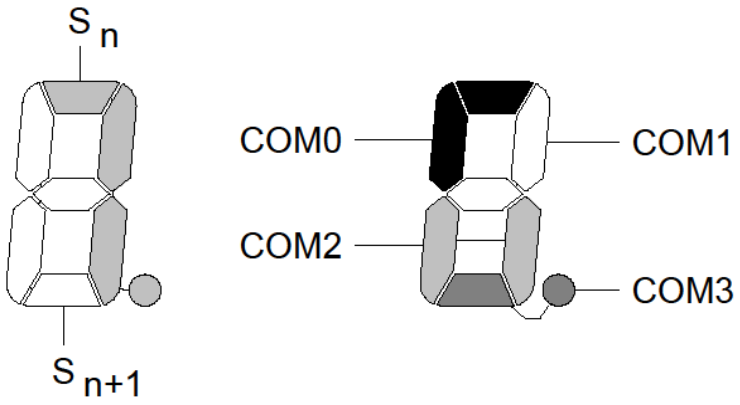


Figure 6: Quadplexed LCD Electrode Connections

The following description assumes the numeral ‘6’ is displayed by the digit formed using segments S_n and S_{n+1}. Segment and common signal connections are made to the digit as shown in Figure 6. It is necessary to apply the select or deselect voltage to the S_n and S_{n+1} pins according to Table 3.11 at the timing of the common signals COM0 to COM3 to form the digit ‘6’.

Table 3.11 LCD Select and Deselect Voltages (COM0 through COM3)

Segment / Common	S_n	S_{n+1}
COM0	Select	Select
COM1	Deselect	Select
COM2	Select	Select
COM3	Select	Select

Based upon Table 3.11, the display data memory locations (COM0 through COM3) corresponding to Sn and Sn+1 must contain ‘1’, except COM2 position Sn must contain ‘0’.

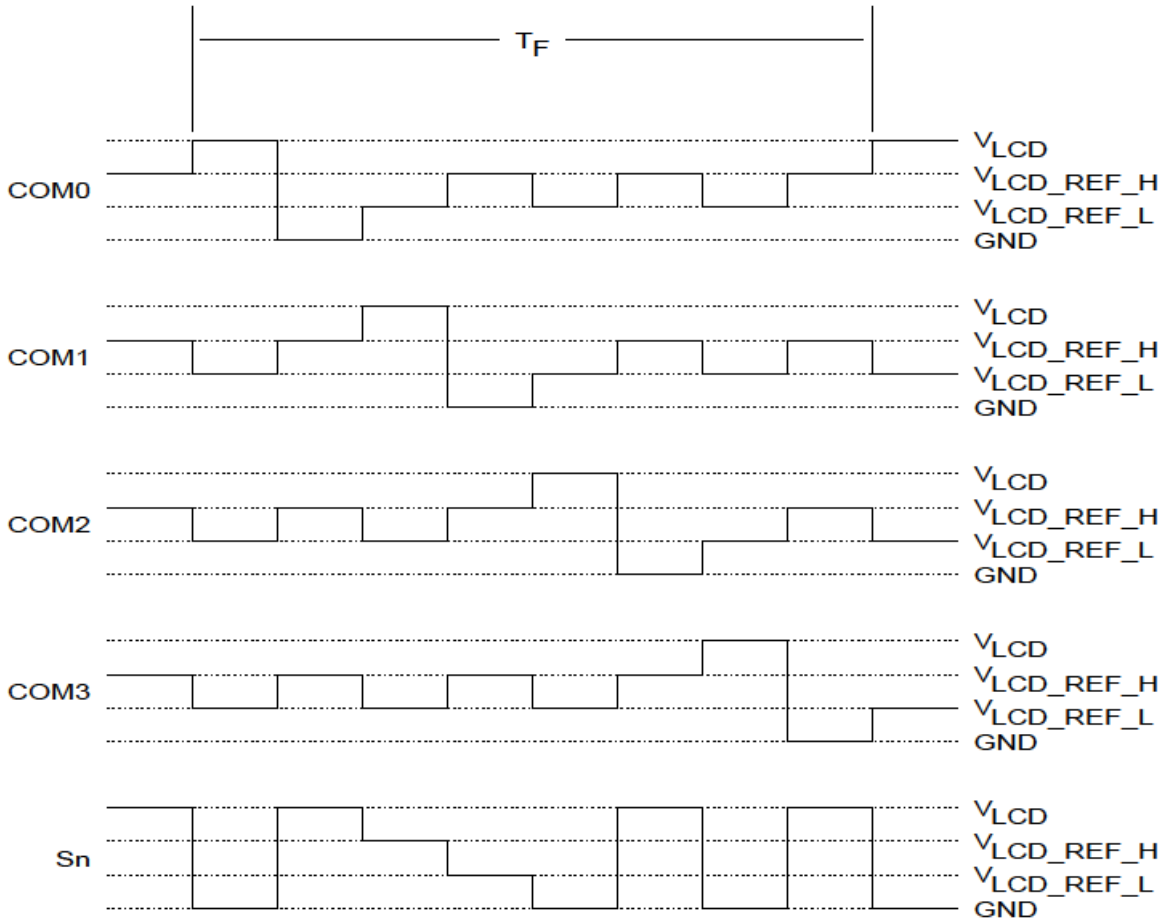


Figure 7: Quadplexed LCD Drive Waveform Example

Figure 7 shows the LCD drive waveforms between the Sn signal and each common signal. When the select voltage is applied to Sn at the timing of COM0, an alternate rectangle waveform, +VLCD/-VLCD, is generated to turn on the corresponding LCD segment. This alternate waveform also occurs at the timing of COM2 and COM3 to turn on the corresponding segments. It does not occur at the timing of COM1, so the alternate waveform is not generated and the corresponding segment remains off.

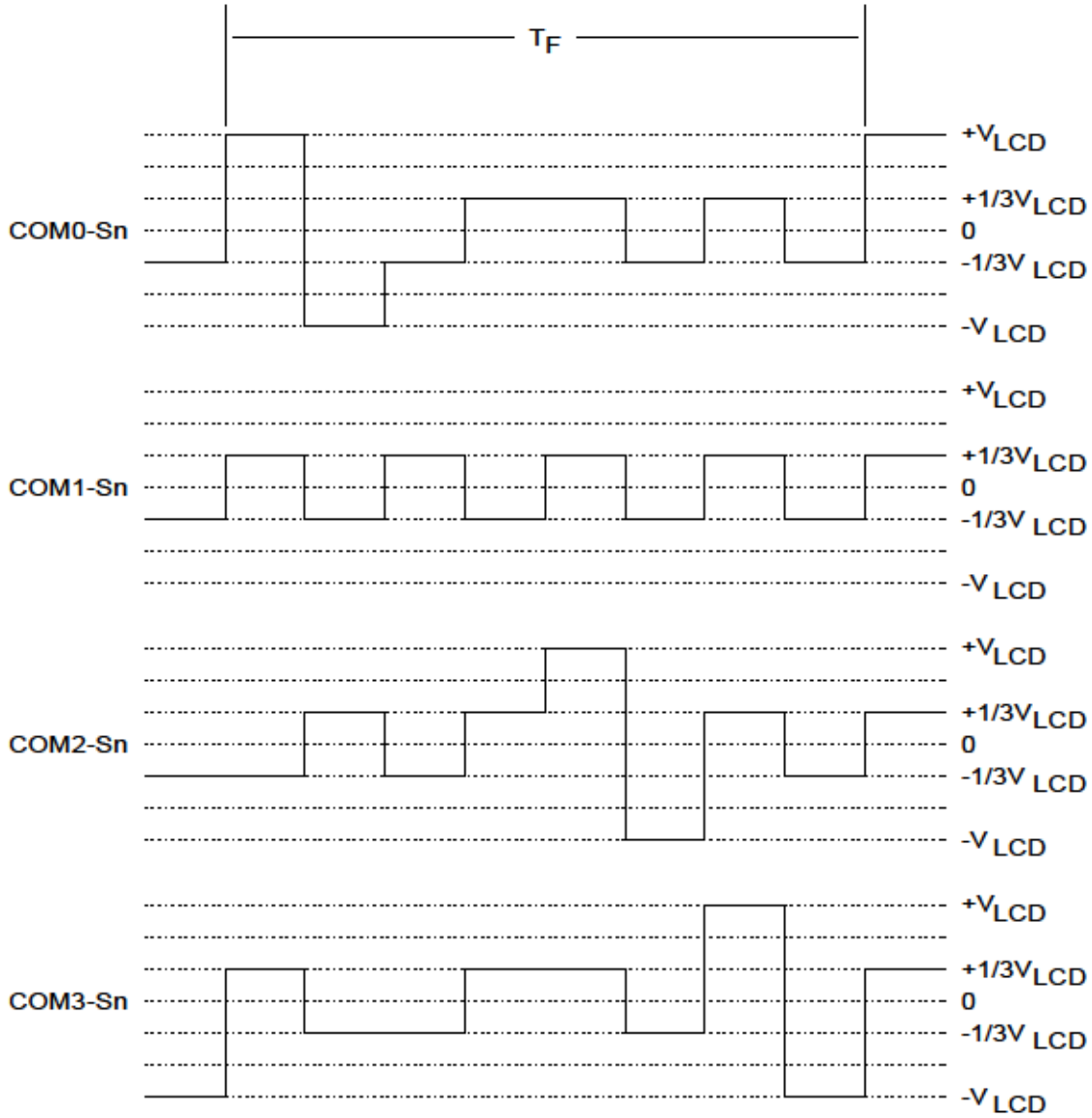


Figure 8: Quadplexed LCD Drive Waveform Example

3.1.12.2 LCD Drive Voltages

The ASIC generates an LCD drive power supply using internal voltage boosting and a voltage divider. When enabled, a regulator/charge pump generates a V_{LCD} voltage of 4.5 V using the available battery voltage or USB supply voltage. The divider then generates LCD_REF_H, and LCD_REF_L according to Table 3.12.

Table 3.12 LCD 1/3 Bias Drive Voltage Levels	
Bias Method / LCD Drive Voltage Pin	1/3 Bias Method
VLCD	VLCD
LCD_REF_H	(2/3) VLCD
LCD_REF_L	(1/3) VLCD

The LCD charge pump requires an external non-polarized charge pump capacitor between pins LCD_CAP_P and LCD_CAP_N. External low-leakage decoupling capacitors are also required on LCD drive voltage pins VLCD, LCD_REF_H, and LCD_REF_L.

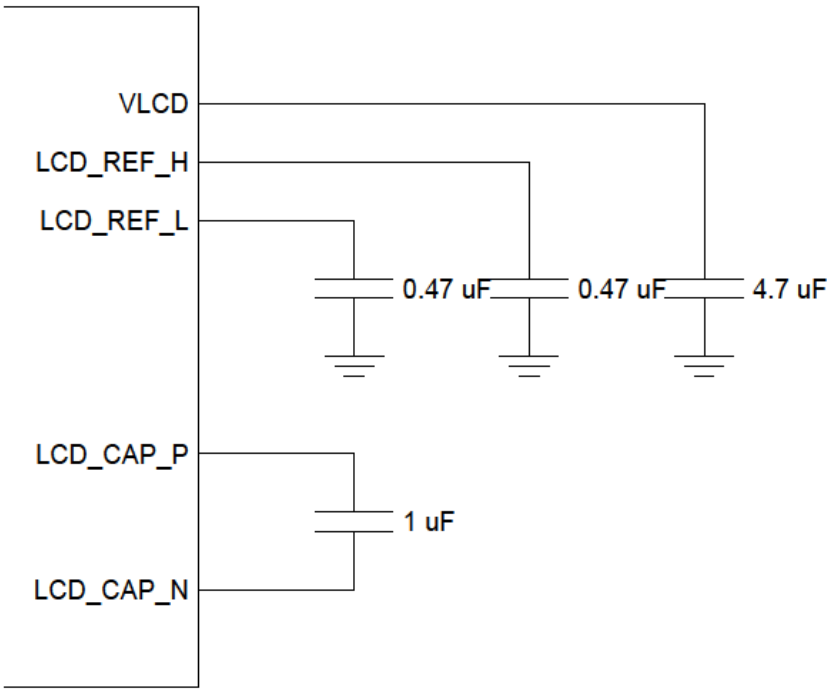


Figure 9: Connecting Pins for LCD Charge Pump and Divider

3.1.12.3 LCD Registers

The following registers control the behavior of the LCD pins:

Table 3.13 LCD Registers			
Address	**Register Name**	**Description**	**Reset Value**
0x500180A0	COM0	COM0 Display Data Register	0xxxxx
0x500180A4	COM1	COM1 Display Data Register	0xxxxx
0x500180A8	COM2	COM2 Display Data Register	0xxxxx
0x500180AC	COM3	COM3 Display Data Register	0xxxxx
0x500180B0	LCDC1	LCD Control Register 1	0x31
0x500180B1	LCDC2	LCD Control Register 2	0xAA

COM0 Display Data Register							
COM0		0x500180A0			0xxxxx		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
S7COM0	S6COM0	S5COM0	S4COM0	S3COM0	S2COM0	S1COM0	S0COM0
S15COM0	S14COM0	S13COM0	S12COM0	S11COM0	S10COM0	S9COM0	S8COM0
S23COM0	S22COM0	S21COM0	S20COM0	S19COM0	S18COM0	S17COM0	S16COM0
Reserved	Reserved	S29COM0	S28COM0	S27COM0	S26COM0	S25COM0	S24COM0
MSB							LSB

Bit31-30 Reserved

Bitn Sn/COM0 LCD display element state (0 <= n <= 29).
0 = Display element blank
1 = Display element visible

COM1 Display Data Register							
COM1		0x500180A4			0xxxxx		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
S7COM1	S6COM1	S5COM1	S4COM1	S3COM1	S2COM1	S1COM1	S0COM1
S15COM1	S14COM1	S13COM1	S12COM1	S11COM1	S10COM1	S9COM1	S8COM1
S23COM1	S22COM1	S21COM1	S20COM1	S19COM1	S18COM1	S17COM1	S16COM1
Reserved	Reserved	S29COM1	S28COM1	S27COM1	S26COM1	S25COM1	S24COM1
MSB							LSB

Bit31-30 Reserved

Bitn Sn/COM1 LCD display element state (0 <= n <= 29).
0 = Display element blank
1 = Display element visible

COM2 Display Data Register							
COM2		0x500180A8			0xxxxx		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
S7COM2	S6COM2	S5COM2	S4COM2	S3COM2	S2COM2	S1COM2	S0COM2
S15COM2	S14COM2	S13COM2	S12COM2	S11COM2	S10COM2	S9COM2	S8COM2
S23COM2	S22COM2	S21COM2	S20COM2	S19COM2	S18COM2	S17COM2	S16COM2
Reserved	Reserved	S29COM2	S28COM2	S27COM2	S26COM2	S25COM2	S24COM2
MSB							LSB
Bit31-30 Reserved Bitn Sn/COM2 LCD display element state (0 <= n <= 29). 0 = Display element blank 1 = Display element visible							

COM3 Display Data Register							
COM3		0x500180AC			0xxxxx		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
S7COM3	S6COM3	S5COM3	S4COM3	S3COM3	S2COM3	S1COM3	S0COM3
S15COM3	S14COM3	S13COM3	S12COM3	S11COM3	S10COM3	S9COM3	S8COM3
S23COM3	S22COM3	S21COM3	S20COM3	S19COM3	S18COM3	S17COM3	S16COM3
Reserved	Reserved	S29COM3	S28COM3	S27COM3	S26COM3	S25COM3	S24COM3
MSB							LSB
Bit31-30 Reserved Bitn Sn/COM3 LCD display element state (0 <= n <= 29). 0 = Display element blank 1 = Display element visible							

LCD Control Register 1							
LCDC1		0x500180B0			0x39		
R/W	Reserved	R/W	Reserved	Reserved	Reserved	Reserved	R/W
EN_SIM	EN_LCD	NCOM[1]	NCOM[0]	LCDTRIM[1]	LCDTRIM[0]	FRATE[1]	FRATE[0]
MSB							LSB

Bit 7 **EN_SIM:** Simultaneous enable for LCD charge pump and display
0 = Delay LCD screen enable by 180 ms to allow charge pump to settle
1 = Simultaneous enable for LCD screen and charge pump

Bit6 **EN_LCD:** Display enable
0 = Display OFF (Display blanked)- lcd_cp_clk disabled
1 = Display ON – lcd_cp_clk enabled

Bit5-4 **NCOM[1:0]:** Number of LCD backplanes / duty control
00 = 1 backplane (COM3, COM2 and COM1 carry same drive signal as COM0)
01 = 2 backplanes (COM3 carries same signal as COM0 and COM2 carries same signal as COM1)
10 = 3 backplanes (COM3 carries same signal as COM0
11 = 4 backplanes (all of COM0-COM3 carry orthogonal signals)

Bit3-2 **LCDTRIM[1:0]:** VLCD Output Voltage Trim (for contrast control)
Note: Values stated below are without LCD screen. Actual voltages depend on LCD screen size and equivalent capacitance and can vary by 100 mV

Frame rate = 32 Hz
00 = 3.34 V
01 = 3.77 V
10 = 4.17 V (default)
11 = 4.54 V

Frame rate = 64 Hz
00 = 3.24 V
01 = 3.65 V
10 = 4.03 V (default)
11 = 4.36 V

Bit1-0 **FRATE[1:0]:** LCD frame rate setting
00 = 64 Hz
01 = 32 Hz (default)
10 = 16 Hz
11 = 8 Hz

LCD Control Register 2							
LCDC2		0x500180B1			0xAA		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
t1[1]	t1[0]	t2[1]	t2[0]	t3[1]	t3[0]	t4[1]	t4[0]
MSB							LSB

Bit7-0 **tx[1:0]:** Reserved for LCD timing control (internal to device manufacturer).

3.1.13 GPIO

Each GPIO may be used as a digital output or a digital input. Additionally, several pins have additional functions. Some are connected to serial interface hardware and PORT C pins 7 to 4 can be connected to a PWM circuit. Each port B through E (port A is not used), has a register which contains the data output to or input through the associated pins. There is a read enable and output enable register for each port and there are two registers per port, which control the activity of a pull up or pull down function to ensure the pin is in a known state in input mode even if the driving source is floating (see Section 3.1.13.1 for exceptions). There are registers that allow configuring each pin to set an interrupt on a state change (except port E).

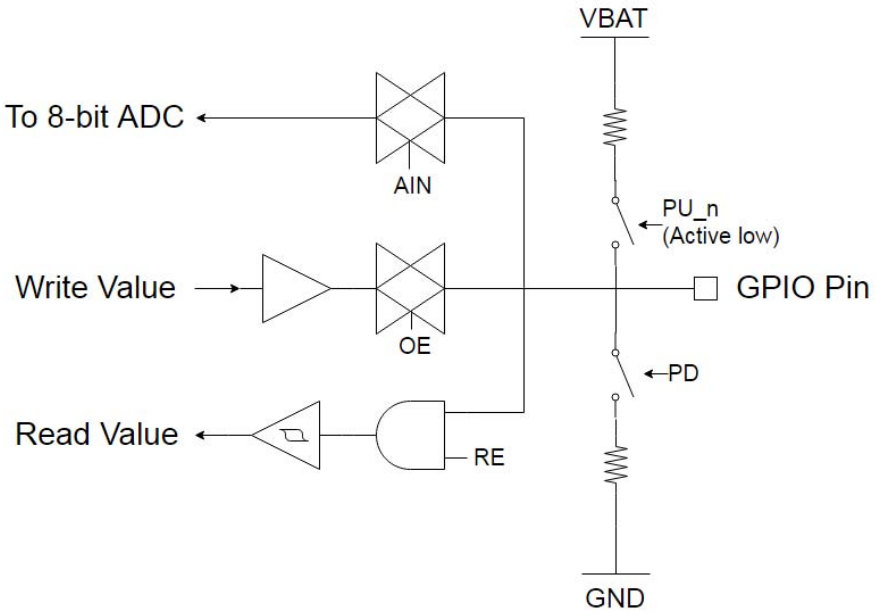


Figure 10: GPIO Functional Diagram

3.1.13.1 Notes to GPIO operation and configuration

Following are certain important aspects of GPIO operation and configuration

1. Pins PD0/SCL and pin PD1/SDA do not support pull-up and pull-down functionality in GPIO mode. These pins are recommended to be used for I2C operation when register PCONF bit MDI2C = 1. In I2C mode only, the on-chip pull-up resistance is configurable though register PCONF bits I2C_RT]1:0].
2. To read any pin on PORT C, one of the pin-change interrupts on a port C pin must be enabled. If none of the pin-change interrupts on PORT C is enabled, then PORT B must be read before PORT C.
3. To read any pin on PORT D, one of the pin-change interrupts on a port D pin must be enabled. If none of the pin-change interrupts on PORT D is enabled, then PORT B must be read before PORT C.

4. Port E does not support read, pull-up or pull-down functionality. Pins of port E (if part of packaging option chosen) are intended for use as power switches eliminating external supply-gating FETs on the PCB in certain cases. Pins of port E can be paralleled to source power to external devices, such as memory chips; each port E pin can source up to 10 mA of current.

Table 3.14 GPIO Port Map

Port	Bit	Pin	Function
B	7	L11	Digital GPIO
	6	L10	Digital GPIO
	5	K11	Digital GPIO
	4	K10	Digital GPIO
	3	J11	Digital GPIO
	2	J10	Digital GPIO
	1	H11	Digital GPIO
	0	H10	Digital GPIO
C	7	H9	Digital GPIO, optional PWM output (default PWM output)
	6	H8	Digital GPIO, optional PWM output
	5	G11	Digital GPIO, optional PWM output
	4	G10	Digital GPIO, optional PWM output
	3	G9	Digital GPIO, optional PWM output, digital real-time output A
	2	G8	Digital GPIO, optional PWM output, digital real-time output B
	1	G7	Digital GPIO, optional PWM output
	0	F8	Digital GPIO, optional PWM output
D	7	E9	Digital GPIO, UART RXD input
	6	E8	Digital GPIO, UART TXD output
	5	D10	SPI_CS-L output
	4	D11	SPI_CK output
	3	C10	SPI_MO output
	2	C11	SPI_MI input
	1	D9	SDA input/output
	0	D8	SCL output
E	2	E10	Digital GPIO, no interrupt capability
	1	F10	Digital GPIO, no interrupt capability
	0	F9	Digital GPIO, no interrupt capability

3.1.13.2 GPIO Registers

Table 3.15 summarizes the registers controlling the behavior of the GPIO pins:

Table 3.15 GPIO Registers

Address	Register Name	Description	Reset Value
0x50000078	PORTB	Port B I/O Register	0x00
0x50000079	PORTC	Port C I/O Register	0x00
0x5000007A	PORTD	Port D I/O Register	0x00
0x5000007B	PORTE	Port E I/O Register	0xx0
0x5000007C	PORTBOE	Port B Output Enable Register	0x00
0x5000007D	PORTCOE	Port C Output Enable Register	0x00
0x5000007E	PORTDOE	Port D Output Enable Register	0x00
0x5000007F	PORTEOE	Port E Output Enable Register	0xx0
0x50018096	PCONF	Port Configuration Register	0x00
0x5001808E	PCPWMCFG	Port C PWM Configuration Register	0x0x
0x50018084	PBINT	Port B Interrupt Enable Register	0xAA02
0x5001808C	PCINT	Port C Interrupt Enable Register	0x0300
0x50018094	PDINT	Port D Interrupt Enable Register	0x0000
0x50018080	PBPUN	Port B Pull up Enable Reg.	0x00
0x50018081	PBPD	Port B Pull down Enable Register	0x00
0x50018082	PBRE	Port B Read Enable Register	0xFF
0x50018088	PCPUN	Port C Pull up Enable Register	0x00
0x50018089	PCPD	Port C Pull down Enable Register	0x00
0x5001808A	PCRE	Port C Read Enable Register	0xFF
0x50018090	PDPUN	Port D Pull up Enable Register	0x00
0x50018091	PDPD	Port D Pull down Enable Register	0x00
0x50018092	PDRE	Port D Read Enable Register	0xFF

Detailed description of each register is shown below

Port B input and output register							
PORTB		0x50000078			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
MSB							LSB

Bit7-0 PB[7:0]: Port B register bits.
When reading, output of the digital Schmitt trigger receiver.
When writing, value driven by the CMOS driver to the pad.
0 = Pin state is ‘0’
1 = Pin state is ‘1’

Port C input and output register							
PORTC		0x50000079			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
MSB							LSB

Bit7-4 PC[7:0]: Port C register bits.
When reading, output of the digital Schmitt trigger receiver.
When writing, value driven by the CMOS driver to the pad.
0 = Pin state is read back as ‘0’
1 = Pin state is read back as ‘1’

Port D input and output register							
PORTD		0x5000007A			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
MSB							LSB

Bit7-0 PD[7:0]: Port D register bits.
When reading, output of the digital Schmitt trigger receiver.
When writing, value driven by the CMOS driver to the pad.
0 = Pin state is ‘0’
1 = Pin state is ‘1’

Port E input and output register							
PORTE		0x5000007B			0xx0		
Reserved	Reserved	Reserved	Reserved	Reserved	R/W	R/W	R/W
-	-	-	-	-	PE2	PE1	PE0
MSB							LSB
Bit7-3 *Reserved* Bit2-0 PORTE[2:0]: Port E register bits. When reading, output of the digital Schmitt trigger receiver. When writing, value driven by the CMOS driver to the pad. 0 = Pin state is ‘0’ 1 = Pin state is ‘1’							

Port B output enable register							
PORTBOE		0x5000007C			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PBOE7	PBOE6	PBOE5	PBOE4	PBOE3	PBOE2	PBOE1	PBOE0
MSB							LSB
Bit7-0 PBOE[7:0]: Port B output enable bits. 0 = Pin is not driven 1 = Pin is driven							

Port C output enable register							
PORTCOE		0x5000007D			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCOE7	PCOE6	PCOE5	PCOE4	PCOE3	PCOE2	PCOE1	PCOE0
MSB							LSB
Bit7-4 PCOE[7:0]: Port C output enable bits. 0 = Pin is not driven 1 = Pin is driven							

Port D output enable register							
PORTDOE		0x5000007E			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PDOE7	PDOE6	PDOE5	PDOE4	PDOE3	PDOE2	PDOE1	PDOE0
MSB							LSB
Bit7-0 PDOE[7:0]: Port D output enable bits. 0 = Pin is not driven 1 = Pin is driven							

Port E output enable register							
PORTEOE		0x5000007F			0xx0		
Reserved	Reserved	Reserved	Reserved	Reserved	R/W	R/W	R/W
-	-	-	-	-	PEOE2	PEOE1	PEOE0
MSB							LSB
Bit7-3 *Reserved* Bit2-0 PEOE[2:0]: Port E output enable bits. 0 = Pin is not driven 1 = Pin is driven							

Port B interrupt enable register							
PBINT		0x50018084			0xAA02		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PB3IE[1]	PB3IE[0]	PB2IE[1]	PB2IE[0]	PB1IE[1]	PB1IE[0]	PB0IE[1]	PB0IE[0]
PB7IE[1]	PB7IE[0]	PB6IE[1]	PB6IE[0]	PB5IE[1]	PB5IE[0]	PB4IE[1]	PB4IE[0]
MSB							LSB
Bit15-0 PbxIE[1:0]: Port B interrupt enable bits. For each pin x, 00 = interrupt disabled 01 = interrupt enabled on rising edge 10 = interrupt enabled on falling edge 11 = interrupt enabled on both edges							

Port C interrupt enable register							
PCINT		0x5001808C			0x0300		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PC3IE[1]	PC3IE[0]	PC2IE[1]	PC2IE[0]	PC1IE[1]	PC1IE[0]	PC0IE[1]	PC0IE[0]
PC7IE[1]	PC7IE[0]	PC6IE[1]	PC6IE[0]	PC5IE[1]	PC5IE[0]	PC4IE[1]	PC4IE[0]
MSB							LSB
Bit15-0 PcxIE[1:0]: Port C interrupt enable bits. For each pin x, 00 = interrupt disabled 01 = interrupt enabled on rising edge 10 = interrupt enabled on falling edge 11 = interrupt enabled on both edges							

Port D interrupt enable register							
PDINT		0x50018094			0x0000		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PD3IE[1]	PD3IE[0]	PD2IE[1]	PD2IE[0]	PD1IE[1]	PD1IE[0]	PD0IE[1]	PD0IE[0]
PD7IE[1]	PD7IE[0]	PD6IE[1]	PD6IE[0]	PD5IE[1]	PD5IE[0]	PD4IE[1]	PD4IE[0]
MSB							LSB
Bit15-0 PdxIE[1:0]: Port D interrupt enable bits. For each pin x, 00 = interrupt disabled 01 = interrupt enabled on rising edge 10 = interrupt enabled on falling edge 11 = interrupt enabled on both edges							

Port B pull up enable register							
PBPUN		0x50018080			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PBPUN7	PBPUN6	PBPUN5	PBPUN4	PBPUN3	PBPUN2	PBPUN1	PBPUN0
MSB							LSB
Bit7-0 PBPUN[7:0]: Port B pull up enable bits (active low). 0 = pull up enabled 1 = pull up disabled							

Port B pull down enable register							
PBPD		0x50018081			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PBPD7	PBPD6	PBPD5	PBPD4	PBPD3	PBPD2	PBPD1	PBPD0
MSB							LSB
Bit7-0 PBPD[7:0]: Port B pull down enable bits. 0 = pull down disabled 1 = pull down enabled							

Port C pull up enable register							
PCPU		0x50018088			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCPUN7	PCPUN6	PCPUN5	PCPUN4	PCPUN3	PCPUN2	PCPUN1	PCPUN0
MSB							LSB
Bit7-0 PCPUN[7:0]: Port C pull up enable bits (active low). 0 = pull up enabled 1 = pull up disabled							

Port C pull down enable register							
PCPD		0x50018089			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCPD7	PCPD6	PCPD5	PCPD4	PCPD3	PCPD2	PCPD1	PCPD0
MSB							LSB
Bit7-4 PCPD[7:0]: Port C pull down enable bits. 0 = pull down disabled 1 = pull down enabled							

Port D pull up enable register							
PDPU		0x50018090			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PDPUN7	PDPUN6	PDPUN5	PDPUN4	PDPUN3	PDPUN2	-	-
MSB							LSB
Bit7-0 PDPUN[7:2]: Port D pull up enable bits (active low) 0 = pull up enabled 1 = pull up disabled							

Port D pull down enable register							
PDPD		0x50018091			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PDPD7	PDPD6	PDPD5	PDPD4	PDPD3	PDPD2	-	-
MSB							LSB
Bit7-0 PDPD[7:2]: Port D pull down enable bits. 0 = pull down disabled 1 = pull down enabled							

Port B Read enable register							
PBRE		0x50018082			0xFF		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PBRE7	PBRE6	PBRE5	PBRE4	PBRE3	PBRE2	PBRE1	PBRE0
MSB							LSB
Bit7-0 PBRE[7:0]: Port B read enable bits. 0 = port cannot be read 1 = port can be read							

Port C Read enable register							
PCRE		0x5001808A			0xFF		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCRE7	PCRE6	PCRE5	PCRE4	PCRE3	PCRE2	PCRE1	PCRE0
MSB							LSB
Bit7-0 PCRE[7:0]: Port C read enable bits. 0 = port cannot be read 1 = port can be read							

Port D Read enable register							
PDRE		0x50018092			0xFF		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PDRE7	PDRE6	PDRE5	PDRE4	PDRE3	PDRE2	PDRE1	PDRE0
MSB							LSB
Bit7-0 PDRE[7:0]: Port D read enable bits. 0 = port cannot be read 1 = port can be read							

Pin configuration register							
PCONF		0x50018096			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
I2C_RT1	I2C_RT0	UPSWAP	UTXPOL	MDSPI	MDI2C	MDUART	MDSPICS
MSB							LSB

Bit7-6 **I2C_RT[1:0]**: I²C Resistor Trim.
00 = Open
01 = 1k
10 = 10k
11 = 100k

Bit5 **UPSWAP:** UART Pins Swap Bit
0 = Pins are not swapped
1 = Pins (TXD and RXD) are swapped

Bit4 **UTXPOL:** UART signals polarity
0 = normal polarity
1 = inverted polarity

Bit3 **MDSPI:** SPI mode enable
0 = GPIO
1 = SPI mode

Bit2 **MDI2C:** I²C mode enable
0 = GPIO
1 = I²C mode

Bit1 **MDUARTE:** UART mode enable
0 = GPIO
1 = UART mode

Bit0 **MDSPICS:** Enable software control over PIO_CS1-L
0 = Hardware controls rise and fall of PIO_CS1-L when MDSPI = 1.
1 = Software controls rise and fall of PIO_CS1-L when MDSPI = 1 (through writing port D).

Port C PWM configuration register							
PCPWMCFG		0x5001808E			0x80		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCPWM7	PCPWM6	PCPWM5	PCPWM4	-	-	-	-
MSB							LSB

Bit7-0 PCPWM[7:4]: Connect Port C to PWM
0 = GPIO mode
1 = PWM mode

For minimal standby current, pull-up for PWM output pin defaulting to a low state should be disabled while in the standby state.

3.1.14 SPI Interface

The digital engine supports an SPI interface for system expansion. The microcontroller serves as the master. A GPIO pin can be used to gate power to an external SPI device. The SPI signal pins, SPI_MI/PD2, SPI_MO/PD3, and SPI_CK/PD4, may also be configured as GPIO pins.

The Serial Peripheral Interface (SPI) is a synchronous full-duplex serial interface. It communicates in master/slave mode where the master initiates the data transfer. For this SPI interface, only the master mode is implemented.

Features of this SPI interface are:

- Compatible with an Industry Standard SPI interface.
- Four-byte deep reception FIFO
- Four-byte deep transmission FIFO
- Interrupt upon events related to transmission, reception, and error:
  - Write Collision
  - Transmission FIFO full and empty
  - Reception FIFO full and empty

Table 3.16 SPI (Master) Signals

Pin Name	Description
SPI_MO/PD3	Serial Peripheral Interface 1 Data Output – Serial data output from the digital engine (master) to an external device, such as EEPROM. This signal pin may alternatively be used as GPIO pin GPIOD3.
SPI_MI/PD2	Serial Peripheral Interface 1 Data Input – Serial data input to the digital engine (master) from an external device, such as EEPROM. This signal pin may alternatively be used as GPIO pin GPIOD2.
SPI_CK/PD4	Serial Peripheral Interface 1 Clock – Serial data clock output from the digital engine to an external device, such as EEPROM. This signal pin may alternatively be used as GPIO pin GPIOD4.
SPI_CS-L/PD5	Serial Peripheral Interface Chip Select – Active-low chip select output from the digital engine used for enabling SPI communication with an external device, such as EEPROM. Normally implemented with GPIOD5 pin. Signal should be maintained high when communication is not intended, including the reset condition. This signal pin may alternatively be used as GPIO pin GPIOD5.

3.1.14.1 SPI Operation

Only the master mode is implemented. The microcontroller configures the clock frequency and generates the serial clock (SPI_CK) for the interface. Data transfer is synchronous with SPI_CK. The SPI is full duplex; data is transmitted and received simultaneously. Information is sent to the slave via the SPI_MO pin and received via the SPI_MI pin. The CPOL and CPHA bits in the SPI control register determine when to sample the data.

When CPOL=0, the base value of the clock is logic '0'. In this case, if CPHA=0, data is captured on the rising edge of SPI_CK and data is propagated on the falling edge of SPI_CK. For CPHA=1, data is captured on the falling edge of SPI_CK and data is propagated on the rising edge of SPI_CK.

If CPOL=1, the base value of the clock is logic ‘1’. In this case, if CPHA=0, data is captured on the falling edge of SPI_CK and data is propagated on the rising edge of SPI_CK. For CPHA=1, data is captured on the rising edge of SPI_CK and data is propagated on the falling edge of SPI_CK.

A timing diagram is shown below.

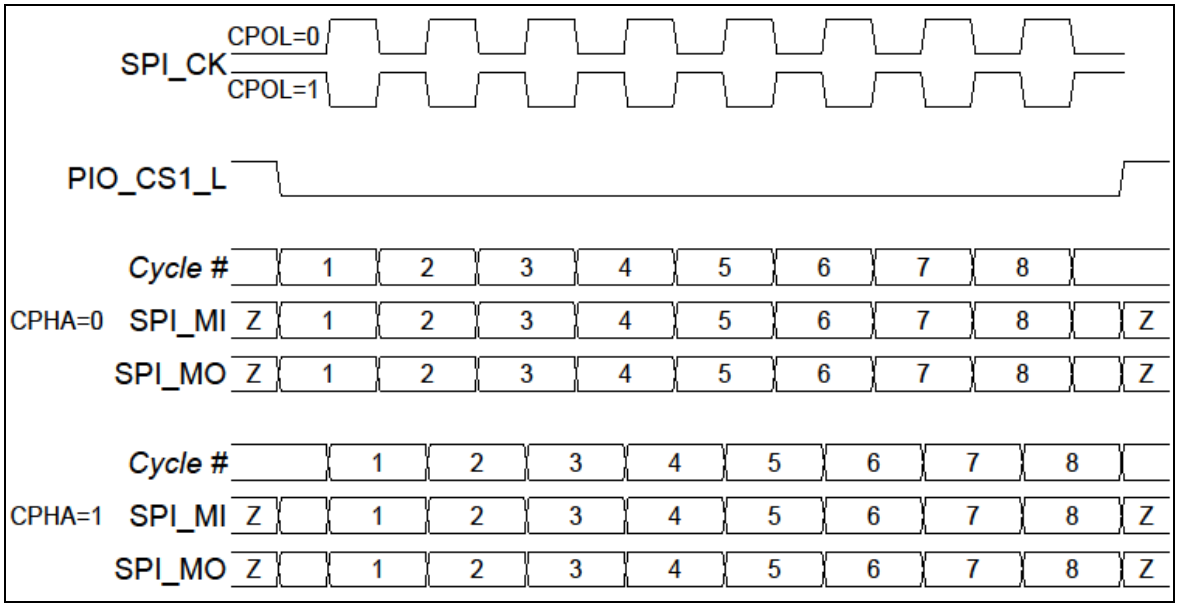


Figure 11: SPI Timing Diagram

After the desired configuration is set through configuration registers, a transfer is initiated by writing to the SPI Data Register (SPDR). The data is input into a 4-deep FIFO before being transmitted. When the data is transmitted, the slave also simultaneously transmits data for the ASIC to receive. The received data is stored in a separate 4-deep FIFO. The received data is accessed by reading the SPDR register.

To operate properly, the following steps must be followed:

- Configure and enable the SPI: Select if the interrupt is enabled, set the polarity, the phase, and the clock divider.
- Enable the interrupt (if required).
- Process the interrupt (if required) and detect the reason for the interrupt (error, transmission or reception related and act accordingly.

3.1.14.2 SPI Registers

The following registers are defined in the SPI interface:

Table 3.17 SPI Registers

Address	Register Name	Description	Reset Value
0x5000001C	SPCR	SPI Control Register	0x10
0x5000001D	SPSR	SPI Status Register	0x00
0x5000001E	SPDR	SPI Data Register	0x00
0x5000001F	SPER	SPI Extension Register	0x00

SPI Control Register							
SPCR		0x5000001C			0x10		
R/W	Reserved	Reserved	R	R/W	R/W	R/W	R/W
SINTE	-	-	MSTR	CPOL	CPHA	SCKSTD1	SCKSTD0
MSB							LSB

Bit7 **SINTE:** SPI Interrupt enable:
0 = Disabled
1 = Enabled
Bit4 **MSTR:** Master Mode Select bit:
SPI always in master mode; Set to logic “1”.
Bit3 **CPOL:** SPI clock polarity:
0 = Base value of clock is zero
1 = Base value of clock is one
Bit2 **CPHA:** SPI clock phase:
0 = Data captured on clock transition from base, propagated on transition to base
1 = Data captured on clock transition to base, propagated on transition from base
Bit1-0 **SCKSTD[1:0]:** SPI standard clock divider selection
Refer to SPER register for system clock

SPI Status Register							
SPSR		0x5000001D			0x00		
R/W	R/W	Reserved	Reserved	R/W	R/W	R/W	R/W
SINTF	SWCOL	-	-	STXFF	STXFE	SRXFF	SRXFE
MSB							LSB

Bit7 **SINTF:** SPI interrupt flag:
0 = Interrupt not asserted
1 = Interrupt asserted
Bit6 **SWCOL:** SPI write collision set when SPDR register written to while transmit FIFO is full:
Bit3 **STXFF:** SPI transmit FIFO full:
0 = Transmit FIFO not full
1 = Transmit FIFO full
Bit2 **STXFE:** SPI transmit FIFO empty:
0 = Transmit FIFO not empty
1 = Transmit FIFO empty
Bit1 **SRXFF:** SPI reception FIFO full:
0 = Reception FIFO not full
1 = Reception FIFO full
Bit0 **SRXFE:** SPI reception FIFO empty:
0 = Reception FIFO not empty
1 = Reception FIFO empty

SPI Data Register							
SPDR		0x5000001E			0xXX		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
SPID7	SPID6	SPID5	SPID4	SPID3	SPID2	SPID1	SPID0
MSB							LSB
Bit7-0 **SPID[7:0]**: SPI data, both transmitted and received.							

SPI Extension Register							
SPER		0x5000001F			0x00		
R/W	R/W	Reserved	Reserved	R/W	R/W	Reserved	Reserved
SICNT1	SINCT0	-	-		SPE	SCKEXT1	SCKEXT0
MSB							LSB

Bit7-6 **SICNT[1:0]:** SPI Interrupt Counter Bits:
00 = SINTF set after every completed transfer
01 = SINTF set after two completed transfers
10 = SINTF set after three completed transfers
11 = SINTF set after four completed transfers

Bit2 **SPE:** SPI enable:
0 = SPI module disabled
1 = SPI module enabled

Bit1-0 **SCKEXT[1:0]:** SPI extended clock divider

SCKSTD	SCKEXT	Result Clock Divider
00	00	= System Clock/2
01	00	= System Clock/4
10	00	= System Clock/16
11	00	= System Clock/32
00	01	= System Clock/8
01	01	= System Clock/64
10	01	= System Clock/128
11	01	= System Clock/256
00	10	= System Clock/512
01	10	= System Clock/1024
10	10	= System Clock/2048
11	10	= System Clock/4096
xx	11	= Reserved

3.1.15 UART Interface

The ASIC includes a Universal Asynchronous Receiver Transmitter (UART) module. The main characteristics are as follows:

- Interrupt available for transmission, reception, and error events
- Reception timeout timer
- Programmable break reception and transmission
- Programmable parity with “stick” parity option
- Number of bits selectable from 5 to 8
- Number of stop-bits selectable as 1, 1½ , or 2
- Programmable loop-back
- Swappable TXD and RXD
- Transmitter Polarity selection

The UART protocol requires two signals (TXD and RXD) assigned to pins TXD and RXD respectively.

Table 3.18 UART Signals

Pin Name	Description
TXD/PD6	UART Transmitted Data Output – Transmitted data output from the digital engine’s UART to an external device, such as an RF module. The signal marking level is high and the spacing level low. This signal pin may alternatively be used as GPIO pin GPIOD6.
RXD/PD7	UART Received Data Input – Received data output to the digital engine’s UART from an external device, such as an RF module. The signal marking level is high and the spacing level low. This signal pin may alternatively be used as GPIO pin GPIOD7.

3.1.15.1 UART Operation

Pins TXD and RXD are allocated for UART communication and must be configured by setting bit MDUARTE in register PCONF

- Select whether pins are to be used in normal configuration or swapped and select polarity. Selection is made via the UPSWAP and UTXPOL bits of Pin configuration register PCONF.
- Define the following parameters using the UART Line Control Register UARTLCTRL:
  - Loop back: If testing, connects the output to the input.
  - Break enable: Pull output down while enabled.
  - Stick parity: Forces parity to stable value.
  - Even/Odd parity selection and enable: Select no, even, or odd parity.
  - Number of stop bits: Select 1, 1½ , or 2 stop bits.

- Data size: Select 5, 6, 7, or 8 data bits.

- Define baud rate using the UART Baud Rate Divider Register UARTDIV. The baud rate is calculated as follows:

$$Baud = \frac{Fclk}{16 * (UARTDIV + 1)}$$

- Enable the UART using UART Enable Register UARTCTRL1 and its interrupt using UART Interrupt Control Register UARTICTCL. The UART may generate an interrupt for events related to:
  - Transmission completed.
  - Reception: Timeout of ≈40 bit-times without reception, and data received (one or three bytes received, programmable).
  - Errors detected: Framing error, parity error, and overrun error.
  - Break signal detected (received).

3.1.15.2 UART Registers

The following registers are defined in the UART interface:

Table 3.19 UART Registers

Address	Register Name	Description	Reset Value
0x50000010	UARTDATA	UART Data Register	0x00
0x50000011	UARTICTRL	UART Interrupt Control Register	0x00
0x50000012	UARTLCTRL	UART Line Control Register	0x00
0x50000013	UARTEN	UART Enable Register	0x00
0x50000014	UARTLSTAT	UART Line Status	0x00
0x50000016-0x50000017	UARTDIV	UART Baud Rate Divider	0x0000

UART Data Register							
UARTDATA		0x50000010			0xXX		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
UARTD7	UARTD6	UARTD5	UARTD4	UARTD3	UARTD2	UARTD1	UARTD0
MSB							LSB
Bit7-0 **UARTD[7:0]**: UART data, both transmitted and received.							

UART Interrupt Control Register							
UARTICTCL		0x50000011			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
UISTTS3	UISTTS2	UISTTS1	UISTTS0	UTOUTIEN	URXERREN	UTXIEN	URXIEN
MSB							LSB

Bit7-4 **UISTTS[3:0]:** UART Interrupt status:
0001 = No Interrupt asserted
0010 = Transmission completed
0100 = Data received
0110 = Reception error
1100 = Reception timeout (≈40 bit-time)

Bit3 **UTOUTIEN:** UART time-out interrupt enable bit:
0 = Disabled
1 = Enabled

Bit2 **URXERREN:** UART reception error interrupt enable bit:
0 = Disabled
1 = Enabled

Bit1 **UTXIEN:** UART transmission completed interrupt enable bit:
0 = Disabled
1 = Enabled

Bit0 **URXIEN:** UART reception interrupt enable bit:
0 = Disabled
1 = Enabled

UART Line Control Register							
UARTLCTCL		0x50000012			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ULOOPEN	UBREAKEN	USTICKEN	UPARITY	UPAREN	USTOP	USIZE	USIZE
MSB							LSB

Bit7 **ULOOPEN:** UART loop back status:
0 = UART loop back disabled
1 = UART loop back enabled
Bit6 **UBREAKEN:** UART break enable:
0 = Disabled
1 = Enabled
Bit5 **USTICKEN:** UART stick parity enable bit:
0 = Disabled
1 = Enabled
Bit4 **UPARITY:** UART parity bit:
0 = Odd parity
1 = Even parity
Bit3 **UPAREN:** UART parity enable bit:
0 = Disabled
1 = Enabled
Bit2 **USTOP:** UART stop bit:
0 = One stop bit
1 = 1.5 stop bits for 5-bit transmission, otherwise 2 stop bits
Bit1-0 **USIZE:** UART transmission size:
00 = 5-bit data
01 = 6-bit data
10 = 7-bit data
11 = 8-bit data

UART Enable Register							
UARTCTRL1		0x50000013			0x00		
Reserved	Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W
-	-	-	-	UARTEN	URXFS	UTXFRST	URXFRST
MSB							LSB

Bit3 **UARTEN:** UART enable:
0 = Disabled
1 = Enabled
Bit2 **URXFS:** UART RX FIFO interrupt level:
0 = UART interrupts after one byte received
1 = UART interrupts after three bytes received
Bit1 **UTXFRST:** UART transmission FIFO reset bit:
0 = TX FIFO not reset
1 = TX FIFO reset
Bit0 **URXFRST:** UART reception FIFO reset bit:
0 = RX FIFO not reset
1 = RX FIFO reset

UART Line Status Register							
UARTSTATUS		0x50000014			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
UERR	UTXEMPTY	UTXFFEMPTY	UBREAKINT	UFRMERR	UPRTYERR	UOVRUNERR	UDTRDY
MSB							LSB

Bit7 **UERR:** UART error:
0 = No error
1 = Error in UART

Bit6 **UTXEMPTY:** UART transmission empty:
0 = Transmitter not empty
1 = Transmitter empty

Bit5 **UTXFFEMPTY:** UART transmission FIFO empty:
0 = TX FIFO not empty
1 = TX FIFO empty

Bit4 **UBREAKINT:** UART break interrupt:
0 = No break interrupt
1 = Break interrupt

Bit3 **UFRMERR:** UART framing error:
0 = No framing error
1 = Framing error

Bit2 **UPRTYERR:** UART overrun error:
0 = No parity error
1 = Parity error

Bit1 **UOVRUNERR:** UART transmission size:
0 = No overrun error
1 = Overrun error

Bit0 **UDTRDY:** UART data ready:
0 = No data ready (reception)
1 = Data ready (reception)

UART Baud Rate Divider Register (16-bit)							
UARTDIV		0x50000016			0x0000		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
UDIV7	UDIV6	UDIV5	UDIV4	UDIV3	UDIV2	UDIV1	UDIV0
UDIV15	UDIV14	UDIV13	UDIV12	UDIV11	UDIV10	UDIV9	UDIV8
MSB							LSB

Bit15-0 **UDIV[15:0]:** UART clock divider

3.1.16 I²C Interface

The digital engine supports an I²C interface for system expansion. The pins are to include a pull-up capability that can be software enabled or disabled. The I2C signal pins, SCL/PD0 and SDA/PD1, may also be configured as GPIO pins. The MDI2C bit must be set in the PCONF register to enable I²C operation.

The main characteristics of the implementation are:

- Support for slave and multi-master modes
- General call support
- 7-bit or 10-bit address
- Address masking

Table 3.20 I2C Signals

Pin Name	Description
SDA/PD1	I2C Serial Data I/O – I2C data signal from the digital engine to external devices. Internal pull-up can be enabled or disabled under software control. This signal pin may alternatively be used as GPIO pin GPIOD1.
SCL/PD0	I2C Serial Clock Output – I2C clock output from the digital engine to external devices. Internal pull-up can be enabled or disabled under software control. This signal pin may alternatively be used as GPIO pin GPIOD0.

3.1.16.1 I²C Operation in Slave Mode

In order to configure Slave Mode, the following steps must be carried out:

- Select the peripheral as slave. Selection is made by clearing the I²C Master/Slave bit IMS of I²C Control Register 2 I2CCTRL2.
- Select the address size. Selection is made by the configuring the I²C selection of address size bit ISTPSIZE of I²C Control Register 1 I2CCTRL1.
- Load the address into I²C Address 0 Register I2CADDR0 and I²C Address 0 Register I2CADDR1.
- Select if general call is to be supported. Selection is made by configuring the general call address bit IGC of I²C Control Register 1 I2CCTRL1.
- Select address masking, if required. If required, the peripheral provides a 5-bit address mask, in I²C Control Register 2 I2CCTRL2, for the lower 5 address bits. Each bit masks the corresponding bit in address comparison when set.
- Enable the interface and its interrupts if used. Enabling is controlled by configuring the I2C Enable bit IEN in I²C Control Register 2 I2CCTRL2.
- Define the interrupt handlers if required.

The several phases of the slave side of the communications are described in the following paragraphs:

3.1.16.1.1 I²C Access Sequence, Slave Mode

I²C addresses are always prefixed by START or Repeated Start condition.

3.1.16.1.1.1 7-bit Address Mode

The slave, once enabled, waits for an I²C Start condition to happen. Once a Start condition is selected, the slave shifts in the next 8 bits into an internal shift register and the following actions take place:

- The IBUF bit is set.
- If the address contained in the internal register matches the one from the register. I2CADDR0, the interface sends back an ACK and an interrupt is asserted.
- At this time, the application must read the I2CSTATUS register and check the IADDRR and IRWBUSY bits.
- The IADDRR should be ‘1’ (indicating that an address has been received).
- The IRWBUSY bit indicates if the operation is a write (‘1’) or a read (‘0’).
- After reading these bits, the application must read the I2CDATA register to clear the buffer.
- If IBUFF is set before receiving the address or IRBUFOVL is set when receiving the address, then the slave will send NACK and issue an error interrupt to notify the application of these errors.

3.1.16.1.1.2 10-bit Address Mode

Two address-byte receptions are required in this mode. The first byte consists of the following sequence:

```
'1 1 1 1 0 A[9] A[8] 0'
```

Where A[9:8] are the upper two bits of the I²C address.

The last bit, R/W (1=read, 0=write), must be 0 so the slave can receive another address byte. If the upper two address bits match, then the Slave sends an ACK and asserts an interrupt.

At this point, the application must read the I2CSTATU register to check IADDRR and IRWBUSY bits, which should be 1 (address byte) and 0 (write operation). The application then needs to read the I2CDATA register to clear the buffer.

The second byte contains address bits A[7:0]. In the same fashion, if the lower 8 address bits match, then the slave sends the ACK and asserts an interrupt.

The application reads the I2CSTATU register to check IADDRR bit, which should be logic 1 (address byte). The application then needs to read the I2CDATA register to clear the buffer.

If it is an I²C read access, then after two address-byte receptions, the slave shall receive a Repeated Start condition and then the first address byte again with last bit, R/W (1=read, 0=write), being 1. The slave sends the ACK bit and asserts an interrupt.

The application reads I2CSTATUS register to check IBUFF and IRWBUSY bits, which should be 1 (address byte) and 1 (read operation). The application then needs to read the I2CDATA register to clear the buffer.

Figure 15 illustrates a 10-bit address mode receiving timing waveform.

3.1.16.1.2 I²C Access Sequence, Write Data Phase

If the received R/W bit (1=read, 0=write) is 0, it is an I²C write access and the slave remains in receiving mode. Every time the slave shifts in a byte, it sends the ACK bit as long as the IBUFF bit is cleared before receiving the data and the IRBUFOVL bit is cleared when receiving the data.

In slave mode, the peripheral asserts an interrupt after receiving each byte from the I²C bus. The application needs to read the I2CSTATUS register to check the status and then the I2CDATA register to fetch the data. The write data phase is concluded when detecting a Stop or Repeated Start condition.

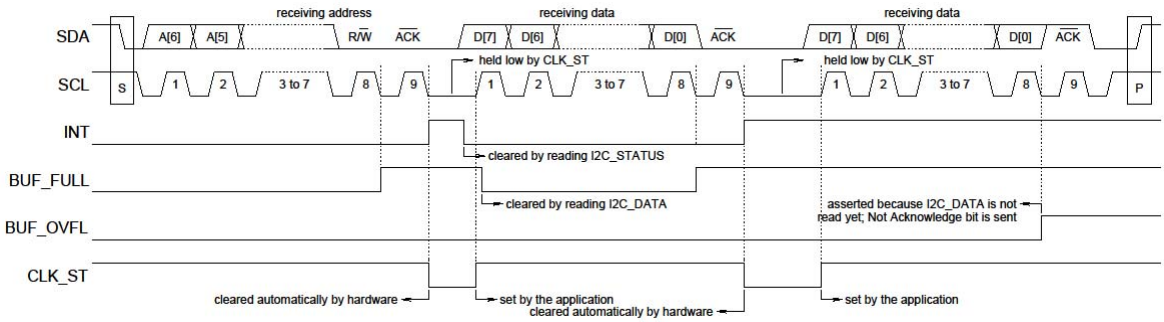


Figure 12: Slave Mode Timing w/CLK_ST_ENB=1 (Reception, 7-bit Address)

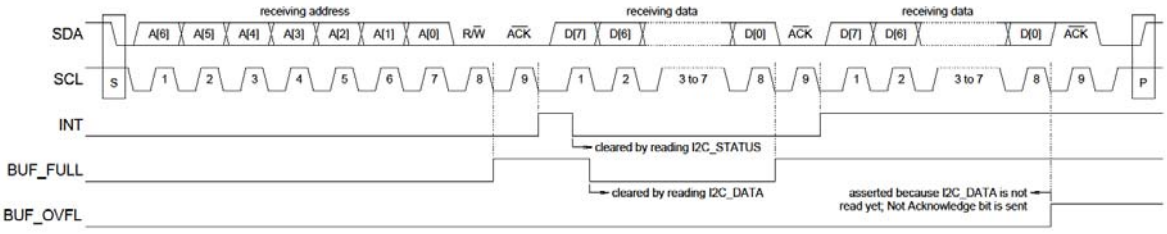


Figure 13: Slave Mode Timing w/CLK_ST_ENB=0 (Reception, 7-bit Address)

3.1.16.1.3 I²C Access Sequence, Read Data Phase

If the received R/W bit (1=read, 0=write) is 1, it is an I²C read access and the slave switched to transmitting mode. Before each byte shifts out, the ICLKSTR bit is automatically cleared to hold the SCL pin low (known as clock stretching). The application needs to load the I2CDATA register with the byte to be transmitted and then set ICLKSTR bit to release SCL pin. Every time the slave shifts out a byte, it receives the ACK/NACK bit. If it receives the ACK bit, the slave automatically clears the ICLKSTR bit, and the application needs to load the I2CDATA register and set the ICLKSTR bit to resume transmission. If it receives the NACK bit, which means the Master device has completed reading data, the Slave releases both SCL and SDA. The Slave asserts an interrupt after receiving the ACK/NACK bit. The read data phase is concluded when receiving the NACK bit or detecting Repeated Start or Stop condition. Figures below illustrates I²C read access.

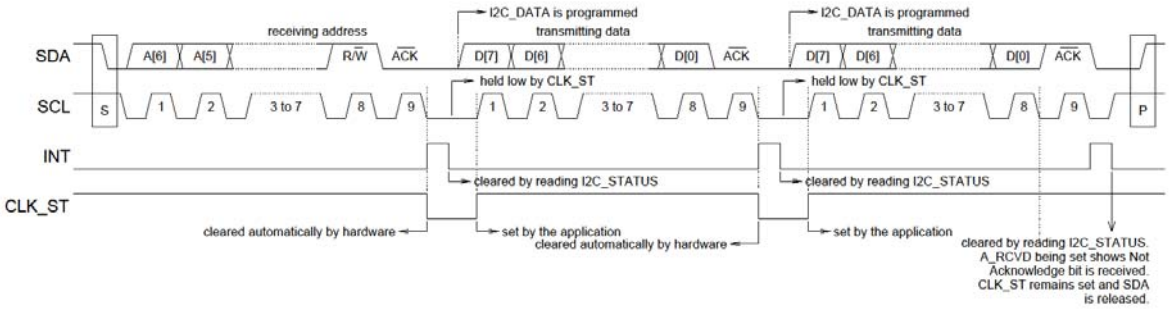


Figure 14: Slave Mode Timing (Transmission, 7-bit Address)

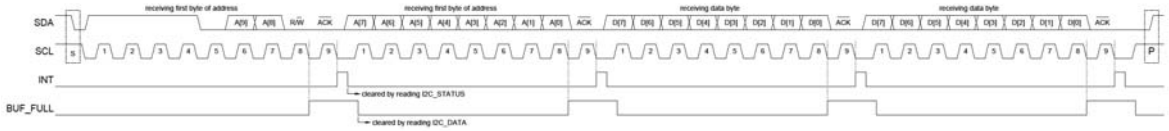


Figure 15: Slave Mode Timing (Reception, 10-bit Address)

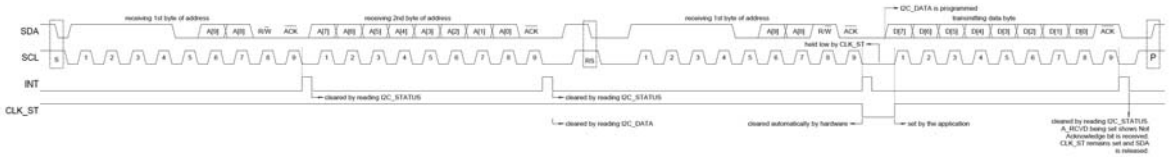


Figure 16: Slave Mode Timing (Transmission, 10-bit Address)

3.1.16.2 I²C Operation in Master Mode

In order to configure Master Mode, the following steps must be carried out:

- Select the master mode. Selection is made by setting the I²C Master/Slave bit IMS of I²C Control Register 2 I2CCTRL2.
- Define the baud rate. The equation defining the baud rate as a function of the system clock is:

$$Fi2c = \frac{Fclk}{2 * (divider + 1)}$$

  Where Fi2c is the frequency of the I²C interface, divider is the I²C divider, and Fclk is the system clock.
- Enable the interface. Enabling is controlled by configuring the I2C Enable bit IEN in I²C Control Register 2 I2CCTRL2.
- Enable the interrupts and define the corresponding handlers.
- Define the interrupt handlers if required.

From this point on, the application must handle the communication. The following paragraphs describe general steps.

3.1.16.2.1 Configuration Settings

The Master controls SCL and SDA when issuing Start, Repeated Start, and Stop conditions. It also drives (release/drain) SCL and SDA when transmitting address/data bytes as well as ACK/NACK bits after receiving data bytes.

When IEN and IMS bits are both set in the I2CCTRL2 register, the interface is configured as an I²C Master.

3.1.16.2.2 Baud Rate Generator Configuration

A baud rate generator (BRG) inside the peripheral serves as an engine to time SCL transitions during a data transfer as well as the transitions of both SCL/SDA during Start, Repeated Start, and Stop conditions.

The BRG consists of an 8-bit counter that, when enabled, loads the value from the I2CADDR0 register and counts down to 0. It then goes back to the I2CADDR0 register value and repeats the counting down process. When the BRG counter counts down to 0, it triggers the SCL transitions during data transfer and SCL/SDA transitions during Start, Repeated Start, and Stop conditions.

The peripherals baud rate is determined by the system clock frequency Fclk and divider. The equation is:

$$Fi2c = \frac{Fclk}{2 * (divider + 1)}$$

3.1.16.2.3 Start Condition

The Master issues a Start condition when the IRSTRB bit is set by the application. When detecting the issued Start condition, the Master asserts an interrupt and clears the ISTRSTRETCH bit. The application reads the I2CSTATUS register to clear the interrupt condition. At this point, the ISTRR bit is set.

Once the ISTRSTRETCH bit is set, if SCL or SDA is already sampled low when the ISTRSTRETCH bit is set, then it is concluded that there is a bus collision due to another I²C bus Master on the bus, and the bus collision interrupt is asserted. The application must read the I2CSTATUS register to clear this interrupt condition.

3.1.16.2.4 Repeated Start Condition

The Master issues a Repeated Start condition when the ISTRSTRETCH bit is set by the application. When detecting the issued Repeated Start condition, the Master asserts an interrupt and clears the IRSTR bit. The application reads the I2CSTATUS register to clear the interrupt condition. At this point, the ISTRR bit is set.

Once the IRSTRS bit is set, if SCL is sampled low before SDA goes low, or if SDA is sampled low when SCL goes from low to high, then it is concluded that there is an I²C bus collision and collision interrupt is asserted. The application reads the I2CSTATUS register to clear the interrupt condition.

3.1.16.2.5 Stop Condition

The Master issues a Stop condition when the ISTPSIZE bit is set by the application. When detecting the issued Stop condition, the Master asserts an interrupt and clears the ISTPSIZE bit. The application reads the I2CSTATUS register to clear the interrupt condition. At this point. The ISTRR bit is set.

Once the ISTPSIZE bit is set, if SDA is sampled low one baud period (T_{br}) after it is released by the peripheral, or, after SCL is released, SCL is sampled low before SDA goes high, then it is concluded that there is an I²C bus collision and collision interrupt is asserted. The application reads the I2CSTATUS register to clear the interrupt condition.

3.1.16.2.6 Acknowledge Bit

The Master transmits ACK/NACK bit when ISACK is set by the application. If the value of ISNACK is 1, a NACK bit is transmitted, otherwise an ACK bit is transmitted. The Master asserts an interrupt and clears ISNACK. The application reads the I2CSTATUS register to clear the interrupt condition.

If the Master transmits a NACK bit but detects an ACK bit, then there is an I²C bus collision and collision interrupt is asserted. The application reads the I2CSTATUS register to clear the interrupt condition.

3.1.16.2.7 I²C Write Access Sequence

The typical I²C write access sequence consists of the following steps:

- The application sets the ISTRETCH bit to issue a Start condition.
- The Master detects the Start condition and asserts an interrupt. The application reads the I2CSTATUS register to clear the interrupt and check the ISTRR bit.
- The application programs the I2CDATA register with the appropriate I²C Slave's address, then the Master starts transmitting the address byte.
- After sampling ACK/NACK sent by the Slave, the Master asserts an interrupt. The application reads the I2CSTATUS register to clear the interrupt and check the IACKR bit.
- The application programs the I2CDATA register with the data byte to be transmitted, then the Master starts transmitting the data byte.
- After sampling the ACK/NACK bit sent by the Slave, the Master asserts an interrupt. The application reads the I2CSTATUS register to clear the interrupt condition and check the IACKR bit.
- Previous two steps are repeated to transmit more bytes.
- The application can access a different I²C Slave or read from the same one by setting the IRSTR bit. This causes the Master to issue a Repeated Start condition. When the condition is sampled, the Master asserts an interrupt. The application reads the I2CSTATUS register to clear the interrupt condition and check the ISTRR bit.
- The application concludes the current transfer by setting the ISTPSIZE bit, and the Master issues a Stop condition. When the condition is sampled, the Master asserts an interrupt. The application reads the I2CSTATUS register to clear the interrupt and check the ISTPR bit.

Figure below shows a timing waveform of Master write access. The only difference between 7-bit and 10-bit address mode is that the application needs to program two address bytes in 10-bit mode.

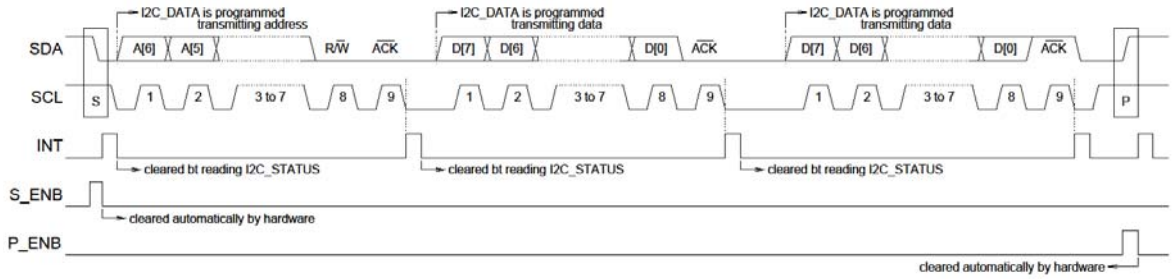


Figure 17: Master Timing Waveform (Transmission)

3.1.16.2.8 I²C Read Access Sequence

The typical I²C read access sequence consists of the following steps:

- The application sets the ISTRETCH bit to issue a Start condition.
- The Master detects the Start condition and asserts an interrupt. The application reads the I2CSTATUS register to clear the interrupt and check the ISTRR bit.
- The application programs the I2CDATA register with the appropriate I²C Slave’s address, then the Master starts transmitting the address byte.
- After sampling ACK/NACK sent by the Slave, the Master asserts an interrupt. The application reads the I2CSTATUS register to clear the interrupt and check the IACKR bit.
- The application programs the I2CDATA register with the data byte to be transmitted, then the Master starts transmitting the data byte.
- After sampling the ACK/NACK bit sent by the Slave, the Master asserts an interrupt. The application reads the I2CSTATUS register to clear the interrupt condition and check the IACKR bit.
- The application sets the IRCSTRT bit, which enables the Master to pulse the SCL pin and shift in a data byte. After shifting in the whole data byte, the Master asserts an interrupt. The application reads the I2CSTATUS register to clear the interrupt. The application then reads the I2CDATA register to fetch the received data byte.
- The application clears the ISNACK bit (Acknowledge bit to be sent) and sets the ISACK bit. The Master transmits the ACK bit. After the transmission, the Master asserts an interrupt. The application reads the I2CSTATUS register to clear the interrupt.
- Previous two steps are repeated to receive more bytes.
- The application can access a different I²C Slave or write to the same one by setting the IRSTR bit. This causes the Master to issue a Repeated Start condition. When the condition is sampled, the Master asserts an interrupt. The application reads the I2CSTATUS register to clear the interrupt condition and check the ISTRR bit.
- If the data byte being received is the last one, after clearing the interrupt, the user sets the ISNACK (Not Acknowledge bit to be set) and IRSTS bits. The Master transmits NACK bit. After the transmission, the MASTER asserts an interrupt. The application reads the I2CSTATUS register to clear the interrupt and check the ISTPR bit.
- The application concludes the current transfer by setting the ISTPSIZE bit, and the Master issues a Stop condition. When the condition is sampled, the Master asserts an interrupt. The application reads the I2CSTATUS register to clear the interrupt and check the ISTPR bit.

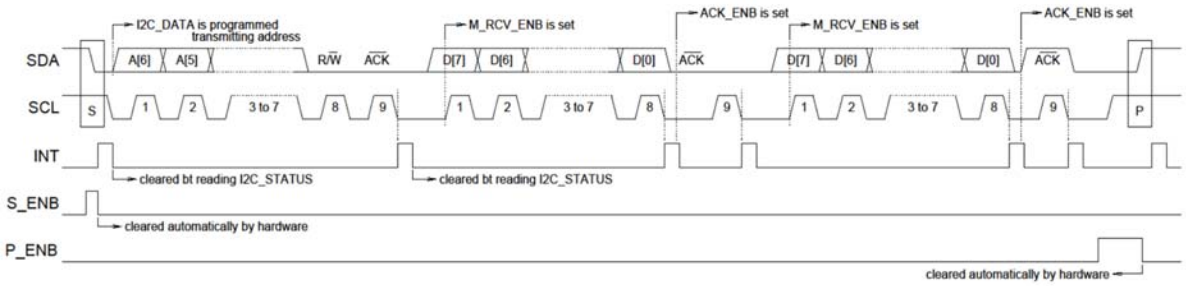


Figure 18: Master Timing Waveform (Reception)

3.1.16.2.9 RW_BUSY Indicator

Whenever detecting a Start/Stop condition, wish it did not issue itself, the Master asserts/de-asserts RW_BUSY to indicate the busy/idle status of the I²C bus. The application can check the IRWBUSY bit before issuing a Start condition so a bus collision can be avoided.

3.1.16.3 I2C Registers

The following registers are used to control the I²C interface:

Table 3.21 I2C Registers

Address	Register Name	Description	Reset Value
0x50000008	I2CSTATUS	I2C Status Register	0x10
0x50000009	I2CCTRL1	I2C Control Register 1	0x40
0x5000000A	I2CCTRL2	I2C Control Register 2	0x02
0x5000000B	I2CDATA	I2C Data Register	0x00
0x5000000C	I2CADDR0	I2C Address Register 0	0x00
0x5000000D	I2CADDR1	I2C Address Register 1	0x00

I²C Status Register							
I2CSTATUS		0x50000008			0x10		
R	R	R	R	R	R	R	R
IACKR	IADDRR	ISTRR	ISTPR	IRWBUSY	IBUFF	IWBUFOVL	IRBUFOVL
MSB							LSB

Bit7 **IACKR:** Acknowledge received
0 = Received
1 = Not received

Bit6 **IADDRR:** Data/Address received (slave mode)
0 = DATA received
1 = ADDRESS received

Bit5 **ISTRR:** Start bit received
0 = Not received
1 = Received

Bit4 **ISTPR:** Stop bit received (slave mode)
0 = No Stop bit received
1 = Stop bit received

Bit3 **IRWBUSY:** Read/Write Busy:
Master Mode:
0 = Bus not being accessed
1 = Bus being accessed
Slave Mode:
0 = **I²C** write operation (Slave receives data)
1 = **I²C** read operation (Slave transmits data)

Bit2 **IBUFF:** Buffer Full
0 = Buffer empty
1 = Buffer full, either it received data or there’s an untransmitted byte

Bit1 **IWBUFOVL:** Write buffer overflow
0 = Buffer empty
1 = Internal shift register is full and the I2CDATA register was written

Bit0 **IRBUFOVL:** Data/Address received (slave mode)
0 = Register was read
1 = Internal shift register full and another byte received from I²C bus

NOTE: While IRBUFOVL is set, the shift-in of bits from bus is stopped.

I²C Control Register 1							
I2CCTRL1		0x50000009			0x40		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
IRSTR	ICLKSTR	IGC	IRCSTRT	ISNACK	ISACK	ISTPSIZE	ISTRSTRE
MSB							LSB

Bit7 **IRSTR:** Repeated Start bit (Master Only)
0 = Repeated start bit disabled
1 = Issue start-bit enable (when set, the I²C transmits Repeated Start bit), cleared by HW.

Bit6 **ICLKSTR:** Clock stretch (Slave Only)
0 = SCL held low
1 = SCL released

Bit5 **IGC:** General call address (Slave Only)
0 = General call address disabled
1 = General call address enabled

Bit4 **IRCSTRT:** Start bit reception (Master Only and cleared by HW)
0 = Receive operation not allowed
1 = Receive operation starts (the receive operation starts when this bit is set)

Bit3 **ISNACK:** ACK bit to be transmitted. (Master Only)
0 = ACK is transmitted upon reception of byte
1 = NACK is transmitted upon reception of byte

Bit2 **ISACK:** ACK bit (Master Only)
0 = No ACK/NACK bit transmitted
1 = Acknowledge (ACK/NACK, defined by ISNACK) is transmitted

Bit1 **ISTPSIZE:** Stop bit or selection of address size
Master Mode:
0 = No stop bit sent
1 = Stop bit sent
Slave Mode:
0 = 7-bit address
1 = 10-bit address

Bit0 **ISTRSTRETCH:** Start and stretch
Master Mode:
0 = No start bit sent
1 = Send start bit
Slave Mode:
0 = No clock stretch
1 = Clock stretched

I²C Control Register 2							
I2CCTRL2		0x5000000A			0x02		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
IMSK4	IMSK3	IMSK2	IMSK1	IMSK0	IEN	IFILTER	IMS
MSB							LSB

Bit7-3 **IMSK[4:0]:** I²C Address mask (Slave Only)
0 = Repeated start bit disabled
1 = Issue start-bit enable (when set, the I²C transmits Repeated Start bit), cleared by HW.

Bit2 **IEN:** I²C Enable bit
0 = I²C disabled
1 = I²C enabled

Bit1 **IFILTER:** I²C filter
0 = Filter disabled
1 = Filter enabled

Bit0 **IMS:** I²C Master/Slave
0 = I²C slave
1 = I²C master

NOTE: In order to ignore glitches on I²C bus, a 3-tab median filter operating at system clock rate is implemented on the incoming SCL and SDA data paths. This filter can be enabled/disabled by setting/clearing IFILTER. The truth table of the median filter is as follows:

Filter Tabs and Output			
Filter tab 0	**Filter tab 1**	**Filter tab 2**	**Filter output**
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

I²C Data Register							
I2CDATA		0x5000000B			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
IDT7	IDT6	IDT5	IDT4	IDT3	IDT2	IDT1	IDT0
MSB							LSB

Bit7-0 **IDT[7:0]:** I²C Data

I²C Address 0 Register							
I2CADDR0		0x5000000C			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
IADDR7	IADDR6	IADDR5	IADDR4	IADDR3	IADDR2	IADDR1	IADDR0
MSB							LSB
Bit7-0 **IADDR[7:0]**: In Master mode, the data represents the 8-bit clock frequency divider counter maximum value. In Slave mode, it represents the lower 8-bit I²C address.							

I²C Address 1 Register							
I2CADDR1		0x5000000D			0x00		
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	R/W	R/W
-	-	-	-	-	-	IADDR9	IADDR8
MSB							LSB
Bit1-0 **IADDR[9:8]**: In Slave mode, the data represents the upper 2-bit I²C address. In Master mode, the data is ignored.							

3.1.17 USB Interface

The digital engine supports a full-speed USB interface for data transfers with an external PC or manufacturing systems. An external USB_5V_IN pin provides access to power from the USB connection and detection of an external USB connection. The rising edge of the USB supply voltage generates an interrupt that can be used to wake the microcontroller. A USB_3p3V_OUT pin is provided for a 1 µF external capacitor used to decouple a regulated 3.3 V supply generated from USB power. When connected to USB power, the device will switch to this source from battery power to conserve battery life, as USB communication is likely to be processor and power intensive. The ARM core will be powered from the USB supply and basic LCD functionality will be maintained using USB power. SPI, I2C, and UART interfaces are fully functional when powered by the USB port. The GPIO power pins, GPIOE0 through GPIOE3, may be used to selectively source power, including USB derived power, to external devices. The device internally support a switched 1.5 KΩ pull-up to 3.0 V through 3.6 V on USBDP to negotiate full-speed operation. The high-speed clock required by the USB interface is derived from the 32.768 KHz oscillator using the high frequency PLL (see Figure 3).

Table 3.22 USB Signals

Pin Name	Description
USB_DM	USB Data Signal Minus – USB data signal from/to digital engine. Series resistor required.
USB_DP	USB Data Signal Plus – USB data signal from/to digital engine. Series resistor required. Internal switched 1.5 KΩ pull-up resistor to 3.3 V required.
USB_5V_IN	USB 5V Supply Input – Used for external powering of USB interface from USB bus. Also used to detect external USB connection. Rising edge to generate an interrupt capable of waking the microcontroller from power-down mode. Should be bypassed with a 1 µF decoupling capacitor.

USB_3p3V_OUT	USB 3.3 V Regulated Output Voltage – Regulated voltage derived from the USB supply voltage. Allows the connection of an external 1 µF decoupling capacitor.

3.1.17.1 USB Digital Engine Features

- Full-speed (12 Mbps, default) and low speed (1.5 Mbps) USB serial interface engine and transceiver
- Support for control, bulk and interrupt transfers only (no isochronous)
- Maximum of two endpoints of up to 64 bytes each for data transfer in interrupt or bulk transfer mode, in addition to the control endpoint 0 for configuration purposes
- Three USB related interrupts
    - USB connection detected (to allow software to lock out analog measurement functions during USB operation)
    - USB resume from suspend interrupt
    - USB communications interrupt
- Transfer of data from USB peripheral to MCU using block mode

3.1.17.2 USB Transceiver (PHY) Features

- Closed loop slew rate control as per USB standard
- On-chip series matching and pull-up resistors for speed identification
- Integrated 48 MHz PLL using 32.768 KHz XTAL as a clock source
- On-chip power management including 3.3 V regulator
- Ability to power digital peripherals based on USB power

Owing to the large amount of protocol-level detail involved in USB implementation, details on USB programming use model and register map are provided in separate documents references [2] and [3].

3.1.17.3 USB Register Map

USB peripheral address space is mapped starting at base address of 0x50011340. The USB register addressing is summarized in Table 6.23. Details regarding usage of the USB port are covered in a separate document. Please contact Indie Semiconductor for an example USB firmware driver.

Table 3.23 USB Registers

Address	Description
0x50011340 – 0x5001173F	USB end-point buffer memory
0x50011340	OUT_7_BASE
0x50011380	IN_7_BASE
0x500113C0	OUT_6_BASE

Table 3.23 USB Registers	
Address	**Description**
0x50011400	IN_6_BASE
0x50011440	OUT_5_BASE
0x50011480	IN_5_BASE
0x500114C0	OUT_4_BASE
0x50011500	IN_4_BASE
0x50011540	OUT_3_BASE
0x50011580	IN_3_BASE
0x500115C0	OUT_2_BASE
0x50011600	IN_2_BASE
0x50011640	OUT_1_BASE
0x50011680	IN_1_BASE
0x500116C0	OUT_0_BASE
0x50011700	IN_0_BASE
0x50011740 – 0x5001174F	USB DMA Registers
0x50011740	DMABUFADDR0_ID
0x50011741	DMABUFADDR1_ID
0x50011742	DMARAMADDR0_ID
0x50011743	DMARAMADDR1_ID
0x50011744	DMARAMADDR2_ID
0x50011745	DMARAMADDR3_ID
0x50011746	DMALEN_ID
0x50011747	DMACTRL_ID
0x50011748	DMALEN_LSB_ID
0x50011749	DMALEN_MSB_ID
0x50011750 – 0x5001175F	Reserved
0x50011760 – 0x5001176F	USB OUT/IN Data I/D
0x50011760	OUT8DATA_ID
0x50011761	OUT9DATA_ID
0x50011762	OUT10DATA_ID
0x50011763	OUT11DATA_ID
0x50011764	OUT12DATA_ID
0x50011765	OUT13DATA_ID
0x50011766	OUT14DATA_ID
0x50011767	OUT15DATA_ID
0x50011768	IN8DATA_ID
0x50011769	IN9DATA_ID
0x5001176A	IN10DATA_ID
0x5001176B	IN11DATA_ID
0x5001176C	IN12DATA_ID
0x5001176D	IN13DATA_ID
0x5001176E	IN14DATA_ID
0x5001176F	IN15DATA_ID
0x50011770 – 0x5001177F	USB OUT/IN BCL I/D
0x50011770	OUT8BCH_ID
0x50011771	OUT8BCL_ID

Table 3.23 USB Registers	
Address	**Description**
0x50011772	OUT9BCH_ID
0x50011773	OUT9BCL_ID
0x50011774	OUT10BCH_ID
0x50011775	OUT10BCL_ID
0x50011776	OUT11BCH_ID
0x50011777	OUT11BCL_ID
0x50011778	OUT12BCH_ID
0x50011779	OUT12BCL_ID
0x5001177A	OUT13BCH_ID
0x5001177B	OUT13BCL_ID
0x5001177C	OUT14BCH_ID
0x5001177D	OUT14BCL_ID
0x5001177E	OUT15BCH_ID
0x5001177F	OUT15BCL_ID
0x50011780 – 0x5001178F	USB OUT/IN Address
0x50011780	Reserved
0x50011781	BOUT1ADDR_ID
0x50011782	BOUT2ADDR_ID
0x50011783	BOUT3ADDR_ID
0x50011784	BOUT4ADDR_ID
0x50011785	BOUT5ADDR_ID
0x50011786	BOUT6ADDR_ID
0x50011787	BOUT7ADDR_ID
0x50011788	BINSTADDR_ID
0x50011789	BIN1ADDR_ID
0x5001178A	BIN2ADDR_ID
0x5001178B	BIN3ADDR_ID
0x5001178C	BIN4ADDR_ID
0x5001178D	BIN5ADDR_ID
0x5001178E	BIN6ADDR_ID
0x5001178F	BIN7ADDR_ID
0x50011790	CLKGATE
0x50011791 – 0x5001179F	Reserved
0x500117A0 – 0x500117AF	USB Interrupts
0x500117A0	ISOERR_ID
0x500117A1	ISOCTL_ID
0x500117A2	ZBCOUT_ID
0x500117A3 – 0x500117A7	Reserved
0x500117A8	IVEC_ID
0x500117A9	IN07IRQ_ID
0x500117AA	OUT07IRQ_ID
0x500117AB	USBIRQ_ID
0x500117AC	IN07IEN_ID
0x500117AD	OUT07IEN_ID
0x500117AE	USBIEN_ID

Table 3.23 USB Registers	
Address	**Description**
0x500117AF	USBBAV_ID
0x500117B0 – 0x500117BF	USB Special Function Register
0x500117B0 – 0x500117B3	Reserved
0x500117B4	EP0CS_ID
0x500117B5	IN0BC_ID
0x500117B6	IN1CS_ID
0x500117B7	IN1BC_ID
0x500117B8	IN2CS_ID
0x500117B9	IN2BC_ID
0x500117BA	IN3CS_ID
0x500117BB	IN3BC_ID
0x500117BC	IN4CS_ID
0x500117BD	IN4BC_ID
0x500117BE	IN5CS_ID
0x500117BF	IN5BC_ID
0x500117C0	IN6CS_ID
0x500117C1	IN6BC_ID
0x500117C2	IN7CS_ID
0x500117C3	IN7BC_ID
0x500117C4	Reserved
0x500117C5	OUT0BC_ID
0x500117C6	OUT1CS_ID
0x500117C7	OUT1BC_ID
0x500117C8	OUT2CS_ID
0x500117C9	OUT2BC_ID
0x500117CA	OUT3CS_ID
0x500117CB	OUT3BC_ID
0x500117CC	OUT4CS_ID
0x500117CD	OUT4BC_ID
0x500117CE	OUT5CS_ID
0x500117CF	OUT5BC_ID
0x500117D0	OUT6CS_ID
0x500117D1	OUT6BC_ID
0x500117D2	OUT7CS_ID
0x500117D3	OUT7BC_ID
0x500117D4 – 0x500117D5	Reserved
0x500117D6	USBCS_ID
0x500117D7	TOGCTL_ID
0x500117D8	USBFRML_ID
0x500117D9	USBFRMH_ID
0x500117DA	Reserved
0x500117DB	FNADDR_ID
0x500117DC	Reserved
0x500117DD	USBAIR_ID
0x500117DE	IN07VAL_ID

Table 3.23 USB Registers	
Address	**Description**
0x500117DF	OUT07VAL_ID
0x500117E0	INISOVAL_ID
0x500117E1	OUTISOVAL_ID
0x500117E2	ISOSTADDR_ID
0x500117E3	ISOSIZE_ID
0x500117E4 – 0x500117E7	Reserved
0x500117E8	SETUPBUF0_ID
0x500117E9	SETUPBUF1_ID
0x500117EA	SETUPBUF2_ID
0x500117EB	SETUPBUF3_ID
0x500117EC	SETUPBUF4_ID
0x500117ED	SETUPBUF5_ID
0x500117EE	SETUPBUF6_ID
0x500117EF	SETUPBUF7_ID
0x500117F0	OUT8ADDR_ID
0x500117F1	OUT9ADDR_ID
0x500117F2	OUT10ADDR_ID
0x500117F3	OUT11ADDR_ID
0x500117F4	OUT12ADDR_ID
0x500117F5	OUT13ADDR_ID
0x500117F6	OUT14ADDR_ID
0x500117F7	OUT15ADDR_ID
0x500117F8	IN8ADDR_ID
0x500117F9	IN9ADDR_ID
0x500117FA	IN10ADDR_ID
0x500117FB	IN11ADDR_ID
0x500117FC	IN12ADDR_ID
0x500117FD	IN13ADDR_ID
0x500117FE	IN14ADDR_ID
0x500117FF	IN15ADDR_ID
0x50011800	USBCONFIG: USB top-level configuration register

3.1.18 Pulse Width Modulator

The device includes a PWM. The main characteristics are:

- Twelve bit resolution – Both period and pulse width.
- Independent pre-scaler
- Programmable active level.

The pulse width modulator can be assigned to any pin in GPIO port C. The PWM_EN bit must be set in the PWMCTRL register. For minimal standby current, pull-up for PWM output pin defaulting to a low state should be disabled while in the standby state.

Table 3.24 PWM Signal	
Pin Name	**Description**
PCx	Pulse Width Modulated Output – Pulse width modulated output signal from the digital engine to an external device. Default: GPIOC7. For minimal standby current, pull-up for PWM output pin defaulting to a low state should be disabled while in the standby state.

3.1.18.1 PWM Usage Description

The PWM circuit generates a wide-range, high-resolution modulated output. It has a total of 4 data and configuration registers to communicate with the microcontroller.

The waveform is controlled by 12-bit period word (PWMPER and PWMEXT) and 12-bit pulse width word (PWMPW and PWMEXT) that are used to determine the output waveform.

The entire waveform can be scaled by adjusting the Prescaler value in PWM2CTRL. The Prescaler can be set to one of eight different settings shown in Table 3.8.1.1.

Table 3.25 PWM Prescaler Divider Values	
PWM_PWC	**Divide Value (f_{XO}/f_{PWM})**
000	1
001	2
010	4
011	8
100	32
101	256
110	8,192
111	262,144

The output period is calculated as follows:

$$Period = \frac{1 + PWMPER \times DIVIDE\_VALUE}{SystemClock}$$

The PWM pulse width is calculated as follows:

$$Pulse\_Width = \frac{1 + PWMPW \times DIVIDE\_VALUE}{SystemClock}$$

To control the active level of the PWM, control bit PWM_INV is used. If this bit is set to one, the PWM output is low level during the pulse and one at other times, including if the PWM is disabled by the user.

Alternatively, if PWM_INV is set to zero, then the PWM outputs a high level during the pulse.

3.1.18.2 PWM Registers

The following registers are used to control the PWM:

Table 3.26 PWM Registers

Address	Register Name	Description	Reset Value
0x500180C0	PWM_CTRL	PWM control	0x00
0x500180C1	PWM_PER	PWM period	0x00
0x500180C2	PWM_PW	PWM pulse width	0x00
0x500180C3	PWM_EXT	PWM period and pulse width	0x00

PWM Control Register

PWM_CTRL		0x500180C0			0x00		
R/W	Reserved	Reserved	R/W	Reserved	R/W	R/W	R/W
PWM_EN	-	-	PWM_INV	-	PRESC2	PRESC1	PRESC0
MSB							LSB

Bit7 PWM_EN: PWM enable bit
0 = PWM2 disabled
1 = PWM2 enabled
Bit4 PWM_INV: PWM output signal polarity
0 = Receive Normal logic
1 = Inverted logic (active low)
Bit2-0 PRESC[2:0}: PWM’s Prescaler
000 = System Clock/1
001 = System Clock/2
010 = System Clock/4
011 = System Clock/8
100 = System Clock/32
101 = System Clock/256
110 = System Clock/8192
111 = System Clock/262144 (2^{18})

PWM Period Low Byte Register

PWM_PER		0x500180C1			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PWM_PER7	PWM_PER6	PWM_PER5	PWM_PER4	PWM_PER3	PWM_PER2	PWM_PER1	PWM_PER0
MSB							LSB

Bit7-0 PWM_PER[7:0]: PWM period low register

PWM Pulse Width Low Byte Register

PWM_PW		0x500180C2			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PWM_PW7	PWM_PW6	PWM_PW5	PWM_PW4	PWM_PW3	PWM_PW2	PWM_PW1	PWM_PW0
MSB							LSB

Bit7-0 PWM_PW[7:0]: PWM pulse width low register

PWM extension with high nibble of period and pulse width							
PWM_EXT		0x500180C3			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PWM_PER11	PWM_PER10	PWM_PER9	PWM_PER8	PWM_PW11	PWM_PW10	PWM_PW9	PWM_PW8
MSB							LSB
Bit7-4 PWM_PER[11:8]: PWM period high nibble Bit3-0 PWM_PW[11:8]: PWM pulse width high nibble							

3.1.19 Internal Interface Bus

All peripherals are mapped into the unified 32-bit address space in the ARM Cortex.

The M0 Core’s internal interface bus supports high speed interfacing with the analog engine and those digital interface functions not resident in the M0 Core.

A block transfer mode is provided to reduce the overhead for data-intensive peripherals such as USB (see Section 3.1.19.1 Block Transfer Mode).

3.1.19.1 Block Transfer Mode

Individual data transfers to certain peripherals have a large accessing overhead due to internal serialization and addressing overhead. If working with large chunks of data in the peripheral space, it is significantly more efficient to have the microcontroller working with the data in SRAM. For instance, if computing a checksum for a received transmission, the recommended sequence is to transfer the data to SRAM first and then compute the checksum from the transferred block. To assist with the transferring of data, a background transfer peripheral has been implemented which allows background data transfers between the MCU and the peripheral to occur with only high level microcontroller interaction. Using this mechanism, the microcontroller can initiate a block transfer between peripheral and SRAM and then go on to other tasks while the data is being transferred. In addition to letting the programmer optimize the code to perform other tasks while the transfer is occurring, the transfer itself completes in less cycles since the addressing overhead is substantially reduced in block transfers. The limitations of using block transfer mode are as follows:

- Communication using block mode must be between the SRAM and a peripheral in the address space from 0x5001000 to 0x5001FFFF.
- Source and destination address blocks must be word-aligned. Only transfers of an integer number of 32-bit words are supported.
- The peripheral address can either be a block or a single address used repeatedly. The latter option is used to support FIFO and queue peripheral interface types.
- The serial bandwidth to the ASIC peripherals is shared between the block transfer peripheral and memory accesses from program code. So if a background block transfer is in progress while the program attempts to access a serialized peripheral, either the block transfer of the program code must stall until the other is finished. The priority can be set by a configuration bit.

The programming use model for block transfer is as follows

MCU program configures:

- base address in SRAM
- base address in peripheral memory space
- number of words to be transferred
- whether to increment peripheral address (or access same address repeatedly). The SRAM address always increments.
- data direction, either SRAM to peripheral or peripheral to SRAM

MCU then writes a start indication to initiate the transfer.

During the block transfer, the serial bus is used by the block transfer according to the following rules: If both program code and block transfer request use of the bus, any transfer already in progress will continue until the transfer completes or the next word boundary is reached (if the transfer in progress is a block transfer). Then if the BLOCKING bit is set then the bus will be allocated to the block transfer and the program code will stall. If the BLOCKING bit is not set, then the block transfer will stall until the program code’s transfer is complete.

3.1.19.2 Block Transfer Mode Registers

Block Transfer Base Peripheral Address							
BXADD		0x50000080			0x0000		
W	W	W	W	W	W	W	W
BXADD7	BXADD6	BXADD5	BXADD4	BXADD3	BXADD2	BXADD1	BXADD0
BXADD15	BXADD14	BXADD13	BXADD12	BXADD11	BXADD10	BXADD9	BXADD8
MSB							LSB
Bit15-0 BXADD[15:0] This register controls the starting address in the peripheral space for the transfer. Legitimate addresses for peripheral side of the transfer are 0x5001xxxx. Write this register with the lower 16 bits of the desired address. Only word-aligned addresses should be written to this register or else undefined behavior may result.							

Block Transfer Number of Words							
BXNUM		0x50000082			0x00		
W	W	W	W	W	W	W	W
BXNUM7	BXNUM6	BXNUM5	BXNUM4	BXNUM3	BXNUM2	BXNUM1	BXNUM0
MSB							LSB
Bit7-0 BXNUM[7:0] This register controls the size of the block to be transferred. Load this register with one less than the number of 32-bit words to transfer. For instance, to transfer 32 words then set BXNUM to a value of 31.							

Block Transfer Configuration							
BXCONF		0x50000083			0x00		
W	W	W	W	W	W	W	W

Reserved	Reserved	Reserved	Reserved	START	BLOCKING	W1R0	INCRADDR
MSB							LSB

Bit7-4 Reserved
Bit3 START Set this bit to initiate the currently configured block transfer. The bit automatically clears once the transfer completes.
Bit2 BLOCKING – Configures priority of block transfer vs. program code
1 = Stall program code in case of conflict
0 = Stall block transfer in case of conflict
Bit1 W1R0
1 = Transfer is from SRAM to peripheral
0 = Transfer is from peripheral to SRAM
Bit0 INCRADDR
1 = Increment peripheral address automatically
0 = Repeatedly access same peripheral address

SRAM Start Address Register							
BXSRAMADDR		0x50000084			0x0000		
W	W	W	W	W	W	W	W
BXSRAM7	BXSRAM6	BXSRAM5	BXSRAM4	BXSRAM3	BXSRAM2	BXSRAM1	BXSRAM0
BXSRAM15	BXSRAM14	BXSRAM13	BXSRAM12	BXSRAM11	BXSRAM10	BXSRAM9	BXSRAM8
MSB							LSB

Bit15-0 EXSRAM[15:0] This register configures the starting address of the SRAM block to be used during the transfer. Valid SRAM addresses are in the form 0x2000xxxx. Write this register with the lower 16 bits of the desired address. Only word-aligned addresses should be written to this register or else undefined behavior may result.

3.2 ANALOG FUNCTIONS

The circuitry of the analog engine (also referred to as "Ernie" hereinafter) provides the measurement interface to the sensor, to the thermistor(s) used for ambient temperature sensing. It contains digital-to-analog converters used to trim the sensor drive voltages. It contains transimpedance amplifiers used to apply drive voltages to the sensor and convert sensor currents to voltage for input to the analog-to-digital converter (ADC). There is analog switching to direct drive voltage to and sense current from the appropriate sensor contacts. It contains an ADC with multiplexed differential input to accept voltage input for measurement from one of two transimpedance amplifiers and produce a 17-bit 2's complement result. The bipolar ADC input allows detection of and software compensation for small negative inputs. It contains a precision voltage reference. It contains an analog charge pump and voltage regulation to support analog engine operation over the specified supply/battery voltage range. There is low-battery monitoring capability with reset generation following loss of battery voltage.

3.2.1 Sensor Configurations

Two types of sensor configurations are supported as described below:

3.2.1.1 SG-referenced configuration

In an SG-referenced configuration, the electrochemical sensor is driven with the DAC voltage relative to an independent voltage, referred to herein as SG. This SG voltage is generated by a 10-bit trim DAC using the same reference as in Section 3.2.3). The output stage of this DAC is able to sink or source currents much larger than the expected sensor currents so that the SG voltage is well regulated.

The target voltage for SG is recommended to be 1.1 V. The associated 10-bit trim DAC code is established during manufacturing calibration of a meter, with the codes retained in non-volatile eFlash memory.

In many systems, the electrochemical sensor only sees the voltage difference between the Source DAC driving the transimpedance amplifier and the SG voltage across its terminals, so efficient calibration can simply measure the SG voltage and then find the Source DAC code word that produces the desired (SG voltage + target drive voltage) at the DAC output. This saves the calibration step of having to find an unnecessarily precise SG voltage first.

3.2.1.2 Ground-referenced configuration

In a ground-referenced configuration, the electrochemical sensor is driven with the DAC voltage relative to the common analog ground on the PCB. For most types of chemistries, this configuration prevents any transimpedance amplifier output from exceeding the battery voltage. However, since ground is typically a common PCB net, this configuration may be more susceptible to noise coupling from other sources. In addition, if sensor current must pass through a switch before flowing to ground, the resistance of the switch must be taken into account more precisely since feedback cannot be used to regulate the ground voltage independent of the current.

3.2.2 Analog Sensor Interface and Switching

The analog engine contains analog switches that allow selective connection of the transimpedance amplifiers, analog common voltage SG, and ground to sensor contacts, a reference resistor, and one thermistor. It is also possible to electrically isolate sensor contacts, the reference resistor, and the thermistors. When a sensor contact is electrically isolated, the leakage current is minimized using a high-isolation transmission gate based design.

A current sensing switch in the pair sees only amplifier input leakage current and allows accurate control of the pin voltage. The amplifier need only provide additional output voltage swing to accommodate the voltage drop across the switch sourcing the load current to the pin.

Switches for selecting alternate gains have an on-resistance that is low in comparison to associated external gain resistors. Switch impedance remains stable so that any impact of switch impedance can be calibrated out of the end application.

Separate double-pole switches are provided for each supported amplifier pin combination. Additional switches allow the feedback loops to be closed when no pin is being driven. The approach is similar for both the transimpedance amplifiers and the SG buffer amplifier.

A detailed connection diagram is shown in Figure 19 for the ground-referenced configuration. Please contact Indie Semiconductor with a sensor pin definition for a recommendation on how to assign and use the analog switch matrix. **NOTE: Pins SG, GHH, GH, C, F, TH2, HM, HH, H, J, D, G, B are available in BGA package only.**

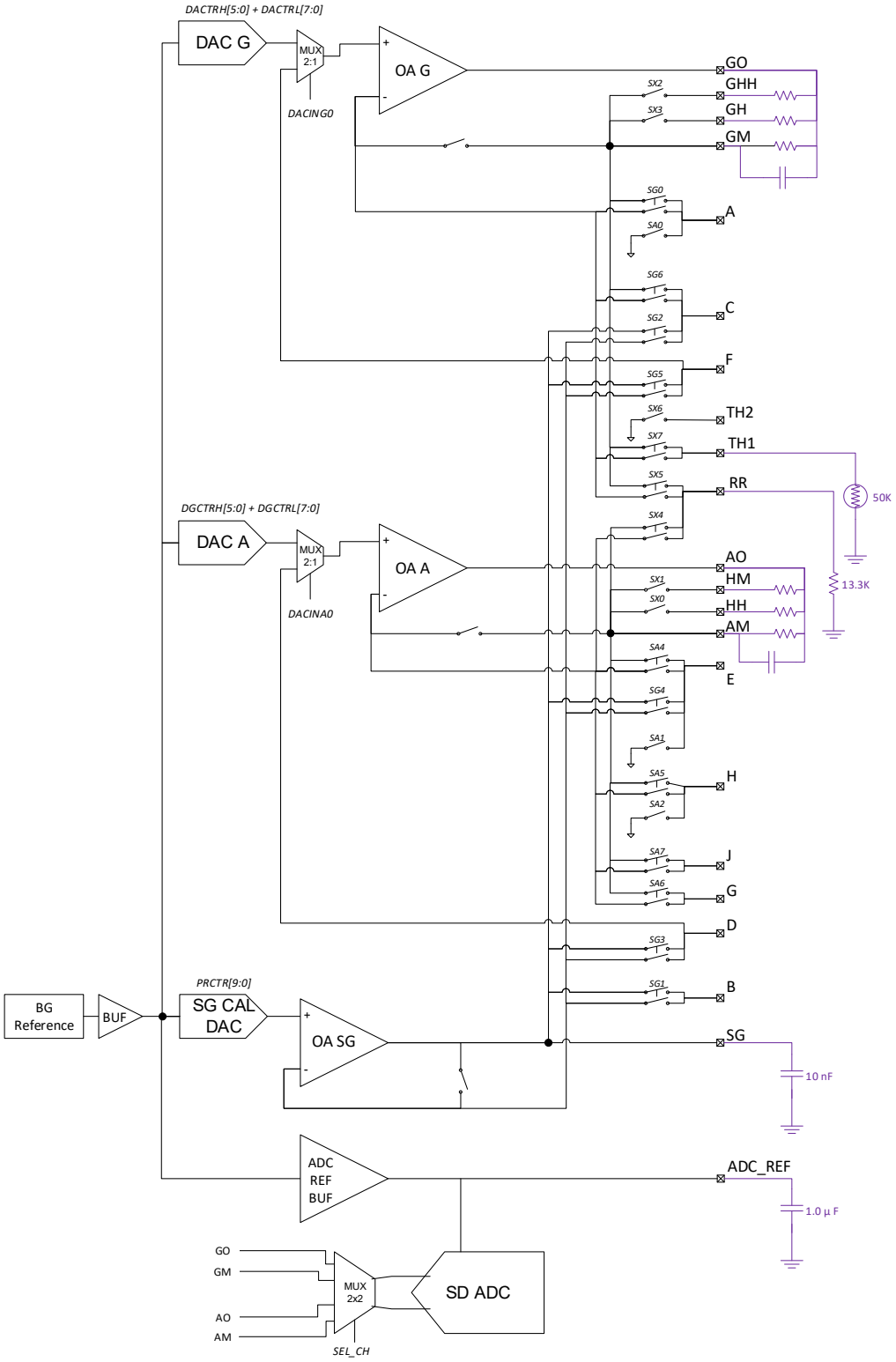


Figure 19: Switch matrix shown with external components for ground-referenced sensor

3.2.2.1 Analog Interface Pins

3.2.2.1.1 Sensor Interface Pins

Table 3.27 Sensor interface	
Pin Name	**Description**
A	A Sensor Contact – Can be electrically isolated, connected to the transimpedance amplifier G voltage output/current input, or connected to system ground GND. This pin is ESD hardened.
C	C Sensor Contact – Can be electrically isolated, connected to the transimpedance amplifier G voltage output/current input, or connected to the analog common voltage SG. NOTE: Available in BGA package only. This pin is ESD hardened.
E	E Sensor Contact – Can be electrically isolated, connected to the transimpedance amplifier A voltage output/current input, connected to the analog common voltage SG, or connected to system ground GND. This pin is ESD hardened.
H	H Sensor Contact – Can be electrically isolated, connected to the transimpedance amplifier A voltage output/current input, or connected to system ground GND. NOTE: Available in BGA package only.
B	B Sensor Contact – Can be electrically isolated or connected to the analog common voltage SG. NOTE: Available in BGA package only.
D	D Sensor Contact – Can be electrically isolated or connected to the analog common voltage SG. NOTE: Available in BGA package only.
F	F Sensor Contact – Can be electrically isolated or connected to the analog common voltage SG. NOTE: Available in BGA package only.
G	G Sensor Contact – Can be electrically isolated or connected to the transimpedance amplifier A voltage output/current input. NOTE: Available in BGA package only. This pin is ESD hardened.
J	J Sensor Contact – Can be electrically isolated or connected to the transimpedance amplifier A voltage output/current input. NOTE: Available in BGA package only.

3.2.2.1.2 Gain Setting Resistor Pins

Transimpedance amplifier gains, and therefore input current ranges, are established by external gain setting resistors.

The gain of transimpedance amplifier G is controlled using resistors connected between pin GO and pins GM, GH, and GHH. Maximum gain, and therefore minimum input current range, is determined by the fixed resistance between pins GO and GM. Switched resistors connected between GO and GH and between GO and GHH allow the feedback resistance to be lower and a higher current input range achieved. Refer to Figure 4 below for typical external component values.

The gain of transimpedance amplifier A is controlled using resistors connected between pin AO and pins AM, HH, and HM. Maximum gain, and therefore minimum input current range, is determined by the fixed resistance between pins AO and AM. Switched resistors connected between AO and HH and between AO and HM allow the feedback resistance to be lower and a higher current input range achieved.

Section 3.2.2.2.3 (SX Control Register, SXCTR) describes the control for switches used to control the gain setting registers.

Table 3.28 Gain Setting Pins

Pin Name	Description
GO	Transimpedance Amplifier G Output – Analog engine I/O pin used as common connection for gain setting resistors. Referred to as GO for the single-sensor meter. Is electrically isolated when analog engine is disabled.
GM	Transimpedance Amplifier G Feedback GM – Analog engine I/O pin used for connection of fixed gain setting resistor GM. Referred to as GM for the single-sensor meter. Is electrically isolated when analog engine is disabled.
GH	Transimpedance Amplifier G Feedback GH – Analog engine I/O pin used for connection of switched gain setting resistor GH. Referred to as GH for the single-sensor meter. Is electrically isolated when analog engine is disabled. NOTE: Available in BGA package only.
GHH	Transimpedance Amplifier G Feedback GHH – Analog engine I/O pin used for connection of switched gain setting resistor GHH. Is electrically isolated when analog engine is disabled. NOTE: Available in BGA package only.
AO	Transimpedance Amplifier A Output – Analog engine I/O pin used as common connection for gain setting resistors of the single-sensor meter. For single-sensor meter, is electrically isolated when analog engine is disabled, including the power-down state
AM	Transimpedance Amplifier A Feedback AM – Analog engine I/O pin used for connection of fixed gain setting resistor AM of the single-sensor meter. Is electrically isolated when analog engine is disabled, including the power-down state.
HM	Transimpedance Amplifier A Feedback HM – Analog engine I/O pin used for connection of switched gain setting resistor HM of the single-sensor meter. Is electrically isolated when analog engine is disabled, including the power-down state. NOTE: Available in BGA package only.
HH	Transimpedance Amplifier A Feedback HH – Analog engine I/O pin used for connection of switched gain setting resistor HH of the single-sensor meter. Is electrically isolated when analog engine is disabled, including the power-down state. NOTE: Available in BGA package only.

3.2.2.1.3 Reference Resistor and Thermistor Pins

Table 3.29 Reference resistor and Thermistor Interface

Pin Name	Description
RR	Reference Resistor – Analog engine I/O pin used to apply voltage to and read current drawn by external reference resistor. Can be electrically isolated, connected to the transimpedance amplifier G voltage output/current input, or connected to the transimpedance amplifier A voltage output/current input.

TH1	Thermistor 1 – Analog engine I/O pin used to apply voltage to and read current drawn by external thermistor 1. Can be electrically isolated or connected to the transimpedance amplifier G voltage output/current input.
TH2	TH2 Sensor Contact – Can be electrically isolated or connected to system ground, GND NOTE: Available in BGA package only.

3.2.2.1.4 Analog Voltage Reference Pins

Table 3.30 Analog Voltage Reference Pins

Pin Name	Description
SG	Analog Common Voltage – SG reference voltage for connection to an external 10 nF filter capacitor referenced to ground. In an SG-referenced system, this establishes the analog common voltage for the DACs and most sensor measurements. Value can be trimmed, recommended to be 1.1 V.
ADC_REF	ADC Reference Voltage Output – SDADC reference voltage output for connection to an external 1 μF decoupling capacitor referenced to ground.

3.2.2.2 Analog Switch Interface Registers

These registers control the analog switches used to connect the transimpedance amplifiers to sensor contacts and address the multiplexer used to select transimpedance amplifier be measured with the ADC. These can be accessed as individual bytes or the four bytes, which are aligned on a word boundary, can be accessed simultaneously as a 32-bit word (0x5000005C – 0x5000005F).

3.2.2.2.1 SG Control Register, SGCTR

This register controls connection of sensor interface pins to transimpedance amplifier G or SG/REF_1.21. The register is cleared following reset, including that associated with the power-down condition. A bit is set to enable connection of a sensor contact for voltage drive and current measurement.

SG Control Register							
SGCTR		0x50000044			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
SG7	SG6	SG5	SG4	SG3	SG2	SG1	SG0
MSB							LSB

```
Bit7    SG7: (Reserved)
Bit6    SG6: Switch G6 control
        0 = C sensor contact isolated
        1 = C sensor contact connected to transimpedance amplifier G
Bit5    SG5: Switch G5 control
        0 = F sensor contact isolated
        1 = F sensor contact connected to SG (1.21V)
Bit4    SG4: Switch G4 control
        0 = E sensor contact isolated
        1 = E sensor contact connected to SG (1.21V)
Bit3    SG3: Switch G3 control
        0 = D sensor contact isolated
        1 = D sensor contact connected to SG (1.21V)
Bit2    SG2: Switch G2 control
        0 = C sensor contact isolated
```

	1 = C sensor contact connected to SG (1.21V)
Bit1	SG1: Switch G1 control
	0 = B sensor contact isolated
	1 = B sensor contact connected to SG (1.21V)
Bit0	SG0: Switch G0 control
	0 = A sensor contact isolated
	1 = A sensor contact connected to transimpedance amplifier G

3.2.2.2.2 SA Control Register, SACTR

This register controls connection of sensor interface pins to transimpedance amplifier A or GND. The register is cleared following reset, including that associated with the power-down condition. A bit is set to enable connection of a sensor contact for voltage drive and current measurement.

SA Control Register							
SACTR		0x50000045			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0
MSB							LSB

Bit7	SA7: Switch A7 control
	0 = J sensor contact isolated
	1 = J sensor contact connected to transimpedance amplifier A
Bit6	SA6: Switch A6 control
	0 = G sensor contact isolated
	1 = G sensor contact connected to transimpedance amplifier A
Bit5	SA5: Switch A5 control
	0 = H sensor contact isolated
	1 = H sensor contact connected to transimpedance amplifier A
Bit4	SA4: Switch A4 control
	0 = E sensor contact isolated
	1 = E sensor contact connected to transimpedance amplifier A
Bit3	SA3: Reserved
Bit2	SA2: Switch A2 control
	0 = H sensor contact isolated
	1 = H sensor contact connected to GND
Bit1	SA1: Switch A1 control
	0 = E sensor contact isolated
	1 = E sensor contact connected to GND
Bit0	SA0: Switch A0 control
	0 = A sensor contact isolated
	1 = A sensor contact connected to GND

3.2.2.2.3 SX Control Register, SXCTR

This register controls connection of gain setting resistor pins to transimpedance amplifier A and G, the connection of reference resistor RR or thermistors TH1 and TH2 to transimpedance amplifier G, and the connection of reference resistor RR to a transimpedance amplifier A. The register is cleared following reset, including that associated with the power-down condition. A bit is set to reference resistor and thermistor connections to enable connection of a device for voltage drive and current measurement. Multiple gain setting bits may be activated at a time.

SX Control Register							
SXCTR		0x50000046			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
SX7	SX6	SX5	SX4	SX3	SX2	SX1	SX0
MSB							LSB

Bit7 SX7: Switch A7 control
0 = TH1 thermistor pin isolated
1 = TH1 thermistor pin connected to transimpedance amplifier G

Bit6 SX6: Switch X6 control
0 = TH2 thermistor pin isolated
1 = TH2 thermistor pin connected to GND
When SX6 = 1, firmware must ensure that one of SG0, SG6, SX5 or SX7 is also set to 1 simultaneously to complete the current path through the external circuit, and prevent amplifier G from railing out.

Bit5 SX5: Switch X5 control
0 = RR reference resistor pin isolated
1 = RR reference resistor pin connected to transimpedance amplifier G

Bit4 SX4: Switch X4 control
0 = RR reference resistor pin isolated
1 = RR reference resistor pin connected to transimpedance amplifier A

Bit3 SX3: Switch SX3 control
0 = GH gain resistor pin isolated
1 = GH gain resistor pin connected to transimpedance amplifier G

Bit2 SX2: Switch X2 control
0 = GHH gain resistor pin isolated
1 = GHH gain resistor pin connected to transimpedance amplifier G

Bit1 SX1: Switch X1 control
0 = HM gain resistor pin isolated
1 = HM gain resistor pin connected to transimpedance amplifier A

Bit0 SX0: Switch X0 control

0 = HH gain resistor pin isolated 1 = HH gain resistor pin connected to transimpedance amplifier A

3.2.2.2.4 Data Select Control Register, DSCTR

This register is used to select the ADC multiplexer input channel. The register also contains bits that control 2-to-1 multiplexers that select the source of the reference voltage supplied to transimpedance amplifier G and transimpedance amplifier A. The register bits are cleared following reset.

Data Select Control Register							
DSCTR		0x50000047			0x00		
Reserved	R/W	Reserved	Reserved	Reserved	Reserved	R/W	R/W
SEL_CH[3]	SEL_CH[2]	SEL_CH[1]	SEL_CH[0]	-	-	DACINAO	DACINGO
MSB							LSB
Bit7-4 SEL_CH[3:0]: ADC channel select control (Ernie_A5 silicon) 4’b0000 = No channel selected (SD ADC input is floating) 4’b0001 = Transimpedance amplifier G selected: Pin GO (positive) vs. Pin GM (negative) 4’b0010 = Transimpedance amplifier A selected: Pin AO (positive) vs. Pin AM (negative) 4’b0100 = Pin F (positive) vs. Pin D (negative) 4’b1000 = Auxiliary mux output vs. on-chip GND (negative) All other combinations = Reserved (should never be set by firmware at any point) Bit3-2 Reserved Bit 1 DACINA0: 0 = DAC selected as reference input for transimpedance amplifier A 1 = D sensor contact pin selected as reference input for transimpedance amplifier A Bit 0 DACING0: 0 = DAC selected as reference input for transimpedance amplifier G 1 = F sensor contact pin selected as reference input for transimpedance amplifier G							

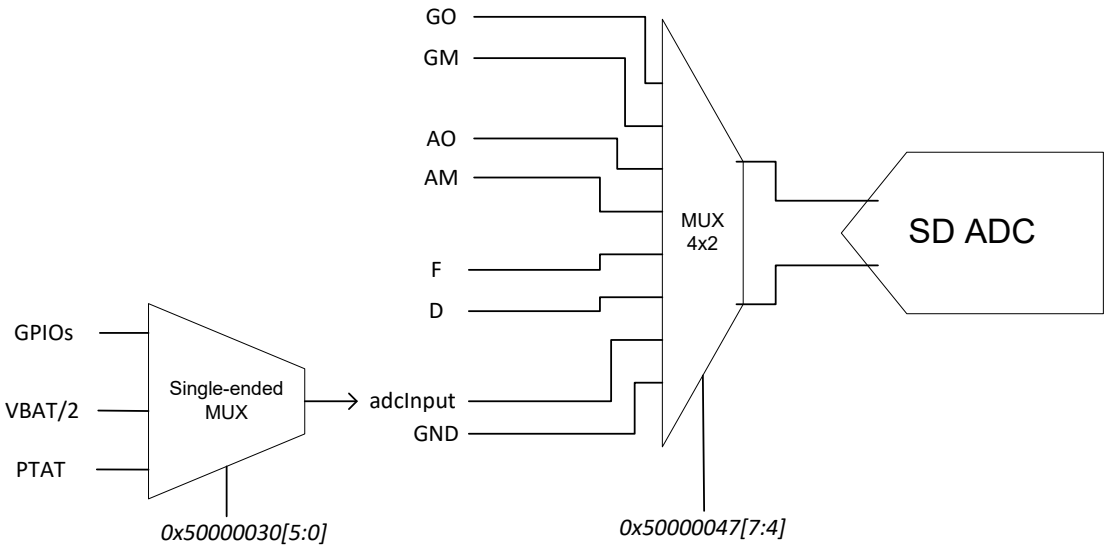


Figure 20: Sigma Delta ADC Input Channel Multiplexing

3.2.3 Voltage Reference

The analog engine contains a 1.21 V bandgap voltage reference used as the common reference for the DACs, and 17-bit ADC. To maintain a stable sensor drive voltage and long-term current measurement accuracy, the voltage reference exhibits good long-term stability and tight supply voltage, load, and temperature regulation.

The precision reference and reference voltage generator circuitry are placed in a minimum current draw configuration when the analog engine is placed in the power-down state.

3.2.4 SG Reference

The internal voltage reference is used to drive a pin SG for SG-referenced systems. This pin voltage can be trimmed between 1.01 V and 1.20 V using a 10-bit DAC as per application needs. The circuit used is as shown in Figure 21. The reference voltage at the input of this calibration DAC circuit is 1.21 V. Therefore, the output voltage, VSG, can be calculated as follows:

$$V_{SG} = Vref \times \frac{S + \frac{Dbg}{1024}}{S + 1} = 1.21 \times \left(\frac{5}{6} + \frac{1}{6} \times \frac{Dbg}{1024}\right)$$

The term S is the ratio R_2 / R_1 and equals 5. The gain, Gr, is 1.0

The term Dbg represents the programmed DAC code.

The nominal trim voltage range is 1.01 V to 1.20 V in 0.2 mV steps.

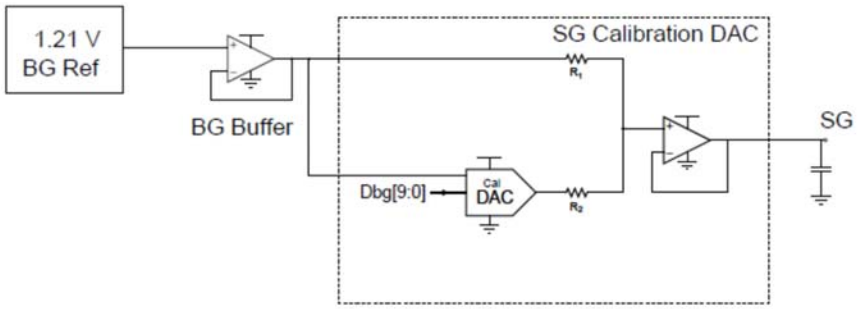


Figure 21: SG Trim DAC Circuit

3.2.4.1.1 SG Reference Control Registers

SG Reference Control Register							
PRCTR		0x50000050-51			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0
-	-	-	-	-	-	PR9	PR8
MSB							LSB
Bit15-10	Reserved						
Bit9-0	PR[9:0] Least-significant 8 bits of 10-bit precision reference trim						

3.2.5 Source DACs

To accommodate the various drive voltages required for measurements made of a sensor, the analog engine contains a pair of sub-ranging DACs that are used to program drive voltage. The DAC can generate 0 to 3 V drive relative to ground. In particular, this includes the range (SG – 1 V) to (SG + 1 V) drive, i.e. ± 1 V relative to SG. This covers both oxidative and reductive reaction chemistries.

An SG-referenced system (as defined in Section 0) avoids operation near the analog supply rails, where unintended shifts in transimpedance amplifier offsets can result more easily. In an SG-referenced system, the source DAC voltage setting during all measurements, including offset measurements, should be limited to the 1 V to 2.5 V range, relative to ground. The DAC setting used during offset readings should match that used during associated sensor, thermistor, and reference measurements, even though the voltage is not applied to a load.

The appropriate DAC settings and resulting voltages are determined when the system is manufactured. They are retained in eFlash memory for the life of the meter. Registers controlling the course and fine adjustments are aligned within a word boundary.

The DACs are placed in a minimum current draw configuration when the analog engine is placed in the power-down state.

3.2.5.1 DAC Output Characteristic

The circuit used for the coarse/fine DACs is as shown in Figure 22. The reference voltage, Vref, at the input of this DAC is 1.21 V and fixed final stage gain, G, is 2.5. The output voltage, DAC_OUT, can be calculated as follows:

$$DAC\_OUT = Vref \times G \times \frac{S\frac{Dc}{64} + \frac{Df}{256}}{S+1}$$

The term S is the ratio R_2 / R_1 and equals 31 in this design. The gain, G, is 2.5.

The terms Dc and Df represent the programmed coarse and fine DAC codes respectively.

Coarse DAC step size is nominally 46 mV and fine DAC step size 0.37mV.

Full scale range is nominally 3.0 V.

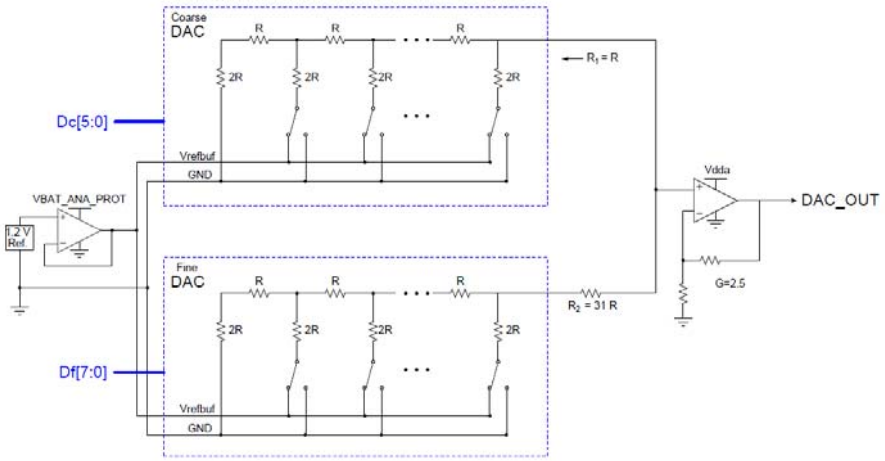


Figure 22: Sub-ranging (Coarse + Fine) DAC Architecture

In use, the DAC voltage is initially set using the formula above. The fine DAC is then trimmed to reduce error around the target to within 0.37 mV. The resulting codes are then stored in non-volatile memory for use during sensor measurements. The process is completed for each applicable measurement voltage.

3.2.5.2 DAC G Control Registers, DGCTRL and DGCTRH

These 8-bit registers, which can be simultaneously accessed as a 16-bit half-word, control setting and reading the state of the bits of the sub-ranging DAC associated with transimpedance amplifier G. There is an 8-bit fine adjustment (DGCTRL) and a 6-bit course adjustment (DGCTRH). The registers are cleared following reset, including that associated with the power-down condition.

DAC G Control Register L							
DGCTRL		0x50000048			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DGF7	DGF6	DGF5	DGF4	DGF3	DGF2	DGF1	DGF0
MSB							LSB
Bit7-0 DGF[7:0]: 8-bit fine DAC G setting							

DAC G Control Register H							
DGCTRH		0x50000049			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
-	-	DGC5	DGC4	DGC3	DGC2	DGC1	DGC0
MSB							LSB
Bits 7-6 Reserved							
Bit 5-0 DGC[5:0]: 6-bit course DAC G setting							

3.2.5.3 DAC A Control Registers, DACTRL and DACTRH

These 8-bit registers, which can be simultaneously accessed as a 16-bit half-word, control setting and reading the state of the bits of the sub-ranging DAC associated with transimpedance amplifier A. There is an 8-bit fine adjustment (DACTRL) and a 6-bit course adjustment (DACTRH). The registers are cleared following reset, including that associated with the power-down condition.

DAC A Control Register L							
DACTRL		0x5000004C			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DAF7	DAF6	DAF5	DAF4	DAF3	DAF2	DAF1	DAF0
MSB							LSB
Bit7-0 DAF[7:0]: 8-bit fine DAC A setting							

DAC A Control Register H							
DACTRH		0x5000004D			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
-	-	DAC5	DAC4	DAC3	DAC2	DAC1	DAC0
MSB							LSB
Bits 7-6 Reserved							
Bits 5-1 DAC[5:0]: 6-bit course DAC A setting							

3.2.6 Transimpedance Amplifiers

The analog engine contains two transimpedance amplifiers. Each is used to apply an associated DAC output voltage to a sensor contact and to allow the measurement of current at that contact. Current flow is normally into the sensor contact from the meter. Transimpedance amplifier G is used to interface with one particular set of sensor contacts. Transimpedance amplifier A is used to interface with another set of sensor contacts. Each transimpedance amplifier converts contact current to a voltage, referenced to the DAC’s output voltage. The transfer function is established by resistance external to the analog engine. To accommodate different full-scale current ranges with a fixed ADC input voltage range, each transimpedance amplifier is accompanied by analog switching that allows the external feedback resistance to be modified.

Up to three gain settings are supported for each transimpedance amplifier. Use of external resistors provides better long-term gain stability and allows potential adjustment of gain ranges. The impedance of enabled analog switches in series with the external resistors must be low in comparison to the external resistance so as not to significantly impact gain.

Since the transimpedance amplifiers run from the output of the analog boost regulator, they support up to 3.25 V output level over the entire battery voltage range of 2.2 V to 3.3 V

The transimpedance amplifiers are placed in a minimum current draw configuration when the analog engine is placed in the power-down state.

Instead of the DAC, external voltages can be used as reference to the transimpedance amplifiers G and A as shown in the simplified figures below:

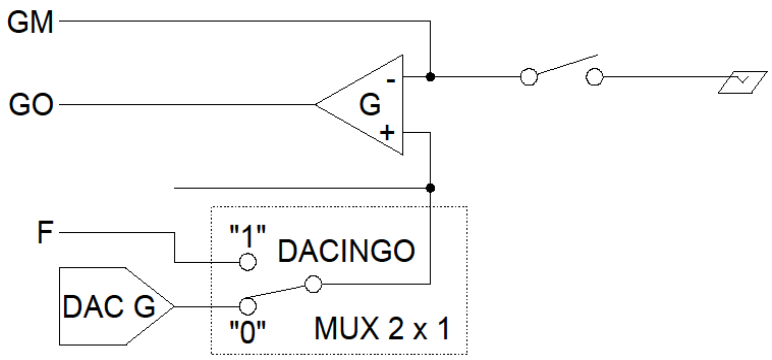


Figure 23: External Voltage Reference for Amplifier G

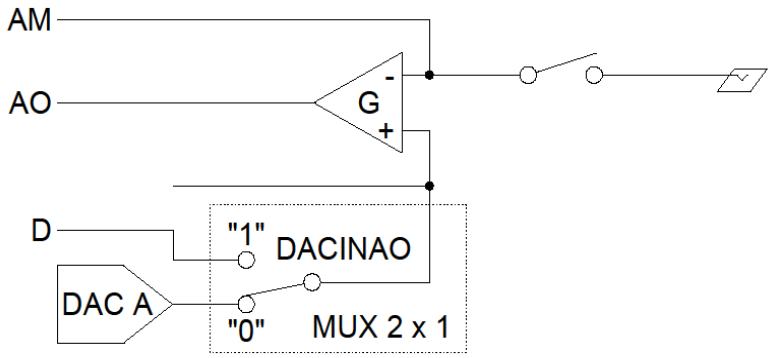


Figure 24: External Voltage Reference for Amplifier A

3.2.7 Sigma Delta ADC

A sigma delta analog-to-digital converter (SDADC) with bipolar input producing a 17-bit two’s complement result is used to quantify the output of the transimpedance amplifiers. Because there are two transimpedance amplifiers and their output voltages are referenced to DAC voltages that vary, the ADC supports differential input and have a differential input multiplexer.

To assure good signal to noise ratio in the SDADC, the maximum transimpedance amplifier differential voltage is ± 1.0 V, subject to the condition that each of the two voltage inputs to the ADC is less than 3.25 V. Firmware has the ability to select conversion times (or, equivalently, sampling rate) between approximately 0.25 ms and 32 ms in power of two steps (i.e. 0.25 ms, 0.5 ms, 1 ms, 2 ms, 4 ms, 8 ms, 16 ms, and 32 ms).

In single sample mode, the SDADC presents a single output codeword representing the average value over the preprogrammed conversion time to the firmware.

In stream mode, the SDADC produces a sequence of output codewords, each of which represents the average value over the preprogrammed conversion time to the firmware. The sequence is initiated and terminated under firmware control.

The sampling window is defined as:

$$T_{sampleWindow} = \frac{256 \times 2\text{^}\{SCALE\_SEL[2:0]\}}{F_{SDADC\_CLK}}$$

There is a fixed one-time latency of 512 ADC clock cycles from the point `START_SAMPLE` is asserted by firmware before the sample window whose conversion result shall be presented to firmware. In stream mode, this latency only applies to the first sample that is produced.

Therefore, the sampling rate in stream mode is $1/T_{sampleWindow}$

For instance, if the system is configured as follows

- `CKSEL[1:0] = 2’b10` (selects the PLL/4 clock)
- `PLLMTRM[10:0] = ’d1536` (sets PLL feedback integer divider)
- `adc_sclk_ctr` = `1’b1` (sets the SDADC-specific clock divider to divide by 12)

- `SCALE_SEL[2:0] = 3'b000` (lowest SDADC integration period)

Then the sampling rate shall be $1/T_{sampleWindow}$ = (32768*1536/4/12)/(256*2^0) = 4096 samples per second, and the accuracy of this sampling rate will be maximized, as it is now crystal-referenced.

The SDADC output is in 2's-complement 17-bit format. The bipolar SDADC input range allows for an opportunity to measure offsets and correct for them algorithmically, even if they are negative. This avoids the need to inject an artificial offset that may have been injected in previous systems by the use of a fixed resistor in parallel with the sensor.

The SDADC reference of 2.0 V is generated on chip and brought to the pin ADC_REF for off-chip decoupling and measurement. The SDADC reference must be enabled 20 ms before any ADC conversion is initiated, to allow it to settle.

The LSB size of the ADC in volts is given approximately by

$$LSB\ size = \frac{2 \times V_{ADC\_REF}}{2^{17} - 1}$$

Measuring and storing the value of the voltage of the ADC reference buffer on ADC_REF pin into eFlash during meter manufacturing allows for a simple method of calibrating the SDADC gain variation. Other methods of optimizing system level accuracy are also possible; please contact Indie Semiconductor for further details.

Conversions are always initiated under digital engine software control. An interrupt is sent to the digital engine when a conversion completes or an error is detected. The interrupt is cleared once the conversion result or error status has been read. A status register also allows the digital engine to poll to determine if the SDADC is conversion has been completed. The digital engine has the ability to terminate an analog-to-digital conversion in progress. Disabling the analog engine will terminate a conversion.

To minimize noise during conversions, the digital engine may be placed into an idle mode by firmware once the conversion is initiated. The SDADC completion interrupt provides a mechanism for resuming program execution.

When not required, software can disable the SDADC circuitry and SDADC reference independently of the remainder of the analog engine circuitry to minimize battery drain between conversions.

3.2.7.1 SDADC Registers

These registers control the SDADC and provide access to the SDADC result. These can be accessed as individual bytes. Four bytes, which are aligned on a word boundary, can also be read simultaneously as a 32-bit word (0x50000020 – 0x50000023).

3.2.7.1.1 SDADC Output/Status Register

ADC Data Output							
ADC16_DATA		0x50000020-23			0x00		
R	R	R	R	R	R	R	R
DSBUF7	DSBUF6	DSBUF5	DSBUF4	DSBUF3	DSBUF2	DSBUF1	DSBUF0
DSBUF15	DSBUF14	DSBUF13	DSBUF12	DSBUF11	DSBUF10	DSBUF9	DSBUF8
-	-	-	-	-	-	-	DSBUF16
-	-	-	-	-	-	-	FINISH
MSB							LSB
Bit31-25 Reserved Bit24 FINISH 0 = ADC conversion not complete							

1 = ADC conversion complete (an interrupt can be triggered when conversion completes) Bit23-17 Reserved Bit16-0 DSBUF[16:0]: ADC output in 17-bit 2's complement format

3.2.7.1.2 SDADC Configuration Registers

This register is used to configure operation of the sigma-delta A/D converter and to initiate a conversion.

ADC Configuration Register							
ADC16_CTR		0x50000024			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	SCALESEL[2]	SCALESEL[2]	SCALESEL[2]	STREAM	START_SAMPLE
MSB							LSB
Bit7-5 Reserved Bit4-2 SCALE_SEL[2:0]: ADC integration window interval in number of ADC_CLK cycles. 000 = 256 100 = 4096 001 = 512 101 = 8192 010 = 1024 110 = 16384 011 = 2048 111 = 32768 Bit1 STREAM: 0 = When START_SAMPLE is asserted, one ADC output word is produced 1 = When START_SAMPLE is asserted, ADC output words are produced continuously until START_SAMPLE is de-asserted by software (last conversion in progress is always completed) Bit0 START_SAMPLE: 2's-complement sign bit of ADC result Write 1 to initiate ADC conversion Write 0 to stop ADC conversions after current one (in stream mode)							

ADC Digital Configuration Register							
ADC16_DIGCONF		0x50000025			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
sample_edge	dithEN[1]	dithEN[0]	chopCtrl[1]	chopCtrl[0]	chopFreq[2]	chopFreq[1]	chopFreq[0]
MSB							LSB
Bit7-0 Reserved for device manufacturer use							

ADC Dither 0 Seed Register							
ADC16_DITH0		0x50000028-29			0x567A		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Seed0[7]	Seed0[6]	Seed0[5]	Seed0[4]	Seed0[3]	Seed0[2]	Seed0[1]	Seed0[0]
Seed0[15]	Seed0[14]	Seed0[13]	Seed0[12]	Seed0[11]	Seed0[10]	Seed0[9]	Seed0[8]
MSB							LSB
Bit15-0 Seed0[15:0] Dither 0 sequence generator seed (reserved for device manufacturer use)							

ADC Dither 1 Seed Register		
ADC16_DITH1	0x5000002A-2B	0xABCF

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Seed1[7]	Seed1[6]	Seed1[5]	Seed1[4]	Seed1[3]	Seed1[2]	Seed1[1]	Seed1[0]
Seed1[15]	Seed1[14]	Seed1[13]	Seed1[12]	Seed1[11]	Seed1[10]	Seed1[9]	Seed1[8]
MSB							LSB
Bit15-0 Seed1[15:0] Dither 1 sequence generator seed (reserved for device manufacturer use)							

ADC16 Power Management Register

ADC16_PMU		0x5000002C			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
adc enadc overRide	adc enadc regBit	adc reset	adc dccal	Reserved	adc enref	adc enbuf	-
MSB							LSB

Bit7 adc_enadc_overRide Over-ride digital control of when ADC modulator is enabled
0 = Digital state machine enables ADC modulator based on writing start_sample
1 = adc_enadc_regBit in this register controls when ADC modulator is enabled
Bit6 adc_enadc_regBit Enable ADC when adc_enadc_overRide in this register is enabled
0 = ADC modulator disabled
1 = ADC modulator enabled
Bit5 adc_reset ADC modulator reset – bit must be toggled in software to enter and exit reset
0 = ADC modulator is not reset
1 = ADC modulator is reset (shorts all feedback integration capacitors inside modulator)
Bit4 adc_dccal ADC DC calibration mode: input is disconnected from pins and shorted to ground
0 = Input is connected to pins
1 = Input is disconnected from pins and shorted to ground
Bit3 Reserved
Bit2 adc_enref ADC reference enable
0 = ADC reference disabled
1 = ADC reference enabled (wait 20 ms for analog buffer driving ADC_REF pin to settle), must be done before start_sample is asserted)
Bit1 adc_enbuf ADC input buffer enable
0 = ADC input buffers disabled
1 = ADC input buffers enabled (must be done before start_sample is asserted)
Bit0 Reserved

ADC16 Misc/Control Trim Register

ADC16_OVR		0x5000002D			0x9A		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
adc_isel [1]	adc_isel [0]	adc_selext	Reserved	Reserved	Reserved	Reserved	Reserved
MSB							LSB

Bit7-6 adc_isel[1:0] ADC modulator current trim (power vs. performance trade-off)
00 = 0.50*Pnom (1.0u)
01 = 0.75*Pnom (1.5u)
10 = 1.00*Pnom (2.0u) (default)
11 = 1.25*Pnom (2.5u)
Bit5 adc_selext Bypass input buffer (manufacturer use only)
0 = Input buffer in signal path

1 = Input buffer bypassed Bit4-0 Reserved

ADC16 Reference Trim Register							
ADC16_REF		0x5000002E			0x17		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
adc_clk_delay[1]	adc_clk_delay[0]	adc_g2x	adc_vbgsel	adc_vref[3]	adc_vref[2]	adc_vref[1]	adc_vref[0]
MSB							LSB
Bit7-6 adc_clk_delay[7:6] Delay ADC modulator sampling clock relative to ADC digital filter clock (reserved for device manufacturer use) 00 = No delay (default) Bit5 adc_g2x Enable double-sampling of ADC input (doubles the ADC modulator gain) 0 = Disabled 1 = Enabled Bit4 adc_vbgsel Choose reference generation mechanism for ADC reference buffer 0 = Vbg/R current selected as an input (converted to voltage locally by matching R) 1 = Vbg voltage directly selected as an input Bit3-0 adc_vref[3:0] ADC reference voltage trim 0000 1.62 V 0001 1.67 V 0010 1.71 V 0011 1.76 V 0100 1.82 V 0101 1.88 V 0110 1.94 V 0111 2.00 V (default) 1000 2.07 V 1001 2.14 V 1010 2.22 V 1011 2.31 V 1100 2.40 V 1101 2.50 V 1110 2.61 V 1111 2.73 V							

ADC16 Coefficient Trim Register							
ADC16_COEFF		0x5000002F			0xAA		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
coeff[7]	coeff[6]	coeff[5]	coeff[4]	coeff[3]	coeff[2]	coeff[1]	coeff[0]
MSB							LSB
Bit7-0 coeff[7:0] ADC modulator coefficient trim (reserved for device manufacturer use)							

3.2.8 Auxiliary 8-bit ADC

A port-connected auxiliary 8-bit analog to digital converter is provided for future system expansion. Its main features include:

- 8-bit resolution
- Single ended input
- Conversion rate of up to 80 KSPS
- Configurable reference (VREF = VREFHI – VREFLO)
- Choice of band-gap voltage or supply voltage (VSUP_DIG)
- Reference may be scaled in order to provide higher absolute resolution around a smaller input voltage range
- Maximum ADC input range is from 0 V to the supply voltage

It can read from a total of 27 channels (25 GPIOs, a PTAT reference to measure temperature, and the battery voltage after reverse battery protection)

This ADC uses a standard charge redistribution technique, with a single-ended input and internally generated positive and negative reference voltages. The ADC accommodates 27 analog input channels. The user can select which input channel to be sampled by setting the ADCCHANNELS register. The ADC has its own internally generated reference voltages (VREFHI and VREFLO).

There are several steps required for the user to use the ADC. The general sequence is described below:

1. Select the input channel to be measured
2. Setup the clock system, including the ADC clock divider
3. Configure references by programming the ADCREFHI, ADCREFLO, ADCPGN, and ADCREFS bits.
4. Start the ADC conversion
5. Check the ADC status bit and read the data

The following section will describe each configuration step in detail.

3.2.8.1 Input Channel Selection

GPIOs (PB[7:0], PC[7:0], PD[7:2], and PE[2:0]), the VBAT voltage after supply protection and resistive division by 2, and the PTAT reference voltage are available as inputs to the ADC. Firmware can control which input is connected for the conversion by programming the control bits, ADDCH[5:0], in the ADDCHANNEL register.

3.2.8.2 ADC Clock and Sampling Period

The conversion algorithm has a basic period of 9 cycles (one for sampling, one for each bit). There is a 2-cycle latency from the last bit measurement until the data becomes available to the user. Additionally, there is a single idle cycle to allow biasing before any conversion is initiated when starting from the converter in a disabled state, and another cycle to indicate conversion start. Thus a single conversion will take 9 + 2 + 1 + 1 = 13 cycles. In streaming mode, new samples will arrive every 9 clock cycles.

The converter uses a single clock cycle to sample the input into an input capacitor. When the channel is selected, the source must drive the sample/hold capacitor through the source resistance of the signal to be measured. The sampling time varies with this source resistance. The input to ADC has sufficiently low driving impedance and settling time to settle the input to within 1 LSB of the data conversion during the input sampling stage. An equivalent circuit and related equations are depicted in the Figure 25 below.

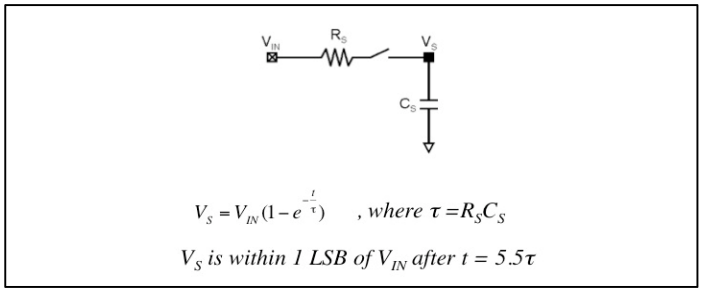


Figure 25: 8-bit ADC Input Settling Time

The ADC clock frequency can be programmed through the ADCCLKDIV register. As an example, if ADC clock is derived from a 12 MHz oscillator divided by 16, the input has 1.3 µs to settle. Since the maximum value of the internal sample/hold capacitor, Cs is 10 pF, the maximum source resistance Rs of the signal to be sampled to guarantee 8-bit performance can be calculated as below:

$$Rs = 1.3\mu / (10p \cdot 5.5) = 23.6\ K\Omega$$

If the source impedance is larger, the user can reduce the ADC clock frequency in order to retain the conversion accuracy.

3.2.8.3 ADC Reference Voltage Configuration

The ADC can generate its own reference voltages (VREFHI and VREFLO) from two different sources, its supply voltage (VDD) or an internal bandgap reference voltage (VBG). The ADCREFS bit in the ADCREG3 register selects the source. Once the reference source is selected, the reference voltages can be programmed through the ADCREFHI, ADCREFLO, and ADCPGN bits according to Figure 26.

ADCREFS=0*	ADCREFS=1
$VREFHI = \frac{ADCREFHI}{ADCPGN} \times VBG$	$VREFHI = \frac{ADCREFHI}{15} \times VDD$
$VREFLO = \frac{ADCREFLO}{ADCPGN} \times VBG$	$VREFLO = \frac{ADCREFLO}{15} \times VDD$
* If $\left(\frac{15}{ADCPGN} \times VBG\right) < (VDD - 0.1V)$	

Figure 26: 8-bit ADC Reference Voltage Equations

Once the reference voltages are established, the ADC data output ADCDT for input voltage VIN can be defined as:

$$ADCDT = floor\left(255 \times \frac{(VIN - VREFLO)}{(VREFHI - VREFLO)}\right)$$

Below are two examples:

1. In a system operating with VDD=3V, there is a signal that moves between 0V and 2.94V. In this case it is recommended that VDD be selected as the reference source and the following setting be made, ADCREFH=15, ADCREFL=0, and ADCPGN=15. This selection would allow for the maximum range of measurement (0V to VDD).

2. In a system operating with VDD=3V and VBG=1.21V, there is a signal that moves between 1.71V and 2.2V. In this case selecting VBG as the reference source and settings of ADCREFH=15, ADCREFL=11, and ADCPGN=8 we can achieve higher resolution. The resolution in this case is:

$$RESOLUTION = \left(\frac{VREFHI - VREFLO}{255}\right) = \left(\frac{\left(\frac{15}{8} \times 1.21\right) - \left(\frac{11}{8} \times 1.21\right)}{255}\right) = \frac{2.27 - 1.66}{255} = 2.38mV$$

Note that in this particular case, the 8-bit ADC effectively generating a digital value with the precision of a 10-bit ADC operating from 0V to VDD.

It is clear from the examples how flexible the ADC can be in a range of applications. The user can devise several schemes to cleverly measure the range of signal of interest and then narrow the reference values to get the optimum resolution if the conversion time is acceptable.

3.2.8.4 ADC Start and Status

Before starting the conversion, the ADC must be enabled and biased. The ADC is enabled by the ADCEN bit in register ADCREG3. The START bit (ADCSTART) starts the conversion process. Once completed the value of the conversion is loaded into the ADCDATA register.

3.2.8.5 Auxiliary 8-bit ADC Registers

The following registers control the behavior of the ADC:

Table 3.31 Auxiliary 8-bit ADC registers

Address	Register Name	Description	Reset Value
0x50000030	ADCCHANNELS	ADC channel select register	
0x50000031	ADCSTART	ADC conversion start	
0x50000032	ADCDATA	ADC data output	
0x50000033	ADCCLKDIV	ADC clock divider	
0x50000034	ADCTRIM0	ADC trim setting	0xF0
0x50000035	ADCTRIM1	ADC trim setting	0x9F
0x50000036	ADCTRIM2	ADC trim setting	0x00

ADC Channel Select Register							
ADCCHANNELS		0x50000030			0x00		
Reserved	Reserved	R/W	R/W	R/W	R/W	R/W	R/W
-	-	ADCCH5	ADCCH4	ADCCH3	ADCCH2	ADCCH1	ADCCH0
MSB							LSB

Bit5-0 ADCCH[5:0]: ADC Channel Select
000000 – 000111 = Reserved
001000 – 001111 = PB0 through PB7
010000 – 010111 = PC0 through PC7
011010 – 011111 = PD2 through PD7
100000 – 100011 = PE0 through PE2
101000 = VBAT/2 (resistive divider on VBAT after reverse supply protection)
101001 = PTAT (for temperature measurement)

ADC Start Register							
ADCSTART		0x50000031			0x00		
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	R/W
-	-	-	-	-	-	-	START
MSB							LSB

Bit0 START: Writing one starts the conversion. Reading returns the status of conversion; ‘0’ means conversion
Is finished and ‘1’ means the conversion is ether pending or in progress

ADC Result Register							
ADCDATA		0x50000032			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADCDT7	ADCDT6	ADCDT5	ADCDT4	ADCDT3	ADCDT2	ADCDT1	ADCDT0

MSB							LSB
Bit7-0 ADCDT[7:0]: ADC Result							

ADC Clock Divider Control Register							
ADCCLKDIV		0x50000033			0x6F		
Reserved	R/W	Reserved	Reserved	R/W	R/W	R/W	R/W
-	-	-	-	-	ADCDIV1	ADCDIV0	ADCCONT
MSB							LSB
Bit2-1 ADCDIV[1:0]: ADC Clock divider 00 = System Clock/8 01 = System Clock/16 10 = System Clock/32 11 = System Clock/64 Bit0 ADCCONT: ADC Stream Mode 0 = Single Conversion 1 = Streaming Mode							

ADC Trim0 Register							
ADCTRIM0		0x50000034			0xF0		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADCREFHI3	ADCREFHI2	ADCREFHI1	ADCREFHI0	ADCREFLO3	ADCREFLO2	ADCREFLO1	ADCREFLO0
MSB							LSB
Bit7-4 ADCREFHI[3:0]: ADC Reference High Setting Bit3-0 ADCREFLO[3:0]: ADC Reference Low Setting							

ADC Trim1 Register							
ADCTRIM1		0x50000035			0x9F		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADCSW1	ADCSW0	ADCCAL	ADCREFS	ADCPGN3	ADCPGN2	ADCPGN1	ADCPGN0
MSB							LSB
Bit7-6 ADCSW[1:0]: ADCs Enable Bit. 00 = Correlated double sampling off 01 = Input offset calibration on 1x = Correlated doubling sampling on (default) Bit5 ADCCAL: ADC Calibration 0 = Normal 1 = Calibration mode Bit4 ADCREFS: ADC Internal Reference Source Selection 0 = Band Gap 1 = VDD Bit3-0 ADCPGN[3:0]: ADC Reference Gain							

ADC Trim2 Register							
ADCTRIM2		0x50000036			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
ADCSCYC2	ADCSCYC1	ADCSCYC0	ADCENFORCE	ADC_ENABLE	Reserved	Reserved	ADCGNDOFF
MSB							LSB
Bit7-5 ADCSCYC[2:0]: ADC Sampling cycle (Sampling period = ADCSCYC + 1 cycles) Bit4 ADCENFORCE: Manual Sampling 0 = Normal mode 1 = Force data sampling Bit3 ADC_ENABLE: ADC reference buffer enable 0 = ADC reference off 1 = ADC reference on (must be 1 if using ADC8) Bit2-1 Reserved Bit0 ADCGNDOFF: Ground offset correction 0 = Bandgap and ADC reference have common ground 1 = difference between bandgap and ADC reference is compensated by switched cap circuit							

3.2.9 Power Management Unit

The ASIC implements a power management unit providing both reverse battery protection and an on-chip max supply selector between CR2032 battery and USB power. This allows system to boot with a discharged or non-existent CR2032 battery, if USB is available. A diagram is shown in Figure 27 below detailing these features.

Power Management primary characteristics:

- Reverse battery protection for VBAT
- On chip max supply selector between battery and USB power
  VSUP = max(VBAT, USB_3p3V_OUT)
- Seamless switching between USB power supply and CR2032 battery supply
- HW reset – resets all registers and the processor
- SW reset – resets processor, but preserves state of all registers in address space 0x500X_XXXX
- Firmware triggered deep sleep mode

Table 3.32 Power Management Signals

Pin Name	Description
RESET_B	Active-low external reset input (with internal pull-up)
VBAT	Supply Voltage Input – This is the positive voltage input for the device, battery or regulated. LCD and analog charge pump supply voltages are derived from this input. This input has reverse-battery protection circuitry. A 10 µF decoupling capacitor should be connected to this pin with provision for parallel 0.1 µF and 1 nF capacitors.
VREG_MCU_PAD1	1.8V Regulated Voltage Output – Used to connect regulated 1.8V supply voltage used by microcontroller core to an external 0.1 µF decoupling capacitor.
GND	Power Return – Return for VBAT and VDDA pins.
VDDA	Regulated Analog Charge Pump Voltage Output – For external 0.1 µF decoupling of the regulated analog supply voltage derived from the ACP_OUT analog charge pump output.
VDDA_FILT	Filtered VDDA input from VDDA decoupling capacitor.
VFLASH_SEL	Flash voltage supplied to MCU.
VSUP	Supply selector output. Bypass with 10 µF, 0.1 µF, and 1 nF capacitors.
VSUP_ANA	Supply selector output route-back for AFE. Bypass with 0.1 µF capacitor.
VSUP_DIG	Supply selector output route-back for Digital

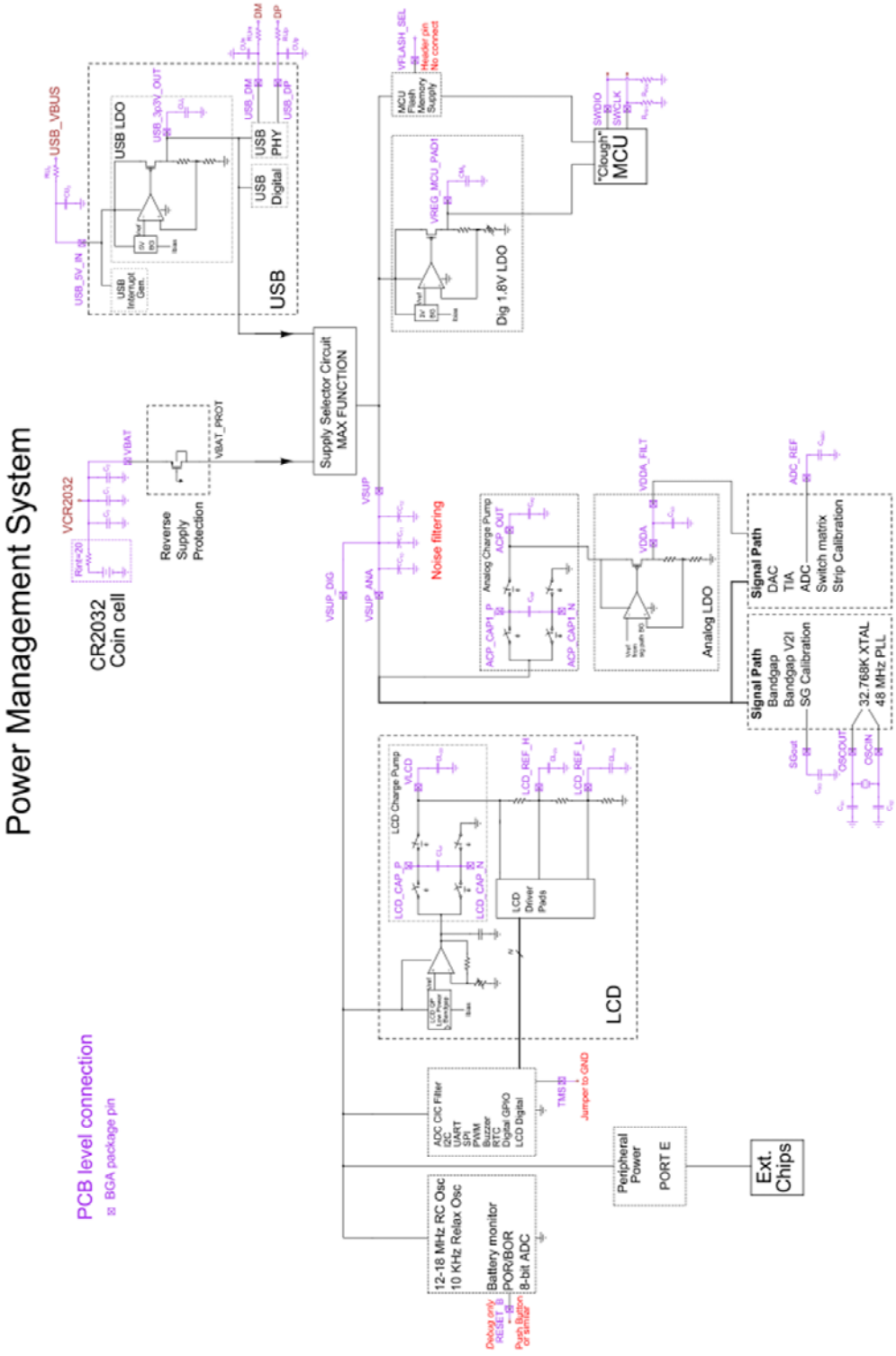


Figure 27: Power Management System Diagram

3.2.9.1 Reset

There are two forms of reset that can be issued:

- Hardware reset: After this reset is issued, the 10 kHz clock source is selected and all other clock sources are disabled.
- Software reset: After this reset is issued, the clock setup is kept unchanged along with the brownout selection.

3.2.9.2 Brown-Out Reset

The PMU controls the Brownout Reset. By default, a brown-out will reset the system. Firmware can configure this behavior and select the brown-out reset threshold upon first power up using the PMUBOR register.

3.2.9.3 Deep Sleep mode

The PMU can set the system into deep sleep mode, which is the lowest current consumption mode available in the system. In deep sleep mode:

- The Clough CPU is halted and its power supply disabled
- Any enabled clock source will continue to operate
- The three timers (Timer0, Timer1, and Timer 2) and the SysTick Timer will stop operating
- Timer3 will continue to operate
- All other peripherals will keep running (if enabled and fed by their required source clock)
- The system will leave the deep sleep mode through a Power On Reset, the wakeup timer, or any peripheral that generates an interrupt independently of the interrupt being enabled by the Nested Vector Interrupt Controller

Note: For those peripherals that have a bit in their registers that locally enables an interrupt, this register has to be enabled in order to reset the system. Example: For the GPIO ports PORTB, PORTC, PORTD, and PORTE, the INTE bits of the I/O pins selected to reset the part upon change must be set.

Note 2: All pending interrupts must be serviced/cleared by firmware before deep sleep mode can be entered successfully.

3.2.9.4 Analog Boost Regulator

To support operation of the analog engine above the specified supply voltage, VBAT, range, the analog engine includes an analog boost regulator with regulated outputs used to power those portions of the analog circuitry that cannot be operated at battery voltage. This analog boost regulator can be enabled and disabled under firmware control. When the analog boost regulator is disabled, its output is equal to the battery voltage VBAT. When the analog boost regulator is enabled, the output is regulated to about 3.6 V to support analog engine operation with signal levels up to 3.25 V at transimpedance amplifier outputs. The allowable charge pump input voltage range is equal to the VBAT range of 2.2 V to 3.3 V. Pins are provided for external decoupling of the regulated analog charge pump output, VDDA.

The pin VDDA is to be connected to VDDA_FILT on the PCB with a decoupling capacitor (e.g. 100 nF) placed close to the VDDA_FILT pin for noise filtering.

Table 3.33 Analog Boost Regulator Signals

Pin Name	Description
ACP_CAP1_N	Analog Boost Regulator Capacitor - – Analog engine interface pin connected to ACP_CAP1_P with an external 0.039 µF charge pump capacitor and used to step up voltage for use by the analog circuitry.
ACP_CAP1_P	Analog Boost Regulator Capacitor + – Analog engine interface pin connected to ACP_CAP1_N with an external 0.039 µF charge pump capacitor and used to step up voltage for use by the analog circuitry.
ACP_OUT	Analog Boost Regulator Output – Analog engine raw boost regulator voltage output. To be connected to a 0.1 µF external holding capacitor.

3.2.9.5 Supply Voltage Level Detection

The present battery voltage can be monitored using an input channel of the auxiliary 8-bit ADC (or the sigma delta ADC).

3.2.9.6 Power Management Registers

The following registers are used to control and determine the status of power management functions

3.2.9.6.1 Processor/Digital Power Management Registers

Processor Control/Status Register							
PMURST		0x50000001			0x01		
W	W	R/W	R	R	R	R	R/W
HWRST	SWRST	DSLEEP	RCMON	USBF	WAKEUPF	BOF	PORF
MSB							LSB

Bit7 **HWRST:** Hardware reset
0 = Idle
1 = Hardware reset (automatically cleared after reset process completed)

Bit6 **SWRST:** Software reset
0 = Idle
1 = Software reset (automatically cleared after reset process completed)

Bit5 **DSLEEP:** Deep sleep (HALT) mode
Writing:
0 = Clear deep sleep flag
1 = Put the system in deep sleep – Halt (all interrupts must be cleared before writing this bit to a 1)
Reading:
0 = Flag cleared
1 = System in deep sleep mode

Bit4 **RCMON:** Main RC Oscillator Monitor
0 = Main RC Oscillator inactive
1 = Main RC Oscillator active

Bit3 **USBF:** USB flag
0 = USB power not detected
1 = USB power is detected

Bit2 **WAKEUPF:** Wake-up flag
0 = Wake-up event not detected
1 = Wake-up event is detected

Bit1 **BOF:** Brownout flag
0 = No brownout
1 = Brownout

Bit0 **PORF:** Power-On Reset flag
Writing:
0 = Clear POR
1 = No effect
Reading:
0 = POR flag already cleared by application
1 = The system just came out of POR or HW Reset

Brownout Reset Control Register							
PMUBOR		0x50000002			0x03		
R/W	R/W	R/W	Reserved	R/W	R/W	R/W	R/W
BOREN	BORRST	BORINT	-	BOR_TRIM3	BOR_TRIM2	BOR_TRIM1	BOR_TRIM0
MSB							LSB

Bit7 **BOREN:** Brownout enable
0 = Brownout disabled
1 = Brownout enabled

Bit6 **BORRST:** Brownout Reset enable
0 = Disable Brownout-based reset
1 = Enable Brownout-based reset

Bit5 **BORINT:** Brownout Interrupt
0 = Brownout interrupt disabled
1 = Brownout interrupt enabled

Bit4 *Reserved*

Bit3-0 **BOR_TRIM[3:0]:** Brownout threshold value

0000	1.75 V	1000	2.13 V
0001	1.79 V	1001	2.19 V
0010	1.83 V	1010	2.26 V
0011	1.87 V	1011	2.33 V
0100	1.92 V	1100	2.40 V
0101	1.97 V	1101	2.56 V
0110	2.02 V	1110	2.74 V
0111	2.08 V	1111	2.95 V

3.2.9.6.2 Analog Front-End Power Management Registers

The bits defined in register APCTR0 below are provided to selectively power down sensor interface circuits. In an SG referenced system, it is assumed that the bit ACP_EN has been set before the DAC and TIA are enabled.

Supply bias/status for AFE Register							
PMUSUPPLY		0x50000006			0x00		
R	Reserved	Reserved	Reserved	Reserved	Reserved	R/W	R/W
V2ION_INDIC	-	-	-	-	-	ANA_LDO_EN	ACP_EN
MSB							LSB

Bit7 **V2ION_INDIC:** Read-only indicator for main AFE current converter bias
0 = V2I disabled
1 = V2I enabled
Bit6-2 *Reserved*
Bit1 **ANA_LDO_EN:** Analog regulator enable
0 = Analog regulator (for VDD) disabled
1 = Analog regulator (for VDD) enabled
Bit0 **ACP_EN:** Analog charge pump enable
0 = Analog charge pump disabled (acp_clk also disabled)
1 = Analog charge pump enabled (acp_clk also enabled)

Analog Power Control Register 0							
APCTR0		0x50000058			0xFD		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
TIA_A_EN	TIA_G_EN	DAC_A_EN	DAC_G_EN	BG_CAL_EN	BG_BUF_EN	BG_V2I_EN	BG_EN
MSB							LSB

Bit7 TIA_A_EN: TIA A power enable
0 = TIA A disabled
1 = TIA A enabled
Bit6 TIA_G_EN: TIA G power enable
0 = TIA A disabled
1 = TIA A enabled
Bit5 DAC_A_EN: DAC A power enable
0 = Battery monitoring low-battery circuitry enabled
1 = Battery monitoring low-battery circuitry enabled
Bit4 DAC_G_EN: DAC G power enable
0 = Battery monitoring reset circuitry disabled
1 = Battery monitoring reset circuitry enabled
Bit3 BG_CAL_EN: Bandgap calibration circuit power enable
0 = Calibration circuit disabled
1 = Calibration circuit enabled
Bit2 BG_BUF_EN: Bandgap buffer circuit power enable
0 = Bandgap voltage-to-current converter disabled
1 = Bandgap voltage-to-current converter enabled
Bit1 BG_V2I_EN: Bandgap-based AFE current bias generator power enable
0 = Bandgap buffer disabled
1 = Bandgap buffer enabled
Bit0 BG_EN: Precision bandgap power enable
0 = Precision bandgap disabled
1 = Precision bandgap enabled

Analog Power Control Register 1							
APCTR1		0x50000059			0x80		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
-	-	-	-	-	-	-	-
MSB							LSB
Bit7-0 Reserved for device manufacturer’s internal use							

3.3 DESIGN FOR TEST FEATURES

To help characterize and debug the system, several design-for-test features are implemented for both the digital and analog engines. This allows bringing internal signals to pins.

RTSELA output comes to pin GPIOC2.

RTSELB output comes to pin GPIOC3.

Pins GPIOC2 and GPIOC3 are designated to bring out digital test signals. To observe these signals, Port C registers should be configured to make GPIOC2 and GPIOC3 pins become digital outputs (i.e. enable output enables, and disable pull-ups, pull-downs, and read-enables for these pins).

Real-time Output Register							
RTO		0x50000005			0x00		
Reserved	Reserved	R/W	R/W	R/W	R/W	R/W	R/W
-	-	RTSELA2	RTSELA1	RTSELA0	RTSELB2	RTSELB1	RTSELB0
MSB							LSB

```
Bit7-6  Reserved
Bit5-3  RTSELA[2:0]: Real-time Output A Select
        000 = None.  Normal pin function.
        001 = 10 kHz oscillator output (divided by 2), expect 5 kHz at the pin
        010 = Main RC oscillator output
        011 = Crystal oscillator clock output (divided by 2), expect 16.384 kHz at the pin
        100 = PLL output clock (undivided)
        101 = Sigma-delta ADC modulator output clock
        110 = Sigma-delta ADC dither output 0
        111 = Brownout reset mask
Bit2-0  RTSELB[2:0]: Real-time Output B Select
        000 = None.  Normal pin function.
        001 = Full rate clock
        010 = Clock provided to MCU
        011 = Clock provided to USB IP
        100 = Clock provided to analog charge pump
        101 = Sigma-delta ADC modulator output data (single bit stream)
        110 = Sigma-delta ADC dither output 1
        111 = Real-time ADC8 output
```

4.0 ELECTRICAL CHARACTERISTICS

4.1 ELECTRICAL CONDITIONS

4.1.1 Absolute Maximum ratings

Table 4.34 Absolute maximum ratings

Parameter	Descriptions	MIN	MAX	UNIT
VBAT	Supply voltage	-3.6	3.6	V
VUSB		-0.3	10	V
Input Voltage	MCU interface	-0.3	VBAT+0.3	V
Storage Temperature Range		-40	+85	°C

4.1.2 Operating conditions

Table 4.35 Operating Conditions

Parameters			MIN	TYP	MAX	UNIT
VBAT	Supply voltage	Per analog performance specifications	2.2	3.0	3.3	V
VUSB		USB supply voltage	4.4	5.0	5.25	V
TA	Operating free-air temperature	Per analog performance specifications	0		50	°C
		Real-time clock circuitry functional	-40		85	°C
TCAL	Calibration temperature		22	25	28	°C
	Humidity range		10%		95%	RH

4.1.3 Electrical Characteristics over Operating Range

4.1.3.1 Supply Current

Table 4.36 Supply current

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
ISUPPLY	Supply current					
	(Analog circuitry enabled, digital engine active w/ 4MHz clock [MO Core 160 µA/MHz plus 100 µA supply bias], peak)	2.2V≤VBAT≤3.3V			6	mA
	(Analog circuitry enabled, digital engine active w/ 4MHz clock [MO Core 160 µA/MHz plus 100 µA supply bias], average)	2.2V≤VBAT≤3.3V			3	mA
	(Analog circuitry disabled, no pull-ups loaded, digital engine active w/ 1MHz clock [MO Core 160 µA/MHz plus 100 µA supply bias])	2.2V≤VBAT≤3.3V 0°C≤TA≤50°C			1.5	mA
ILCD	LCD driver circuit current draw, 32.768 KHz oscillator operation	2.2V≤VBAT≤3.3V 0°C≤TA≤50°C		70		µA
IANLG	Analog circuitry current draw when enabled	2.2V≤VBAT≤3.3V 0°C≤TA≤50°C			2	mA
ISTBY	Standby Current (Analog circuitry disabled, no pull-ups loaded, digital engine deep sleep, RTC, PWM and peripherals disabled)	2.2V≤VBAT≤3.3V 0°C≤TA≤50°C		1.1		µA
IRTC	Real-time clock circuit current draw, 32.768 KHz oscillator operation	2.2V≤VBAT≤3.3V 0°C≤TA≤50°C		0.9		µA
RRBP	Reverse-battery protection circuit ON impedance. Guaranteed by design.	1.8V≤VBAT≤3.3V 0°C≤TA≤50°C 0mA≤ISUPPLY≤10mA			5	Ω
tCPSU	Analog charge pump start-up time	2.2V≤VBAT≤3.3V 0°C≤TA≤50°C			2	ms

4.1.3.2 Supply Voltage Monitor

Table 4.37 Supply Voltage monitor

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
VlowBatt	Low-battery Voltage (VBAT referenced to GND)	0°C≤TA≤50°C		2.4		V
Vbrown-out	Brown-out (Dead-battery) Voltage (VBAT referenced to GND) Interrupt Level, Post Trim	-40°C≤TA≤85°C		2.2		V

4.1.3.3 General Purpose I/O

(-40°C≤T_A≤85°C, 1.8V≤VBAT≤3.3V)

Table 4.38 General Purpose I/O

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
Outputs:						
VOL	Low-level output voltage	@ IOL minimum			0.1 x VBAT	V
VOH	High-level output voltage	@ IOH minimum	0.9 x VBAT			V
IOL	Low-level output current	VOL=0.1 x VBAT		2		mA
IOLPWM	Low-level output current, PWM Output	VOL=0.1 x VBAT		10		mA
IOH	High-level output current	VOH=0.9 x VBAT		2		mA
IOHPP	High-level output current, Peripheral power pin, PIO_PP10, PIO_PP11, PIO_PP20, & PIO_PP21.	VOH=0.9 x VBAT		10		mA
Inputs:						
VIL	Maximum low-level input voltage				0.3 x VBAT	V
VIH	Minimum high-level input voltage		0.7 x VBAT			V

4.1.3.4 Buzzer Driver

(BUZZER_HI & BUZZER_LO, 0°C≤TA≤50°C, 2.2V≤VBAT≤3.3V)

Table 4.39 Buzzer Driver

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
VBUZZDRV	Buzzer drive voltage (leg relative to GND)			tbd		V
RBUZZDRV	Buzzer series impedance (per leg, external to the device)	-40°C≤TAA≤85°C 0°C≤TA≤50°C	tbd		 tbd	Ω Ω
fBUZZDRV	Buzzer frequency	CPIEZO=70000pF ±30%	900		2000	kHz
VPPIEZO	Piezo generated voltage withstand voltage	CPIEZO=70000pF ±30%, 50µsec rise, 200µsec duration			±100	V
CPIEZO	Piezo capacitance range		3500	70000	100000	pF

4.1.3.5 Source DAC Voltage

(A thru H, RR, TH1)

Table 4.40 Source DAC Voltage

	PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
VSG	Analog Common Voltage, SG, Post Calibration (Referenced to GND)	VBAT=2.75V 20°C≤TA≤30°C	1.01	1.1	1.2	V
VSGSTEP	Analog Common Voltage, SG, Step Size	VBAT=2.75V 20°C≤TA≤30°C		0.2		mV
VDDA	Regulated Analog Supply Voltage, Enabled (Referenced to GND)	VBAT=2.75V 20°C≤TA≤30°C	3.5			V
VSFS	Full-scale Source DAC Voltage (Referenced to GND)	Fine DAC code=FFH Course DAC code =3FH 2.2V≤VBAT≤3.3V 0°C≤TA≤50°C	2.5	3.0		V
	Recommended Source DAC Voltage Range (Referenced to GND) for measurements and offset readings		1.2		2.5	V
VDACSTEP	Source DAC Voltage Step Size	VBAT=2.75V 20°C≤TA≤30°C		0.37	1	mV
ISMAX	Maximum source current	2.2V≤VBAT≤3.3V 0≤VOUT≤VSFS 0°C≤TA≤50°C	250	300		µA
	Programming Resolution					
	Coarse DAC		6			bits
	Fine DAC		8			bits
	Effective linearity, best-fit straight line	2.2V≤VBAT≤3.3V 0°C≤TA≤50°C			±0.75	LSB
ΔVS	Worst-case source voltage error at sensor (driven pin to an analog common pin) versus manufacturing target calibration voltage over device life and operating extremes	2.2 V ≤VBAT≤ 3.3 V 0°C ≤TA≤ 50°C 0 ≤IS≤ 150 µA Worst-case offsets Worst-case gain errors	-5		5	mV
		2.2 V ≤VBAT ≤3.3 V 5°C ≤TA≤ 40°C 200 mV ≤VS≤ 500 mV 0 ≤IS≤ 8 µA Typical offsets and gain errors		< ±2.5		mV
ΔVS25,WC	Worst-case source voltage error at sensor (driven pin to GND referenced pin) versus manufacturing target calibration voltage of 2.5 V over device life and operating extremes	2.2 V ≤VBAT≤ 3.3 V 0°C ≤TA≤ 50°C 0 ≤IS≤ 250 µA Worst-case offsets Worst-case gain errors	-25		25	mV

Table 4.40 Source DAC Voltage						
	PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
ΔVS25	Typical-case source voltage error at sensor (driven pin to GND referenced pin) versus manufacturing target calibration voltage of 2.5 V over device life and operating extremes	2.2 V ≤VBAT≤ 3.3 V 5°C ≤TA≤ 40°C 0 ≤IS≤ 80 μA Typical offsets and gain errors		< ±4		mV
ΔVSL	Load regulation at fixed VBAT and TA. Guaranteed by design.	DAC code=3FFH 2.2V≤VBAT≤3.3V 0°C≤TA≤50°C 0≤IS≤150 μA	-2.5		2.5	mV
ISLOFF	Source voltage OFF leakage current (per output channel) Guaranteed by design.	0≤VOUT≤VSFS 2.2V≤VBAT≤3.3V 0°C≤TA≤50°C	-10	< ±2	10	nA
tsw	Source voltage switching time, 3 time constants	0≤VOUT≤VSFS 2.2V≤VBAT≤3.3V 0°C≤TA≤50°C			1	ms

4.1.3.6 Transimpedance Amplifier

Table 4.41 Transimpedance Amplifier						
	PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
VIO	Input offset voltage	0°C≤TA≤50°C			±0.5	mV
IIB	Input bias current Guaranteed by design.	0°C≤TA≤50°C	-0.1		0.1	nA
	Maximum amplifier output voltage		3.25			V
	Output load current capability		-3		3	mA

4.1.3.7 Switch Matrix

Table 4.42 Switch Matrix						
	PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
RGSW	Switch resistance for switches connecting to GND	2.2V≤VBAT≤3.3V 0°C≤TA≤50°C		60	90	Ω

4.1.3.8 Sigma Delta ADC

Table 4.43 Sigma Delta ADC

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
	Resolution (bipolar two’s complement result)		17			bits
	SNR	Integration window = 16384 ADC clock cycles of 0.5 MHz		116		dB
	Flicker-free Code Resolution	Integration window = 16384 ADC clock cycles of 0.5 MHz		19		bits
		Integration window = 1024 ADC clock cycles of 0.5 MHz		16.2		bits
	Sampling frequency			0.5		MHz
	Integral non-linearity				6	bits
	Quantization error				0.75	LSB
Tint	A/D conversion time (selectable, power of 2 steps)	2.2V≤VBAT≤3.3V -40°C≤TA≤85°C	1.024		65.536	ms
	Differential input voltage range, minimum	2.2V≤VBAT≤3.3V 0°C≤TA≤50°C	-0.75		0.75	V
	Common mode input voltage range	2.2V≤VBAT≤3.3V 0°C≤TA≤50°C	3.25			V
	ADC gain error	2.2V≤VBAT≤3.3V 0°C≤TA≤50°C	-3.5		3.5	%FS
	ADC start-up time	2.2V≤VBAT≤3.3V 0°C≤TA≤50°C	1000			µs

4.1.3.9 LCD Characteristics

(1/3 bias method,, 0°C≤TA≤50°C, 2.1V≤VBAT≤3.3V)

Table 4.44 LCD

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
VLCD	LCD drive voltage		3.2	4.0	4.5	V
VBAT	Battery Voltage		2.2	3.0	3.3	V
CL	Load Capacitance, 4x30 LCD segments, 100 pF per LCD segment				3	nF
Settling time	99% rise time to 2/3VLCD and 1/3VLCD				180	ms

fLCDframe	Frame frequency (4 time slices per frame)		8	32	64	Hz

5.0 PACKAGE

Ernie Ics are packaged within a 120 balls, 8x8mm BGA, 0.65 mm pitch for low cost manufacturing. Figure 28 shows the ball assignment. Table 5.45 describes each signal. Please refer to individual sections of the spec for further details.

5.1 BALL ASSIGNMENT

	1	2	3	4	5	6	7	8	9	10	11
A	LCD_REF_H	LCD_SEG29	LCD_SEG28	LCD_SEG23	LCD_SEG19	LCD_SEG15	LCD_SEG12	LCD_SEG9	LCD_SEG6	LCD_SEG3	LCD_SEG1
B	LCD_REF_L	LCD_COM3	LCD_SEG27	LCD_SEG22	LCD_SEG18	LCD_SEG14	LCD_SEG11	LCD_SEG8	LCD_SEG5	LCD_SEG2	LCD_SEG0
C	VLCD	LCD_COM2	LCD_SEG26	LCD_SEG21	LCD_SEG17	LCD_SEG13	LCD_SEG10	LCD_SEG7	LCD_SEG4	SPI_MO/PD3	SPI_MI/PD2
D	LCD_CAP_P	LCD_COM1	LCD_SEG25	LCD_SEG20	LCD_SEG16	VBAT	VBAT	SCL/PD0	SDA/PD1	SPI_CS-L/PD5	SPI_CK/PD4
E	LCD_CAP_N	LCD_COM0	LCD_SEG24	USB_5V_IN	USB_5V_IN	GND	USB_3p3V_OUT	TXD/PD6	RXD/PD7	PE2	VSUP_DIG
F	ACP_OUT	ACP_CAP1_P	ACP_CAP1_N	VREG_MCU_PAD1	VSUP	NC	GND	PC0/CS2-L	PE0	PE1	GND
G	USB_DM	SWDIO	GND	VFLASH_SEL	GND	GND	PC1/CS3-L	PC2/CS4-L	PC3/CS5-L	PC4/DETECT	PC5/BUZZER_LO
H	USB_DP	SWCLK	OSCOUT	OSCIN	GND	GND	GND	PC6/BUZZER_HI	PC7/PWM	PB0/SW1	PB1/SW2
J	GHH	GO	HM	TH1	VSUP_ANA	VDDA	VDDA_FILT	J	ADC_REF	PB2/SW3	PB3/SW4
K	GH	AO	HH	RR	G	A	C	F	TMS	PB4/SW5	PB5/SW6
L	GM	AM	TH2	SG	B	D	E	H	RESET_B	PB6/SW7	PB7/SW8

Figure 28: BGA Package Ball Assignment (Top View)

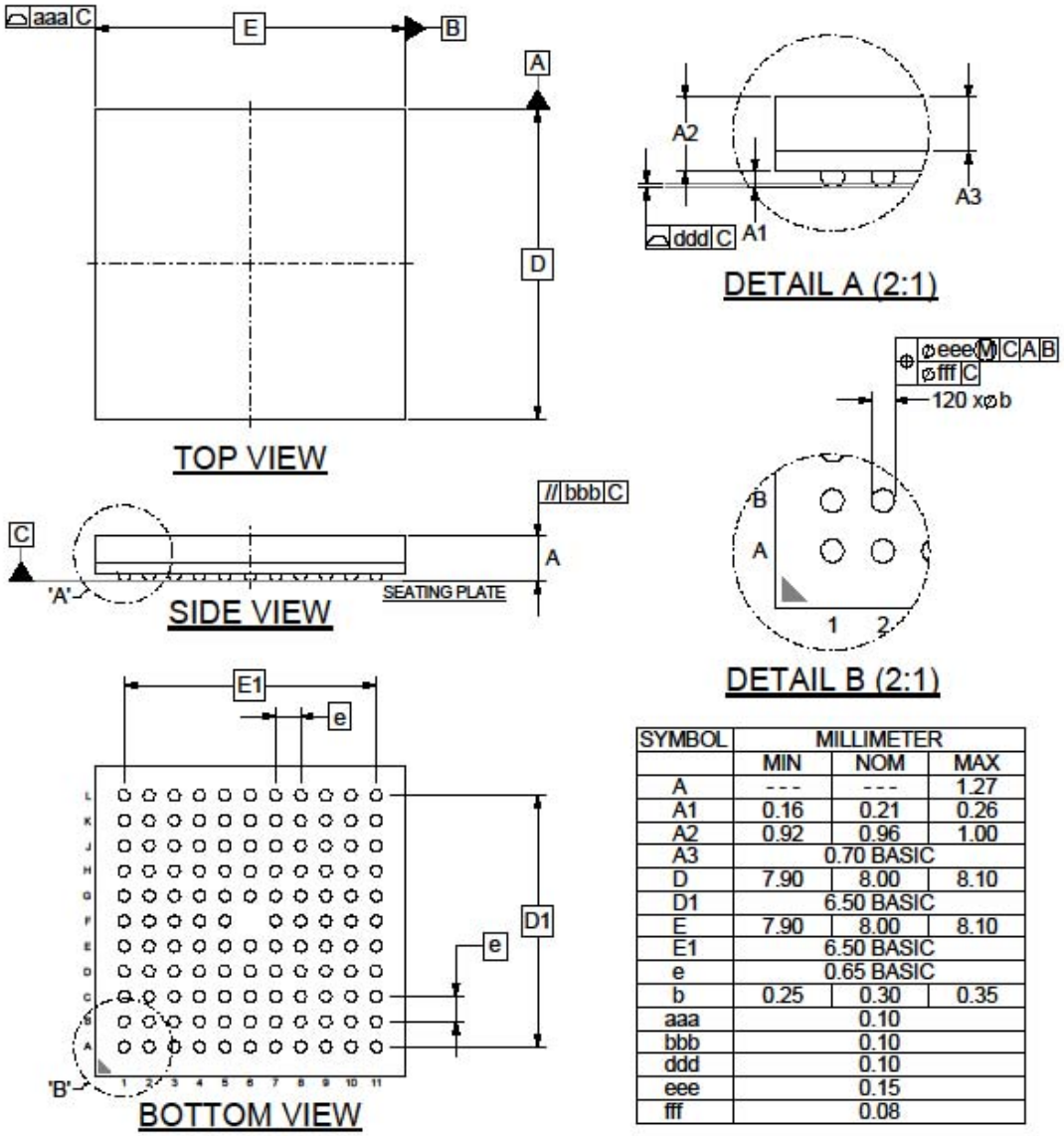
5.2 SIGNAL DESCRIPTIONS

Table 5.45 Package Signal Descriptions			
Ball #	**Signal Name**	**I/O**	**Functional Description**
E2	LCD_COM0	Digital O	LCD COM0 driver output
D2	LCD_COM1	Digital O	LCD COM1 driver output
C2	LCD_COM2	Digital O	LCD COM2 driver output
B2	LCD_COM3	Digital O	LCD COM3 driver output
B11	LCD_SEG0	Digital O	LCD SEG0 driver output
A11	LCD_SEG1	Digital O	LCD SEG1 driver output
B10	LCD_SEG2	Digital O	LCD SEG2 driver output
A10	LCD_SEG3	Digital O	LCD SEG3 driver output
C9	LCD_SEG4	Digital O	LCD SEG4 driver output
B9	LCD_SEG5	Digital O	LCD SEG5 driver output
A9	LCD_SEG6	Digital O	LCD SEG6 driver output
C8	LCD_SEG7	Digital O	LCD SEG7 driver output
B8	LCD_SEG8	Digital O	LCD SEG8 driver output
A8	LCD_SEG9	Digital O	LCD SEG9 driver output
C7	LCD_SEG10	Digital O	LCD SEG10 driver output
B7	LCD_SEG11	Digital O	LCD SEG11 driver output
A7	LCD_SEG12	Digital O	LCD SEG12 driver output
C6	LCD_SEG13	Digital O	LCD SEG13 driver output
B6	LCD_SEG14	Digital O	LCD SEG14 driver output
A6	LCD_SEG15	Digital O	LCD SEG15 driver output
D5	LCD_SEG16	Digital O	LCD SEG16 driver output
C5	LCD_SEG17	Digital O	LCD SEG17 driver output
B5	LCD_SEG18	Digital O	LCD SEG18 driver output
A5	LCD_SEG19	Digital O	LCD SEG19 driver output
D4	LCD_SEG20	Digital O	LCD SEG20 driver output
C4	LCD_SEG21	Digital O	LCD SEG21 driver output
B4	LCD_SEG22	Digital O	LCD SEG22 driver output
A4	LCD_SEG23	Digital O	LCD SEG23 driver output
A2	LCD_SEG24	Digital O	LCD SEG24 driver output
E3	LCD_SEG25	Digital O	LCD SEG25 driver output
D3	LCD_SEG26	Digital O	LCD SEG26 driver output
C3	LCD_SEG27	Digital O	LCD SEG27 driver output
B3	LCD_SEG28	Digital O	LCD SEG28 driver output
A3	LCD_SEG29	Digital O	LCD SEG29 driver output
C1	VLCD	Analog I/O	LCD charge pump 4.7 µF bypass capacitor connection.

Table 5.45 Package Signal Descriptions			
Ball #	**Signal Name**	**I/O**	**Functional Description**
A1	LCD_REF_H	Analog I/O	LCD bias network 0.47 µF capacitor connection.
B1	LCD_REF_L	Analog I/O	LCD bias network 0.47 µF capacitor connection.
E1	LCD_CAP_N	Analog I/O	LCD charge pump 1 µF charge pump capacitor connection, negative
D1	LCD_CAP_P	Analog I/O	LCD charge pump 1 µF charge pump capacitor connection, positive
C10	SPI_MO/PD3	Digital I/O	SPI data output (GPIOD3)
C11	SPI_MI/PD2	Digital I/O	SPI data input (GPIOD2)
D11	SPI_CK/PD4	Digital I/O	SPI clock output (GPIOD4)
E8	TXD/PD6	Digital I/O	UART transmitted data (GPIOD6)
E9	RXD/PD7	Digital I/O	UART received data (GPIOD7)
D9	SDA/PD1	Digital I/O	I2C serial data I/O (GPIOD1)
D8	SCL/PD0	Digital I/O	I2C serial clock output (GPIOD0)
G1	USB_DM	Digital I/O	USB data signal minus
H1	USB_DP	Digital I/O	USB data signal plus
E7	USB_3p3V_OUT	O Power	USB 3.3 V output with 1 µF decoupling capacitor
E4, E5	USB_5V_IN	I Power	USB 5V supply input / USB detection with 1 µF decoupling capacitor.
H9	PC7	Digital I/O	GPIOC7 (default PWM output)
H3	OSCOUT	Analog I/O	32.768 KHz oscillator output. Compensate with 10 pF capacitor.
H4	OSCIN	Analog I/O	32.768 KHz oscillator input. Compensate with 10 pF capacitor.
H8	PC6	Digital I/O	GPIOC6
G11	PC5	Digital I/O	GPIOC5
L10	PB6	Digital I/O	GPIOB6
L11	PB7	Digital I/O	GPIOB7
K10	PB4	Digital I/O	GPIOB4
K11	PB5	Digital I/O	GPIOB5
H10	PB0	Digital I/O	GPIOB0
H11	PB1	Digital I/O	GPIOB1
J10	PB2	Digital I/O	GPIOB2
J11	PB3	Digital I/O	GPIOB3
D10	SPI_CS-L/PD5	Digital I/O	GPIOD5 (Peripheral (SPI) CS output 1)
F8	PC0	Digital I/O	GPIOC0
G7	PC1	Digital I/O	GPIOC1
G8	PC2	Digital I/O	GPIOC2
G9	PC3	Digital I/O	GPIOC3
F9	PE0	Digital I/O	GPIOE0
F10	PE1	Digital I/O	GPIOE1
E10	PE2	Digital I/O	GPIOE2
G10	PC4	Digital I/O	GPIOC4
F4	VREG_MCU_PAD1	O Power	1.8V microcontroller core regulator 0.1 µF decoupling capacitor
D6,D7	VBAT	I Power	Supply voltage input with 10 µF, 0.1 µF, and 1 nF bypass capacitors.
J6	VDDA	O Power	Regulated analog supply 0.1 µF decoupling capacitor
E6,F7, F11,G3, G5,G6, H5,H6, H7	GND	Power	Power return(s) – Ground pins

Table 5.45 Package Signal Descriptions			
Ball #	**Signal Name**	**I/O**	**Functional Description**
F2	ACP_CAP1_P	Analog I/O	Step-up regulator 0.039 µF analog charge pump capacitor connection, positive
F3	ACP_CAP1_N	Analog I/O	Step-up regulator 0.039 µF analog charge pump capacitor connection, negative
F1	ACP_OUT	Power O	Analog charge pump output 0.1 µF holding capacitor
J9	ADC_REF	Analog I/O	ADC reference supply voltage 1µF decoupling capacitor
J4	TH1	Analog I/O	Thermistor 1
L3	TH2	Analog I/O	Sensor contact TH2
K4	RR	Analog I/O	Reference resistor
J1	GHH	Analog I	Transimpedance Amplifier G Feedback GHH
K1	GH	Analog I	Transimpedance Amplifier G Feedback GH
L1	GM	Analog I	Transimpedance Amplifier G Feedback GM
J2	GO	Analog O	Transimpedance Amplifier G Output
K2	AO	Analog O	Transimpedance Amplifier A Output
L2	AM	Analog I	Transimpedance Amplifier A Feedback AM
J3	HM	Analog I	Transimpedance Amplifier A Feedback HM
K3	HH	Analog I	Transimpedance Amplifier A Feedback HH
K5	G	Analog I/O	Sensor contact G
L5	B	Analog I/O	Sensor contact B
K6	A	Analog I/O	Sensor contact A
L6	D	Analog I/O	Sensor contact D
K7	C	Analog I/O	Sensor contact C
L7	E	Analog I/O	Sensor contact E
K8	F	Analog I/O	Sensor contact F
L8	H	Analog I/O	Sensor contact H
J8	J	Analog I/O	Sensor contact J
L4	SG	Analog I/O	Analog reference voltage with 10 nF decoupling capacitor
H2	SWCLK	Digital I	SerialWire clock with 10 KΩ external pull-down resistor
G2	SWDIO	Digital I/O	SerialWire data with 10 KΩ external pull-down resistor
K9	TMS	Digital I	Test input: connect to GND on application PCB
L9	RESET_B	Digital I	Reset input
G4	VFLASH_SEL	Power O	Flash voltage supplied to MCU
J7	VDDA_FILT	Power O	Filtered VDDA input from VDDA decoupling capacitor.
F5	VSUP	Power O	Supply selector output bypassed with 10 µF, 0.1 µF, and 1 nF capacitors
J5	VSUP_ANA	Power O	Supply selector output route-back for AFE bypassed with 0.1 µF capacitor
E11	VSUP_DIG	Power O	Supply selector output route-back for Digital

5.3 PACKAGE DRAWING BGA PACKAGE



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	---	---	1.27
A1	0.16	0.21	0.26
A2	0.92	0.96	1.00
A3	0.70 BASIC		
D	7.90	8.00	8.10
D1	6.50 BASIC		
E	7.90	8.00	8.10
E1	6.50 BASIC		
e	0.65 BASIC		
b	0.25	0.30	0.35
aaa	0.10		
bbb	0.10		
ddd	0.10		
eee	0.15		
fff	0.08		

Figure 29: BGA Package Outline Drawing

5.4 LEAD ASSIGNMENT (QFN 10X10, 88 LEAD)

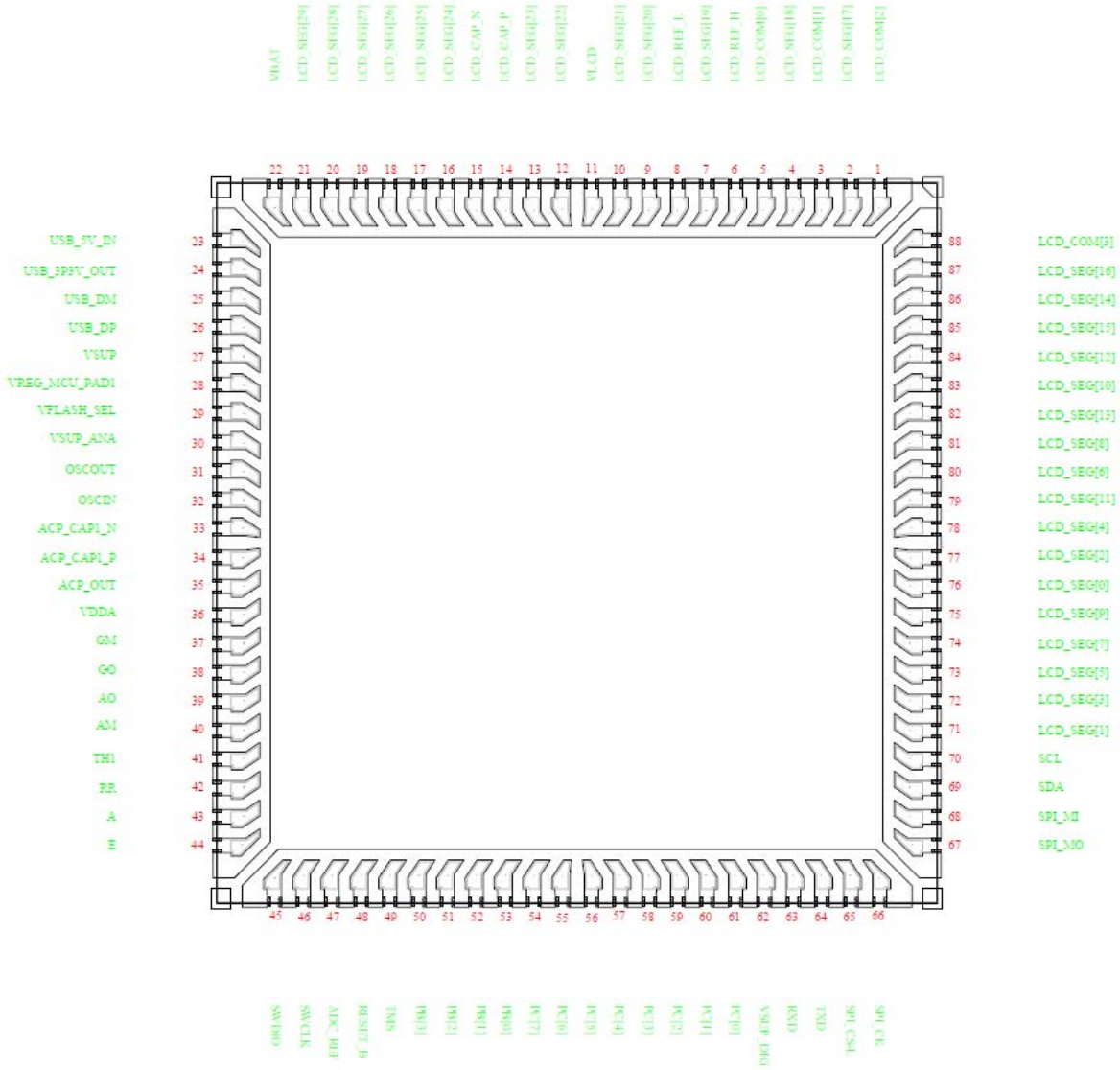


Figure 30: QFN Package Ball Assignment (Top View)

5.5 SIGNAL DESCRIPTION – QFN PACKAGE

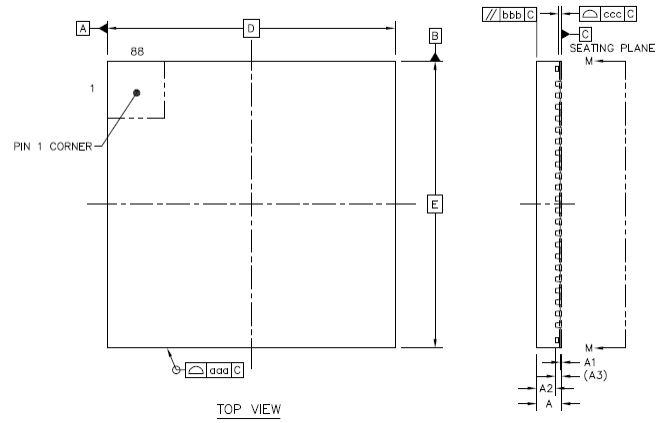
Table 5.46 QFN Package Signal Descriptions

Lead#	Signal Name	I/O	Functional Description
1	LCD_COM2	Digital O	LCD COM2 driver output
2	LCD_SEG17	Digital O	LCD SEG17 driver output
3	LCD_COM1	Digital O	LCD COM1 driver output
4	LCD_SEG18	Digital O	LCD SEG18 driver output
5	LCD_COM0	Digital O	LCD COM0 driver output
6	LCD_REF_H	Analog I/O	LCD bias network 0.47 µF capacitor connection.
7	LCD_SEG19	Digital O	LCD SEG19 driver output
8	LCD_REF_L	Analog I/O	LCD bias network 0.47 µF capacitor connection.
9	LCD_SEG20	Digital O	LCD SEG20 driver output
10	LCD_SEG21	Digital O	LCD SEG21 driver output
11	VLCD	Analog I/O	LCD charge pump 4.7 µF bypass capacitor connection.
12	LCD_SEG22	Digital O	LCD SEG22 driver output
13	LCD_SEG23	Digital O	LCD SEG23 driver output
14	LCD_CAP_P	Analog I/O	LCD charge pump 1 µF charge pump capacitor connection, positive
15	LCD_CAP_N	Analog I/O	LCD charge pump 1 µF charge pump capacitor connection, negative
16	LCD_SEG24	Digital O	LCD SEG24 driver output
17	LCD_SEG25	Digital O	LCD SEG25 driver output
18	LCD_SEG26	Digital O	LCD SEG26 driver output
19	LCD_SEG27	Digital O	LCD SEG27 driver output
20	LCD_SEG28	Digital O	LCD SEG28 driver output
21	LCD_SEG29	Digital O	LCD SEG29 driver output
22	VBAT	I Power	Supply voltage input with 10 µF, 0.1 µF, and 1 nF bypass capacitors.
23	USB_5V_IN	I Power	USB 5V supply input / USB detection with 1 µF decoupling capacitor.
24	USB_3p3V_OUT	O Power	USB 3.3 V output with 1 µF decoupling capacitor
25	USB_DM	Digital I/O	USB data signal minus
26	USB_DP	Digital I/O	USB data signal plus
27	VSUP	Power O	Supply selector output bypassed with 10 µF, 0.1 µF, and 1 nF capacitors
28	VREG_MCU_PAD1	O Power	1.8V microcontroller core regulator 0.1 µF decoupling capacitor
29	VFLASH_SEL	Power O	Flash voltage supplied to MCU
30	VSUP_ANA	Power O	Supply selector output route-back for AFE bypassed with 0.1 µF capacitor
31	OSCOUT	Analog I/O	32.768 KHz oscillator output. Compensate with 10 pF capacitor.
32	OSCIN	Analog I/O	32.768 KHz oscillator input. Compensate with 10 pF capacitor.
33	ACP_CAP1_N	Analog I/O	Step-up regulator 0.039 µF analog charge pump capacitor connection, negative
34	ACP_CAP1_P	Analog I/O	Step-up regulator 0.039 µF analog charge pump capacitor connection, positive
35	ACP_OUT	Power O	Analog charge pump output 0.1 µF holding capacitor
36	VDDA	O Power	Regulated analog supply 0.1 µF decoupling capacitor
37	GM	Analog I	Transimpedance Amplifier G Feedback GM
38	GO	Analog O	Transimpedance Amplifier G Output
39	AO	Analog O	Transimpedance Amplifier A Output

Table 5.46 QFN Package Signal Descriptions			
Lead#	**Signal Name**	**I/O**	**Functional Description**
40	AM	Analog I	Transimpedance Amplifier A Feedback AM
41	TH1	Analog I/O	Thermistor 1
42	RR	Analog I/O	Reference resistor
43	A	Analog I/O	Sensor contact A
44	E	Analog I/O	Sensor contact E
45	SWDIO	Digital I/O	SerialWire data with 10 KΩ external pull-down resistor
46	SWCLK	Digital I	SerialWire clock with 10 KΩ external pull-down resistor
47	ADC_REF	Analog I/O	ADC reference supply voltage 1µF decoupling capacitor
48	RESET_B	Digital I	Reset input
49	TMS	Digital I	Test input: connect to GND on application PCB
50	PB3	Digital I/O	GPIOB3
51	PB2	Digital I/O	GPIOB2
52	PB1	Digital I/O	GPIOB1
53	PB0	Digital I/O	GPIOB0
54	PC7	Digital I/O	GPIOC7 (default PWM output)
55	PC6	Digital I/O	GPIOC6
56	PC5	Digital I/O	GPIOC5
57	PC4	Digital I/O	GPIOC4
58	PC3	Digital I/O	GPIOC3
59	PC2	Digital I/O	GPIOC2
60	PC1	Digital I/O	GPIOC1
61	PC0	Digital I/O	GPIOC0
62	VSUP_DIG	Power O	Supply selector output route-back for Digital
63	RXD/PD7	Digital I/O	UART received data (GPIOD7)
64	TXD/PD6	Digital I/O	UART transmitted data (GPIOD6)
65	SPI_CS-L/PD5	Digital I/O	GPIOD5 (Peripheral (SPI) CS output 1)
66	SPI_CK/PD4	Digital I/O	SPI clock output (GPIOD4)
67	SPI_MO/PD3	Digital I/O	SPI data output (GPIOD3)
68	SPI_MI/PD2	Digital I/O	SPI data input (GPIOD2)
69	SDA/PD1	Digital I/O	I2C serial data I/O (GPIOD1)
70	SCL/PD0	Digital I/O	I2C serial clock output (GPIOD0)
71	LCD_SEG1	Digital O	LCD SEG1 driver output
72	LCD_SEG3	Digital O	LCD SEG1 driver output
73	LCD_SEG5	Digital O	LCD SEG1 driver output
74	LCD_SEG7	Digital O	LCD SEG1 driver output
75	LCD_SEG9	Digital O	LCD SEG1 driver output
76	LCD_SEG0	Digital O	LCD SEG1 driver output
77	LCD_SEG2	Digital O	LCD SEG1 driver output
78	LCD_SEG4	Digital O	LCD SEG1 driver output
79	LCD_SEG11	Digital O	LCD SEG1 driver output
80	LCD_SEG6	Digital O	LCD SEG1 driver output
81	LCD_SEG8	Digital O	LCD SEG1 driver output
82	LCD_SEG13	Digital O	LCD SEG1 driver output
83	LCD_SEG10	Digital O	LCD SEG1 driver output
84	LCD_SEG12	Digital O	LCD SEG1 driver output

Table 5.46 QFN Package Signal Descriptions			
Lead#	**Signal Name**	**I/O**	**Functional Description**
85	LCD_SEG15	Digital O	LCD SEG1 driver output
86	LCD_SEG14	Digital O	LCD SEG1 driver output
87	LCD_SEG16	Digital O	LCD SEG1 driver output
88	LCD_COM3	Digital O	LCD COM3 driver output

5.6 PACKAGE DRAWING – QFN PACKAGE



		SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS		A	0.8	0.85	0.9
STAND OFF		A1	0	0.035	0.05
MOLD THICKNESS		A2	---	0.65	0.67
L/F THICKNESS		A3		0.203 REF	
LEAD WIDTH		b	0.15	0.2	0.25
BODY SIZE	X	D		10 BSC	
	Y	E		10 BSC	
LEAD PITCH		e		0.4 BSC	
EP SIZE	X	J	8	8.1	8.2
	Y	K	8	8.1	8.2
LEAD LENGTH		L	0.35	0.4	0.45
PACKAGE EDGE TOLERANCE		aaa		0.1	
MOLD FLATNESS		bbb		0.1	
COPLANARITY		ccc		0.08	
LEAD OFFSET		ddd		0.1	
EXPOSED PAD OFFSET		eee		0.1	

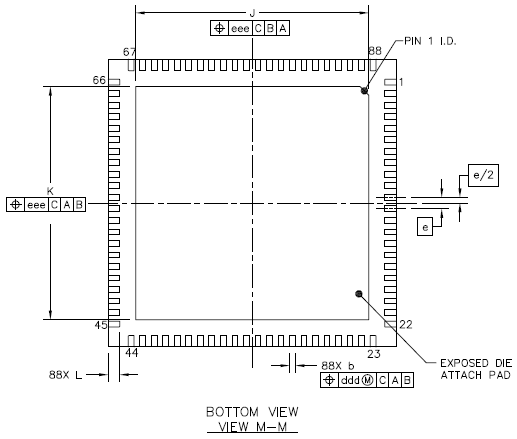


Figure 31: QFN Package Outline Drawing

6.0 REVISION HISTORY

Table 6.47 Revision History			
Rev #	**Date**	**Action**	**By**
0.1	11/27/2012	Original system specification	SM
0.2	01/12/2014	Indie version	PH
0.3	09/10/2015	Indie open market version	CR
0.4	2/18/2016	Minor updates to remove non-essential, internal information	ER
0.5	3/18/2016	Updates to Interrupts, PMU, LCD, GPIOs, Timers, RTC to provide additional details on the functionality of each of these blocks	ER
0.6	4/1/2016	Correct revision numbering, updated clock tree information in 5.1.5 and 5.1.6, correct FLASH endurance information, corrected Register 6 ADC Trim2 register table, added switch matrix diagram, added PMU diagram showing USB and battery capabilities.	ER
0.7	5/4/2016	Imported high resolution diagrams for switch matrix and power management, corrected pin description table, removed register numbering which was confusing, fixed font and column width on interrupt vector table, RTC description was updated with more detailed information, various correction to GPIO descriptions/features, RC18TRIM register and clock source information was added	ER
0.8	7/14/2016	Changes to update documentation for A4 revision of ASIC die. Major changes to document organization	SM
0.9	9/20/2016	Update for A5 revision of ASIC die. Added information on I2C, SPI and UART operation and registers	SM
0.91	1/5/2017	Added references to alternate package availability	ER
0.92	2/24/2017	Updated sleep currents	ER

Table 6.47 Revision History			
Rev #	**Date**	**Action**	**By**
1.0	8/22/2017	Updated LCD drive voltages Updated RTC, Timer3 and LCD description to receive XTAL clock divided by 2. Updated 10 kHz RC clock source to indicate internal division by 2 Updated Figure 3 Updated WKPTIME register description (formula) Added Section 3.1.13.1: Notes to GPIO operation and configuration Updated PDPD and PDPU register descriptions Updated PCPWMCFG register description Updated SX Control Register description of SX6 Updated bit names in ADC Trim0 register to be consistent with remaining document Update PMURST description to emphasize that all interrupts must be cleared before initiating deep sleep Update PMUBOR description to correct threshold table	SM

7.0 REFERENCES

[1] Indie semiconductor ARM CORTEX M0 core programming guide, *Aydeekay_Core_160_8.pdf*

[2] USB Hardware Interface Design Specification (pdf document)

[3] USB Software Stack Interface User Manual (pdf document)

8.0 CONTACTS

United States
32 Journey
Aliso Viejo, California 92656, USA
Tel: +1 949-608-0854
sales@indiesemi.com

China
232 Room, Donghai Wanhao Plaza,
South Hi-tech 11th Road, Hi-tech Industry Park,
Nanshan District, Shenzhen, China.
Tel: +86 755-86116939

Scotland
MWB Business Exchange
9-10 St. Andrew Square
Edinburgh EH2 2AF, Scotland
Tel: +44 131 718 6378

http://www.indiesemi.com/

Important Notice

indie semiconductor reserves the right to make changes, corrections, enhancements, modifications, and improvements to indie semiconductor products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on indie semiconductor products before placing orders. indie semiconductor products are sold pursuant to indie semiconductor’s terms and conditions of sale in place at the time of order acknowledgement. Purchasers are solely responsible for the choice, selection, and use of indie semiconductor products and services described herein. indie semiconductor assumes no liability for the choice, selection, application assistance or the design of Purchasers’ products.

No license, express or implied, to any intellectual property right is granted by indie semiconductor by this document.

The materials, products and information are provided “as is” without warranty of any kind, whether express, implied, statutory, or otherwise, including fitness for a particular purpose or use, merchantability, performance, quality or non-infringement of any intellectual property right. Indie semiconductor does not warrant the accuracy or completeness of the information, text, graphics or other items contained herein. indie semiconductor shall not be liable for any damages, including but not limited to any special, indirect, incidental, statutory, or consequential damages, including without limitation, lost of revenues or lost profits that may result from the use of the materials or information , whether or not the recipient of material has been advised of the possibility of such damage.

Unless expressly approved in writing by two authorized indie semiconductor representatives, indie semiconductor products are not designed, intended, warranted, or authorized for use as components in military, space, or aircraft, in systems intended to support or sustain life, or for any other application in which the failure or malfunction of the indie semiconductor product may result in personal injury, death, or severe property or environmental damage.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.