iND83211 Datasheet Rev 1.14

1 Revision History

Table 1 Revision History

Rev #	Date	Action
1.1	05/21/2021	1. Added Memory Description
1.2	05/24/2021	1. Fixed buck description issue in system overview for keeping in line with the design: a) Operation Freq adjustable: 290KHz~760KHz 2. Output voltage adjustable: < 6V
1.3	05/28/2021	Replaced the package marking
1.4	06/03/2021	LIN Master Description updated: Added a limitation that LIN Master only supports auto-baudrate LIN Slave.
1.5	09/16/2021	Update Pin Configuration
1.6	11/1/2021	Package Thermal Resistance Modified :(ThetaJB) form 30K/W to 10°C /W Added :(ThetaJA) 27°C /W
1.7	2/22/2022	Updated the Table 14 Tempsensor Output voltage vs Junction Temp
1.8	6/6/2022	Added WDTA descripition in ASIC Features
1.9	6/9/2022	Fixed SSC_DEEP description
1.10	10/27/2022	1. Added 9.2.1 Power on sequence; 2. added figures of the clock generation.
1.11	3/6/2023	1. Fixed ADC synchronization diagram 2. Replace logo with Indiemicro logo 3. Updated pin configuration diagram appearance
1.12	3/24/2023	1. Fixed ADC synchronization diagram 2. Add detailed description of the global ADC attenuator
1.13	9/19/2023	1. Update the spec. of bandgap voltage 2. Add MSL level information of package
1.14	11/2/2023	SDIO timing description updated in 7.2.23.11 FLASH->TRIM

2 TABLE OF CONTENTS

3 LIST OF TABLES

4 LIST OF FIGURES

Figure of Contents

5 ORDERING INFORMATION

Part number	Package	MSL	Shipping
iND83211	QFN48	Level 3	3000pcs/Tape&Reel

6 SYSTEM OVERVIEW

"iND83211" IC is an automotive LED lighting integrated device that combines together a 32bit MCU (Cortex M0, indie 'Verne' core) with a flexible power management unit including a step down buck converter and associated regulators, 24x high voltage programmable current open drain IO with PWM along with specific monitoring features, 2x LIN slave transceivers and their controllers, 1x UART ,1x SPI Master and an integrated 10 bit ADC for monitoring purpose. The IC can withstand 45V load dump from the car battery on its VBAT pin while has the option to operate with a 12V and a 5V input supplies saving the buck passive components entirely.

- Full automotive qualification AEC-Q100 Grade1
- Functional Safety Enhancements:
    - Two independent hardware LIN TX monitors for LINS/LINM to prevent a dominant bus caused by internal malfunction.
    - Buck
        - Period & VDAC configuration hardware check.
        - Buck Clock is independent from system clock configuration.
    - LIN Bus Idle timeout monitors
        - Always active, even the chip is in hibernate mode.
        - For preventing a fast discharge of the car battery, if a short to ground is detected, the following options are available:
            - Automatically switch off LIN slave's pullup.
            - Reduce LIN master's pullup from 1K to 30K.
            - Disable lin switch automatically.
        - Auto-recovery if the failure condition disappears.
- CPU architecture:
    - ARM Cortex M0 processor
    - System Tick Timer (Systick, 24bits, interruptible)
    - Serial Wire Debugger (ARM)
    - Built-in Nested Vectored Interrupt Controller (NVIC)
    - 32bits multiplier
    - Programmable Watch-Dog Timer
    - 3 programmable timers

- Memory:
    - 64kBytes of Flash Program Memory, 10 years retention in automotive environment
    - 16kBytes of SRAM
- Peripherals/Digital Features
    - Clock and Reset Manager
        - RCO: system and always on (wake up support)
        - Reset POR and BOR (no external reset)
    - One SAE J2602/LIN2.2 LIN Slave Controller and Transceiver
        - Supports LIN auto-addressing through an internal LIN switch.
    - One LIN Master Controller and Transceiver
        - Only available when the internal LIN switch is not used.
        - Only supports auto-baudrate LIN Slaves.
    - Watch dog timer (ASIC side)
    - 24x16bits PWM required to control LED current driver:
        - Common prescaler and 16bit timer
        - Support power balance with independent rise/fall timing configuration
    - 5x16bits PWM required to control GPIOs, with 2 shared prescaler and 16bit timer.
    - Serial Communication interfaces:
        - Integrated UART Controller with two independent fifos(TX:32 bytes; RX: 32 bytes)
        - Integrated SPI master controller
    - Up to 9 GPIOs with flexible configurations:
        - General GPIO with pullup & pulldown
        - Analog Input mode
        - Support Edge detects
        - Wakeup sources in hibernate mode
- Peripherals/Analog Features:
    - 24 Programmable 45mA max constant current Low voltage(VDD5P0) IO open drain
        - Drain voltage that ensures < 1V
        - Integrated pullup capability

- PN voltage measurement for temp compensation:
  - Integrated a dedicated 2.5mA current source
  - Fully differential measurement
- Temperature Sensor/Monitor with ADC
- Battery voltage detection and monitoring
- Hardware over temperature protection
- 10 bits SAR ADC with 38 channels
  - Buffered bandgap voltage
  - Junction Temperature
  - Forward voltages of 24 external LEDs
  - GPIO1~9
  - One accurate VBAT channel
  - MCU Core/IO Voltage
  - Buck Output voltage
- Fully integrated step-down buck converter
  - Operation Freq adjustable: 290KHz~760KHz
  - Output voltage adjustable: < 6V
  - Overcurrent/Dead short protection with hiccup support
- Integrated voltage regulators
  - LDO 3.3Vout (ASIC Core and IO supply + MCU I/O)
  - LDO 1.5VOut (MCU Core/Flash)

6.1 Application Block Diagram

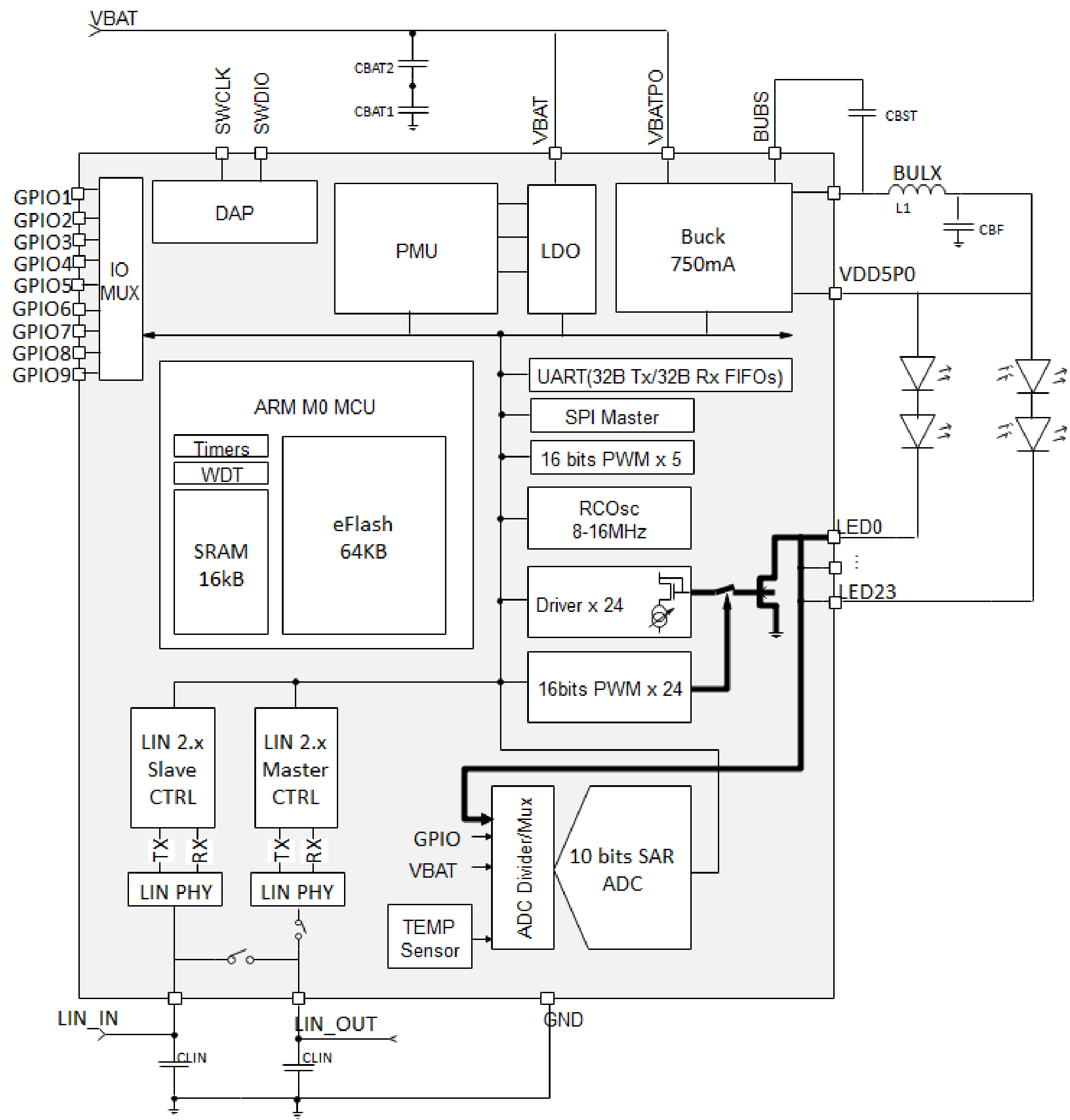


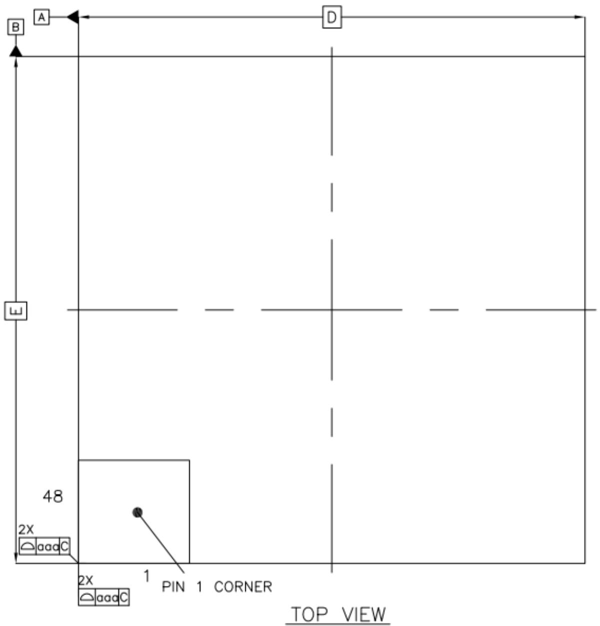
Figure 1 IC block diagram with external components

Note: The application block diagram does not include components used to qualify the system against ISO7637-2/-3.

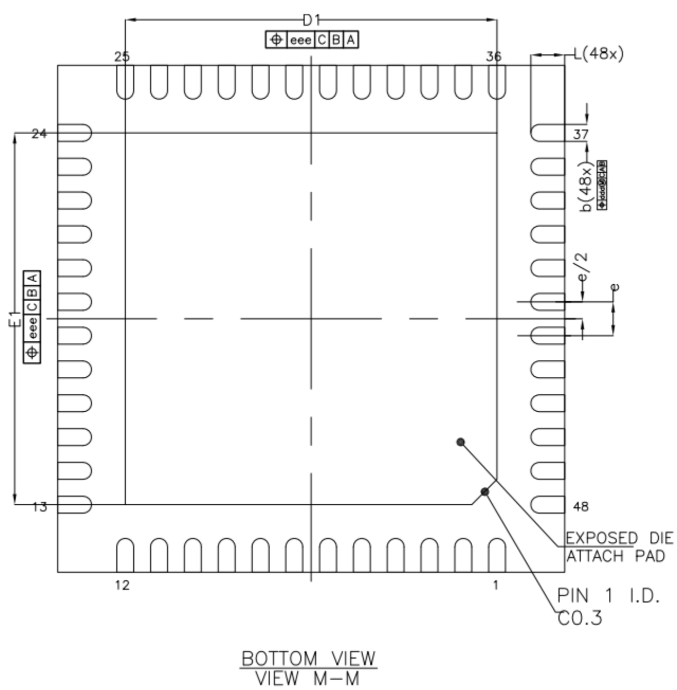
6.2 PACKAGE OVERVIEW AND PIN DESCRIPTION

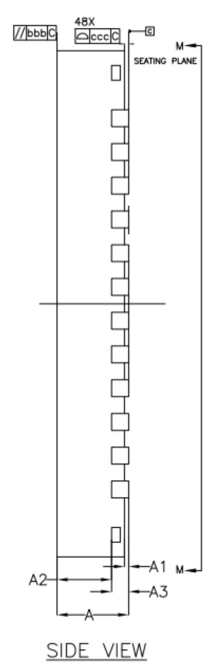
6.2.1 QFN48 Package Outline

QFN 6x6 48 pins 0.4mm pitch



DESCRIPTION		SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS		A	0.80	0.85	0.90
STAND OFF		A1	0.00	---	0.05
MOLD THICKNESS		A2	0.60	0.65	0.70
L/F THICKNESS		A3	0.203 REF		
LEAD WIDTH		b	0.14	0.20	0.26
BODY SIZE	X	D	5.90	6.00	6.10
	Y	E	5.90	6.00	6.10
LEAD PITCH		e	0.40 BSC		
EP SIZE	X	D1	4.35	4.40	4.45
	Y	E1	4.35	4.40	4.45
LEAD LENGTH		L	0.30	0.40	0.50
Tolerance of form and position					
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		

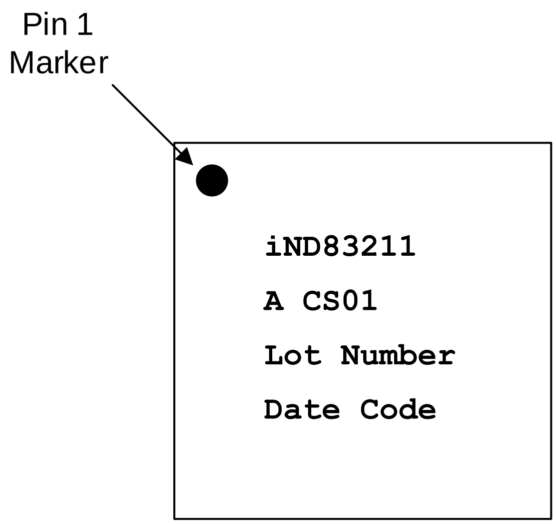




6.2.2 Part Number

Part Number: iND83211

Package Branding:



6.3 IO PIN DESCRIPTIONS

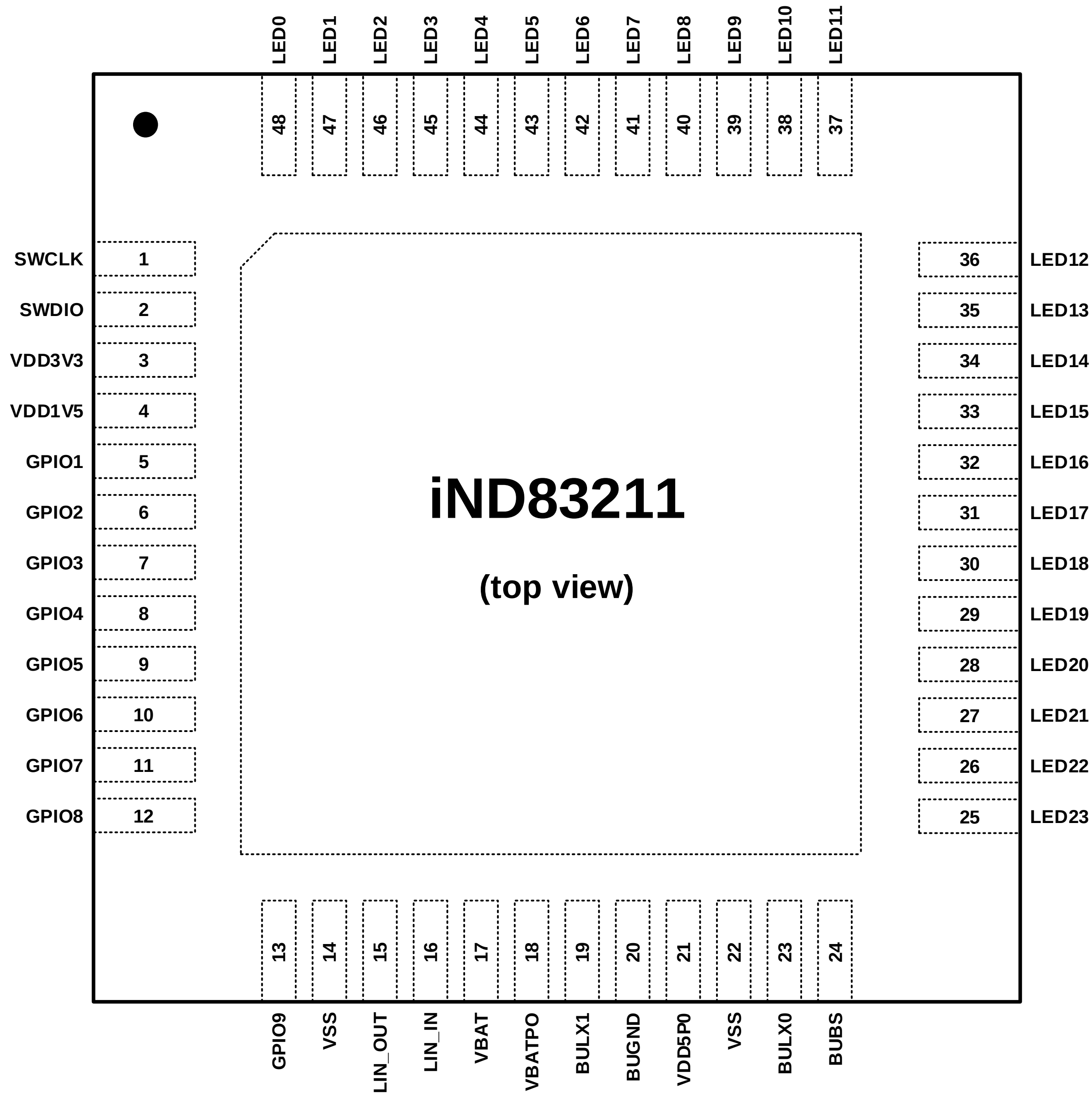


Figure 2 Pin Configuration

Table 2 Pin List

#	Pin Name	Type	Voltage	Direction	Description
1	SWCLK	GPIO	VDD3V3	Input	SWD clock
2	SWDIO	GPIO	VDD3V3	I/O	SWD data
3	VDD3V3	Output	VDD3V3	Analog	Voltage regulator decoupling cap
4	VDD1V5	Output	VDD1V5	Analog	Voltage regulator decoupling cap
5	GPIO1	GPIO	VDD3V3	I/O	PWM0/LINM RXD/General purpose IO
6	GPIO2	GPIO	VDD3V3	I/O	PWM1/SPI_SS/ LINM TXD/General purpose IO
7	GPIO3	GPIO	VDD3V3	I/O	PWM2/SPI_SCLK/LINS RXD /General purpose IO
8	GPIO4	GPIO	VDD3V3	I/O	PWM3/SPI_MOSI/UART RX /LINS TXD/General purpose IO
9	GPIO5	GPIO	VDD3V3	I/O	PWM4/SPI_MISO/UART TX/ LINS TXD/General purpose IO
10	GPIO6	GPIO	VDD3V3	I/O	PWM1/SPI_SS/ LINM TXD/General purpose IO
11	GPIO7	GPIO	VDD3V3	I/O	PWM2/SPI_SCLK/ LINS RXD/General purpose IO
12	GPIO8	GPIO	VDD3V3	I/O	PWM3/SPI_MOSI/UART RX /LINS TXD/General purpose IO
13	GPIO9	GPIO	VDD3V3	I/O	PWM4/SPI_MISO/UART TX/LINS TXD/General purpose IO

#	Pin Name	Type	Voltage	Direction	Description
14	VSS	Supply	Ground	Analog	Bonding with ground plane
15	LIN_OUT	IO	VBAT	I/O	LIN Master pin/LIN Switch Out
16	LIN_IN	IO	VBAT	I/O	LIN slave pin
17	VBAT	Supply	VBAT	n/a	Input VBAT
18	VBATPO	Supply	VBAT	n/a	Input VBAT for buck (power)
19	BULX1	Analog	VBAT	n/a	connect to external inductor
20	BUGND	Ground	Ground	Analog	Buck power gound
21	VDD5P0	Analog	VDD5P0	Analog	buck 5V output(inner), also is feedback input(external)
22	VSS	Supply	Ground	Analog	Bonding with ground plane
23	BULX0	Analog	VBAT	n/a	inner connect to BULX1, external connect to bootstrap cap(100nF)
24	BUBS	Analog	BULX0+3.3V	Analog	Buck Bootstrap cap (100nF) bootstrap 3.3V w.r.t BULX0
25	LED23	Analog	VDD5P0	Analog	Current Regulated Sink
26	LED22	Analog	VDD5P0	Analog	Current Regulated Sink
27	LED21	Analog	VDD5P0	Analog	Current Regulated Sink
28	LED20	Analog	VDD5P0	Analog	Current Regulated Sink
29	LED19	Analog	VDD5P0	Analog	Current Regulated Sink
30	LED18	Analog	VDD5P0	Analog	Current Regulated Sink
31	LED17	Analog	VDD5P0	Analog	Current Regulated Sink
32	LED16	Analog	VDD5P0	Analog	Current Regulated Sink

#	Pin Name	Type	Voltage	Direction	Description
33	LED15	Analog	VDD5P0	Analog	Current Regulated Sink
34	LED14	Analog	VDD5P0	Analog	Current Regulated Sink
35	LED13	Analog	VDD5P0	Analog	Current Regulated Sink
36	LED12	Analog	VDD5P0	Analog	Current Regulated Sink
37	LED11	Analog	VDD5P0	Analog	Current Regulated Sink
38	LED10	Analog	VDD5P0	Analog	Current Regulated Sink
39	LED9	Analog	VDD5P0	Analog	Current Regulated Sink
40	LED8	Analog	VDD5P0	Analog	Current Regulated Sink
41	LED7	Analog	VDD5P0	Analog	Current Regulated Sink
42	LED6	Analog	VDD5P0	Analog	Current Regulated Sink
43	LED5	Analog	VDD5P0	Analog	Current Regulated Sink
44	LED4	Analog	VDD5P0	Analog	Current Regulated Sink
45	LED3	Analog	VDD5P0	Analog	Current Regulated Sink
46	LED2	Analog	VDD5P0	Analog	Current Regulated Sink
47	LED1	Analog	VDD5P0	Analog	Current Regulated Sink
48	LED0	Analog	VDD5P0	Analog	Current Regulated Sink
*	EPAD	Supply	Ground	n/a	Ground

Note: the above pinout is not representing the final pin allocation for the IC. The exact pin allocation will be defined during the design phase including floorplan studies and PCB constraints provided by the customer.

6.3.1 **Pin state upon Power-on Reset**

- Unless otherwise noted, all pins default to tristate/Isolation mode (Hi-Z) upon power-on reset. Electrical Characteristics

7 MEMORY DESCRIPTION

7.1 TOP LEVEL MEMORY MAP

The chip uses a unified memory model with a linear address space (Von Neumann architecture) including Flash and RAM memories as well as registers address space. The implementation of the Cortex M0 core uses a high density 64KB Flash cell along with 16KB of SRAM.

Table 3 Top Level Memory Map

Address	Memory	Description
0x00000000 – 0x0000FFFF	Flash	64Kbytes of Flash Memory, user programmable.
0x00010000 – 0x0003FFFF	N/A	Reserved
0x00040000 – 0x000400FF	N/A	Reserved
0x00040100 – 0x000401FF	N/A	Reserved
0x00040200 – 0x1FFFFFFF	N/A	Reserved
0x20000000 – 0x20003FFF	SRAM	16Kbytes of SRAM
0x20004000 – 0x4FFFFFFF	N/A	Reserved
0x50000000 - 0x5000003F	CRGA	Clock & Reset Generator
0x50000040 - 0x5000005F	PMU	Power Management Unit
0x50000060 - 0x5000007F	EVTHOLD	Event Hold Control

0x50000080 - 0x500000FF	BTE	Block Transfer Engine registers
0x50010100 - 0x500101FF	WICA	WakeUp Interrupt Controller
0x50010200 - 0x500102FF	WDTA	Watchdog Timer Registers
0x50010300 - 0x500103FF	LINS	LIN slave interface registers
0x50010400 - 0x500104FF	LINM	LIN master interface registers
0x50010500 - 0x500105FF	BUCKCTRL	Buck Interface Control Registers
0x50010600 - 0x500106FF	ADC	ADC Control
0x50010700 - 0x500107FF	SPI	SPIM Interface registers
0x50010800 - 0x500108FF	UART	UART Interface registers
0x50010900 - 0x500109FF	PWM_AUX	Control (and status) registers for the pulse width modulation waveform generator.
0x50011000 - 0x50011FFF	PWM	Control (and status) registers for the pulse width modulation waveform generator.
0x50012000 - 0x50012FFF	IOCTRLA	I/O configuration and DFT pin control
0x50013000 - 0x50013FFF	SYSCTRLA	System configuration and retention memory
0x50014000 - 0x50017FFF	GPIO	GPIO bit control and configuration

0x50018000 - 0x5001FFFF	BUCK_TRIM	Trim Memory for BUCK Regulator
0x50020000 - 0x50020007	TIMER0	General purpose timer 0
0x50020008 - 0x5002000F	TIMER1	General purpose timer 1
0x50020010 - 0x50020017	TIMER2	General purpose timer 2
0x50020018 - 0x5002001F	WDT1	The watchdog timer that is local to the MCU
0x50020020 – 0x5002005F	Flash	Flash Programming/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

7.2 Register Description

7.2.1 CRGA

Clock & Reset Generator

7.2.1.1 CRGA->LFCLKCTRL

CRGA	**LFCLKCTRL**	0x50000000
	Low frequency clock control	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																							A								B

#	Bit(s)	Field	Description	Type	Reset
A	8	CLKLFSEL	*LF Clock Source select* Used to select lf osc mode between 0x**0** — CLK_LF(LF OSC out) is 32KHz, slow system clock freq is Freq_CLK_LF/2 0x**1** — CLK_LF(LF OSC out) is 256KHz, slow system clock freq is Freq_CLK_LF/16. Recommended for more accurate LIN Wakeup detection in hibernate mode.	r/w	0
B	0	LFRCSTS	*Slow oscillator status* Will be high when the 16KHz oscillator is selected	*ro*	0

7.2.1.2 CRGA->SYSCLKCTRL

CRGA	**SYSCLKCTRL**	0x50000004
	System clock control	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													A										B							C	D

#	Bit(s)	Field	Description	Type	Reset
A	18:16	DIVSYSCLK	*Clock div select* Select the divider ratio on the system clock when using fast oscillator 0x**0** — No Division. Full Clock speed 0x**1** — Div by 2. 0x**2** — Div by 3. 0x**3** — Div by 4. 0x**4** — Div by 5. 0x**5** — Div by 6. 0x**6** — Div by 7. 0x**7** — Div by 8.	r/w	0
B	8	SYSCLKSEL	*Clock select* Used to switch between the fast and slow system clocks 0x**0** — Slow clock (16 KHz) 0x**1** — Fast clock (16 MHz)	r/w	0
C	1	HFRCSTS	*Fast oscillator status* Will be high when the 16MHz oscillator is enabled	*ro*	0
D	0	HFRCENA	*Fast oscillator enable* Setting this bit when the 16MHz oscillator is not running will cause the oscillator to start (the PMU may	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			have already started it). Even though the fast oscillator is running, its output is only used when selected via the clock mux - see CLKSEL. This bit is cleared automatically on entering SLEEP mode		

7.2.1.3 **CRGA->RESETCTRL**

CRGA	**RESETCTRL**	0x50000008
	Reset control	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							A								B		C	D	E		F		G		H	I	J		K		L

#	Bit(s)	Field	Description	Type	Reset
A	24	SOFTRSTREQ	*Soft reset request* Set to trigger a soft reset of chip – cleared automatically after each write	*wo*	0
B	16	HARDRSTREQ	*Hard reset request* Set to trigger a hard reset of chip	*wo*	0
C	14	OVTEMPFLAGCLR	*OVTEMP flag clear* Set to clear the OVTEMP flag	*wo*	0
D	13	WDTFLAGCLR	*WDT flag clear* Set to clear the WDT flag	*wo*	0
E	12	BOR1V5FLAGCLR	*BOR 1v5 clear*	*wo*	0

#	Bit(s)	Field	Description	Type	Reset
			Set to clear the 1.5V brownout detected flag		
F	10	BOR3V3FLAGCLR	*BOR 3v3 clear* Set to clear the 3.3V brownout detected flag	*wo*	0
G	8	PORFLAGCLR	*POR flag clear* Set to clear the POR flag	*wo*	0
H	6	OVTEMPFLAG	*Over Temp Violation flag* Set by the hardware when the over temp condition is detected	*ro*	n/a
I	5	WDTFLAG	*Watchdog bark flag* Set by the hardware when the watchdog barks	*ro*	n/a
J	4	BOR1V5FLAG	*BOR 1v5 flag* Set by the hardware when a brownout of the 1.5V supply is detected	*ro*	0
K	2	BOR3V3FLAG	*BOR 3v3 flag* Set by the hardware when a brownout of the 3.3V supply is detected	*ro*	0
L	0	PORFLAG	*Power on reset flag* Set by the hardware during power-on reset	*ro*	n/a

7.2.1.4 **CRGA->MODULERST**

CRGA	**MODULERST**	0x5000000C
	Module Reset control	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								A							

#	Bit(s)	Field	Description	Type	Reset
A	7:0	MODULERSTREQ	*Module Soft reset request* Set to trigger a soft reset of module below 0x**1** — ADC Soft reset request 0x**2** — BUCK Soft reset request 0x**4** — LINM Soft reset request 0x**8** — LINS Soft reset request 0x**10** — PWMAUX soft reset request 0x**20** — PWM soft reset request 0x**40** — SPIM soft reset request 0x**80** — UART soft reset request – cleared automatically after each write	*wo*	0

7.2.1.5 **CRGA->BORACTION**

CRGA	**BORACTION**	0x50000010
	BOR action	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A								B																		C				D	

#	Bit(s)	Field	Description	Type	Reset
A	31	BOR_1V5_LOCK	Set Only bit. Set this bit to lock BOR_1V5_ACTION & BOR_1V5_THRESH bits	r/w	0
B	23	BOR_3V3_LOCK	Set Only bit. Set this bit to lock BOR_3V3_ACTION & BOR_3V3_THRESH bits	r/w	0
C	5:4	VDD1V5	*BOR 1v5 action* Defines the consequences of brown-out condition on the 1v5 supply being detected by the hardware 0x**0** — Hard reset generated 0x**1** — IRQ generated 0x**2** — No action — read-only, unless 'bor_1v5 access' is enabled	r/w	0
D	1:0	VDD3V3	*BOR 3v3 action* Defines the consequences of brown-out condition on the 3v3 supply being detected by the hardware 0x**0** — Hard reset generated 0x**1** — IRQ generated 0x**2** — No action — read-only, unless 'bor_3v3 access' is enabled	r/w	0

7.2.1.6 **CRGA->BORCONFIG**

CRGA	**BORCONFIG**	0x50000014
	BOR configuration	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			A				B						C															D			

#	Bit(s)	Field	Description	Type	Reset
A	28	BORBIASOVERRIDESEL	*BOR bias override select* Independently controllable test bit which can be used to override the BOR bias enable signal. Set 1 to enable the use of the BOR_REF_ENA register test bit 0x**0** — Functional Mode 0x**1** — Override Mode- Use BOR_BIAS_OVERRIDE_ENA register to control bias. — only accessible when 'debug access' is enabled	r/w	0
B	24	BORBIASOVERRIDEENA	*BOR bias override bit* Independently controllable test bit which can be used to override the BOR bias enable signal. Set 1 to enable the use of this test bit — only accessible when 'debug access' is enabled	r/w	*1*
C	18:16	BOR1V5THRESH	*BOR 1v5 threshold* Select the BOR threshold voltage level for the 1v5 regulator. Following typical value(under 27degC) can be used as reference: 0x**0** — Vr: 1.317V Vf: 1.261V	r/w	*3*

#	Bit(s)	Field	Description	Type	Reset
			0x**1** — Vr: 1.347V Vf: 1.290V 0x**2** — Vr: 1.379V Vf: 1.320V 0x**3** — Vr: 1.412V Vf: 1.352V 0x**4** — Vr: 1.446V Vf: 1.385V 0x**5** — Vr: 1.482V Vf: 1.420V 0x**6** — Vr: 1.520V Vf: 1.456V 0x**7** — Vr: 1.560V Vf: 1.495V — read-only, unless 'bor_1v5 access' is enabled		
D	3:0	BOR3V3THRESH	*BOR 3v3 threshold* Select the BOR threshold voltage level for the 3v3 regulator, Following typical value(under 27degC) can be used as reference: NOTE: Invalid Configuration will be forced to default value '0x6' 0x**3** — Vr: 2.392V Vf: 2.293V 0x**4** — Vr: 2.466V Vf: 2.365V 0x**5** — Vr: 2.546V Vf: 2.442V 0x**6** — Vr: 2.631V Vf: 2.524V 0x**7** — Vr: 2.721V Vf: 2.612V 0x**8** — Vr: 2.819V Vf: 2.706V 0x**9** — Vr: 2.924V Vf: 2.807V 0x**A** — Vr: 3.037V Vf: 2.916V 0x**B** — Vr: 3.159V Vf: 3.034V	r/w	*6*

#	Bit(s)	Field	Description	Type	Reset
			0x**C** — Vr: 3.291V Vf: 3.162V — read-only, unless 'bor_3v3 access' is enabled		

7.2.1.7 **CRGA->WDTACTION**

CRGA	**WDTACTION**	0x50000018
	Watchdog action	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															A					B											

#	Bit(s)	Field	Description	Type	Reset
A	16	WDTBARKCNTCLR	*WatchDog Bark Counter Clear* Set to clear watchdog bark counter	*wo*	0
B	11:8	WDTBARKCNT	*WatchDog Bark Counter* Read to reflect the WDT bark counter value	*ro*	0

7.2.1.8 **CRGA->LFCLKKILL**

CRGA	**LFCLKKILL**	0x5000001C
	Low frequency clock kill	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															A

#	Bit(s)	Field	Description	Type	Reset
A	0	KILLLFRC	*Kill slow RC oscillator* Setting this bit gates the low frequency RC oscillator input — only accessible when ‘debug access’ is enabled	r/w	0

7.2.1.9 **CRGA->CPCLKCTRL**

CRGA	**CPCLKCTRL**	0x50000020
	Charge pump clock control The clock divider and the charge pump clock gate use the system clock as the clock source. The charge pump clock divider is fed by the gated clock.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																									A					B	C

#	Bit(s)	Field	Description	Type	Reset
A	6:4	PMUCPDIVSEL	*PMU charge pump divider select* Selects the divide value for the PMU Charge Pump Clock 0x**0** — No Division. Full Clock speed. 0x**1** — Div by 2. 0x**2** — Div by 4. 0x**3** — Div by 8. 0x**4** — Div by 16. 0x**5** — Div by 32.	r/w	*2*
B	1	PMUCPREG	*PMU charge pump override register*	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			when PMU_CP_SEL=1: 1 enables PMU charge pump clock, 0 gates PMU charge pump clock		
C	0	PMUCPSEL	*PMU charge pump select* Setting this bit will override the enable signal to the PMU charge pump clock gate 0x**0** — The PMU charge pump clock gate is enabled by two sources by default: 1. The PMU state machine 'bring-up/active' states (i.e. hardware driven) 2. The High Frequency Oscillator being active *Note* both #1 and #2 soures must be active. 0x**1** — The value of the PMU_CP_REG field is what is used to drive the enable signal on the PMU charge pump clock gate. *Note* this allows the charge pump clock to be driven even with the slow RC oscillator (16KHz).	r/w	0

7.2.1.10 CRGA->OVTEMPACTION

CRGA	**OVTEMPACTION**	0x50000024
	OVTEMP action	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																														B	

#	Bit(s)	Field	Description	Type	Reset
A	31	OVTEMP_LOCK	Set Only bit. Set this bit to lock OVTEMP related bits	r/w	0
B	1:0	OVTEMP	*Over Temperature action*	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			Defines the consequences of over temp condition detected by the hardware 0x**0** — Hard reset generated 0x**1** — IRQ generated 0x**2** — No action — read-only, unless ‘ovtemp access’ is enabled		

7.2.1.11 **CRGA->OVTEMPCONFIG**

CRGA	**OVTEMPCONFIG**	0x50000028
	OVTEMP configuration	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						A	B																					C			

#	Bit(s)	Field	Description	Type	Reset
A	25	DISTEMPSENS	*Disable Temp Sensor Analog Part* Set to disable Temperature Sensor analog circuit only if OVTEMP_ENA is clearred. Notice that the internal temperature sensor is shared by over temp monitor. BGAUXREG in SYSCTRLA should be enabled before using TempSensor in hibernate mode	r/w	0
B	24	OVTEMPENA	*OverTemp Monitor Enable bit* — read-only, unless ‘ovtemp access’ is enabled	r/w	0
C	3:0	OVTEMPLEVEL	*Over Temp threshold*	r/w	*0xB*

#	Bit(s)	Field	Description	Type	Reset
			Select the OVTEMP threshold level for the monitor 0x**0** — 90C 0x**1** — 95C 0x**2** — 100C 0x**3** — 105C 0x**4** — 110C 0x**5** — 115C 0x**6** — 120C 0x**7** — 125C 0x**8** — 130C 0x**9** — 135C 0x**A** — 140C 0x**B** — 145C 0x**C** — 150C 0x**D** — 155C 0x**E** — 160C 0x**F** — 165C — read-only, unless ‘ovtemp access’ is enabled		

7.2.2 **PMUA**

Power Management Unit

7.2.2.1 **PMUA->CTRL**

PMUA								**CTRL**						0x50000040																	
								Control																							
31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
																												A		B	C

#	Bit(s)	Field	Description	Type	Reset
A	3	FASTMCUON	*Fast MCU Power On* Set to enable VDD1V5 LDO during hibernate mode for fast mcu power on sequennce. Clr to disable VDD1V5 LDO during hibernate mode for saving power	r/w	*1*
B	1	FASTBOOT	*Fast boot* Set to enable used of the fast clock during subsequent power-up sequences (including the portion consumed by the mcu boot sequence). The set value brings the system up with the slow clock to make the initial boot and any boot after a hard reset (e.g. after a brownout) as safe as possible	r/w	*1*
C	0	HIBERNATE	Set to put the chip into HIBERATE mode. Before setting this bit, ensure that wake interrupt controller HOLD bit has been set (and that a corresponding Lullaby interrupt has been received)	*wo*	0

7.2.2.2 **PMUA->DEBUG**

PMUA	**DEBUG**	0x50000044

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															A

#	Bit(s)	Field	Description	Type	Reset
A	0	IGNORE_CIFS	*Ignore QACKs* Setting a bit in this register prevents PMUA from waiting for the assertion of the corresponding 'Quiescent State Acknowlege' signal when before transitioning towards the Hibernate state	r/w	0

7.2.2.3 **PMUA->DWELL**

PMUA	**DWELL**	0x50000048
	Minimum times spent in various PMUA states. Please Note- The STARTUP_BIAS_DWELL state timeout is hardcoded to a value of 0xC. A value of 0xC in the STARTUP_BIAS_DWELL state @ 16KHz yields a delay of 1.5 milliseconds	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								A				B				C				D				E				F			

#	Bit(s)	Field	Description	Type	Reset
A	23:20	ENABLE_MAIN_REG	*Enable main regulator dwell time* Defines the amount of time spent in the 'Enable main reg' state. Allows the main 3v3 regulator to power up before the first load (the bias) is enabled — read-only, unless 'trim access' is enabled	r/w	*0xC*
B	19:16	POWER_DOWN_MCU	*Power down MCU dwell time*	r/w	*0xC*

#	Bit(s)	Field	Description	Type	Reset
			Defines the amount of time spent in the 'Power down MCU' state. Pausing here allows the MCU supplies to discharge (to guarantee subsequent POR) — read-only, unless 'trim access' is enabled		
C	15:12	ATTACH_3V3	*Attach 3.3V dwell time* Defines the amount of time spent in the 'Attach 3V3' state. State Attaches 3.3V to MCU — read-only, unless 'trim access' is enabled	r/w	*0xC*
D	11:8	ATTACH_1V5	*Attach 1.5V dwell time* Defines the amount of time spent in the 'Attach 1V5' state. State Attaches 1.5V to MCU — read-only, unless 'trim access' is enabled	r/w	*0xC*
E	7:4	ENABLE_1V5	*Enable 1.5V dwell time* Defines the amount of time spent in the 'Enable 1V5' state. Allows 3v3 and 1v5 regs to settle — read-only, unless 'trim access' is enabled	r/w	*0xC*
F	3:0	ENABLE_BIAS	*Enable bias dwell time* Defines the amount of time spent in the 'Enable bias' state. Allows the 3v3 reg to settle	r/w	*0xC*

#	Bit(s)	Field	Description	Type	Reset
			— read-only, unless 'trim access' is enabled		

7.2.2.4 **PMUA->VBATCTRL**

PMUA	**VBATCTRL**	0x5000004C
	VBAT Monitor Register	

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
				A	B	C	D																					E	F	G	H

#	Bit(s)	Field	Description	Type	Reset
A	27	HIGH_DBNC	*Battery Voltage High Status after deboucing* Battery monitor over voltage event signal coming from the debouncer of analog comparator circuit	*ro*	0
B	26	LOW_DBNC	*Battery Voltage Low Status after debouncing* Battery monitor under voltage event signal coming from the debouncer of analog comparator circuit	*ro*	0
C	25	HIGH	*Battery Voltage High Status* RAW battery monitor over voltage event signal coming from the analog comparator circuit	*ro*	0
D	24	LOW	*Battery Voltage Low Status*	*ro*	0

#	Bit(s)	Field	Description	Type	Reset
			RAW battery monitor under voltage event signal coming from the analog comparator circuit		
E	3	OV_POL	Battery Voltage Monitor Over Voltage Interrupt Event Polarity. Flips the over voltage event signal coming from the analog comparator circuit which feeds into the interrupt generator 0x**0** — Native Polarity 0x**1** — Flip Polarity	r/w	0
F	2	UV_POL	Battery Voltage Monitor Under Voltage Interrupt Event Polarity. Flips the under voltage event signal coming from the analog comparator circuit which feeds into the interrupt generator 0x**0** — Native Polarity 0x**1** — Flip Polarity	r/w	0
G	1	OV_MONITOR_ENA	*Battery Over Voltage Monitor Enable* Set to enable the vbat over voltage monitor analog comparator circuit	r/w	0
H	0	LOW_MONITOR_ENA	*Battery Under Voltage Monitor Enable* Set to enable the vbat under voltage monitor analog comparator circuit	r/w	0

7.2.2.5 **PMUA->VBATTRIM**

PMUA	**VBATTRIM**	0x50000050

VBAT Monitor Trim Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						A						B										C				D					

#	Bit(s)	Field	Description	Type	Reset
A	25:24	OVHYS	*Battery Voltage Monitor Over Voltage Hysterisis Select* Selects the hysterisis level for the Over Voltage monitor 0x**0** — 0.44V 0x**1** — 1.13V 0x**2** — 1.77V 0x**3** — 2.36V	r/w	*1*
B	19:16	OVLEVEL	*Battery Voltage Monitor Over Voltage Select* Selects the reference level for the Over Voltage monitor. If OVLEVEL>9, the actual analog trim value is the same as OVLEVEL==0 0x**0** — Over Voltage Threshold- Vr: 14.72V Vf: 13.97V 0x**1** — Over Voltage Threshold- Vr: 15.21V Vf: 14.42V 0x**2** — Over Voltage Threshold- Vr: 15.74V Vf: 14.90V 0x**3** — Over Voltage Threshold- Vr: 16.30V Vf: 15.41V 0x**4** — Over Voltage Threshold- Vr: 16.90V Vf: 15.96V 0x**5** — Over Voltage Threshold- Vr: 17.55V Vf: 16.55V 0x**6** — Over Voltage Threshold- Vr: 18.24V Vf: 17.18V 0x**7** — Over Voltage Threshold- Vr: 19.00V Vf: 17.87V 0x**8** — Over Voltage Threshold- Vr: 19.82V Vf: 18.61V 0x**9** — Over Voltage Threshold- Vr: 20.72V Vf: 19.42V	r/w	*7*

#	Bit(s)	Field	Description	Type	Reset
C	9:8	UVHYS	*Battery Voltage Monitor Under Voltage Hysterisis Select* Selects the hysterisis level for the Under Voltage monitor 0x**0** — 127mV 0x**1** — 199mV 0x**2** — 273mV 0x**3** — 351mV	r/w	*1*
D	5:0	UVLEVEL	*Battery Voltage Monitor Under Voltage Select* Selects the reference level for the Under Voltage monitor. If UVLEVEL>39, the actual analog trim value is the same as UVLEVEL==0. The UVLEVEL threshold is mononically increased with the setting 0x**0** — UV Threshold - Vr: 4.82V Vf: 4.66V 0x**9** — UV Threshold - Vr: 5.55V Vf: 5.36V	r/w	*9*

7.2.2.6 **PMUA->VBATDBNC**

PMUA	**VBATDBNC**	0x50000054
	VBAT Debounce Register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																										A	B	C	D	E	F

#	Bit(s)	Field	Description	Type	Reset
A	5	OVSTRB1SEL	*Low frequency strobing select for debouncing* Enables strobing 1 with the Low frequency clock(16K) instead of just clocking with High frequency Clock(16M). Default is set to 0 for quick acknowledgement of OV event	r/w	0

#	Bit(s)	Field	Description	Type	Reset
B	4	OVSTRB0SEL	*Low frequency strobing select for debouncing* Enables strobing 0 with the Low frequency clock(16K) instead of just clocking with High frequency Clock(16M). Default is set to 1 for debouncing of OV	r/w	*1*
C	3	UVSTRB1SEL	*Low frequency strobing select for debouncing* Enables strobing 1 with the Low frequency clock(16K) instead of just clocking with High frequency Clock(16M). Default is set to 0 for quick acknowledgement of UV event	r/w	0
D	2	UVSTRB0SEL	*Low frequency strobing select for debouncing* Enables strobing 0 with the Low frequency clock(16K) instead of just clocking with High frequency Clock(16M). Default is set to 1 for debouncing of UV	r/w	*1*
E	1	OV	*over voltage signal debounce enable* if set to ‘1, debounces the over voltage signal before going to over voltage interrupt generation else the signal bypass the debouncer	r/w	*1*
F	0	UV	*under voltage signal debounce enable* if set to ‘1, debounces the under voltage signal before going to under voltage interrupt generation else the signal bypass the debouncer	r/w	*1*

7.2.2.7 **PMUA->VBATDBNCTHRES**

PMUA	**VBATDBNCTHRES**	0x50000058

	VBAT Monitor Threshold Register																														
31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
A								B								C								D							

#	Bit(s)	Field	Description	Type	Reset
A	31:24	OVTHRES0	*Over Voltage debouncing threshold for 1to0 Transition* Sets the threshold for debouncing the over voltage event going from 1 to 0. It will require the ov event to stay high for the (OVTHRES0+1) number of LF/HF clocks to transit from a 1 to 0	r/w	*0xFF*
B	23:16	UVTHRES0	*Under Voltage debouncing threshold for 1to0 Transition* Sets the threshold for debouncing the under voltage event going from 1 to 0. It will require the uv event to stay high for the (UVTHRES0+1) number of LF/HF clocks to transit from a 1 to 0	r/w	*0xFF*
C	15:8	OVTHRES1	*Over Voltage debouncing threshold for 0to1 Transition* Sets the threshold for debouncing the over voltage event going from 0 to 1. It will require the ov event to stay high for the (OVTHRES1+1) number of LF/HF clocks to transit from a 0 to 1	r/w	*1*
D	7:0	UVTHRES1	*Under Voltage debouncing threshold for 0to1 Transition* Sets the threshold for debouncing the under voltage event going from 0 to 1. It will require the uv event to stay high for the (UVTHRES1+1) number of LF/HF clocks to transit from a 0 to 1	r/w	*1*

7.2.2.8 **PMUA->PMUIRQ**

PMUA	**PMUIRQ**	0x5000005C
	Voltage Monitor interrupts	

	Contains the the enable, clear, status and active flag for the Battery Voltage interrupt sources.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						A	B							C	D							E	F							G	H

#	Bit(s)	Field	Description	Type	Reset
A	25	OV	*over voltage interrupt active*	*ro*	0
B	24	UV	*under voltage interrupt active*	*ro*	0
C	17	OV	*over voltage interrupt status*	*ro*	0
D	16	UV	*under voltage interrupt status*	*ro*	0
E	9	OV	*over voltage interrupt clear* – cleared automatically after each write	*wo*	0
F	8	UV	*under voltage interrupt clear* – cleared automatically after each write	*wo*	0
G	1	OV	*over voltage interrupt enable*	r/w	0
H	0	UV	*under voltage interrupt enable*	r/w	0

7.2.3 **EVTHOLD**

Event Hold Control

7.2.3.1 **EVTHOLD->HOLD**

EVTHOLD	**HOLD**	0x50000060

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															A

#	Bit(s)	Field	Description	Type	Reset
A	0	HOLD	Set to prevent serialisation of new non-wakeup events in prepartion for hibernate mode. At the point of becoming set, a request to send the lullaby interrupt is automatically generated. The lullaby handler can then safely assert the PMUA->CTRL.HIBERNATE bit in order to put the device into hibernate mode	*wo*	0

7.2.4 **BTE**

7.2.4.1 **BTE->BTE_CTRL**

BTE	**BTE_CTRL**	0x50000080
	BTE Control Register This register can only be written if there is no ongoing transfer. If the BTE is transferring data, any writes to this register will be ignore	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				A	B	C	D	E								F															

#	Bit(s)	Field	Description	Type	Reset
A	27	START	An operation is initiated when this bit is set. The bit auto-clears when the block is complete	r/w	0
B	26	BLOCKING	If set then the block transfer has priority over the MCU. If the MCU tries to use the bus, it will stall until the block transfer is complete. If not set, then the MCU waits only until the remainder of the current word completes and then waits until the bus is idle again before continuing	r/w	0

#	Bit(s)	Field	Description	Type	Reset
C	25	TX_DIR	Transfer direction. If set then SRAM->ASIC otherwise ASIC->SRAM	r/w	0
D	24	INC_ADDR	if set then ASIC die address increments at the end of each transfer. Set to zero if the peripheral is a FIFO	r/w	0
E	23:16	BXNUM	*Number of 32-bit words to transfer*	r/w	0
F	15:0	BXADD	*Address of the ASIC die (LSB)* This is the lower 16 bits of the ASIC die address. The MSBs are 0x5001	r/w	0

7.2.4.2 **BTE->BTE_SRAM_ADDR**

BTE	**BTE_SRAM_ADDR**	0x50000084
	BTE SRAM Address Register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																A															

#	Bit(s)	Field	Description	Type	Reset
A	15:0	BXSRAMADDR	*Address of the SRAM (LSB)* This is the lower 16 bits of the SRAM address. The MSBs are 0x2000	r/w	0

7.2.5 WICA

WakeUp Interrupt Controller

7.2.5.1 **WICA->CTRL**

WICA	**CTRL**	0x50010100
	Wakeup Control Register This is the control register for wakeup via gpio or lin or wut	

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
																A	B		C	D	E		F	G				H	I		J

#	Bit(s)	Field	Description	Type	Reset
A	15	LINMIRQCLR	*clear the wulin_irq* writting a ‘1 to this register will clear the wulins_irq	*wo*	0
B	14	TIMERIRQCLR	*clear the wutimer_irq* writting a ‘1 to this register will clear the wutimer_irq	*wo*	0
C	12	LINSIRQCLR	*clear the wulin_irq* writting a ‘1 to this register will clear the wulins_irq	*wo*	0
D	11	LINMIRQENA	*LIN Master Wakeup Interrupt Enable* if set, wulinm_irq is asserted if a wakeup signal is detected on the LIN_OUT bus	r/w	0
E	10	TIMERIRQENA	*Timer Wakeup Interrupt Enable* if set, wutimer_irq is asserted if wakeup timer matches the tapsel	r/w	*1*
F	8	LINSIRQENA	*LIN Slave Wakeup Interrupt Enable*	r/w	*1*

#	Bit(s)	Field	Description	Type	Reset
			if set, wulin_irq is asserted if a wakeup signal is detected on the LIN_IN bus		
G	7:4	TIMERTAPSEL	*WakeUp Timer Tap Select* Wakeup Time = 2^(WUT_TAPSEL) x Tlfclk(62.5us)	r/w	*4*
H	3	LINMENA	*LIN Master Wakeup Enable* it enables the detect of a wakeup signal on the LIN_OUT bus	r/w	0
I	2	TIMERENA	*Wakeup Timer Enable* it enables the wakeup timer	r/w	0
J	0	LINSENA	*LIN Slave Wakeup Enable* it enables the detect of a wakeup signal on the LIN_IN bus	r/w	0

7.2.5.2 **WICA->STATUS**

WICA	**STATUS**	0x50010104
	Wakeup Status Register This is the status register for wakeup via gpio or lin or wut	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																												B	C		D

#	Bit(s)	Field	Description	Type	Reset
A	31:16	TIMERCNT	*Wakeup Timer Counter Value* Counter Value of the Wakeup Timer	*ro*	0
B	3	LINM	*LIN Master Wakeup Status* This gets set if a wakeup signal is detected on the LIN_OUT bus when LINM SLEEP bit is set. CLRIRQ clears this Register	*ro*	0
C	2	TIMER	*Wakeup Timer Status* This gets set if a wakeup timer is enabled and the count matches the tapsel setting during hibernate, CLRIRQ clears this Register	*ro*	0
D	0	LINS	*LIN Slave Wakeup Status* This gets set if a wakeup signal is detected on the LIN_IN bus when LINS SLEEP bit is set. CLRIRQ clears this Register	*ro*	0

7.2.6 **WDTA**

Watchdog Timer (ASIC)

7.2.6.1 **WDTA->CTRL**

WDTA	**CTRL**	0x50010200
	Control	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																	B	C		D	E									F	G

#	Bit(s)	Field	Description	Type	Reset
A	31	STOP_LOCK	Set Only bit. Set this bit to lock STOP bits	r/w	0
B	14	WDTA_WINOPEN_FLAG	*Window open flag* A flag that indicates when the watchdog window is open. It only can be cleared by Reg CLEAR! 0x**0** — Window is Closed 0x**1** — Window is Open	*ro*	0
C	13:12	WDTA_WINOPEN_SEL	*Window Mode open select* Defines the watchdog window open time (the time between the watchdog start and the window open) 0x**0** — 1/2 * WDT timeout 0x**1** — 1/4 * WDT timeout 0x**2** — 1/8 * WDT timeout 0x**3** — 1/16* WDT timeout	r/w	0
D	11	WDTA_WINOPEN_ENA	*Window Mode Enable* Enables Window Mode. 1’b1: Enable the Window mode of Watchdog, if the WDT is cleared before the time window opens, the WDT will issue a system reset.	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			1'b0: Disable the Window mode of Watchdog		
E	10:8	WDTA_TIMEOUT_SEL	*Timeout select* Defines the watchdog timeout period (the time between a clear operation and the next timeout) 0x**0** — 2^{11} * 62.5us ~= 128 ms 0x**1** — 2^{12} * 62.5us ~= 256 ms 0x**2** — 2^{13} * 62.5us ~= 512 ms 0x**3** — 2^{14} * 62.5us ~= 1.024s 0x**4** — 2^{15} * 62.5us ~= 2.048s 0x**5** — 2^{16} * 62.5us ~= 4.096s 0x**6** — 2^{17} * 62.5us ~= 8.192s 0x**7** — 2^{18} * 62.5us ~= 16.384s	r/w	*7*
F	1	UPDATE	*Window Mode Enable* Set to update Analog-Watchdog Configurations. NOTE: DO NOT change the CTRL register when it is high, which indicates there is an update in progress, It gets cleared by the core when the current update is done	*dual*	0
G	0	WDTA_RUNNING	*Analog-Watchdog Running status* A flag that indicates when the Analog-Watchdog timer is enabled	*ro*	0

#	Bit(s)	Field	Description	Type	Reset
			0x**0** — Analog-Watchdog timer is stopped and cleared 0x**1** — Analog-Watchdog timer is running		

7.2.6.2 **WDTA->STOP**

WDTA	**STOP**	0x50010204

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
A																															

#	Bit(s)	Field	Description	Type	Reset
A	31:0	STOP	Write the *stop* code (0x6da475c3) to this register to reset the timer and disable the watchdog (e.g. during debug). If any other value is written to this register the watchdog will be enabled — read-only, unless 'stop access' is enabled	r/w	0

7.2.6.3 **WDTA->CLEAR**

WDTA	**CLEAR**	0x50010208

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
A																															

#	Bit(s)	Field	Description	Type	Reset
A	31:0	CLEAR	Write the value 0x3c574ad6 (as a single word access) to reset the watchdog timer. Periodically performing this action is the expected method of preventing the watchdog from timing out (and resetting the MCU)	*wo*	0

7.2.6.4 **WDTA->CNTVAL**

WDTA	**CNTVAL**	0x5001020C
	Counter value	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																															

#	Bit(s)	Field	Description	Type	Reset
A	31:0	CNTVAL	*Counter value* The instantaneous value of watchdog timeout counter	*ro*	0

7.2.6.5 **WDTA->INT**

WDTA	**INT**	0x50010210
	WDTA Interrupts Contains the ENABLE, CLEAR, STATUS and IRQ for the UART interrupt sources.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						A	B							C	D							E	F							G	H

#	Bit(s)	Field	Description	Type	Reset
A	25	WINOPEN	*Window Open Interrupt* Set by WDTA timeout	*ro*	0
B	24	WDTA	*WDTA timeout Interrupt* Set by WDTA timeout	*ro*	0
C	17	WINOPEN	*Window Open Status*	*ro*	0

#	Bit(s)	Field	Description	Type	Reset
			Set by WDTA timeout		
D	16	WDTA	*WDTA timeout Status* Set by WDTA timeout	*ro*	0
E	9	WINOPEN	*Window Open Interrupt Clear* – cleared automatically after each write	*wo*	0
F	8	WDTA	*WDTA timeout Interrupt Clear* – cleared automatically after each write	*wo*	0
G	1	WINOPEN	*Window Open Interrupt Enable*	r/w	0
H	0	WDTA	*WDTA timeout Interrupt Enable*	r/w	0

7.2.7 **LINS**

LINS_ATHENS

7.2.7.1 **LINS->DATABYTE1**

LINS	**DATABYTE1**	0x50010300
	Data Byte 1	

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
																								A							

#	Bit(s)	Field	Description	Type	Reset
A	7:0	DATABUF1	*Data Buffer 1* 1st byte of the 8-byte Data Buffer	r/w	0

7.2.7.2 **LINS->DATABYTE2**

LINS	**DATABYTE2**	0x50010304
	Data Byte 2	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								A							

#	Bit(s)	Field	Description	Type	Reset
A	7:0	DATABUF2	*Data Buffer 2* 2nd byte of the 8-byte Data Buffer	r/w	0

7.2.7.3 **LINS->DATABYTE3**

LINS	**DATABYTE3**	0x50010308
	Data Byte 3	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								A							

#	Bit(s)	Field	Description	Type	Reset
A	7:0	DATABUF3	*Data Buffer 3* 3rd byte of the 8-byte Data Buffer	r/w	0

7.2.7.4 **LINS->DATABYTE4**

LINS	**DATABYTE4**	0x5001030C
	Data Byte 4	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								A							

#	Bit(s)	Field	Description	Type	Reset
A	7:0	DATABUF4	*Data Buffer 4* 4th byte of the 8-byte Data Buffer	r/w	0

7.2.7.5 **LINS->DATABYTE5**

LINS	**DATABYTE5**	0x50010310
	Data Byte 5	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								A							

#	Bit(s)	Field	Description	Type	Reset
A	7:0	DATABUF5	*Data Buffer 5* 5th byte of the 8-byte Data Buffer	r/w	0

7.2.7.6 **LINS->DATABYTE6**

LINS	**DATABYTE6**	0x50010314
	Data Byte 6	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								A							

#	Bit(s)	Field	Description	Type	Reset
A	7:0	DATABUF6	*Data Buffer 6* 6th byte of the 8-byte Data Buffer	r/w	0

7.2.7.7 **LINS->DATABYTE7**

LINS	**DATABYTE7**	0x50010318
	Data Byte 7	

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
																								A							

#	Bit(s)	Field	Description	Type	Reset
A	7:0	DATABUF7	*Data Buffer 7* 7th byte of the 8-byte Data Buffer	r/w	0

7.2.7.8 **LINS->DATABYTE8**

LINS	**DATABYTE8**	0x5001031C
	Data Byte 8	

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
																								A							

#	Bit(s)	Field	Description	Type	Reset
A	7:0	DATABUF8	*Data Buffer 8* 8th byte of the 8-byte Data Buffer	r/w	0

7.2.7.9 **LINS->CTRL**

LINS	**CTRL**	0x50010320
	Control Register	

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
																								A	B	C	D	E	F	G	

#	**Bit(s)**	**Field**	**Description**	**Type**	**Reset**
A	7	STOP	*Stop Register* The host controller of the LIN slave has set this register if it handles a data request interrupt and can not make use of the frame content with the received identifier(e.g. extended identifiers). For that case the LIN slave stops the processing of the LIN communication until the next SYNC BREAK is detected. A read access to this bit delivers always the value 0	*wo*	0
B	6	SLEEP	*Sleep Request* The bit is used by the LIN core to determine whether the LIN bus is in Sleep Mode or not. The host controller has to set the bit after sending or receiving a Sleep Mode frame. If DIS_AUTOSLEEP = 0, the bit will be set automatically by the LIN core if a bus idle timeout is detected. The host controller has to clear the bit after a wakeup signal is detected	r/w	0
C	5	TRANSMIT	*Transmit Operation* The bit determines whether the current frame is a transmit frame or a receive frame for the LIN node. It has to be set by the host controller	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			0x**0** — receive operation 0x**1** — transmit operation		
D	4	DATAACK	*Data Acknowledgement* The bit has to be set by the host controller of a LIN slave after handling a data request interrupt (compare STATUS.DATA_REQ register). The bit will be reset by the LIN core	r/w	0
E	3	RSTINT	*Reset interrupt* The host controller has to set this bit to reset the STATUS.INTR register and the interrupt request output of the LIN core. A read access to this bit delivers always the value 0	*wo*	0
F	2	RSTERR	*Reset Error* The host controller has to set this bit to reset the error bits in status register and error register. A read access to this bit delivers always the value 0	*wo*	0
G	1	WAKEUPREQ	*WakeUp Request* The bit has to be set by the host controller to terminate the Sleep Mode of the LIN bus by sending a Wakeup signal. The bit will be reset by the LIN core	r/w	0

7.2.7.10 **LINS->STATUS**

LINS	**STATUS**	0x50010324

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								A	B	C	D	E	F	G	H

#	Bit(s)	Field	Description	Type	Reset
A	7	ACTIVE	*Lin Bus Active* The bit indicates whether the LIN bus is active or not. Note: For the LIN slave, this bit is set after the detection of a correct SYNC BREAK / SYNC FIELD sequence and it is reset at the end of the transmission or if the processing of the current frame is stopped by the host controller 0x**0** — no Lin bus activity 0x**1** — transmission on the LIN bus is active	*ro*	0
B	6	BUSIDLETIMEOUT	*BUS Idle Timeout* This bit is set by the LIN core if LIN is in hardware mode and no bus activity is detected for 4s~10s. In addition, an interrupt request to the host controller is generated in that case. After that, It is assumed that the LIN bus is in sleep mode and CTRL.SLEEP register will be set by the LIN core. The bit has to be reset by the host controller by setting the bit CTRL.RST_INT register	*ro*	0
C	5	ABORTED	The bit is set if the processing of the current frame has been stopped by setting CTRL.STOP register. The bit is cleared by the LIN core after receiving a correct SYNC BREAK / SYNC FIELD sequence	*ro*	0
D	4	DATAREQ	*Data Request*	*ro*	0

#	Bit(s)	Field	Description	Type	Reset
			The LIN core slave sets the bit after receiving the Identifier and requests an interrupt to the host controller. The host controller has to decode the Identifier to decide whether the current frame is a transmit or a receive operation. It has to adjust CTRL.TRANSMIT register and to load the data length. For transmit operations it has to load the data buffer too. After that the host controller has to set CTRL.DATA_ACK register		
E	3	INTR	*Interupt Request* The LIN core sets the bit when it requests an interrupt to the host controller. It has the same value as the interrupt output INTR. The bit has to be reset by the host controller by setting the bit CTRL.RST_INT register	*ro*	0
F	2	ERROR	*Lin Error* The LIN core sets the bit if an error has been detected (compare error register). The bit has to be reset by the host controller by setting the bit CTRL.RST_ERR register	*ro*	0
G	1	WAKEUP	The bit is set when the LIN core is transmitting a Wakeup signal	*ro*	0
H	0	COMPLETE	The LIN core will set the bit after a transmission has been successfully finished and it will reset it at the start of a transmission	*ro*	0

7.2.7.11 LINS->ERROR

LINS	**ERROR**	0x50010328
	Error Register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																									A	B	C	D	E	F	G

#	Bit(s)	Field	Description	Type	Reset
A	6	FRAMEERR	*Byte Field Framing Error* This bit is set by the LIN core slave if a transmission is aborted after the beginning of the data field due to a timeout or an incomplete frame	*ro*	0
B	5	SBITERR	*Start Bit Error in Byte field* Start Bit Error in Byte field, i.e., invalid start bit	*ro*	0
C	4	BITMON	*Bit Monitor Error* The Bit value monitored on the bus is different from the sent bit value	*ro*	0
D	3	PARITY	*Parity Error* Identifier parity error	*ro*	0
E	2	TIMEOUT	*Timeout Error* There are several reason that can cause a timeout error: The master detects a timeout error if it is expecting data from the bus but no slave does respond. If the slave responds to late and the frame is not finished within the maximum frame length TFRAME_MAX a timeout error will be detected too.	*ro*	0

#	Bit(s)	Field	Description	Type	Reset
			The slave detects a timeout error if it is requesting a data acknowledge to the host controller (for selecting receive or transmit, data length and loading data), and the host controller does not set CTRL.DATA_ACK or CTRL.STOP register until the end of the reception of the first byte after the identifier. The slave detects a timeout error if it has transmitted a wakeup signal and it detects no sync field (from the master) within 150 ms. Note: The slave does not perform an exact check of the frame length TFRAME_MAX but a timeout is detected after 200 bit times, if the slave is in receive mode and there are missing data fields or a missing ID field from the master		
F	1	CHK	*Checksum Error* Checksum Error	*ro*	0
G	0	BITERR	*Bit Error in Byte field* Bit Error in Byte field, i.e., invalid stop bit	*ro*	0

7.2.7.12 **LINS->DL**

LINS	**DL**	0x5001032C
	DATA Length Register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								A	B	C		D			

#	Bit(s)	Field	Description	Type	Reset
A	7	ENHCHK	*Enhancement Check*	r/w	0

<table>
<tr><th>#</th><th>Bit(s)</th><th>Field</th><th>Description</th><th>Type</th><th>Reset</th></tr>
<tr><td></td><td></td><td></td><td>The host controller has to set the checksum type used in the current frame by adjusting this register

0x0 — for classic checksum

0x1 — for enhanced checksum</td><td></td><td></td></tr>
<tr><td>B</td><td>6</td><td>DISBITMON</td><td>Disable Bit Monitor

Set to disable the bit monitor during transmission. The bit must be set in case that RXD/TXD are separated</td><td>r/w</td><td>0</td></tr>
<tr><td>C</td><td>5</td><td>DISAUTOSLEEP</td><td>Disable Auto Sleep

Set to Disable auto sleep</td><td>r/w</td><td>0</td></tr>
<tr><td>D</td><td>3:0</td><td>LENGTH</td><td>Data Length

The host controller has to define the length of the data field of the current LIN frame by adjusting the data length register. If the data length is loaded with the value 1111b the length of the data field is decoded from Bit 5 and 4 of the identifier register id according to the Table below (e.g. compatibility to LIN specification 1.1). Otherwise the amount of data bytes can be written directly to the data length register (supported values are 0..8).

<table>
<tr><th>ID(Bit 5)</th><th>ID(Bit 4)</th><th>Number of Bytes in the data field</th></tr>
<tr><td>0</td><td>0</td><td>2</td></tr>
<tr><td>0</td><td>1</td><td>2</td></tr>
<tr><td>1</td><td>0</td><td>4</td></tr>
<tr><td>1</td><td>1</td><td>8</td></tr>
</table></td><td>r/w</td><td>0</td></tr>
</table>

7.2.7.13 LINS->BTDIV07

LINS	**BTDIV07**	0x50010330

	Bit time Divider Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								A							

#	Bit(s)	Field	Description	Type	Reset
A	7:0	BTDIV07	*Bt Div LSBs* Bit time divider [7:0]	r/w	*0xFF*

7.2.7.14 **LINS->BITTIME**

LINS	**BITTIME**	0x50010334
	Control Settings	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								A							B

#	Bit(s)	Field	Description	Type	Reset
A	7:6	PRESCL	*Prescaler Register* Prescaler Setting	r/w	*3*
B	0	BTDIV8	*Bt Div Most Significant bit* Bit time divider [8]	r/w	*1*

7.2.7.15 **LINS->ID**

LINS	**ID**	0x50010338
	ID Register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																										A					

#	Bit(s)	Field	Description	Type	Reset
A	5:0	ID	ID register	r/w	0

7.2.7.16 **LINS->BUSTIME**

LINS	**BUSTIME**	0x5001033C
	Lin Bus Timing Register	

Bit3	Bit2	Bit1	Bit0	Time
0	0	0	0	Reset value
0	0			4s (bus_inactivity_time)
0	1			6s (bus_inactivity_time)
1	0			8s (bus_inactivity_time)
1	1			10s (bus_inactivity_time)
		0	0	180ms (wup_repeat_time)
		0	1	200ms (wup_repeat_time)
		1	0	220ms (wup_repeat_time)
		1	1	240ms (wup_repeat_time)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																											A	B		C	

#	Bit(s)	Field	Description	Type	Reset
A	4	BUSDOMINANTRELEASEWUPENA	*Bus Dominant Release Wakeup Enable*	r/w	0
B	3:2	BUSINACTIVE	*Bus Inactivity Time*	r/w	0
C	1:0	WUPREPEAT	*wakeup repeat time*	r/w	0

7.2.7.17 **LINS->STATUSEXT**

LINS	**STATUSEXT**	0x50010340

	Extended Status

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																												A		B	C

#	Bit(s)	Field	Description	Type	Reset
A	3:2	BUSIDLEMONITOR	*Bus Idle Monitor Status* If LIN is in hardware mode, BIT[0] is set by the lin core if the bus has no activity for 4s~10s and BIT[1] is set by the lin core if LIN is stuck at dominant inactivity state for 4s~10s. Any bus transition will clear these two bits. LIN Slave's pullup will be disabled & LIN Master's pullup will be reduced to 30K when BIT[1] is set	*ro*	0
B	1	BUSIDLETIMEOUTDOMINANT	*Dominant Bus Idle Timeout* The bit is set by the lin core if LIN is in hardware mode & the bus is stuck at dominant inactivity state for 4s~10s. The bit has to be reset by the host controller by setting the bit CTRL.RST_INT register	*ro*	0
C	0	COMPLETETX	*Complete TX* The LIN core will set the bit after a TX transmission has been successfully finished and it will reset it at the start of a transmission	*ro*	0

7.2.7.18 **LINS->WUPDETECTTHRES**

LINS	**WUPDETECTTHRES**	0x50010344
	Wakeup Detection Threshold	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																										A					

#	Bit(s)	Field	Description	Type	Reset
A	5:0	WUPDETECTTHRES	*Wakeup Detection Threshold* Threshold setting(LF_CLK_PREDIV clock cycls) of lin wakeup signal. Only take effect when CLK_LF_SEL = 1; For instace, if target threshold is 150us@LF_CLK_PREDIV=256KHz, WUP_DETECT_THRES = 150/(1000/256)+1	r/w	*0x29*

7.2.8 **LINM**

LINM_ATHENS

7.2.8.1 **LINM->DATABYTE1**

LINM	**DATABYTE1**	0x50010400
	Data Byte 1	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								A							

#	Bit(s)	Field	Description	Type	Reset
A	7:0	DATABUF1	*Data Buffer 1*	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			1st byte of the 8-byte Data Buffer		

7.2.8.2 **LINM->DATABYTE2**

LINM	**DATABYTE2**	0x50010404
	Data Byte 2	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								A							

#	Bit(s)	Field	Description	Type	Reset
A	7:0	DATABUF2	*Data Buffer 2* 2nd byte of the 8-byte Data Buffer	r/w	0

7.2.8.3 **LINM->DATABYTE3**

LINM	**DATABYTE3**	0x50010408
	Data Byte 3	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								A							

#	Bit(s)	Field	Description	Type	Reset
A	7:0	DATABUF3	*Data Buffer 3* 3rd byte of the 8-byte Data Buffer	r/w	0

7.2.8.4 **LINM->DATABYTE4**

LINM	**DATABYTE4**	0x5001040C

	Data Byte 4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								A							

#	Bit(s)	Field	Description	Type	Reset
A	7:0	DATABUF4	*Data Buffer 4* 4th byte of the 8-byte Data Buffer	r/w	0

7.2.8.5 **LINM->DATABYTE5**

LINM	**DATABYTE5**	0x50010410
	Data Byte 5	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								A							

#	Bit(s)	Field	Description	Type	Reset
A	7:0	DATABUF5	*Data Buffer 5* 5th byte of the 8-byte Data Buffer	r/w	0

7.2.8.6 **LINM->DATABYTE6**

LINM	**DATABYTE6**	0x50010414
	Data Byte 6	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								A							

#	Bit(s)	Field	Description	Type	Reset
A	7:0	DATABUF6	*Data Buffer 6* 6th byte of the 8-byte Data Buffer	r/w	0

7.2.8.7 **LINM->DATABYTE7**

LINM	**DATABYTE7**	0x50010418
	Data Byte 7	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								A							

#	Bit(s)	Field	Description	Type	Reset
A	7:0	DATABUF7	*Data Buffer 7* 7th byte of the 8-byte Data Buffer	r/w	0

7.2.8.8 **LINM->DATABYTE8**

LINM	**DATABYTE8**	0x5001041C
	Data Byte 8	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								A							

#	Bit(s)	Field	Description	Type	Reset
A	7:0	DATABUF8	*Data Buffer 8* 8th byte of the 8-byte Data Buffer	r/w	0

7.2.8.9 **LINM->CTRL**

LINM	**CTRL**	0x50010420
	Control Register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								A	B	C	D	E	F	G	H

#	Bit(s)	Field	Description	Type	Reset
A	7	STOP	*Stop Register* The host controller of the LIN slave has set this register if it handles a data request interrupt and can not make use of the frame content with the received identifier(e.g. extended identifiers). For that case the LIN slave stops the processing of the LIN communication until the next SYNC BREAK is detected. A read access to this bit delivers always the value 0	*wo*	0
B	6	SLEEP	*Sleep Request* The bit is used by the LIN core to determine whether the LIN bus is in Sleep Mode or not. The host controller has to set the bit after sending or receiving a Sleep Mode frame. If DIS_AUTOSLEEP = 0, the bit will be set automatically by the LIN core if a bus idle timeout is detected. The host controller has to clear the bit after a wakeup signal is detected	r/w	0
C	5	TRANSMIT	*Transmit Operation* The bit determines whether the current frame is a transmit frame or a receive frame for the LIN node. It has to be set by the host controller	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			0x**0** — receive operation 0x**1** — transmit operation		
D	4	DATAACK	*Data Acknowledgement* The bit has to be set by the host controller of a LIN slave after handling a data request interrupt (compare STATUS.DATA_REQ register). The bit will be reset by the LIN core	r/w	0
E	3	RSTINT	*Reset interrupt* The host controller has to set this bit to reset the STATUS.INTR register and the interrupt request output of the LIN core. A read access to this bit delivers always the value 0	*wo*	0
F	2	RSTERR	*Reset Error* The host controller has to set this bit to reset the error bits in status register and error register. A read access to this bit delivers always the value 0	*wo*	0
G	1	WAKEUPREQ	*WakeUp Request* The bit has to be set by the host controller to terminate the Sleep Mode of the LIN bus by sending a Wakeup signal. The bit will be reset by the LIN core	r/w	0
H	0	STARTREQ	*Start Request* The bit has to be set by the host controller of a LIN master to start the LIN transmission after loading identifier,data length and data buffer. The bit will be	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			reset by the LIN core after the transmission is finished or an error is occurred		

7.2.8.10 **LINM->STATUS**

LINM	**STATUS**	0x50010424

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
																								A	B	C	D	E	F	G	H

#	Bit(s)	Field	Description	Type	Reset
A	7	ACTIVE	*Lin Bus Active* The bit indicates whether the LIN bus is active or not. Note: For the LIN slave, this bit is set after the detection of a correct SYNC BREAK / SYNC FIELD sequence and it is reset at the end of the transmission or if the processing of the current frame is stopped by the host controller 0x**0** — no Lin bus activity 0x**1** — transmission on the LIN bus is active	*ro*	0
B	6	BUSIDLETIMEOUT	*BUS Idle Timeout* This bit is set by the LIN core if LIN is in hardware mode and no bus activity is detected for 4s~10s. In addition, an interrupt request to the host controller is generated in that case. After that, It is assumed that the LIN bus is in sleep mode and CTRL.SLEEP register will be set by the LIN core. The bit has to be reset by the host controller by setting the bit CTRL.RST_INT register	*ro*	0

#	Bit(s)	Field	Description	Type	Reset
C	5	ABORTED	The bit is set if the processing of the current frame has been stopped by setting CTRL.STOP register. The bit is cleared by the LIN core after receiving a correct SYNC BREAK / SYNC FIELD sequence	*ro*	0
D	4	DATAREQ	*Data Request* The LIN core slave sets the bit after receiving the Identifier and requests an interrupt to the host controller. The host controller has to decode the Identifier to decide whether the current frame is a transmit or a receive operation. It has to adjust CTRL.TRANSMIT register and to load the data length. For transmit operations it has to load the data buffer too. After that the host controller has to set CTRL.DATA_ACK register	*ro*	0
E	3	INTR	*Interupt Request* The LIN core sets the bit when it requests an interrupt to the host controller. It has the same value as the interrupt output INTR. The bit has to be reset by the host controller by setting the bit CTRL.RST_INT register	*ro*	0
F	2	ERROR	*Lin Error* The LIN core sets the bit if an error has been detected (compare error register). The bit has to be reset by the host controller by setting the bit CTRL.RST_ERR register	*ro*	0

#	Bit(s)	Field	Description	Type	Reset
G	1	WAKEUP	The bit is set when the LIN core is transmitting a Wakeup signal	*ro*	0
H	0	COMPLETE	The LIN core will set the bit after a transmission has been successfully finished and it will reset it at the start of a transmission	*ro*	0

7.2.8.11 **LINM->ERROR**

LINM	**ERROR**	0x50010428
	Error Register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																									A	B	C	D	E	F	G

#	Bit(s)	Field	Description	Type	Reset
A	6	FRAMEERR	*Byte Field Framing Error* This bit is set by the LIN core slave if a transmission is aborted after the beginning of the data field due to a timeout or an incomplete frame	*ro*	0
B	5	SBITERR	*Start Bit Error in Byte field* Start Bit Error in Byte field, i.e., invalid start bit	*ro*	0
C	4	BITMON	*Bit Monitor Error* The Bit value monitored on the bus is different from the sent bit value	*ro*	0
D	3	PARITY	*Parity Error*	*ro*	0

#	Bit(s)	Field	Description	Type	Reset
			Identifier parity error		
E	2	TIMEOUT	*Timeout Error* There are several reason that can cause a timeout error: The master detects a timeout error if it is expecting data from the bus but no slave does respond. If the slave responds to late and the frame is not finished within the maximum frame length TFRAME_MAX a timeout error will be detected too. The slave detects a timeout error if it is requesting a data acknowledge to the host controller (for selecting receive or transmit, data length and loading data), and the host controller does not set CTRL.DATA_ACK or CTRL.STOP register until the end of the reception of the first byte after the identifier. The slave detects a timeout error if it has transmitted a wakeup signal and it detects no sync field (from the master) within 150 ms. Note: The slave does not perform an exact check of the frame length TFRAME_MAX but a timeout is detected after 200 bit times, if the slave is in receive mode and there are missing data fields or a missing ID field from the master	*ro*	0
F	1	CHK	*Checksum Error* Checksum Error	*ro*	0
G	0	BITERR	*Bit Error in Byte field* Bit Error in Byte field, i.e., invalid stop bit	*ro*	0

7.2.8.12 **LINM->DL**

LINM	**DL**	0x5001042C
	DATA Length Register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								A	B	C		D			

#	Bit(s)	Field	Description	Type	Reset
A	7	ENHCHK	*Enhancement Check* The host controller has to set the checksum type used in the current frame by adjusting this register 0x**0** — for classic checksum 0x**1** — for enhanced checksum	r/w	0
B	6	DISBITMON	*Disable Bit Monitor* Set to disable the bit monitor during transmission. The bit must be set in case that RXD/TXD are separated	r/w	0
C	5	DISAUTOSLEEP	*Disable Auto Sleep* Set to Disable auto sleep	r/w	0
D	3:0	LENGTH	*Data Length* The host controller has to define the length of the data field of the current LIN frame by adjusting the data length register. If the data length is loaded with the value 1111b the length of the data field is decoded from Bit 5 and 4 of the identifier register id according to the Table below (e.g. compatibility to LIN specification 1.1). Otherwise the amount of	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			data bytes can be written directly to the data length register (supported values are 0..8).		

ID(Bit 5)	ID(Bit 4)	Number of Bytes in the data field
0	0	2
0	1	2
1	0	4
1	1	8

7.2.8.13 **LINM->BTDIV07**

LINM	**BTDIV07**	0x50010430
	Bit time Divider Register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								A							

#	Bit(s)	Field	Description	Type	Reset
A	7:0	BTDIV07	*Bt Div LSBs* Bit time divider [7:0]	r/w	*0xFF*

7.2.8.14 **LINM->BITTIME**

LINM	**BITTIME**	0x50010434
	Control Settings	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								A		B					C

#	Bit(s)	Field	Description	Type	Reset
A	7:6	PRESCL	*Prescaler Register* Prescaler Setting	r/w	*3*
B	5:1	BTMULT	*Bt Div Most Significant bit* Bit time multiplier [4:0]	r/w	*1*
C	0	BTDIV8	*Bt Div Most Significant bit* Bit time divider [8]	r/w	*1*

7.2.8.15 **LINM->ID**

LINM	**ID**	0x50010438
	ID Register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																										A					

#	Bit(s)	Field	Description	Type	Reset
A	5:0	ID	ID register	r/w	0

7.2.8.16 **LINM->BUSTIME**

LINM	**BUSTIME**	0x5001043C
	Lin Bus Timing Register	

Bit3	Bit2	Bit1	Bit0	Time
0	0	0	0	Reset value
0	0			4s (bus_inactivity_time)
0	1			6s (bus_inactivity_time)
1	0			8s (bus_inactivity_time)
1	1			10s (bus_inactivity_time)

		0	0	180ms (wup_repeat_time)
		0	1	200ms (wup_repeat_time)
		1	0	220ms (wup_repeat_time)
		1	1	240ms (wup_repeat_time)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																											A	B		C	

#	Bit(s)	Field	Description	Type	Reset
A	4	BUSDOMINANTRELEASEWUPENA	*Bus Dominant Release Wakeup Enable*	r/w	0
B	3:2	BUSINACTIVE	*Bus Inactivity Time*	r/w	0
C	1:0	WUPREPEAT	*wakeup repeat time*	r/w	0

7.2.8.17 **LINM->STATUSEXT**

LINM	**STATUSEXT**	0x50010440
	Extended Status	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																												A		B	C

#	Bit(s)	Field	Description	Type	Reset
A	3:2	BUSIDLEMONITOR	*Bus Idle Monitor Status* If LIN is in hardware mode, BIT[0] is set by the lin core if the bus has no activity for 4s~10s and BIT[1] is set by the lin core if LIN is stuck at dominant inactivity state for 4s~10s. Any bus transition will clear these two bits. LIN Slave's pullup will be	*ro*	0

#	Bit(s)	Field	Description	Type	Reset
			disabled & LIN Master's pullup will be reduced to 30K when BIT[1] is set		
B	1	BUSIDLETIMEOUTDOMINANT	*Dominant Bus Idle Timeout* The bit is set by the lin core if LIN is in hardware mode & the bus is stuck at dominant inactivity state for 4s~10s. The bit has to be reset by the host controller by setting the bit CTRL.RST_INT register	*ro*	0
C	0	COMPLETETX	*Complete TX* The LIN core will set the bit after a TX transmission has been successfully finished and it will reset it at the start of a transmission	*ro*	0

7.2.8.18 **LINM->WUPDETECTTHRES**

LINM	**WUPDETECTTHRES**	0x50010444
	Wakeup Detection Threshold	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																										A					

#	Bit(s)	Field	Description	Type	Reset
A	5:0	WUPDETECTTHRES	*Wakeup Detection Threshold* Threshold setting(LF_CLK_PREDIV clock cycls) of lin wakeup signal. Only take effect when CLK_LF_SEL = 1; For instace, if target threshold	r/w	*0x29*

#	Bit(s)	Field	Description	Type	Reset
			is 150us@LF_CLK_PREDIV=256KHz, WUP_DETECT_THRES = 150/(1000/256)+1		

7.2.9 **BUCKCTRL**

BUCK CONTRL Registers

7.2.9.1 **BUCKCTRL->CTRL**

BUCKCTRL	**CTRL**	0x50010500
	Buck Control Register Register Controls for the Buck	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																							A						B	C	D

#	Bit(s)	Field	Description	Type	Reset
A	8	ENIBOS2P6U	*Power Sense Bias Select* 1: Fast setup time;0: Slow setup time	r/w	0
B	2	ENBYPASSBT	*Enable CBT Charge* 1: Enable CBT charging in advance before Buck is enabled;0: No effects. If the chip is not debug mode, the bit will be cleared automatically if buck is enabled	*dual*	0
C	1	HICCUPENA	*Request to Enable Buck HICCUP*	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			setting this bit to ‘1 will enable the hiccup control of buck FSM		
D	0	ENAREQ	*Request to Enable the Buck* setting this bit to ‘1 will kickstart the statup sequence for the buck regulator. setting this bit to ‘0 will start the off sequence for the buck regulator. Auto clearred if hiccup is disabled & error conditions are encountered	*dual*	0

7.2.9.2 **BUCKCTRL->CLOCK**

BUCKCTRL	**CLOCK**	0x50010504
	Buck Clock setting Register Controls for setting the buck clock period & spreading options	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								A													B			C	D						

#	Bit(s)	Field	Description	Type	Reset
A	23:16	SSCDIV	*Buck SSC Clock Divider* Buck SSC Freq = Buck_Freq/[(SSC_DIV+1)*(SSC_DEEP+1)*4]	r/w	0
B	10:8	SSCDEEP	*Buck Clock Period Spread Depth Configuration* If Buck Clock spreading is enabled,	r/w	*3*

#	Bit(s)	Field	Description	Type	Reset
			the fastest buck_clk period =((PERIOD-SSC_DEEP-1)+1)*(HFOSC Period), the slowest buck_clk period = ((PERIOD+SSC_DEEP+1)+1)*(HFOSC Period)		
C	7	SPREAD	*Clock Spreading Enable* This enbales the buck clock spreading. Spreading if enabled should reduce the EMI effects of the buck due to switching(if any). If system clock SSC is enabled and the SSC freq is much lower than Buck Freq, the EMI effects of the buck will be also reduced	r/w	0
D	6:0	PERIOD	*Buck Clock Period* Buck clock Period is determined by this setting. if (PERIOD < 20) or (PERIOD > 54), PERIOD will be forced to safe value 35. Buck Clock will be of a period of (Buck clock Period+1)*(HFOSC Period). The bits will be changed if buck SSC is enabled. It is requird to reload the bits if SSC is disabled while the bits has been changed	*dual*	*0x17*

7.2.9.3 **BUCKCTRL->TDELAY**

BUCKCTRL	**TDELAY**	0x50010508
	Time Delay Setting Register Register Control for setting up various time delays	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																A		B			C	D				E					

#	Bit(s)	Field	Description	Type	Reset
A	15:14	DEBOUNCEOVERCURSTRBSEL	*strobing point for debouncing the buck overcur signal* this setting will decide when to look at the synchronized buck overcur signal for debouncing purpose 0x**0** — at every system clock 0x**1** — just before the buck clock posedge 0x**2** — just before the buck clock negedge 0x**3** — at both pos and neg edge of buck clock	r/w	0
B	13:12	ENAL	*time after ENAREQ goes low* this setting will decide the number buck clocks to buck after ENAREQ goes low before going IDLE 0x**0** — 1 buck clock period 0x**1** — 2 buck clock period 0x**2** — 3 buck clock period 0x**3** — 4 buck clock period	r/w	*2*
C	10	STARTH	*Buck Start signal High time* buck_start will stay high for at least 2 buck clock periods independent of buck_ready, at the end of 2 buck clock periods if buck_ready is low, it will stay high and it will go low when the buck_ready signal detected high, but if the buck_ready doesn't	r/w	*1*

#	Bit(s)	Field	Description	Type	Reset
			go high for a set timeout then buck_start goes low anyway 0x**0** — timeout after 8 buck clock periods 0x**1** — timeout after 16 buck clock periods		
D	9:8	BYPASSL	*Buck Bypass signal Low time* this setting will decide the time between buck bypass signal going low and start signal getting asserted. It's defined interms of number of buck clock periods 0x**0** — 1 buck clock period 0x**1** — 2 buck clock period 0x**2** — 3 buck clock period 0x**3** — 4 buck clock period	r/w	0
E	5:0	BYPASSH	*Buck Bypass Signal High time* this setting will decide the buck bypass signal high time. the bypass high time = (system clock period*128)*(BYPASSH+1) for example: for system clock at 16M, to set a bypass high time close to 100us, set BYPASSH to 0xC, this will have a delay of 8*(12+1)=104us	r/w	*0xC*

7.2.9.4 **BUCKCTRL->VDACCODE**

BUCKCTRL	**VDACCODE**	0x5001050C

	VDAC code used for Buck Trimming

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																		A						B							

#	Bit(s)	Field	Description	Type	Reset
A	13	RAMPRATE	*Ramp Rate of vdac_3v3 to reach to set vdac_trim* This register decides the rate at which the vdac_3v3 ramp up/down to reach the set vdac_trim 0x**0** — 1LSB per 2 buck clock period 0x**1** — 1LSB per 4 buck clock period	r/w	*1*
B	7:0	VDACTRIM	*TRIM code for VDAC* Control for trimming the VDAC. VBUCK_OUT = 1.375V/256*((VDAC_TRIM[7:0])+1)*6.25 and for limiting the upper output voltage, VDAC_TRIM[7:6] == 0x3 is not valid and will be taken as 0x2 — read-only, unless ‘trim access’ is enabled	r/w	*0x91*

7.2.9.5 **BUCKCTRL->BUCKIRQ**

BUCKCTRL	**BUCKIRQ**	0x50010510
	Buck interrupts Contains the the enable, clear, status and active flag for the Buck interrupt sources.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	A	B	C	D	E	F	G		H	I	J	K	L	M	N		O	P	Q	R	S	T	U		V	W	X	Y	Z	a	b

#	Bit(s)	Field	Description	Type	Reset
A	30	START_ERROR	*buck startup error errorinterrupt active*	*ro*	0
B	29	BUCK_OK	*voltage ok interrupt active*	*ro*	0
C	28	BUCK_OV	*over voltage interrupt active*	*ro*	0
D	27	BUCK_UV	*under voltage interrupt active*	*ro*	0
E	26	OVERCUR	*Buck Overcurrent detected interrupt active*	*ro*	0
F	25	DISABLED	*Buck Disable Complete interrupt active*	*ro*	0
G	24	READY	*Buck Ready interrupt active*	*ro*	0
H	22	START_ERROR	*Startup Error interrupt status*	*ro*	0
I	21	BUCK_OK	*voltage OK interrupt status*	*ro*	0
J	20	BUCK_OV	*over voltage interrupt status*	*ro*	0
K	19	BUCK_UV	*under voltage interrupt status*	*ro*	0
L	18	OVERCUR	*Buck Overcurrent detected interrupt status*	*ro*	0
M	17	DISABLED	*Buck Disable Complete interrupt status*	*ro*	0
N	16	READY	*Buck Ready interrupt status*	*ro*	0
O	14	START_ERROR	*Buck startup error interrupt/flags clear* – cleared automatically after each write	*wo*	0
P	13	BUCK_OK	*voltage ok interrupt clear* – cleared automatically after each write	*wo*	0
Q	12	BUCK_OV	*over voltage interrupt clear*	*wo*	0

#	Bit(s)	Field	Description	Type	Reset
			– cleared automatically after each write		
R	11	BUCK_UV	*under voltage interrupt clear* – cleared automatically after each write	*wo*	0
S	10	OVERCUR	*Buck Overcurrent detected interrupt clear* – cleared automatically after each write	*wo*	0
T	9	DISABLED	*Buck Disable Complete interrupt clear* – cleared automatically after each write	*wo*	0
U	8	READY	*Buck Ready interrupt clear* – cleared automatically after each write	*wo*	0
V	6	START_ERROR	*Buck Startup Error interrupt enable*	r/w	0
W	5	BUCK_OK	*voltage ok interrupt enable*	r/w	0
X	4	BUCK_OV	*over voltage interrupt enable*	r/w	0
Y	3	BUCK_UV	*under voltage interrupt enable*	r/w	0
Z	2	OVERCUR	*Buck Overcurrent detected interrupt enable*	r/w	0
a	1	DISABLED	*Buck Disable Complete interrupt enable*	r/w	0
b	0	READY	*Buck Ready interrupt enable*	r/w	0

7.2.9.6 **BUCKCTRL->STATUS**

BUCKCTRL	**STATUS**	0x50010514

	Status Register Status register for Buck & controller

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																			A									B			C

#	Bit(s)	Field	Description	Type	Reset
A	12:8	STATE	*Buck Controller State* Indicates the current state of the buck controller state machine 0x**0** — IDLE 0x**1** — HICCUP 0x**2** — PRE_ENA 0x**3** — DET_LOW 0x**4** — BYPASS_H 0x**5** — BYPASS_L 0x**6** — START_H 0x**7** — START_L_DBNC 0x**8** — START_L 0x**9** — START_L_OK 0x**A** — START_UV_DBNC 0x**B** — START_OV_DBNC 0x**C** — ENA_L 0x**D** — ENA_L_ERR	*ro*	0
B	3:1	BUCK_ERROR	*BUCK Startup Error Bits*	*ro*	0

#	Bit(s)	Field	Description	Type	Reset
			The flags of the buck startup error. Cleared by setting START_ERROR_CLR bit 0x**1** — vbuck_low not detected 0x**2** — vbuck_monitor_uv detected 0x**4** — vbuck_monitor_ok not detected		
C	0	BUCKOFF	*BUCK off status Register* the status of the buck_off signal	*ro*	0

7.2.9.7 **BUCKCTRL->VBUCKCTRL**

BUCKCTRL	**VBUCKCTRL**	0x50010518
	VBUCK Monitor Register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A	B		C								D									E									F	G	

#	Bit(s)	Field	Description	Type	Reset
A	31	HIGH	*Buck Voltage OV Status* RAW battery monitor over voltage event signal coming from the analog comparator circuit	*ro*	0
B	30	LOW	*Buck Voltage UV Status* RAW battery monitor under voltage event signal coming from the analog comparator circuit	*ro*	0

#	Bit(s)	Field	Description	Type	Reset
C	28:24	OKLEVEL_SEL	*Buck Voltage Monitor Target Voltage Select* Selects the reference level for the target Voltage monitor	r/w	*0x10*
D	20:16	OVLEVEL_SEL	*Buck Voltage Monitor Over Voltage Select* Selects the reference level for the Over Voltage monitor 0x**0** — OV Threshold - 1.44 V 0x**1** — OV Threshold - 1.65 V 0x**2** — OV Threshold - 1.85 V 0x**3** — OV Threshold - 2.06 V 0x**4** — OV Threshold - 2.26 V 0x**5** — OV Threshold - 2.47 V 0x**6** — OV Threshold - 2.68 V 0x**7** — OV Threshold - 2.87 V 0x**8** — OV Threshold - 3.08 V 0x**9** — OV Threshold - 3.29 V 0x**A** — OV Threshold - 3.50 V 0x**B** — OV Threshold - 3.69 V 0x**C** — OV Threshold - 3.90 V 0x**D** — OV Threshold - 4.11 V 0x**E** — OV Threshold - 4.31 V 0x**F** — OV Threshold - 4.52 V 0x**10** — OV Threshold - 4.72 V 0x**11** — OV Threshold - 4.93 V	r/w	*0x1C*

#	Bit(s)	Field	Description	Type	Reset
			0x**12** — OV Threshold - 5.12 V 0x**13** — OV Threshold - 5.22 V 0x**14** — OV Threshold - 5.32 V 0x**15** — OV Threshold - 5.42 V 0x**16** — OV Threshold - 5.53 V 0x**17** — OV Threshold - 5.65 V 0x**18** — OV Threshold - 5.77 V 0x**19** — OV Threshold - 5.90 V 0x**1A** — OV Threshold - 6.03 V 0x**1B** — OV Threshold - 6.16 V 0x**1C** — OV Threshold - 6.31 V 0x**1D** — OV Threshold - 6.46 V 0x**1E** — OV Threshold - 6.61 V 0x**1F** — OV Threshold - 6.78 V		
E	11:8	UVLEVEL	*Buck Voltage Monitor Under Voltage Select* Selects the reference level for the Under Voltage monitor 0x**0** — UV Threshold - 4.43 V 0x**1** — UV Threshold - 4.21 V 0x**2** — UV Threshold - 4.02 V 0x**3** — UV Threshold - 3.84 V 0x**4** — UV Threshold - 3.62 V 0x**5** — UV Threshold - 3.43 V 0x**6** — UV Threshold - 3.24 V	r/w	*6*

#	Bit(s)	Field	Description	Type	Reset
			0x**7** — UV Threshold - 3.05 V 0x**8** — UV Threshold - 2.85 V 0x**9** — UV Threshold - 2.64 V 0x**A** — UV Threshold - 2.46 V 0x**B** — UV Threshold - 2.25 V 0x**C** — UV Threshold - 2.07 V 0x**D** — UV Threshold - 1.86 V 0x**E** — UV Threshold - 1.67 V 0x**F** — UV Threshold - 1.47 V		
F	2	OK_MONITOR_ENA	*Buck OK Voltage Monitor Enable* Set to enable the vbat ok voltage monitor	r/w	0
G	1	OV_MONITOR_ENA	*Buck Over Voltage Monitor Enable* Set to enable the vbat over voltage monitor	r/w	0

7.2.9.8 **BUCKCTRL->VBUCKDBNC**

BUCKCTRL	**VBUCKDBNC**	0x5001051C
	VBUCK Debounce Register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A													B				C				D			E				F			

#	Bit(s)	Field	Description	Type	Reset
A	31:24	BUCKSTRBDBNCSEL	*Strobing select for debouncing*	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			Debouncer Strobe Selects for LOW/UV/OK/OV detection 0x**0** — SysClk As strobe 0x**1** — SysClk 2Div As strobe 0x**2** — SysClk 4Div As strobe 0x**3** — SysClk 8Div As strobe		
B	18:16	EXTRADBNC	*Extra debounce enables* Enable the extra debounce controls for preventing unintentional restart from dirty environment	r/w	*7*
C	14:12	OVTHRES1	*Over Voltage debouncing threshold for 0 to 1 Transition* Sets the threshold for debouncing the over voltage event going from 0 to 1 this can be in the range of 1 to 7, This means, it will require the ov event to stay high for the (OVTHRES1+1) number of LF/HF clocks to transit from a 0 to 1. The threshold for debouncing the over voltage event going from 1 to 0 is fixed at 0, that means the ov event requires 8 LF/HF clocks from transitioning from a 1 to 0	r/w	*7*
D	10:8	UVTHRES1	*Under Voltage debouncing threshold for 0to1 Transition* Sets the threshold for debouncing the under voltage event going from 0 to 1 this can be in the range of 1 to 7, This means, it will require the uv event to stay high for the (UVTHRES1+1) number of LF/HF clocks to transit from a 0 to 1. to stay high for the (UVTHRES1+1) number of LF/HF clocks to transit from a 0 to 1. The threshold for debouncing the under voltage event going from 1 to 0 is fixed at 0, that means the uv	r/w	*7*

#	Bit(s)	Field	Description	Type	Reset
			event requires 8 LF/HF clocks from transitioning from a 1 to 0		
E	7:4	LFENA	*Low frequency strobing enable for debouncing* Enables strobing the debouncer with the Low frequency clock(16K) instead of just clocking with High frequency Clock(2~16M)	r/w	0
F	3:0	BUCKDBNCENA	*voltage signal debounce enable* if set to '1, debounces the voltage signals else the signals bypass the debouncer for LOW/UV/OK/OV	r/w	*0xF*

7.2.9.9 **BUCKCTRL->VBUCKTIMEOUT**

BUCKCTRL	**VBUCKTIMEOUT**	0x50010520
	VBUCK TIMEOUT Register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				A				B				C				D				E			F				G				

#	Bit(s)	Field	Description	Type	Reset
A	27:24	PREENADWELL	*PRE_ENA Timeout Setting* Timeout wait time = BIT[3:0]*2*LF_Tcyc	r/w	*1*
B	23:20	DETLOWDWELL	*DET_LOW Timeout Setting* Timeout wait time = BIT[3:0]*32*LF_Tcyc	r/w	*4*
C	19:16	STARTLDWELL	*START_L Timeout Setting*	r/w	*8*

#	Bit(s)	Field	Description	Type	Reset
			Timeout wait time = BIT[3:0]*8*LF_Tcyc		
D	15:12	STARTDBNCDWELL	*START_L Debounce Timeout Setting* Timeout wait time = BIT[3:0]*8*LF_Tcyc	r/w	*4*
E	11:8	UVDBNCDWELL	*UV Debounce Timeout Setting* Timeout wait time = BIT[3:0]*2*LF_Tcyc	r/w	*8*
F	7:4	OVDBNCDWELL	*OV Debounce Timeout Setting* Timeout wait time = BIT[3:0]*2*LF_Tcyc	r/w	*4*
G	3:0	HICCUPDWELL	*Hiccup Timeout Setting* Timeout wait time = BIT[3:0]*32*LF_Tcyc	r/w	*5*

7.2.9.10 **BUCKCTRL->VBUCKDEBUG**

BUCKCTRL	**VBUCKDEBUG**	0x50010524
	VBUCK Debug Register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																									A						

#	Bit(s)	Field	Description	Type	Reset
A	6:0	BUCKIGNORE	*Ignore the control signals for debugging purpose* BIT[0] : Ignore EN_VOUT_PD. BIT[1] : Ignore VBUCK_LOW_DETECTED. BIT[3] : Ignore VBUCK_MONITOR_OK. BIT[4] : Ignore	r/w	*0x60*

#	Bit(s)	Field	Description	Type	Reset
			VBUCK_MONITOR_OV. BIT[5] : Ignore VBAT_UV. BIT[6] : Ignore VBAT_OV		

7.2.10 ADC

ADC Control

7.2.10.1 ADC->CONF

ADC	**CONF**	0x50010600
	configuration settings for the ADC set this up before starting a conversion.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									A											B					C			D	E	F	G

#	Bit(s)	Field	Description	Type	Reset
A	22:16	SAMPCYC	*Sample cycle* setting sampling time to SAMPCYC+2 adc clock cycles	r/w	*8*
B	11:8	SYNCENA	*Sync Enable* Need to set this bit if the conversion needs to be in sync with an sync input 0x**1** — Triggered by the posedge of the pwm signal 0x**2** — Triggered by the negedge of the pwm signal. 0x**4** — Triggered by the period pulse of pwm.	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			0x**8** — Triggered by Soft Sync Input.		
C	6	SWSYNCIN	*Soft Sync Input* Write 1 to trigger an adc conversion if SYNC_ENA[3] = 1; This register will be used mainly for debug purpose with hw sync disabled we can emulate the hw sync signal in this register	*wo*	0
D	3	BIASENA	*ADC Bias Enable* Set to enable ADC Bias when the chip is active, otherwise ADC bias is only enabled during conversion	r/w	0
E	2	ATTEN	*ADC input Attenuation setting* if set ADC will convert Vin/3	r/w	0
F	1	AUTOEN	***Bias Enable mode*** 0x**0** — Bias is Enabled Continuously 0x**1** — Bias is Enabled only after strb	r/w	0
G	0	MODE	*ADC mode select* **Selects the ADC operating mode** 0x**0** — Diffrential Mode A/D Conversion 0x**1** — Single Mode A/D Conversion	r/w	*1*

7.2.10.2 ADC->CNTRL

ADC	**CNTRL**	0x50010604
	ADC Data Conversion Control Register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		A				B			C					D				E						F	G	H				I	J

#	Bit(s)	Field	Description	Type	Reset
A	29:26	CH4SEL	*Channel4 Selection* Selects the analog outputs of GPIO1~5 connected to CH4 0x**0** — GPIO1 connected to CH4 0x**1** — GPIO2 connected to CH4 0x**2** — GPIO3 connected to CH4 0x**3** — GPIO4 connected to CH4 0x**4** — GPIO5 connected to CH4 0x**5** — GPIO6 connected to CH4 0x**6** — GPIO7 connected to CH4 0x**7** — GPIO8 connected to CH4 0x**8** — GPIO9 connected to CH4	r/w	0
B	25:24	CH3SEL	*Channel3 Selection* **Channel3 Selects** 0x**0** — Accurate VBAT(1/32) 0x**1** — VDD5V(1/8) 0x**2** — Internal Temperature sensor 0x**3** — Reserved	r/w	*2*
C	22:18	CH2SEL	*Channel2 Selection*	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			Selects the forward voltage of the external LEDs connected to CH2; The selected LED CH_NO = CH2_SEL. CH2_SEL value 24~31 are reserved		
D	17:16	CH1SEL	*Channel1 Selection* **Channel1 Selects** 0x**0** — Accurate VBAT(1/32) 0x**1** — VDD5V(1/8) 0x**2** — Internal Temperature sensor 0x**3** — Reserved	r/w	*1*
E	13:8	CHSEQ	*Channel Sequence* Selects the sequence of channels to be converted: CH0: Buffered bandgap voltage CH1: Accurate VBAT(1/32)/VDD5V(1/8)/Internal Temperature sensor CH2: VFW(The forward voltage of the external LEDs), equals (VDD5V-VLED)/5.5 CH3: Accurate VBAT(1/32)/VDD5V(1/8)/Internal Temperature sensor CH4: analog port of GPIO1~5 CH5: VDD1V5 0x**0** — Only CH0 0x**1** — Only CH1 0x**2** — Only CH2 0x**4** — Only CH3 0x**8** — Only CH4 0x**10** — Only CH5 0x**21** — CH1 followed by CH2 0x**29** — CH1 followed by CH3	r/w	*0x3F*

#	Bit(s)	Field	Description	Type	Reset
			0x**22** — CH2 followed by CH3 0x**2A** — CH2 followed by CH1 0x**24** — CH3 followed by CH1 0x**2C** — CH3 followed by CH2 0x**31** — CH1 followed by CH2 followed by CH3 0x**39** — CH1 followed by CH3 followed by CH2 0x**32** — CH2 followed by CH3 followed by CH1 0x**3A** — CH2 followed by CH1 followed by CH3 0x**34** — CH3 followed by CH1 followed by CH2 0x**3C** — CH3 followed by CH2 followed by CH1		
F	7	IRQCLR	*IRQ Clear*	*wo*	0
G	6	IRQENA	*IRQ Enable*	r/w	0
H	5:4	STUPDLY	*Startup Delay* Delay between adc getting enabled and the 1st strbi(command to start a conversion) 0x**0** — 1us Delay 0x**1** — 8us Delay 0x**2** — 12us Delay 0x**3** — 16us Delay	r/w	*2*
I	1	CONT	*Continuous Convesion Enable* if set enables the continuous conversion mode, else it's a single conversion. This is only checked at the end of current conversion	r/w	0

#	Bit(s)	Field	Description	Type	Reset
J	0	CONVERT	*ADC START/STATUS Register* Set to start a conversion, gets cleared at the end of single conversion. If CONT is set then this doesn't get cleared at the end of conversion. This can be read to check the current status of ADC conversion	r/w	0

7.2.10.3 **ADC->TSET**

ADC	**TSET**	0x50010608
	Settling Time settings Register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																				A				B				C			

#	Bit(s)	Field	Description	Type	Reset
A	11:8	TGUARD	*TGUARD setting* It's the guard time where there is no channel is selected, while switching from one channel in the sequence to the other to avoid any overlap. Guard Time = (TGUARD+1)x 250ns	r/w	*3*
B	7:4	TCHNL	*TCHNL setting* It's the time to wait after the guard time for the conversions of 2nd or 3rd channel in the sequence, to allow settling of the channel before start of the conversion Channel Time = (TCHNL+1)x 250ns	r/w	0
C	3:0	TCURR	*TCURR setting*	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			if SYNCENA is set, It's the time to wait after the posedge of the sync input before starting the 1st conversion of the sequence, or in case of a sequence without SYNCENA its the time between CONVERT goes high and the start of ADC conversion, basically it allows time for the first channel in the sequence to settle Current Time = (TCURR+1)x 250ns		

7.2.10.4 ADC->DATA1

ADC	**DATA1**	0x5001060C
	Data Out of CH1,	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																				A											

#	Bit(s)	Field	Description	Type	Reset
A	11:0	DATA1	The result of ADC conversion of CH1	*ro*	0

7.2.10.5 ADC->DATA2

ADC	**DATA2**	0x50010610
	Data Out of CH2,	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																				A											

#	Bit(s)	Field	Description	Type	Reset
A	11:0	DATA2	The result of ADC conversion of CH2	*ro*	0

7.2.10.6 ADC->DATA0345

ADC	**DATA0345**	0x50010614
	Data Out of CH0/CH3/CH4/CH5,	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																				A											

#	Bit(s)	Field	Description	Type	Reset
A	11:0	DATA0345	The result of ADC conversion of CH0/CH3/CH4/CH5 depending on CHSEQ setting	*ro*	0

7.2.10.7 ADC->STATUS

ADC	**STATUS**	0x50010618
	Status Register,	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																									A						B

#	Bit(s)	Field	Description	Type	Reset
A	6:1	FSM	*current state of the ADC Sequencer*	*ro*	0
B	0	CONVDONE	*The set Sequence of Conversions is Done* if set gets cleared with IRQCLR	*ro*	0

7.2.11 SPI

SPIM

7.2.11.1 SPI->SPIRXDATA

SPI	**SPIRXDATA**	0x50010700
	Serial Peripheral Rx Data Register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								A							

#	Bit(s)	Field	Description	Type	Reset
A	7:0	RXDATA	*Rx Data buffer* Received Data Register. A read from this register will return data from current read pointer on the receive FIFO. A read from this register will increment the read pointer on the receive FIFO	*ro*	0

7.2.11.2 SPI->SPITXDATA

SPI	**SPITXDATA**	0x50010704
	Serial Peripheral Tx Data Register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								A							

#	Bit(s)	Field	Description	Type	Reset
A	7:0	TXDATA	*Tx Data buffer* Transmit Data Register. Used for data that is to be transmitted.	*wo*	0

#	Bit(s)	Field	Description	Type	Reset
			A write to this register will place data into the transmit FIFO and increment the write pointer for the transmit FIFO		

7.2.11.3 **SPI->SPICTRL**

SPI	**SPICTRL**	0x50010708
	Serial Peripheral Control Register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							A								B							C		D		E	F	G		H	I

#	Bit(s)	Field	Description	Type	Reset
A	24	FIFOSOFTRESET	*FIFO Soft Reset* FIFO Soft Reset bit (active high). Writing a one here will clear the FIFO pointers. This bit automatically clears to zero	*wo*	0
B	16	LPBK	***Port wired-OR mode bit*** 0x**0** — master_in = miso 0x**1** — master_in = mosi	r/w	0
C	9:8	ICNT	***Interrupt counter bits*** 0x**0** — Interrupt is set after every completed transfer 0x**1** — Interrupt is set after two completed transfers	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			0x**2** — Interrupt is set after three completed transfers 0x**3** — Interrupt is set after four completed transfers		
D	7:6	SCKEXT	*Extended clock divider* SPR - Standard Clock Rate Divider Select SPRE (SCKEXT) - Extended Clock Rate Select Bits {SPRE, SPR} = Clock Divide Result {00, 00}: System Clock / 2 {00, 01}: System Clock / 4 {00, 10}: System Clock / 16 {00, 11}: System Clock / 32 {01, 00}: System Clock / 8 {01, 01}: System Clock / 64 {01, 10}: System Clock / 128 {01, 11}: System Clock / 256 {10, 00}: System Clock / 512 {10, 01}: System Clock / 1024 {10, 10}: System Clock / 2048 {10, 11}: System Clock / 4096 {11, xx}: Reserved	r/w	0

#	Bit(s)	Field	Description	Type	Reset
E	5	CPOL	***SPI clock polarity (Motorola SPI Frame Format)*** 0x**0** — The base value of the clock is zero 0x**1** — The base value of the clock is one	r/w	0
F	4	CPHA	***SPI clock phase*** 0x**0** — Data is captured on clock transition from base and data is propagated on the clock transition to base 0x**1** — Data is captured on clock transition to base and data is propagated on the clock transition from base	r/w	0
G	3:2	SPR	*Standard clock divider selection* Please refer to SPRE register for system clock	r/w	0
H	1	ENA_STS	*SPI enable status* Status of SPI enable	*ro*	0
I	0	ENA_REQ	*SPI enable request* Enables the SPI interface. When SPI enable is deasserted the Tx and Rx FIFO's are still operable. Careful- If data is present in the Tx FIFO prior to enabling SPI, then once SPI is enabled it will begin transmitting the data present in the Tx FIFO	r/w	0

7.2.11.4 SPI->SPISTATUS

SPI	**SPISTATUS**	0x5001070C
	Serial Peripheral Status Register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														A								B			C	D	E	F	G	H	I

#	Bit(s)	Field	Description	Type	Reset
A	17:16	TXFIFOCOUNT	*Transmit FIFO Count* Signal indicates active number of bytes in the FIFO. Increments on Push. Decrements on Pop	*ro*	0
B	9:8	RXFIFOCOUNT	*Reception FIFO Count* Signal indicates active number of bytes in the FIFO. Increments on Push. Decrements on Pop	*ro*	0
C	6	TXFIFOFULL	***Transmit FIFO full*** 0x**0** — FIFO is NOT full 0x**1** — FIFO is full	*ro*	0
D	5	TXFIFOEMPTY	***Transmit FIFO empty*** 0x**0** — FIFO is NOT empty 0x**1** — FIFO is empty	*ro*	*1*
E	4	RXFIFOFULL	***Reception FIFO full*** 0x**0** — FIFO is NOT full 0x**1** — FIFO is full	*ro*	0
F	3	RXFIFOEMPTY	***Reception FIFO empty***	*ro*	*1*

#	Bit(s)	Field	Description	Type	Reset
			0x**0** — FIFO is NOT empty 0x**1** — FIFO is empty		
G	2	TXFIFOOF	***Transmit FIFO overflow*** 0x**0** — overflow inactive 0x**1** — overflow active	*ro*	0
H	1	RXFIFOUF	***Reception FIFO underflow*** 0x**0** — underflow inactive 0x**1** — underflow active	*ro*	0
I	0	XFERCNT	*Completed Transfer Count* Signal is set after a programmable amount of completed transfer. See SPI_CTRL register field ICNT	*ro*	0

7.2.11.5 **SPI->SPIINTSTATUS**

SPI	**SPIINTSTATUS**	0x50010710
	SPI Interrupt Status Read this register to establish the reason for an SPI interrupt. This interrupt status is reporting the status after the interrupt enable. All interrupts are OR'ed together into a single SPI IRQ signal.	

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
																													A	B	C

#	Bit(s)	Field	Description	Type	Reset
A	2	INTTXFIFOOF	*Transmit FIFO Overflow Interrupt* Interrupt is set when Transmit FIFO is full and a write transaction occurs. (WRITE COLLISION)	*ro*	0
B	1	INTRXFIFOUF	*Reception FIFO Underflow Interrupt* Interrupt is set when Receive FIFO is empty and a read transaction occurs. (READ COLLISION)	*ro*	0
C	0	INTXFERCNT	*SPI Transfer Count Interrupt* Interrupt is set after a programmable amount of completed transfer. See SPI_CTRL register field ICNT	*ro*	0

7.2.11.6 **SPI->SPIINTENABLE**

SPI	**SPIINTENABLE**	0x50010714
	SPI Interrupt Enable SPI Interrupt Enable Register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																													A	B	C

#	Bit(s)	Field	Description	Type	Reset
A	2	INTTXFIFOOFENA	*Transmit FIFO Overflow Interrupt Enable*	r/w	0
B	1	INTRXFIFOUFENA	*Reception FIFO Underflow Interrupt Enable*	r/w	0
C	0	INTXFERCNTENA	*SPI Transfer Count Interrupt Enable*	r/w	0

7.2.11.7 SPI->SPIINTCLEAR

SPI	**SPIINTCLEAR**	0x50010718
	SPI Interrupt Clear SPI Interrupt Clear Register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																													A	B	C

#	Bit(s)	Field	Description	Type	Reset
A	2	INTTXFIFOOFCLR	*Transmit FIFO Overflow Interrupt* – cleared automatically after each write	*wo*	0
B	1	INTRXFIFOUFCLR	*Reception FIFO Underflow Interrupt* – cleared automatically after each write	*wo*	0
C	0	INTXFERCNTCLR	*SPI Transfer Count Interrupt* – cleared automatically after each write	*wo*	0

7.2.12 UART

7.2.12.1 UART->DATA

UART	**DATA**	0x50010800

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																				A			B								

#	Bit(s)	Field	Description	Type	Reset
A	11:9	DATARECEIVESTATUS	*Data Receive Status* Mirror for DATA_RECEIVE_STATUS: This register contains the receive status associated with the current byte read from the UART_DATA register. The Data Receive Status register is updated only after a read from the UART_DATA register. Example- Read UART_DATA byte first. Then read UART_DATA_RECEIVE_STATUS register second	*ro*	0
B	8:0	BYTE	*Data* Used for both received data(ONLY bit0~7 are available) and data that is to be transmitted(bit0~8 are available): Transmit: the MSB-bit[8] is only available for Transmitter, and it indicates this data used for sending BREAK or Not 1'b1 : this DATA is used for sending BREAK; 1'b0 : this DATA is used for sending Normal Data.	*dual*	0

#	Bit(s)	Field	Description	Type	Reset
			Receive: the MSB-bit[7] indicates the mode Match Address: 1'b0 : Normal mode, the following Rxd Datas will be received with the condition of 7bits-[6:0] matched; 1'b1 : Broadcast mode, the following Rxd Datas will be received without the condition of 7bits-[6:0] matched		

7.2.12.2 **UART->UARTDATARECEIVESTATUS**

UART	**UARTDATARECEIVESTATUS**	0x50010804
	Data Receive Status This register contains the receive status associated with the current byte read from the UART_DATA register. The Data Receive Status register is updated only after a read from the UART_DATA register. Example- Read UART_DATA byte first. Then read UART_DATA_RECEIVE_STATUS register second.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																													A	B	C

#	Bit(s)	Field	Description	Type	Reset
A	2	BREAKERROR	*Break Error*	*ro*	0

#	Bit(s)	Field	Description	Type	Reset
			This bit is set to 1 if a break condition was detected, indicating that the received data input was held LOW for longer than a full-word transmission time (defined as start, data, parity, and stop bits)		
B	1	PARITYERROR	*Parity Error* When this bit is set to 1, it indicates that the parity of the received data character does not match the parity selected as defined by bits 2 and 7 of the UARTLCR_H register	*ro*	0
C	0	FRAMEERROR	*Framing Error* When this bit is set to 1, it indicates that the received byte did not have a valid stop bit (a valid stop bit is 1)	*ro*	0

7.2.12.3 **UART->MSGCTRL**

UART	**MSGCTRL**	0x50010808
	Message control	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	A							B										C	D	E	F	G		H		I	J	K	L	M	N

#	Bit(s)	Field	Description	Type	Reset
A	30:24	ADDR_MATCH	*Match Address* Match Address, Only 7bit is available	r/w	0
B	23	LOOPENA	*Loopback enable*	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			Set to enable loopback		
C	13	STICKENA	*Sticky parity enable* Set to enable sticky parity	r/w	0
D	12	PARODD	*Odd parity* Set for odd parity (see also PARENA)	r/w	0
E	11	PARENA	*Parity enable* Set to enable parity (see PARODD for odd/even)	r/w	0
F	10	STOP	***Stop bit control*** 0x**0** — One stop bit 0x**1** — If a 5-bit transmission it selects 1.5 stop bits, otherwise 2 stop bits (6, 7 & 8 bits)	r/w	0
G	9:8	SIZE	***Transmission word size*** 0x**0** — 5-bit(only available for Receive mode) 0x**1** — 6-bit(only available for Receive mode) 0x**2** — 7-bit(available for Receive and Transmit mode) 0x**3** — 8-bit(available for Receive and Transmit mode)	r/w	*3*

#	Bit(s)	Field	Description	Type	Reset
H	7:6	**MAX_BYTES_RXD** r/w *3*	Set Max Byte number of DATA in a rxd frame 0x**0** — 15-Bytes 0x**1** — 31-Bytes 0x**2** — 63-Bytes 0x**3** — 127-Bytes		
I	5	ENA_ADDR_MATCH	Set to enable Address Match	r/w	0
J	4	TXXFERCNTCLR	*TX Transfer Counter Clear* Clear TX Multi Transfer Counter to zero – cleared automatically after each write	*wo*	0
K	3	RXXFERCNTCLR	*RX Transfer Counter Clear* Clear RX Multi Transfer Counter to zero – cleared automatically after each write	*wo*	0
L	2	UFIFOSOFTRESET	*FIFO SOFT RESET* Resets FIFO pointers to zero and initializes FIFO contents to zero – cleared automatically after each write	*wo*	0
M	1	ENABLE_STS	*Enable status* Status of UART enable	*ro*	0
N	0	ENABLE	Set to enable the UART	r/w	0

7.2.12.4 UART->UARTINT

UART	**UARTINT**	0x5001080C
	UART Interrupts Contains the enable, status and clear for the UART interrupt sources.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	a	b	c	d	e	f

#	Bit(s)	Field	Description	Type	Reset
A	31	TXMULTDONE	*Multiple Transmit Transactions Done Interrupt* Set by the UART when the programmed number of transmit transactions have completed. See UART_FIFO_LEVEL_CTL register	*ro*	0
B	30	RXMULTDONE	*Multiple Receive Transactions Done Interrupt* Set by the UART when the programmed number of receive transactions have completed. See UART_FIFO_LEVEL_CTL register	*ro*	0
C	29	TXDONE	*Transmission done Interrupt* Set by the UART when the transmission is done	*ro*	0
D	28	BREAKKERR	*Break Error Interrupt* Set by the UART when a break error occurs	*ro*	0
E	27	PRTYERR	*Parity Error Interrupt* Set by the UART when a parity error occurs	*ro*	0

#	Bit(s)	Field	Description	Type	Reset
F	26	FRMERR	*Framing error Interrupt* Set by the UART when a framing error occurs	*ro*	0
G	25	OVRUNERR	*Overrun error Interrupt* Set by the UART when an overrun error occurs	*ro*	0
H	24	RXDONE	*Rx Data ready Interrupt* Set by the UART when Rx data is ready	*ro*	0
I	23	TXMULTDONE	*Multiple Transmit Transactions Done* Set by the UART when the programmed number of transmit transactions have completed. See UART_FIFO_LEVEL_CTL register	*ro*	0
J	22	RXMULTDONE	*Multiple Receive Transactions Done* Set by the UART when the programmed number of receive transactions have completed. See UART_FIFO_LEVEL_CTL register	*ro*	0
K	21	TXDONE	*Transmission is done* Set by the UART when the transmit is done	*ro*	0
L	20	BREAKKERR	*Break IRQ* Set by the UART when a break error occurs	*ro*	0
M	19	PRTYERR	*Parity Error*	*ro*	0

#	Bit(s)	Field	Description	Type	Reset
			Set by the UART when a parity error occurs		
N	18	FRMERR	*Framing error* Set by the UART when a framing error occurs	*ro*	0
O	17	OVRUNERR	*Overrun error* Set by the UART when an overrun error occurs	*ro*	0
P	16	RXDONE	*Rx Data ready* Set by the UART when Rx data is ready	*ro*	0
Q	15	TXMULTDONE	*Multiple Transmit Transactions Done Interrupt Clear* – cleared automatically after each write	*wo*	0
R	14	RXMULTDONE	*Multiple Receive Transactions Done Interrupt Clear* – cleared automatically after each write	*wo*	0
S	13	TXDONE	*Transmission done Interrupt Clear* – cleared automatically after each write	*wo*	0
T	12	BREAKKERR	Break Error Interrupt Clear – cleared automatically after each write	*wo*	0
U	11	PRTYERR	*Parity Error Interrupt Clear* – cleared automatically after each write	*wo*	0

#	Bit(s)	Field	Description	Type	Reset
V	10	FRMERR	*Framing error Interrupt Clear* – cleared automatically after each write	*wo*	0
W	9	OVRUNERR	*Overrun error Interrupt Clear* – cleared automatically after each write	*wo*	0
X	8	RXDONE	*Rx Data ready Interrupt Clear* – cleared automatically after each write	*wo*	0
Y	7	TXMULTDONE	*Multiple Transmit Transactions Done Interrupt Enable*	r/w	0
Z	6	RXMULTDONE	*Multiple Receive Transactions Done Interrupt Enable*	r/w	0
a	5	TXDONE	*Transmission done Interrupt Enable*	r/w	0
b	4	BREAKKERR	*Break Error Interrupt Enable*	r/w	0
c	3	PRTYERR	*Parity Error Interrupt Enable*	r/w	0
d	2	FRMERR	*Framing error Interrupt Enable*	r/w	0
e	1	OVRUNERR	*Overrun error Interrupt Enable*	r/w	0
f	0	RXDONE	*Rx Data ready Interrupt Enable*	r/w	0

7.2.12.5 **UART->UARTINT2**

UART	**UARTINT2**	0x50010810
	Extended UART Interrupts Contains the enable, status and clear for the extended UART interrupt sources.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							A								B								C								D

#	Bit(s)	Field	Description	Type	Reset
A	24	RXTOUT	*Rx Time-out Interrupt* Set by the UART when RXTOUT flag is set and RXOUT interrupt is enabled	*ro*	0
B	16	RXTOUT	*Rx Time-out* Set by the UART when timeout of 4 data frame times without reception, and data received(one or three bytes received,programmable)	*ro*	0
C	8	RXTOUT	*Rx Time-out Interrupt Clear* – cleared automatically after each write	*wo*	0
D	0	RXTOUT	*Rx Time-out Interrupt Enable*	r/w	0

7.2.12.6 **UART->UARTBAUD**

UART	**UARTBAUD**	0x50010814
	UART Baud rate The controls in this register define the relationship between the system clock and the UART baud rate. The baud rate is given by the following equation. baud_rate = Fclk/(OSR*(BAUDDIV+1+FDIV/8)) where Fclk is the frequency of the system clock, and OSR , FDIV and BAUDDIV are the fields of this register.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		A	B		C						D					E															

#	Bit(s)	Field	Description	Type	Reset
A	29	URETARD	*Retard Register* Retards the sample window by 1 cycle. For Debug	r/w	0
B	28	UADVANCE	*Advance Register* Advances the sample window by 1 cycle. For Debug	r/w	0
C	26:24	FDIV	*Fractional divider*	r/w	0
D	20:16	OSR	*Over-sampling ratio* Valid OSR Range: 6 to 16	r/w	*0x10*
E	15:0	BAUDDIV	*Baud rate divider*	r/w	0

7.2.12.7 **UART->UARTFIFOSTATUS**

UART	**UARTFIFOSTATUS**	0x50010818
	UART FIFO STATUS The FIFO status register is read-only and gives the current state of the Receive and Transmit FIFO's.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				A									B	C	D			E											F	G	H

#	Bit(s)	Field	Description	Type	Reset
A	27:24	TXCOUNT	*Transmit FIFO current count*	*ro*	0
B	18	TXFULL	*Transmit FIFO full*	*ro*	0
C	17	TXEMPTY	*Transmit FIFO empty*	*ro*	0

#	Bit(s)	Field	Description	Type	Reset
D	16	TXOF	*Transmit FIFO overflow*	*ro*	0
E	13:8	RXCOUNT	*Reception FIFO current count*	*ro*	0
F	2	RXFULL	*Reception FIFO empty*	*ro*	0
G	1	RXEMPTY	*Reception FIFO empty*	*ro*	0
H	0	RXUF	*Reception FIFO underflow*	*ro*	0

7.2.12.8 **UART->UARTFIFOLEVELCTL**

UART	**UARTFIFOLEVELCTL**	0x5001081C
	UART FIFO Level Control The FIFO Level Control register defines the trigger points for the transmit and receive interrupts.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																				A						B					

#	Bit(s)	Field	Description	Type	Reset
A	11:8	TXMULTIPLEXFERDONECNT	*Transmit Data Count Interrupt* Number of bytes transmitted from the transmit FIFO before a INT_TX_MULTIPLE_XFER_DONE interrupt is asserted. Example: A value of 4 would be used to compare to the number of completed transfers (pops from the Tx FIFO).	r/w	*1*

#	Bit(s)	Field	Description	Type	Reset
			Valid Range: 1 to 8, NOTE: Invalid configuration value ‘0’ will be forced to ‘1’		
B	5:0	RXMULTIPLEXFERDONECNT	*Receive Data Count Interrupt* Minimum number of bytes in the receive FIFO before a INT_RX_MULTIPLE_XFER_DONE interrupt is asserted. Example: A value of 4 would be used to compare to the number of receive transactions (pushes to the Rx FIFO). Valid Range: 1 to 32, NOTE: Invalid configuration value ‘0’ will be forced to ‘1’	r/w	*1*

7.2.13 **PWM_AUX**

Pulse width modulation waveform generator

7.2.13.1 **PWM_AUX->BASE0**

PWM_AUX	**BASE0**	0x50010900
	Base 0 functions	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																					B										

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PERIOD0	*Period* Specifies the period of the output waveform in terms of a number of prescaler output cycles	r/w	0
B	10:8	PRESCALESEL0	*Prescaler select* Defines the ratio between the system clock and the clock used for the waveform generator 0x**0** — Divide by 1 0x**1** — Divide by 2 0x**2** — Divide by 4 0x**3** — Divide by 8 0x**4** — Divide by 16 0x**5** — Divide by 64 0x**6** — Divide by 256 0x**7** — Divide by 1024	r/w	0

7.2.13.2 PWM_AUX->PWMCNT0

PWM_AUX	**PWMCNT0**	0x50010904
	PWM Count Value	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																A															

#	Bit(s)	Field	Description	Type	Reset
A	15:0	PWMCNT0	PWM counter value to give a sense about the current period	*ro*	0

7.2.13.3 **PWM_AUX->BASE1**

PWM_AUX	**BASE1**	0x50010908
	Base 1 functions	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																					B										

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PERIOD1	*Period* Specifies the period of the output waveform in terms of a number of prescaler output cycles	r/w	0
B	10:8	PRESCALESEL1	*Prescaler select* Defines the ratio between the system clock and the clock used for the waveform generator 0x**0** — Divide by 1 0x**1** — Divide by 2 0x**2** — Divide by 4 0x**3** — Divide by 8 0x**4** — Divide by 16 0x**5** — Divide by 64 0x**6** — Divide by 256 0x**7** — Divide by 1024	r/w	0

7.2.13.4 PWM_AUX->PWMCNT1

PWM_AUX	**PWMCNT1**	0x5001090C
	PWM Count Value	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																A															

#	Bit(s)	Field	Description	Type	Reset
A	15:0	PWMCNT1	PWM counter value to give a sense about the current period	*ro*	0

7.2.13.5 PWM_AUX->BASESEL

PWM_AUX	**BASESEL**	0x50010910
	Base Timer Select for individual Channels	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																											A				

#	Bit(s)	Field	Description	Type	Reset
A	4:0	BASE_SEL	Base Timer Selects:1=Select BASE1, 0=Select BASE0	r/w	0

7.2.13.6 PWM_AUX->ENAREQ

PWM_AUX	**ENAREQ**	0x50010914
	Enable request	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					A	B	C																				D				

#	Bit(s)	Field	Description	Type	Reset
A	26	FORCEINACTIVE	Set to force PWM signals return to initial value immediately	r/w	0
B	25	CLRREQALL	Write 1 to clear all ENA_REQ bits; Write 0 has no effects	*wo*	0
C	24	ENAREQALL	Write 1 to enable all ENA_REQ bits; Write 0 has no effects	*wo*	0
D	4:0	ENAREQ	Set to enable the waveform generator	r/w	0

7.2.13.7 **PWM_AUX->ENASTS**

PWM_AUX	**ENASTS**	0x50010918
	Enable status	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																											A				

#	Bit(s)	Field	Description	Type	Reset
A	4:0	ENASTS	Status of enable in the waveform generator	*ro*	0

7.2.13.8 **PWM_AUX->INIT**

PWM_AUX	**INIT**	0x5001091C
	Intial State of Outputs	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																											A				

#	Bit(s)	Field	Description	Type	Reset
A	4:0	INIT	Set to initialise the output waveform	r/w	0

7.2.13.9 PWM_AUX->INV

PWM_AUX	**INV**	0x50010920
	Invert	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																											A				

#	Bit(s)	Field	Description	Type	Reset
A	4:0	INVERT	Set to invert the output waveform	r/w	0

7.2.13.10 PWM_AUX->UPDATE

PWM_AUX	**UPDATE**	0x50010924

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																														A	

#	Bit(s)	Field	Description	Type	Reset
A	1:0	UPDATE	Set to trigger consumption of new PULSE parameters (invert,prescale_sel,period,pulse start & stop). The flag is automatically cleared by the hardware when the settings are consumed, so reading a high value indicates that an update is still pending	*dual*	0

7.2.13.11 PWM_AUX->PULSE0

PWM_AUX	**PULSE0**	0x50010928
	PWM0 pulse setup	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																B															

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PRISE0	*Pulse Rise* Specifies the pulse rise of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse fall timing	r/w	0
B	15:0	PFALL0	*Pulse Fall* Specifies the pulse fall of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse rise timing	r/w	0

7.2.13.12 **PWM_AUX->PULSE1**

PWM_AUX	**PULSE1**	0x5001092C
	PWM1 pulse setup	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																B															

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PRISE1	*Pulse Rise* Specifies the pulse rise of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse fall timing	r/w	0
B	15:0	PFALL1	*Pulse Fall* Specifies the pulse fall of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse rise timing	r/w	0

7.2.13.13 PWM_AUX->PULSE2

PWM_AUX	**PULSE2**	0x50010930
	PWM2 pulse setup	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																B															

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PRISE2	*Pulse Rise* Specifies the pulse rise of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse fall timing	r/w	0
B	15:0	PFALL2	*Pulse Fall* Specifies the pulse fall of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse rise timing	r/w	0

7.2.13.14 PWM_AUX->PULSE3

PWM_AUX	**PULSE3**	0x50010934
	PWM3 pulse setup	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																B															

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PRISE3	*Pulse Rise*	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			Specifies the pulse rise of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse fall timing		
B	15:0	PFALL3	*Pulse Fall* Specifies the pulse fall of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse rise timing	r/w	0

7.2.13.15 **PWM_AUX->PULSE4**

PWM_AUX	**PULSE4**	0x50010938
	PWM4 pulse setup	

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
A																B															

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PRISE4	*Pulse Rise* Specifies the pulse rise of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse fall timing	r/w	0
B	15:0	PFALL4	*Pulse Fall* Specifies the pulse fall of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse rise timing	r/w	0

7.2.13.16 PWM_AUX->INTPOSEDGENA

PWM_AUX	**INTPOSEDGENA**	0x5001093C
	PWM posedge interrupt enable Contains the enable for the PWM posedge interrupt sources.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																											A				

#	Bit(s)	Field	Description	Type	Reset
A	4:0	INTPOSEDGENA	*Interrupt enable* bit[4:0]: posedge interrupt enable	r/w	0

7.2.13.17 PWM_AUX->INTNEGEDGENA

PWM_AUX	**INTNEGEDGENA**	0x50010940
	PWM negedge interrupt enable Contains the enable for the PWM negedge interrupt sources.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																											A				

#	Bit(s)	Field	Description	Type	Reset
A	4:0	INTNEGEDGENA	*Interrupt enable* bit[4:0]: negedge interrupt enable	r/w	0

7.2.13.18 PWM_AUX->INTPOSEDGCLR

PWM_AUX	**INTPOSEDGCLR**	0x50010944

	PWM posedge interrupt control Contains the clear for the PWM posedge interrupt sources.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																											A				

#	Bit(s)	Field	Description	Type	Reset
A	4:0	INTPOSEDGCLR	*Interrupt clear* bit[4:0] : posedge interrupt clear	*wo*	0

7.2.13.19 **PWM_AUX->INTNEGEDGCLR**

PWM_AUX	**INTNEGEDGCLR**	0x50010948
	PWM negedge interrupt control Contains the clear for the PWM negedge interrupt sources.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																											A				

#	Bit(s)	Field	Description	Type	Reset
A	4:0	INTNEGEDGCLR	*Interrupt clear* bit[4:0] : negedge interrupt clear	*wo*	0

7.2.13.20 **PWM_AUX->INTPOSEDGSTS**

PWM_AUX	**INTPOSEDGSTS**	0x5001094C
	PWM posedge interrupt status	

	Contains the status for the PWM posedge interrupt sources.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																											A				

#	Bit(s)	Field	Description	Type	Reset
A	4:0	INTPOSEDGSTS	*Interrupt status* bit[4:0] : posedge interrupt status	*ro*	n/a

7.2.13.21 **PWM_AUX->INTNEGEDGSTS**

PWM_AUX	**INTNEGEDGSTS**	0x50010950
	PWM negedge interrupt status Contains the status for the PWM negedge interrupt sources.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																											A				

#	Bit(s)	Field	Description	Type	Reset
A	4:0	INTNEGEDGSTS	*Interrupt status* bit[4:0] : negedge interrupt status	*ro*	n/a

7.2.13.22 **PWM_AUX->INTPOSEDGIRQ**

PWM_AUX	**INTPOSEDGIRQ**	0x50010954
	PWM posedge interrupt active Contains the active for the PWM posedge interrupt sources.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																											A				

#	Bit(s)	Field	Description	Type	Reset
A	4:0	INTPOSEDGIRQ	*Interrupt active* bit[4:0] : posedge interrupt active	*ro*	n/a

7.2.13.23 **PWM_AUX->INTNEGEDGIRQ**

PWM_AUX	**INTNEGEDGIRQ**	0x50010958
	PWM negedge interrupt active Contains the active for the PWM negedge interrupt sources.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																											A				

#	Bit(s)	Field	Description	Type	Reset
A	4:0	INTNEGEDGIRQ	*Interrupt active* bit[4:0] : negedge interrupt active	*ro*	n/a

7.2.13.24 **PWM_AUX->INTPERIOD**

PWM_AUX	**INTPERIOD**	0x5001095C
	PWM Period interrupt control Contains the enable, clear, status and active for the PWM period interrupt sources.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						A								B								C								D	

#	Bit(s)	Field	Description	Type	Reset
A	25:24	PERIOD	*Period Interrupt active*	*ro*	n/a
B	17:16	PERIOD	*Period Interrupt status*	*ro*	n/a
C	9:8	PERIOD	*Period Interrupt clear*	*wo*	0
D	1:0	PERIOD	*Period Interrupt enable*	r/w	0

7.2.13.25 PWM_AUX->INTUPDATED

PWM_AUX	**INTUPDATED**	0x50010960
	PWM Updated interrupt control Contains the enable, clear, status and active for the PWM updated interrupt sources.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						A								B								C								D	

#	Bit(s)	Field	Description	Type	Reset
A	25:24	UPD	*Updated Interrupt active*	*ro*	n/a
B	17:16	UPD	*Updated Interrupt status*	*ro*	n/a
C	9:8	UPD	*Updated Interrupt clear*	*wo*	0
D	1:0	UPD	*Updated Interrupt enable*	r/w	0

7.2.14 **PWM**

Pulse width modulation waveform generator

7.2.14.1 **PWM->BASE**

PWM	**BASE**	0x50011000
	Base functions	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																					B										

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PERIOD	Specifies the period of the output waveform in terms of a number of prescaler output cycles	r/w	0
B	10:8	PRESCALESEL	*Prescaler select* Defines the ratio between the system clock and the clock used for the waveform generator 0x**0** — Divide by 1 0x**1** — Divide by 2 0x**2** — Divide by 4 0x**3** — Divide by 8 0x**4** — Divide by 16 0x**5** — Divide by 64 0x**6** — Divide by 256 0x**7** — Divide by 1024	r/w	0

7.2.14.2 **PWM->PWMCNT**

PWM	**PWMCNT**	0x50011004
	PWM Count Value	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																A															

#	Bit(s)	Field	Description	Type	Reset
A	15:0	PWMCNT	PWM counter value to give a sense about the current period	*dual*	0

7.2.14.3 **PWM->ENAREQ**

PWM	**ENAREQ**	0x50011008
	Enable request	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					A	B	C	D																							

#	Bit(s)	Field	Description	Type	Reset
A	26	FORCEINACTIVE	Set to force PWM signals return to initial value immediately	r/w	0
B	25	CLRREQALL	Write 1 to clear all ENA_REQ bits; Write 0 has no effects	*wo*	0
C	24	ENAREQALL	Write 1 to enable all ENA_REQ bits; Write 0 has no effects	*wo*	0
D	23:0	ENAREQ	Set to enable the waveform generator	r/w	0

7.2.14.4 **PWM->ENASTS**

PWM	**ENASTS**	0x5001100C
	Enable status	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								A																							

#	Bit(s)	Field	Description	Type	Reset
A	23:0	ENASTS	Status of enable in the waveform generator	*ro*	0

7.2.14.5 **PWM->INIT**

PWM	**INIT**	0x50011010
	Intial State of Outputs	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								A																							

#	Bit(s)	Field	Description	Type	Reset
A	23:0	INIT	Set to initialise the output waveform	r/w	0

7.2.14.6 **PWM->INV**

PWM	**INV**	0x50011014
	Invert	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								A																							

#	Bit(s)	Field	Description	Type	Reset
A	23:0	INVERT	Set to invert the output waveform	r/w	0

7.2.14.7 **PWM->UPDATE**

PWM	**UPDATE**	0x50011018

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															A

#	Bit(s)	Field	Description	Type	Reset
A	0	UPDATE	Set to trigger consumption of new PULSE parameters (invert,prescale_sel,period,pulse start & stop). The flag is automatically cleared by the hardware when the settings are consumed, so reading a high value indicates that an update is still pending	*dual*	0

7.2.14.8 **PWM->PULSE0**

PWM	**PULSE0**	0x5001101C
	PWM0 pulse setup	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																B															

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PRISE0	*Pulse Rise* Specifies the pulse rise of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse fall timing	r/w	0
B	15:0	PFALL0	*Pulse Fall* Specifies the pulse fall of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse rise timing	r/w	0

7.2.14.9 **PWM->PULSE1**

PWM	**PULSE1**	0x50011020
	PWM1 pulse setup	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																B															

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PRISE1	*Pulse Rise* Specifies the pulse rise of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse fall timing	r/w	0
B	15:0	PFALL1	*Pulse Fall* Specifies the pulse fall of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse rise timing	r/w	0

7.2.14.10 **PWM->PULSE2**

PWM	**PULSE2**	0x50011024
	PWM2 pulse setup	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																B															

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PRISE2	*Pulse Rise* Specifies the pulse rise of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse fall timing	r/w	0
B	15:0	PFALL2	*Pulse Fall*	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			Specifies the pulse fall of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse rise timing		

7.2.14.11 **PWM->PULSE3**

PWM	**PULSE3**	0x50011028
	PWM3 pulse setup	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																B															

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PRISE3	*Pulse Rise* Specifies the pulse rise of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse fall timing	r/w	0
B	15:0	PFALL3	*Pulse Fall* Specifies the pulse fall of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse rise timing	r/w	0

7.2.14.12 **PWM->PULSE4**

PWM	**PULSE4**	0x5001102C
	PWM4 pulse setup	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																B															

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PRISE4	*Pulse Rise* Specifies the pulse rise of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse fall timing	r/w	0
B	15:0	PFALL4	*Pulse Fall* Specifies the pulse fall of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse rise timing	r/w	0

7.2.14.13 **PWM->PULSE5**

PWM	**PULSE5**	0x50011030
	PWM5 pulse setup	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																B															

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PRISE5	*Pulse Rise* Specifies the pulse rise of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse fall timing	r/w	0
B	15:0	PFALL5	*Pulse Fall*	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			Specifies the pulse fall of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse rise timing		

7.2.14.14 **PWM->PULSE6**

PWM	**PULSE6**	0x50011034
	PWM6 pulse setup	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																B															

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PRISE6	*Pulse Rise* Specifies the pulse rise of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse fall timing	r/w	0
B	15:0	PFALL6	*Pulse Fall* Specifies the pulse fall of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse rise timing	r/w	0

7.2.14.15 **PWM->PULSE7**

PWM	**PULSE7**	0x50011038
	PWM7 pulse setup	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																B															

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PRISE7	*Pulse Rise* Specifies the pulse rise of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse fall timing	r/w	0
B	15:0	PFALL7	*Pulse Fall* Specifies the pulse fall of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse rise timing	r/w	0

7.2.14.16 **PWM->PULSE8**

PWM	**PULSE8**	0x5001103C
	PWM8 pulse setup	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																B															

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PRISE8	*Pulse Rise* Specifies the pulse rise of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse fall timing	r/w	0
B	15:0	PFALL8	*Pulse Fall*	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			Specifies the pulse fall of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse rise timing		

7.2.14.17 **PWM->PULSE9**

PWM	**PULSE9**	0x50011040
	PWM9 pulse setup	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																B															

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PRISE9	*Pulse Rise* Specifies the pulse rise of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse fall timing	r/w	0
B	15:0	PFALL9	*Pulse Fall* Specifies the pulse fall of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse rise timing	r/w	0

7.2.14.18 **PWM->PULSE10**

PWM	**PULSE10**	0x50011044
	PWM10 pulse setup	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																B															

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PRISE10	*Pulse Rise* Specifies the pulse rise of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse fall timing	r/w	0
B	15:0	PFALL10	*Pulse Fall* Specifies the pulse fall of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse rise timing	r/w	0

7.2.14.19 **PWM->PULSE11**

PWM	**PULSE11**	0x50011048
	PWM11 pulse setup	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																B															

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PRISE11	*Pulse Rise* Specifies the pulse rise of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse fall timing	r/w	0
B	15:0	PFALL11	*Pulse Fall*	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			Specifies the pulse fall of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse rise timing		

7.2.14.20 **PWM->PULSE12**

PWM	**PULSE12**	0x5001104C
	PWM12 pulse setup	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																B															

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PRISE12	*Pulse Rise* Specifies the pulse rise of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse fall timing	r/w	0
B	15:0	PFALL12	*Pulse Fall* Specifies the pulse fall of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse rise timing	r/w	0

7.2.14.21 **PWM->PULSE13**

PWM	**PULSE13**	0x50011050
	PWM13 pulse setup	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																B															

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PRISE13	*Pulse Rise* Specifies the pulse rise of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse fall timing	r/w	0
B	15:0	PFALL13	*Pulse Fall* Specifies the pulse fall of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse rise timing	r/w	0

7.2.14.22 **PWM->PULSE14**

PWM	**PULSE14**	0x50011054
	PWM14 pulse setup	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																B															

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PRISE14	*Pulse Rise* Specifies the pulse rise of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse fall timing	r/w	0
B	15:0	PFALL14	*Pulse Fall*	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			Specifies the pulse fall of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse rise timing		

7.2.14.23 **PWM->PULSE15**

PWM	**PULSE15**	0x50011058
	PWM15 pulse setup	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																B															

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PRISE15	*Pulse Rise* Specifies the pulse rise of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse fall timing	r/w	0
B	15:0	PFALL15	*Pulse Fall* Specifies the pulse fall of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse rise timing	r/w	0

7.2.14.24 **PWM->PULSE16**

PWM	**PULSE16**	0x5001105C
	PWM16 pulse setup	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																B															

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PRISE16	*Pulse Rise* Specifies the pulse rise of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse fall timing	r/w	0
B	15:0	PFALL16	*Pulse Fall* Specifies the pulse fall of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse rise timing	r/w	0

7.2.14.25 **PWM->PULSE17**

PWM	**PULSE17**	0x50011060
	PWM17 pulse setup	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																B															

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PRISE17	*Pulse Rise* Specifies the pulse rise of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse fall timing	r/w	0
B	15:0	PFALL17	*Pulse Fall*	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			Specifies the pulse fall of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse rise timing		

7.2.14.26 **PWM->PULSE18**

PWM	**PULSE18**	0x50011064
	PWM18 pulse setup	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																B															

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PRISE18	*Pulse Rise* Specifies the pulse rise of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse fall timing	r/w	0
B	15:0	PFALL18	*Pulse Fall* Specifies the pulse fall of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse rise timing	r/w	0

7.2.14.27 **PWM->PULSE19**

PWM	**PULSE19**	0x50011068
	PWM19 pulse setup	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																B															

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PRISE19	*Pulse Rise* Specifies the pulse rise of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse fall timing	r/w	0
B	15:0	PFALL19	*Pulse Fall* Specifies the pulse fall of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse rise timing	r/w	0

7.2.14.28 **PWM->PULSE20**

PWM	**PULSE20**	0x5001106C
	PWM20 pulse setup	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																B															

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PRISE20	*Pulse Rise* Specifies the pulse rise of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse fall timing	r/w	0
B	15:0	PFALL20	*Pulse Fall*	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			Specifies the pulse fall of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse rise timing		

7.2.14.29 PWM->PULSE21

PWM	**PULSE21**	0x50011070
	PWM21 pulse setup	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																B															

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PRISE21	*Pulse Rise* Specifies the pulse rise of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse fall timing	r/w	0
B	15:0	PFALL21	*Pulse Fall* Specifies the pulse fall of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse rise timing	r/w	0

7.2.14.30 PWM->PULSE22

PWM	**PULSE22**	0x50011074
	PWM22 pulse setup	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																B															

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PRISE22	*Pulse Rise* Specifies the pulse rise of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse fall timing	r/w	0
B	15:0	PFALL22	*Pulse Fall* Specifies the pulse fall of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse rise timing	r/w	0

7.2.14.31 **PWM->PULSE23**

PWM	**PULSE23**	0x50011078
	PWM23 pulse setup	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																B															

#	Bit(s)	Field	Description	Type	Reset
A	31:16	PRISE23	*Pulse Rise* Specifies the pulse rise of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse fall timing	r/w	0
B	15:0	PFALL23	*Pulse Fall*	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			Specifies the pulse fall of the output waveform in terms of a number of prescaler output cycles. However, if INVERT is set it determines the pulse rise timing		

7.2.14.32 **PWM->INTPOSEDGENA**

PWM	**INTPOSEDGENA**	0x5001107C
	PWM posedge interrupt enable Contains the enable for the PWM posedge interrupt sources.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								A																							

#	Bit(s)	Field	Description	Type	Reset
A	23:0	INTPOSEDGENA	*Interrupt enable* bit[23:0]: posedge interrupt enable	r/w	0

7.2.14.33 **PWM->INTNEGEDGENA**

PWM	**INTNEGEDGENA**	0x50011080
	PWM negedge interrupt enable Contains the enable for the PWM negedge interrupt sources.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								A																							

#	Bit(s)	Field	Description	Type	Reset
A	23:0	INTNEGEDGENA	*Interrupt enable* bit[23:0]: negedge interrupt enable	r/w	0

7.2.14.34 **PWM->INTPOSEDGCLR**

PWM	**INTPOSEDGCLR**	0x50011084
	PWM posedge interrupt control Contains the clear for the PWM posedge interrupt sources.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								A																							

#	Bit(s)	Field	Description	Type	Reset
A	23:0	INTPOSEDGCLR	*Interrupt clear* bit[23:0] : posedge interrupt clear	*wo*	0

7.2.14.35 **PWM->INTNEGEDGCLR**

PWM	**INTNEGEDGCLR**	0x50011088
	PWM negedge interrupt control Contains the clear for the PWM negedge interrupt sources.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								A																							

#	Bit(s)	Field	Description	Type	Reset
A	23:0	INTNEGEDGCLR	*Interrupt clear*	*wo*	0

#	Bit(s)	Field	Description	Type	Reset
			bit[23:0] : negedge interrupt clear		

7.2.14.36 **PWM->INTPOSEDGSTS**

PWM	**INTPOSEDGSTS**	0x5001108C
	PWM posedge interrupt status Contains the status for the PWM posedge interrupt sources.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								A																							

#	Bit(s)	Field	Description	Type	Reset
A	23:0	INTPOSEDGSTS	*Interrupt status* bit[23:0] : posedge interrupt status	*ro*	n/a

7.2.14.37 **PWM->INTNEGEDGSTS**

PWM	**INTNEGEDGSTS**	0x50011090
	PWM negedge interrupt status Contains the status for the PWM negedge interrupt sources.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								A																							

#	Bit(s)	Field	Description	Type	Reset
A	23:0	INTNEGEDGSTS	*Interrupt status*	*ro*	n/a

#	Bit(s)	Field	Description	Type	Reset
			bit[23:0] : negedge interrupt status		

7.2.14.38 **PWM->INTPOSEDGIRQ**

PWM	**INTPOSEDGIRQ**	0x50011094
	PWM posedge interrupt active Contains the active for the PWM posedge interrupt sources.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								A																							

#	Bit(s)	Field	Description	Type	Reset
A	23:0	INTPOSEDGIRQ	*Interrupt active* bit[23:0] : posedge interrupt active	*ro*	n/a

7.2.14.39 **PWM->INTNEGEDGIRQ**

PWM	**INTNEGEDGIRQ**	0x50011098
	PWM negedge interrupt active Contains the active for the PWM negedge interrupt sources.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								A																							

#	Bit(s)	Field	Description	Type	Reset
A	23:0	INTNEGEDGIRQ	*Interrupt active*	*ro*	n/a

#	Bit(s)	Field	Description	Type	Reset
			bit[23:0] : negedge interrupt active		

7.2.14.40 **PWM->INTPWM**

PWM	**INTPWM**	0x5001109C
	PWM interrupt control Contains the enable, clear, status and active for the PWM period & updated interrupt sources.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						A	B							C	D							E	F							G	H

#	Bit(s)	Field	Description	Type	Reset
A	25	UPD	*Updated Interrupt active*	*ro*	n/a
B	24	PERIOD	*Period Interrupt active*	*ro*	n/a
C	17	UPD	*Updated Interrupt status*	*ro*	n/a
D	16	PERIOD	*Period Interrupt status*	*ro*	n/a
E	9	UPD	*Updated Interrupt clear*	*wo*	0
F	8	PERIOD	*Period Interrupt clear*	*wo*	0
G	1	UPD	*Updated Interrupt enable*	r/w	0
H	0	PERIOD	*Period Interrupt enable*	r/w	0

7.2.15 **IOCTRLA**

I/O control

7.2.15.1 **IOCTRLA->GPIO1**

IOCTRLA	**GPIO1**	0x50012000
	GPIO Pin 1 Control GPIO Pin 1 has four seperate drivers: GPIO Controller, PWM Controller ,Testmux and LINM_RXD.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		A																								B	C	D	E		

#	Bit(s)	Field	Description	Type	Reset
A	29:24	**MUXSEL**	Selects debug signal to be output on gpio1 0x**0** — PMUA (Power Management Unit Asic) qacks[0] 0x**1** — PMUA (Power Management Unit Asic) daffodil[0] 0x**2** — PMUA (Power Management Unit Asic) snowflake[0] 0x**3** — CHARGE PUMP CLOCKS pmu_CPclk4M 0x**4** — CRGA (Clock Reset Generation Asic) lf_rc_clk 0x**5** — CRGA (Clock Reset Generation Asic) hf_rc_clk 0x**6** — CRGA (Clock Reset Generation Asic) lf_rc_sts 0x**7** — CRGA (Clock Reset Generation Asic) hf_rc_sts 0x**8** — CRGA (Clock Reset Generation Asic) clk_sys_gated 0x**9** — CRGA (Clock Reset Generation Asic) (a_por_n) 0x**A** — CRGA (Clock Reset Generation Asic) (bor_3v3_n) 0x**B** — CRGA (Clock Reset Generation Asic) (ovtemp_flag)	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			0x**C** — CRGA (Clock Reset Generation Asic) (bor_1v5_n) 0x**D** — CRGA (Clock Reset Generation Asic) (wdt_bark) 0x**E** — ADC ctrl (Analog to Digital Converter Controller) sync_rcvd[0] 0x**F** — LINM rxd 0x**10** — LINM txd 0x**11** — LINS rxd 0x**12** — LINS txd 0x**13** — BOR CONTROL STATE MACHINE state[0] 0x**14** — BOR CONTROL STATE MACHINE state[1] 0x**15** — BOR CONTROL STATE MACHINE bor_bias_ena 0x**16** — BOR CONTROL STATE MACHINE bor_bias_ena_l 0x**17** — BOR CONTROL STATE MACHINE pmua_bor_bias_ena 0x**18** — BOR CONTROL STATE MACHINE hf_clk_allowed 0x**19** — BOR CONTROL STATE MACHINE hf_active 0x**1A** — BOR CONTROL STATE MACHINE pmua_bor_arm_sync 0x**1B** — BATTERY VOLTAGE MONITOR raw output from vbat low comp 0x**1C** — BATTERY VOLTAGE MONITOR raw output from vbat high comp 0x**1D** — BATTERY VOLTAGE MONITOR synchronised vbat low signal with polarity adjusted		

#	Bit(s)	Field	Description	Type	Reset
			0x**1E** — BATTERY VOLTAGE MONITOR synchronised vbat high signal with polarity adjusted 0x**1F** — BATTERY VOLTAGE MONITOR debounced synchronised vbat low signal 0x**20** — BATTERY VOLTAGE MONITOR debounced synchronised vbat high signal 0x**21** — BUCK Controller state[0] 0x**22** — BUCK Controller state[1] 0x**23** — BUCK Controller state[2] 0x**24** — BUCK Controller state[3] 0x**25** — BUCK Controller state[4] 0x**26** — BUCK Controller fine_count[0] 0x**27** — BUCK Controller fine_count[1] 0x**28** — BUCK Controller fine_count[2] 0x**29** — BUCK Controller fine_count[3] 0x**2A** — BUCK Controller fine_count[4] 0x**2B** — BUCK Controller fine_count[5] 0x**2C** — BUCK Controller fine_count[6] 0x**2D** — BUCK Controller buck_clk 0x**2E** — BUCK Controller buck_overcur 0x**2F** — BUCK Controller buck_ready 0x**30** — BUCK Controller buck_off 0x**31** — BUCK dtb[0] 0x**32** — BUCK dtb[1] 0x**33** — BUCK dtb[2]		

#	Bit(s)	Field	Description	Type	Reset
			0x**37** — LIN Slave PHY Input Txd 0x**3B** — LIN Master Input PHY Txd		
B	5	RDENA	***read enable*** 0x**0** — Disable Read path on the GPIO 0x**1** — Enable Read path on the GPIO	r/w	0
C	4	PDENA	***pulldown enable*** 0x**0** — Disable 100K Ω Pull Down 0x**1** — Enable 100K Ω Pull Down	r/w	0
D	3	PUENA	***pullup enable (active-low)*** 0x**0** — Enable 100K Ω Pull Up 0x**1** — Disable 100K Ω Pull Up	r/w	*1*
E	2:0	HWMODE	***hardware mode*** 0x**0** — GPA[0] Mode. 'GPIO Barium' reads and writes data to the GPIO. Output Enable is controlled by GPIO Barium via the direction bit. Read Enable, Pull-Up, and Pull-Down are controlled by GPIO register in IOCTRLA. 0x**1** — PWM[0] Mode. 'PWM Barium' writes data to the GPIO. 0x**2** — Testmux Mode. 'Digital Testmux' writes data to the GPIO. DEBUG Access must be enabled in the system control block to allow testmux outputs.	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			0x**3** — LINM RXD Input from GPIO(External PHY). 0x**4** — LINM Single wire mode, LINM RXD/LINM TXD, Open-drain output. 0x**5** — Monitor 2nd Stage output of LINS RX Glitch filter. 0x**6** — Monitor 2nd Stage output of LINM RX Glitch filter.		

7.2.15.2 **IOCTRLA->GPIO2**

IOCTRLA	**GPIO2**	0x50012004
	GPIO Pin 2 Control GPIO Pin 2 has five seperate drivers: GPIO Controller, PWM Controller ,Testmux , LINM_TXD and SPI_SS.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																										A	B	C	D		

#	Bit(s)	Field	Description	Type	Reset
A	5	RDENA	***read enable*** 0x**0** — Disable Read path on the GPIO 0x**1** — Enable Read path on the GPIO	r/w	0
B	4	PDENA	***pulldown enable*** 0x**0** — Disable 100K Ω Pull Down 0x**1** — Enable 100K Ω Pull Down	r/w	0

#	Bit(s)	Field	Description	Type	Reset
C	3	PUENA	***pullup enable (active-low)*** 0x**0** — Enable 100K Ω Pull Up 0x**1** — Disable 100K Ω Pull Up	r/w	*1*
D	2:0	HWMODE	***hardware mode*** 0x**0** — GPA[1] Mode. 'GPIO Barium' reads and writes data to the GPIO. Output Enable is controlled by GPIO Barium via the direction bit. Read Enable, Pull-Up, and Pull-Down are controlled by GPIO register in IOCTRLA. 0x**1** — PWM[1] Mode. 'PWM Barium' writes data to the GPIO. 0x**2** — Testmux Mode. GPIO input connects to LINS TXD. DEBUG Access must be enabled in the system control block to allow testmux outputs. 0x**3** — LINM TXD Output to GPIO. 0x**4** — SPI Master Slave Select. 0x**5** — Testmux Mode. GPIO output 1st Stage output of LINS RX Glitch filter. 0x**6** — Monitor LINS Input of 1st Stage Glitch Filter Rxd.	r/w	0

7.2.15.3 **IOCTRLA->GPIO3**

IOCTRLA	**GPIO3**	0x50012008
	GPIO Pin 3 Control	

	GPIO Pin 3 has six seperate drivers: GPIO Controller, PWM Controller , Testmux , LINS_RXD and SPI_SCK.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																										A	B	C	D		

#	Bit(s)	Field	Description	Type	Reset
A	5	RDENA	***read enable*** 0x**0** — Disable Read path on the GPIO 0x**1** — Enable Read path on the GPIO	r/w	0
B	4	PDENA	***pulldown enable*** 0x**0** — Disable 100K Ω Pull Down 0x**1** — Enable 100K Ω Pull Down	r/w	0
C	3	PUENA	***pullup enable (active-low)*** 0x**0** — Enable 100K Ω Pull Up 0x**1** — Disable 100K Ω Pull Up	r/w	*1*
D	2:0	HWMODE	***hardware mode*** 0x**0** — GPA[2] Mode. 'GPIO Barium' reads and writes data to the GPIO. Output Enable is controlled by GPIO Barium via the direction bit. Read Enable, Pull-Up, and Pull-Down are controlled by GPIO register in IOCTRLA. 0x**1** — PWM[2] Mode. 'PWM Barium' writes data to the GPIO. 0x**2** — Testmux Mode. LINM RXD connects to GPIO output. DEBUG Access must be enabled in	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			the system control block to allow testmux outputs. 0x**3** — LINS RXD Input from GPIO(External PHY). 0x**4** — LINS Single wire mode, LINS RXD/LINS TXD, Open-drain output. 0x**5** — SPI Master Clock Out. 0x**6** — Monitor 1st Stage output of LINS RX Glitch filter. 0x**7** — Monitor 1st Stage output of LINM RX Glitch filter.		

7.2.15.4 **IOCTRLA->GPIO4**

IOCTRLA	**GPIO4**	0x5001200C
	GPIO Pin 4 Control GPIO Pin 4 has five seperate drivers: GPIO Controller, PWM Controller ,UART_RXD_IN ,LINS_TXD and SPI_MOSI.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																										A	B	C	D		

#	Bit(s)	Field	Description	Type	Reset
A	5	RDENA	***read enable*** 0x**0** — Disable Read path on the GPIO 0x**1** — Enable Read path on the GPIO	r/w	0
B	4	PDENA	***pulldown enable***	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			0x**0** — Disable 100K Ω Pull Down 0x**1** — Enable 100K Ω Pull Down		
C	3	PUENA	***pullup enable (active-low)*** 0x**0** — Enable 100K Ω Pull Up 0x**1** — Disable 100K Ω Pull Up	r/w	*1*
D	2:0	HWMODE	***hardware mode*** 0x**0** — GPA[3] Mode. 'GPIO Barium' reads and writes data to the GPIO. Output Enable is controlled by GPIO Barium via the direction bit. Read Enable, Pull-Up, and Pull-Down are controlled by GPIO register in IOCTRLA. 0x**1** — PWM[3] Mode. 'PWM Barium' writes data to the GPIO. 0x**2** — UART RXD Input from GPIO 0x**3** — UART Single Wire Mode 0x**4** — SPI Master MOSI 0x**5** — Testmux Mode. GPIO input connects to LINM TXD. 0x**6** — Monitor LINS Core Rxd input. 0x**7** — Monitor LINM Core Rxd input.	r/w	0

7.2.15.5 **IOCTRLA->GPIO5**

IOCTRLA	**GPIO5**	0x50012010
	GPIO Pin 5 Control	

	GPIO Pin 5 has five seperate drivers: GPIO Controller, PWM Controller ,UART_TXD_OUT , LINS_TXD and SPI MISO.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																							A			B	C	D	E		

#	Bit(s)	Field	Description	Type	Reset
A	8	UART_TXD_IDLE_DIR	***Control the direction of PAD for UART TXD Idle*** 0x**0** — Hold the UART TXD Pad to output when TXD is idle, 0x**1** — Switch the TXD Pad to input When TXD is idle.	r/w	0
B	5	RDENA	***read enable*** 0x**0** — Disable Read path on the GPIO 0x**1** — Enable Read path on the GPIO	r/w	0
C	4	PDENA	***pulldown enable*** 0x**0** — Disable 100K Ω Pull Down 0x**1** — Enable 100K Ω Pull Down	r/w	0
D	3	PUENA	***pullup enable (active-low)*** 0x**0** — Enable 100K Ω Pull Up 0x**1** — Disable 100K Ω Pull Up	r/w	*1*
E	2:0	HWMODE	***hardware mode*** 0x**0** — GPA[4] Mode. ‘GPIO Barium’ reads and writes data to the GPIO. Output Enable is controlled by GPIO	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			Barium via the direction bit. Read Enable, Pull-Up, and Pull-Down are controlled by GPIO register in IOCTRLA. 0x**1** — PWM[4] Mode. 'PWM Barium' writes data to the GPIO. 0x**2** — UART TXD Output to GPIO 0x**3** — LINS TXD Output to GPIO. 0x**4** — SPI Master MISO. 0x**5** — Monitor LINM Input of 1st Stage Glitch Filter Rxd. 0x**6** — Monitor 1st Stage output of LINS RX Glitch filter. 0x**7** — Monitor 1st Stage output of LINM RX Glitch filter.		

7.2.15.6 **IOCTRLA->GPIO6**

IOCTRLA	**GPIO6**	0x50012014
	GPIO Pin 6 Control GPIO Pin 6 has five seperate drivers: GPIO Controller, PWM Controller ,Testmux , LINM_TXD and SPI_SS.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																									A	B	C	D	E		

#	Bit(s)	Field	Description	Type	Reset
A	6	UART_RXO_SW	***UART RX output Switch*** 0x**0** — Bypass UART RXD 0x**1** — Force UART RXD output to Reset Value-'1'	r/w	0
B	5	RDENA	***read enable*** 0x**0** — Disable Read path on the GPIO 0x**1** — Enable Read path on the GPIO	r/w	0
C	4	PDENA	***pulldown enable*** 0x**0** — Disable 100K Ω Pull Down 0x**1** — Enable 100K Ω Pull Down	r/w	0
D	3	PUENA	***pullup enable (active-low)*** 0x**0** — Enable 100K Ω Pull Up 0x**1** — Disable 100K Ω Pull Up	r/w	*1*
E	2:0	HWMODE	***hardware mode*** 0x**0** — GPA[5] Mode. 'GPIO Barium' reads and writes data to the GPIO. Output Enable is controlled by GPIO Barium via the direction bit. Read Enable, Pull-Up, and Pull-Down are controlled by GPIO register in IOCTRLA. 0x**1** — PWM[1] Mode. 'PWM Barium' writes data to the GPIO.	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			0x**2** — Testmux Mode. GPIO output 2st Stage output of LINS RX Glitch filter. 0x**3** — LINM TXD Output to GPIO. 0x**4** — SPI Master MISO. 0x**5** — UART RXD Output to GPIO. 0x**6** — Monitor 2nd Stage output of LINS RX Glitch filter. 0x**7** — Monitor 2nd Stage output of LINM RX Glitch filter.		

7.2.15.7 **IOCTRLA->GPIO7**

IOCTRLA	**GPIO7**	0x50012018
	GPIO Pin 7 Control GPIO Pin 7 has six seperate drivers: GPIO Controller, PWM Controller , Testmux , LINS_RXD and SPI_SCK.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																										A	B	C	D		

#	Bit(s)	Field	Description	Type	Reset
A	5	RDENA	***read enable*** 0x**0** — Disable Read path on the GPIO 0x**1** — Enable Read path on the GPIO	r/w	0
B	4	PDENA	***pulldown enable*** 0x**0** — Disable 100K Ω Pull Down	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			0x**1** — Enable 100K Ω Pull Down		
C	3	PUENA	***pullup enable (active-low)*** 0x**0** — Enable 100K Ω Pull Up 0x**1** — Disable 100K Ω Pull Up	r/w	*1*
D	2:0	HWMODE	***hardware mode*** 0x**0** — GPA[6] Mode. 'GPIO Barium' reads and writes data to the GPIO. Output Enable is controlled by GPIO Barium via the direction bit. Read Enable, Pull-Up, and Pull-Down are controlled by GPIO register in IOCTRLA. 0x**1** — PWM[2] Mode. 'PWM Barium' writes data to the GPIO. 0x**2** — Testmux Mode. GPIO output 2st Stage output of LINM RX Glitch filter. 0x**3** — LINS RXD Input from GPIO(External PHY). 0x**4** — LINS Single wire mode, LINS RXD/LINS TXD, Open-drain output. 0x**5** — SPI Master Clock Out. 0x**6** — UART TXD Input from GPIO.	r/w	0

7.2.15.8 **IOCTRLA->GPIO8**

IOCTRLA	**GPIO8**	0x5001201C
	GPIO Pin 8 Control	

	GPIO Pin 8 has five seperate drivers: GPIO Controller, PWM Controller ,UART_RXD_IN ,LINS_TXD and SPI_MOSI.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																										A	B	C	D		

#	Bit(s)	Field	Description	Type	Reset
A	5	RDENA	***read enable*** 0x**0** — Disable Read path on the GPIO 0x**1** — Enable Read path on the GPIO	r/w	0
B	4	PDENA	***pulldown enable*** 0x**0** — Disable 100K Ω Pull Down 0x**1** — Enable 100K Ω Pull Down	r/w	0
C	3	PUENA	***pullup enable (active-low)*** 0x**0** — Enable 100K Ω Pull Up 0x**1** — Disable 100K Ω Pull Up	r/w	*1*
D	2:0	HWMODE	***hardware mode*** 0x**0** — GPA[7] Mode. 'GPIO Barium' reads and writes data to the GPIO. Output Enable is controlled by GPIO Barium via the direction bit. Read Enable, Pull-Up, and Pull-Down are controlled by GPIO register in IOCTRLA. 0x**1** — PWM[3] Mode. 'PWM Barium' writes data to the GPIO. 0x**2** — UART RXD Input from GPIO	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			0x**3** — UART Single Wire Mode 0x**4** — SPI Master MOSI 0x**5** — Monitor LINS Core Rxd input. 0x**6** — Monitor LINM Core Rxd input.		

7.2.15.9 **IOCTRLA->GPIO9**

IOCTRLA	**GPIO9**	0x50012020
	GPIO Pin 9 Control GPIO Pin 9 has five seperate drivers: GPIO Controller, PWM Controller ,UART_TXD_OUT , LINS_TXD and SPI MISO.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																										A	B	C	D		

#	Bit(s)	Field	Description	Type	Reset
A	5	RDENA	***read enable*** 0x**0** — Disable Read path on the GPIO 0x**1** — Enable Read path on the GPIO	r/w	0
B	4	PDENA	***pulldown enable*** 0x**0** — Disable 100K Ω Pull Down 0x**1** — Enable 100K Ω Pull Down	r/w	0
C	3	PUENA	***pullup enable (active-low)*** 0x**0** — Enable 100K Ω Pull Up	r/w	*1*

#	Bit(s)	Field	Description	Type	Reset
			0x**1** — Disable 100K Ω Pull Up		
D	2:0	HWMODE	***hardware mode*** 0x**0** — GPB[0] Mode. ‘GPIO Barium’ reads and writes data to the GPIO. Output Enable is controlled by GPIO Barium via the direction bit. Read Enable, Pull-Up, and Pull-Down are controlled by GPIO register in IOCTRLA. 0x**1** — PWM[4] Mode. ‘PWM Barium’ writes data to the GPIO. 0x**2** — UART TXD Output to GPIO 0x**3** — LINS TXD Output to GPIO. 0x**4** — SPI Master Slave Select. 0x**6** — Monitor LINM Input of 1st Stage Glitch Filter Rxd.	r/w	0

7.2.15.10 **IOCTRLA->LIN**

IOCTRLA	**LIN**	0x50012024
	LIN Pin Control	

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
A							B				C	D	E	F		G	H	I	J				K		L	M		N			O

#	Bit(s)	Field	Description	Type	Reset
A	31	PMODE	*LIN Power Mode* Control LINS/LINM power state in hibernate mode	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			0x**0** — Regardless of the related enable bits, LIN TX analog parts will be shut-down in hibernate mode, unless any LIN dominant signal is detected. 0x**1** — LIN TX analog parts are still controlled by theirs corresponding enable bits.		
B	24	SWON_LOCK	*SWON Lock Bit*	r/w	0
C	19	LINM_PUOFF_TIMEOUT	*LINM 1K Pullup disable in dominant TimeOut condition* Set to disable LINM 1K pullup in case that lin bus is shorted to ground(Bus idle dominant timeout is detected) for saving power. LINS 1K Pullup will be recovered automatically if bus idle dominant timeout is released by any bus activity. If this bit is set & SWON = 1, LINM's 30K pullup will be enabled for preventing bus floating. Only reset by power-on sequence	r/w	*1*
D	18	LINS_PUOFF_TIMEOUT	*LINS Pullup Disable in dominant TimeOut condition* Set to disable LINS 30K pullup in case that lin bus is shorted to ground(Bus idle dominant timeout is detected) for saving power. LINS Pullup will be recovered automatically if bus idle dominant	r/w	*1*

#	Bit(s)	Field	Description	Type	Reset
			timeout is released by any bus activity. Only reset by power-on sequence		
E	17	SWOFF_TIMEOUT	*Switch off in dominant TimeOut condition* Set to disconnect lin switch in case that lin bus is shorted to ground(Bus idle dominant timeout is detected) for saving power. LIN Switch will be recovered automatically if bus idle dominant timeout is released by any bus activity. Only reset by power-on sequence	r/w	*1*
F	16	SWON	*LIN Dual Mode Switch On* Set to enable dual-mode lin switch. Only reset by power-on sequence. This bit can be locked by setting SWON_LOCK — read-only, unless 'lin_swon access' is enabled	r/w	*1*
G	14	LINM_RXENA	LIN receive enable	r/w	0
H	13	LINM_TXENA	*LIN transmit enable*	r/w	0
I	12	LINM_PU1K_ENA	LIN 1K pullup enable	r/w	*1*
J	11	LINM_PU30K_ENA	LIN 30K pullup enable	r/w	0
K	8	LINM_HWMODE	***LIN Master hardware mode*** 0x**0** — Hardware Mode Disabled. GPIO Barium Peripheral GPB[2] writes/reads the LIN I/O pin.	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			0x**1** — Hardware Mode Enabled. LIN peripheral writes/read the LIN I/O pin.		
L	6	LINS_RXENA	LIN receive enable	r/w	0
M	5	LINS_TXENA	*LIN transmit enable*	r/w	0
N	3	LINS_PU30K_ENA	*LIN 30K pullup enable*	r/w	*1*
O	0	LINS_HWMODE	***LIN Slave hardware mode*** 0x**0** — Hardware Mode Disabled. GPIO Barium Peripheral GPB[1] writes/reads the LIN I/O pin. 0x**1** — Hardware Mode Enabled. LIN peripheral writes/reads the LIN I/O pin.	r/w	0

7.2.15.11 **IOCTRLA->LINSGFCONF**

IOCTRLA	**LINSGFCONF**	0x50012028
	LINS Glitch Filter Configuration in active mode	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									A								B											C			

#	Bit(s)	Field	Description	Type	Reset
A	22:16	LINS_DBNC_THRES1_1ST	*1st Stage LINS Debounce Threshold for 0 to 1*	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			1st Stage LINS Debounce Threshold for 0 to 1, detect ‘1’ width of (T_clksys*resetvalue = 62.5ns*0 = 0[us]		
B	14:8	LINS_DBNC_THRES0_1ST	*1st Stage LINS Debounce Threshold for 1 to 0* 1st Stage LINS Debounce Threshold for 1 to 0, detect ‘0’ width of (T_clksys*resetvalue = 62.5ns*8 = 0.5[us]	r/w	*8*
C	3:0	LINS_RX_GF_ENA	*LINS RXD Glitch Filter enables* Filter 1st stage; bit-1: Enable LINS Glitch Filter 2nd stage; bit-2: Enable LINS Glitch Filter 3rd stage; bit-3: Enable LINS Glitch Filter for Sleep Mode	r/w	*0xF*

7.2.15.12 **IOCTRLA->LINSGFCONF1**

IOCTRLA	**LINSGFCONF1**	0x5001202C
	LINS Glitch Filter Configuration in active mode	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	A								B								C								D						

#	Bit(s)	Field	Description	Type	Reset
A	30:24	LINS_DBNC_THRES1_3RD	*3rd Stage LINS Debounce Threshold for 0 to 1* 3rd Stage LINS Debounce Threshold for 0 to 1, detect ‘1’ width of	r/w	*0x28*

#	Bit(s)	Field	Description	Type	Reset
			(T_clksys*resetvalue = 62.5ns*40 = 2.5[us]		
B	22:16	LINS_DBNC_THRES0_3RD	*3rd Stage LINS Debounce Threshold for 1 to 0* 3rd Stage LINS Debounce Threshold for 1 to 0, detect '0' width of (T_clksys*resetvalue = 62.5ns*40 = 2.5[us]	r/w	*0x28*
C	14:8	LINS_DBNC_THRES1_2ND	*2nd Stage LINS Debounce Threshold for 0 to 1* 2nd Stage LINS Debounce Threshold for 0 to 1, detect '1' width of (T_clksys*resetvalue = 62.5ns*32 = 2[us]	r/w	*0x20*
D	6:0	LINS_DBNC_THRES0_2ND	*2nd Stage LINS Debounce Threshold for 1 to 0* 2nd Stage LINS Debounce Threshold for 1 to 0, detect '0' width of (T_clksys*resetvalue = 62.5ns*16 = 1[us]	r/w	*0x10*

7.2.15.13 **IOCTRLA->LINMGFCONF**

IOCTRLA	**LINMGFCONF**	0x50012030
	LINM Glitch Filter Configuration in active mode	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									A								B											C			

#	Bit(s)	Field	Description	Type	Reset
A	22:16	LINM_DBNC_THRES1_1ST	*LINM Debounce Threshold for 0 to 1* 1st Stage LINM Debounce Threshold for 0 to 1, detect '1' width of (T_clksys*resetvalue = 62.5ns*0 = 0[us]	r/w	0
B	14:8	LINM_DBNC_THRES0_1ST	*LINM Debounce Threshold for 1 to 0* 1st Stage LINM Debounce Threshold for 1 to 0, detect '0' width of (T_clksys*resetvalue = 62.5ns*8 = 0.5[us]	r/w	*8*
C	3:0	LINM_RX_GF_ENA	*LINM RXD Glitch Filter enables* Filter 1st stage; bit-1: Enable LINM Glitch Filter 2nd stage; bit-2: Enable LINM Glitch Filter 3rd stage; bit-3: Enable LINM Glitch Filter for Sleep Mode	r/w	*0xF*

7.2.15.14 **IOCTRLA->LINMGFCONF1**

IOCTRLA	**LINMGFCONF1**	0x50012034
	LINM Glitch Filter Configuration in active mode	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	A								B								C								D						

#	Bit(s)	Field	Description	Type	Reset
A	30:24	LINM_DBNC_THRES1_3RD	*3rd Stage LINM Debounce Threshold for 0 to 1*	r/w	*0x28*

#	Bit(s)	Field	Description	Type	Reset
			3rd Stage LINM Debounce Threshold for 0 to 1, detect ‘1’ width of (T_clksys*resetvalue = 62.5ns*40 = 2.5[us]		
B	22:16	LINM_DBNC_THRES0_3RD	*3rd Stage LINM Debounce Threshold for 1 to 0* 3rd Stage LINM Debounce Threshold for 1 to 0, detect ‘0’ width of (T_clksys*resetvalue = 62.5ns*40 = 2.5[us]	r/w	*0x28*
C	14:8	LINM_DBNC_THRES1_2ND	*2nd Stage LINM Debounce Threshold for 0 to 1* 2nd Stage LINM Debounce Threshold for 0 to 1, detect ‘1’ width of (T_clksys*resetvalue = 62.5ns*32 = 2[us]	r/w	*0x20*
D	6:0	LINM_DBNC_THRES0_2ND	*2nd Stage LINM Debounce Threshold for 1 to 0* 2nd Stage LINM Debounce Threshold for 1 to 0, detect ‘0’ width of (T_clksys*resetvalue = 62.5ns*16 = 1[us]	r/w	*0x10*

7.2.15.15 IOCTRLA->LINTXDMONITOR

IOCTRLA	**LINTXDMONITOR**	0x50012038
	LIN TXD Dominant Timeout	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																						A	B							C	D

#	Bit(s)	Field	Description	Type	Reset
A	9	LINMTXDTIMEOUTDOMINANT	*Tx Dominant Timeout* The bit is set by LINM TxD monitor if LINM's TxD is stuck at dominant output for 64ms. A dominant to recessive transition of the TxD will clear this bit. LIN's TX will be disabled automatically when this bit is set	*ro*	0
B	8	LINSTXDTIMEOUTDOMINANT	*Tx Dominant Timeout* The bit is set by LINS TxD monitor if LINS's TxD is stuck at dominant output for 64ms. A dominant to recessive transition of the TxD will clear this bit. LIN's TX will be disabled automatically when this bit is set	*ro*	0
C	1	LINMTXDMONITORENA	*LINM TxD Monitor enable*	r/w	*1*
D	0	LINSTXDMONITORENA	*LINS TxD Monitor enable*	r/w	*1*

7.2.15.16 **IOCTRLA->LEDPIN**

IOCTRLA	**LEDPIN**	0x5001203C
	LED Pin Control	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A		B					C																								D

#	Bit(s)	Field	Description	Type	Reset
A	31	VFW_ENA	*LED Forward Voltage Current Enable* Set to enable the independent LED VFW current source(maximum=5mA). When ADC CH2 measurement is active, the LED channel selected by CH2_SEL will be driven by LED_VFW current source	r/w	0
B	29:25	SENSE_CTRL	*LED* LED Sense Control bits for override control/debug	r/w	0
C	24	SENSE_ENA	*LED Forward Voltage Sense Enable* Set to enable LED forward voltage sense module	r/w	0
D	0	HWMODE	*LED hardware mode* LED Hardware Mode Enable. Each bit controls the corresponding LED Channel respectively 0x**0** — Hardware Mode Disabled. LED_DATA register drives the LED Data Output pin. Read is not available on this pin. 0x**1** — Hardware Mode Enabled. PWM_BARIUM peripheral drives the LED Data Output pin. Read is not available on this pin.	r/w	0

7.2.15.17 **IOCTRLA->LEDDATA**

IOCTRLA	**LEDDATA**	0x50012040
	LED Data Out	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								A																							

#	Bit(s)	Field	Description	Type	Reset
A	23:0	LEDDATA	*LED Data Out* When the LED hardware mode is disabled, then the data in this register bit is used to drive the LED driver. Each bit controls the corresonding LED Channel respectively	r/w	0

7.2.15.18 **IOCTRLA->LEDPUENA**

IOCTRLA	**LEDPUENA**	0x50012044
	LED Data Out	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								A																							

#	Bit(s)	Field	Description	Type	Reset
A	23:0	LEDPUENA	*LED Pullup Enable* Set to enable LED Pullup when no LED current source is connected	r/w	0

7.2.15.19 **IOCTRLA->ANALOGTESTMUXOVERRIDE1**

IOCTRLA	**ANALOGTESTMUXOVERRIDE1**	0x50012048
	Analog Testmux Override 1 This register controls the multiplexers for analog signals.	

The select bit allows firmware to control the corresponding select field (in other words, firmware control).

The following table is intended to be a helpful guide in what data should be written to this register in order to connect a source and target together.

Note- Care should be taken to write zero to this register between connection changes. This ensures a clean break between selections.

Data to write at address 0x50011018 to enable connection

Source Description	Target Description	Data to write to connect Source to Target
3.3V Digital Supply	GPIO1	0x0101_0006
3.3V Analog Supply	GPIO1	0x0102_0006
1.5V MCU Supply	GPIO1	0x0104_0006
Band Gap-vbg_buffered raw	GPIO1	0x0108_0006
VDD Limit = Measured Voltage * 4	GPIO2	0x0210_0006
VDD Protected = Measured Voltage * 9	GPIO2	0x0220_0006
Band Gap	ADC	0x0000_0101
Accurate VBAT	ADC	0x0000_0201
LED Forward Voltage	ADC	0x0000_0401
Temperature Sensor or accurate_vbat	ADC	0x0000_0801
GPIO1/2/3/4	ADC	0x0000_1001

	VDD1.5V	ADC	0x0000_2001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																	A												B	C	D

#	Bit(s)	Field	Description	Type	Reset
A	14:8	ADCCONREG	*Firmware Debug Value* Contains the output value when the ADC_CON_SEL firmware select bit is set 0x**1** — Select Major Bandgap to ADC 0x**2** — Select Accurate VBAT to ADC 0x**4** — Select LED0~23(determind by CH2_SEL in ADC_CTRL) Forward Voltage to ADC 0x**8** — Select Tempsensor to ADC 0x**10** — Select GPIO1~9(determined by CH4_SEL in ADC_CTRL) to ADC 0x**20** — Select VDD1.5v to ADC 0x**40** — Select VDD5.0v to ADC	r/w	0
B	2	ADCSELSEL	***Hardware/Firmware Select*** 0x**0** — Hardware Controlled. 0x**1** — ADC_SEL_REG controls output. — only accessible when 'debug access' is enabled	r/w	0
C	1	GPIOCONSEL	***Hardware/Firmware Select*** 0x**0** — Hardware Controlled. 0x**1** — GPIO_CON_REG controls output.	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			— only accessible when 'debug access' is enabled		
D	0	ADCCONSEL	***Hardware/Firmware Select*** 0x**0** — Hardware Controlled. 0x**1** — ADC_CON_REG controls output. — only accessible when 'debug access' is enabled	r/w	0

7.2.15.20 **IOCTRLA->ANALOGTESTMUXOVERRIDE2**

IOCTRLA	**ANALOGTESTMUXOVERRIDE2**	0x5001204C
	Analog Testmux Override 2	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							A															B									

#	Bit(s)	Field	Description	Type	Reset
A	24:16	ADCSELREG	*Firmware Debug Value* Need to be configured with gpio_con_reg to enable gpio analog output to ADC 0x**1** — Enable GPIO1 analog connection 0x**2** — Enable GPIO2 analog connection 0x**4** — Enable GPIO3 analog connection 0x**8** — Enable GPIO4 analog connection 0x**10** — Enable GPIO5 analog connection 0x**20** — Enable GPIO6 analog connection 0x**40** — Enable GPIO7 analog connection	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			0x**80** — Enable GPIO8 analog connection 0x**100** — Enable GPIO9 analog connection		
B	9:0	GPIOCONREG	*Firmware Debug Value* GPIO1~9 test MUX select. ADC_SEL_SEL/ADC_SEL_REG need to be configured at the same time to enable required GPIO connection 0x**1** — Select 3.3V digital supply to GPIO1 0x**2** — Select 3.3v analog supply to GPIO1 0x**4** — Select 1.5v MCU supply to GPIO1 0x**8** — Select Major Bandgap supply to GPIO1 0x**10** — Select vdd_lim/4 to GPIO2 0x**20** — Select vdd_prt/9 to GPIO2 0x**40** — Select tempsensor to GPIO2 0x**80** — Select TBD to GPIO2 0x**100** — Select ADC_VCM to GPIO2 0x**200** — Select BUCK_ATB1 to GPIO3 and BUCK_ATB2 to GPIO4.	r/w	0

7.2.15.21 **IOCTRLA->IRQ**

IOCTRLA	**IRQ**	0x50012050
	IOCTRLA LINS/LINM TXD Dominant Monitor interrupts Contains the the enable, clear, status and active flag for the LINS/LINM TXD Dominant Monitor interrupt sources.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						A	B							C	D							E	F							G	H

#	Bit(s)	Field	Description	Type	Reset
A	25	LINM_TXD_DOM	*LINM TXD Dominant Monitor interrupt active*	*ro*	0
B	24	LINS_TXD_DOM	*LINS TXD Dominant Monitor interrupt active*	*ro*	0
C	17	LINM_TXD_DOM	*LINM TXD Dominant Monitor interrupt status*	*ro*	0
D	16	LINS_TXD_DOM	*LINS TXD Dominant Monitor interrupt status*	*ro*	0
E	9	LINM_TXD_DOM	*LINM TXD Dominant Monitor interrupt clear* – cleared automatically after each write	*wo*	0
F	8	LINS_TXD_DOM	*LINS TXD Dominant Monitor interrupt clear* – cleared automatically after each write	*wo*	0
G	1	LINM_TXD_DOM	*LINM TXD Dominant Monitor interrupt enable*	r/w	0
H	0	LINS_TXD_DOM	*LINS TXD Dominant Monitor interrupt enable*	r/w	0

7.2.16 **SYSCTRLA**

System control

7.2.16.1 **SYSCTRLA->RETAIN0**

SYSCTRLA	**RETAIN0**	0x50013000
	Retained data 0	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																												A			

#	Bit(s)	Field	Description	Type	Reset
A	3:0	RETAIN0	*Firmware scratch register 0* Only reset at power-on (e.g contents retained in Hibernate mode and retained despite any hard or soft resets)	r/w	0

7.2.16.2 **SYSCTRLA->RETAIN1**

SYSCTRLA	**RETAIN1**	0x50013004
	Retained data 1	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																												A			

#	Bit(s)	Field	Description	Type	Reset
A	3:0	RETAIN1	*Firmware scratch register 1 (0x1)* Contents retained in Hibernate mode - but lost after any hard or soft reset	r/w	0

7.2.16.3 **SYSCTRLA->DEBUG_ACCESS_KEY**

SYSCTRLA	**DEBUG_ACCESS_KEY**	0x50013008
	Debug access key	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																												B			

#	Bit(s)	Field	Description	Type	Reset
A	31	DEBUG_LOCK	Set Only bit. Set this bit to lock DEBUG_CODE bits	r/w	0
B	3:0	DEBUG_ACCESS_KEY	Write the value 0x5 to this register to enable debug options. Write any other value to disable the debug options — read-only, unless 'debug_code access' is enabled	r/w	0

7.2.16.4 **SYSCTRLA->DEBUG_ACCESS_ENABLED**

SYSCTRLA	**DEBUG_ACCESS_ENABLED**	0x5001300C
	Debug access enabled	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															A

#	Bit(s)	Field	Description	Type	Reset
A	0	DEBUG_ACCESS_ENABLED	A status flag that is set when debug access is enabled	*ro*	0

7.2.16.5 **SYSCTRLA->TRIM_ACCESS_KEY**

SYSCTRLA	**TRIM_ACCESS_KEY**	0x50013010
	Trim access key	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																												B			

#	Bit(s)	Field	Description	Type	Reset
A	31	TRIM_LOCK	Set Only bit. Write 1 to this bit to lock TRIM_CODE bits	r/w	0
B	3:0	TRIM_ACCESS_KEY	Write the value 0xe to this register to enable ‘trim access’ (which allows write access to various trim settings and production test options). Write any other value to disable trim access — read-only, unless ‘trim_code access’ is enabled	r/w	0

7.2.16.6 **SYSCTRLA->TRIM_ACCESS_ENABLED**

SYSCTRLA	**TRIM_ACCESS_ENABLED**	0x50013014
	Trim access enabled	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															A

#	Bit(s)	Field	Description	Type	Reset
A	0	TRIM_ACCESS_ENABLED	A status flag that is set when trim access is enabled	*ro*	0

7.2.16.7 **SYSCTRLA->PMU_TRIM**

SYSCTRLA	**PMU_TRIM**	0x50013018
	PMU trim values	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																		A							B						

#	Bit(s)	Field	Description	Type	Reset
A	13:8	RESISTOR_TRIM	*V2I Resistor Trim* Selects the resistor trim value for the V2I circuit. The larger RESISTOR_TRIM value, the lower V2I current — read-only, unless 'trim access' is enabled	r/w	*0x1E*
B	6:4	TRIM	*Band Gap Trim* Selects the trim value for the band gap reference circuit. The larger TRIM value, the higher bandgap voltage — read-only, unless 'trim access' is enabled	r/w	*4*

7.2.16.8 **SYSCTRLA->LF_OSC_TRIM**

SYSCTRLA	**LF_OSC_TRIM**	0x5001301C
	Trim controls for the low frequency (32k/250KHz) oscillators	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								A							

#	Bit(s)	Field	Description	Type	Reset
A	7:0	TRIM_LF_RC	*LF RC oscillator trim* Following value only used as a reference: 0x**0** — freq(32k)=19.06kHz , frea(250k)=131.1Khz 0x**AF** — freq(32k)=38.57kHz , frea(250k)=255.9Khz 0x**FF** — freq(32k)=66.06kHz , frea(250k)=421.7Khz — read-only, unless 'trim access' is enabled	r/w	*0xAF*

7.2.16.9 SYSCTRLA->HF_OSC_TRIM

SYSCTRLA	**HF_OSC_TRIM**	0x50013020
	Trim controls for the high frequency (16MHz) oscillator	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								A												B			C	D							

#	Bit(s)	Field	Description	Type	Reset
A	23:16	SSCDIV	*SSC Clock Divider* SSC Freq = SYS_FREQ/[(SSC_DIV+1)*(SSC_DEEP+1)*4]	r/w	*0x28*
B	11:9	SSCDEEP	*SSC Depth Configuration*	r/w	0
C	8	SSCENA	*SSC Enable*	r/w	0
D	7:0	TRIM_HF_RC	*High Frequency RC Oscillator trim* TRIM bits will be changed if SSC is enabled. Reload the trim bits from Flash if SSC is disabled while the change has happened. Following value only used as a reference: 0x**A** — freq=26.08Mhz 0x**80** — freq=9.62Mhz 0x**FA** — freq=5.94Mhz	r/w	*0x34*

7.2.16.10 SYSCTRLA->BIAS

SYSCTRLA	**BIAS**	0x50013024
	Bias Control	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																														A	B

#	Bit(s)	Field	Description	Type	Reset
A	1	LEDBIASREG	*High Voltage LED bias select register* If LED_BIAS_SEL is set, then LED_BIAS_REG allows override access to the LED BIAS signal	r/w	0
B	0	LEDBIASSEL	***High Voltage LED bias select*** 0x**0** — The LED Bias is enabled and disabled by the pmu hardware state machine by default. 0x**1** — The value of the LED_BIAS_REG field is what is used to drive the LED_BIAS signal. (override mode)	r/w	0

7.2.16.11 **SYSCTRLA->TRIMLED0**

SYSCTRLA	**TRIMLED0**	0x50013028
	High Voltage LED trim	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							A																								

#	Bit(s)	Field	Description	Type	Reset
A	24:16	TRIM0	*LED trim (100uA step)*	r/w	*0x78*

7.2.16.12 **SYSCTRLA->TRIMLED1**

SYSCTRLA	**TRIMLED1**	0x5001302C
	High Voltage LED trim	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							A																								

#	Bit(s)	Field	Description	Type	Reset
A	24:16	TRIM1	*LED trim (100uA step)*	r/w	*0x78*

7.2.16.13 **SYSCTRLA->TRIMLED2**

SYSCTRLA	**TRIMLED2**	0x50013030
	High Voltage LED trim	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							A																								

#	Bit(s)	Field	Description	Type	Reset
A	24:16	TRIM2	*LED trim (100uA step)*	r/w	*0x78*

7.2.16.14 **SYSCTRLA->TRIMLED3**

SYSCTRLA	**TRIMLED3**	0x50013034
	High Voltage LED trim	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							A																								

#	Bit(s)	Field	Description	Type	Reset
A	24:16	TRIM3	*LED trim (100uA step)*	r/w	*0x78*

7.2.16.15 **SYSCTRLA->TRIMLED4**

SYSCTRLA	**TRIMLED4**	0x50013038
	High Voltage LED trim	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							A																								

#	Bit(s)	Field	Description	Type	Reset
A	24:16	TRIM4	*LED trim (100uA step)*	r/w	*0x78*

7.2.16.16 **SYSCTRLA->TRIMLED5**

SYSCTRLA	**TRIMLED5**	0x5001303C
	High Voltage LED trim	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							A																								

#	Bit(s)	Field	Description	Type	Reset
A	24:16	TRIM5	*LED trim (100uA step)*	r/w	*0x78*

7.2.16.17 **SYSCTRLA->TRIMLED6**

SYSCTRLA	**TRIMLED6**	0x50013040
	High Voltage LED trim	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							A																								

#	Bit(s)	Field	Description	Type	Reset
A	24:16	TRIM6	*LED trim (100uA step)*	r/w	*0x78*

7.2.16.18 **SYSCTRLA->TRIMLED7**

SYSCTRLA	**TRIMLED7**	0x50013044
	High Voltage LED trim	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							A																								

#	Bit(s)	Field	Description	Type	Reset
A	24:16	TRIM7	*LED trim (100uA step)*	r/w	*0x78*

7.2.16.19 **SYSCTRLA->TRIMLED8**

SYSCTRLA	**TRIMLED8**	0x50013048
	High Voltage LED trim	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							A																								

#	Bit(s)	Field	Description	Type	Reset
A	24:16	TRIM8	*LED trim (100uA step)*	r/w	*0x78*

7.2.16.20 **SYSCTRLA->TRIMLED9**

SYSCTRLA	**TRIMLED9**	0x5001304C
	High Voltage LED trim	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							A																								

#	Bit(s)	Field	Description	Type	Reset
A	24:16	TRIM9	*LED trim (100uA step)*	r/w	*0x78*

7.2.16.21 **SYSCTRLA->TRIMLED10**

SYSCTRLA	**TRIMLED10**	0x50013050
	High Voltage LED trim	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							A																								

#	Bit(s)	Field	Description	Type	Reset
A	24:16	TRIM10	*LED trim (100uA step)*	r/w	*0x78*

7.2.16.22 **SYSCTRLA->TRIMLED11**

SYSCTRLA	**TRIMLED11**	0x50013054
	High Voltage LED trim	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							A																								

#	Bit(s)	Field	Description	Type	Reset
A	24:16	TRIM11	*LED trim (100uA step)*	r/w	*0x78*

7.2.16.23 **SYSCTRLA->TRIMLED12**

SYSCTRLA	**TRIMLED12**	0x50013058
	High Voltage LED trim	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							A																								

#	Bit(s)	Field	Description	Type	Reset
A	24:16	TRIM12	*LED trim (100uA step)*	r/w	*0x78*

7.2.16.24 **SYSCTRLA->TRIMLED13**

SYSCTRLA	**TRIMLED13**	0x5001305C
	High Voltage LED trim	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							A																								

#	Bit(s)	Field	Description	Type	Reset
A	24:16	TRIM13	*LED trim (100uA step)*	r/w	*0x78*

7.2.16.25 **SYSCTRLA->TRIMLED14**

SYSCTRLA	**TRIMLED14**	0x50013060
	High Voltage LED trim	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							A																								

#	Bit(s)	Field	Description	Type	Reset
A	24:16	TRIM14	*LED trim (100uA step)*	r/w	*0x78*

7.2.16.26 **SYSCTRLA->TRIMLED15**

SYSCTRLA	**TRIMLED15**	0x50013064
	High Voltage LED trim	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							A																								

#	Bit(s)	Field	Description	Type	Reset
A	24:16	TRIM15	*LED trim (100uA step)*	r/w	*0x78*

7.2.16.27 **SYSCTRLA->TRIMLED16**

SYSCTRLA	**TRIMLED16**	0x50013068
	High Voltage LED trim	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							A																								

#	Bit(s)	Field	Description	Type	Reset
A	24:16	TRIM16	*LED trim (100uA step)*	r/w	*0x78*

7.2.16.28 **SYSCTRLA->TRIMLED17**

SYSCTRLA	**TRIMLED17**	0x5001306C
	High Voltage LED trim	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							A																								

#	Bit(s)	Field	Description	Type	Reset
A	24:16	TRIM17	*LED trim (100uA step)*	r/w	*0x78*

7.2.16.29 **SYSCTRLA->TRIMLED18**

SYSCTRLA	**TRIMLED18**	0x50013070
	High Voltage LED trim	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							A																								

#	Bit(s)	Field	Description	Type	Reset
A	24:16	TRIM18	*LED trim (100uA step)*	r/w	*0x78*

7.2.16.30 **SYSCTRLA->TRIMLED19**

SYSCTRLA	**TRIMLED19**	0x50013074
	High Voltage LED trim	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							A																								

#	Bit(s)	Field	Description	Type	Reset
A	24:16	TRIM19	*LED trim (100uA step)*	r/w	*0x78*

7.2.16.31 **SYSCTRLA->TRIMLED20**

SYSCTRLA	**TRIMLED20**	0x50013078
	High Voltage LED trim	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							A																								

#	Bit(s)	Field	Description	Type	Reset
A	24:16	TRIM20	*LED trim (100uA step)*	r/w	*0x78*

7.2.16.32 **SYSCTRLA->TRIMLED21**

SYSCTRLA	**TRIMLED21**	0x5001307C
	High Voltage LED trim	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							A																								

#	Bit(s)	Field	Description	Type	Reset
A	24:16	TRIM21	*LED trim (100uA step)*	r/w	*0x78*

7.2.16.33 **SYSCTRLA->TRIMLED22**

SYSCTRLA	**TRIMLED22**	0x50013080
	High Voltage LED trim	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							A																								

#	Bit(s)	Field	Description	Type	Reset
A	24:16	TRIM22	*LED trim (100uA step)*	r/w	*0x78*

7.2.16.34 SYSCTRLA->TRIMLED23

SYSCTRLA	**TRIMLED23**	0x50013084
	High Voltage LED trim	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							A																								

#	Bit(s)	Field	Description	Type	Reset
A	24:16	TRIM23	*LED trim (100uA step)*	r/w	*0x78*

7.2.16.35 SYSCTRLA->TRIMVFW

SYSCTRLA	**TRIMVFW**	0x50013088
	VFW Current Trim	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								A							

#	Bit(s)	Field	Description	Type	Reset
A	7:0	TRIMVFW	*PN Forward Voltage Current trim (10uA step)*	r/w	*0xC8*

7.2.16.36 SYSCTRLA->LIN

SYSCTRLA	**LIN**	0x5001308C
	LIN IO Control (Trim access need to be enabled before Written)	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																						A	B	C		D		E		F	

#	Bit(s)	Field	Description	Type	Reset
A	9	TXLINMRISESLOPE	*LIN Slave IO Rise Slope select* — read-only, unless 'trim access' is enabled	r/w	*1*
B	8	TXLINSRISESLOPE	*LIN Slave IO Rise Slope select* — read-only, unless 'trim access' is enabled	r/w	*1*
C	7:6	LINMTX_BIAS_BOOST	*LIN Master IO TX Bias select* Select LIN IO TX Pull Down Current 0x**0** — ~61 mA 0x**1** — ~83 mA 0x**2** — ~105 mA 0x**3** — ~126 mA — read-only, unless 'trim access' is enabled	r/w	0
D	5:4	LINSTX_BIAS_BOOST	*LIN Slave IO TX Bias select* Select LIN IO TX Pull Down Current 0x**0** — ~61 mA 0x**1** — ~83 mA 0x**2** — ~105 mA 0x**3** — ~126 mA — read-only, unless 'trim access' is enabled	r/w	0

#	Bit(s)	Field	Description	Type	Reset
E	3:2	TXLINM_DR_SLOPE	*LIN Master IO Drive Slope select* Slew from 40%*Vbat to 60%*Vbat @Vbat=13V,Cbus=1nF 0x**0** — when TXLINM_RISE_SLOPE=0: slewRate_pos = 2.1V/us, slewRate_neg = 2.2V/us 0x**1** — when TXLINM_RISE_SLOPE=0: slewRate_pos = 4.0V/us, slewRate_neg = 4.3V/us 0x**2** — when TXLINM_RISE_SLOPE=0: slewRate_pos = 4.9V/us, slewRate_neg = 6.3V/us 0x**3** — when TXLINM_RISE_SLOPE=0: slewRate_pos = 5.1V/us, slewRate_neg = 30.3V/us 0x**0** — when TXLINM_RISE_SLOPE=1: slewRate_pos = 5.1V/us, slewRate_neg = 2.2V/us 0x**1** — when TXLINM_RISE_SLOPE=1: slewRate_pos = 5.1V/us, slewRate_neg = 4.3V/us 0x**2** — when TXLINM_RISE_SLOPE=1: slewRate_pos = 5.1V/us, slewRate_neg = 6.3V/us 0x**3** — when TXLINM_RISE_SLOPE=1: slewRate_pos = 5.1V/us, slewRate_neg = 30.3V/us — read-only, unless 'trim access' is enabled	r/w	*2*

#	Bit(s)	Field	Description	Type	Reset
F	1:0	TXLINS_DR_SLOPE	*LIN Slave IO Drive Slope select* Slew from 40%*Vbat to 60%*Vbat @Vbat=13V,Cbus=1nF 0x**0** — when TXLINS_RISE_SLOPE=0: slewRate_pos = 2.1V/us, slewRate_neg = 2.2V/us 0x**1** — when TXLINS_RISE_SLOPE=0: slewRate_pos = 4.0V/us, slewRate_neg = 4.3V/us 0x**2** — when TXLINS_RISE_SLOPE=0: slewRate_pos = 4.9V/us, slewRate_neg = 6.3V/us 0x**3** — when TXLINS_RISE_SLOPE=0: slewRate_pos = 5.1V/us, slewRate_neg = 30.3V/us 0x**0** — when TXLINS_RISE_SLOPE=1: slewRate_pos = 5.1V/us, slewRate_neg = 2.2V/us 0x**1** — when TXLINS_RISE_SLOPE=1: slewRate_pos = 5.1V/us, slewRate_neg = 4.3V/us 0x**2** — when TXLINS_RISE_SLOPE=1: slewRate_pos = 5.1V/us, slewRate_neg = 6.3V/us 0x**3** — when TXLINS_RISE_SLOPE=1: slewRate_pos = 5.1V/us, slewRate_neg = 30.3V/us — read-only, unless 'trim access' is enabled	r/w	*2*

7.2.16.37 SYSCTRLA->DFTCODE

SYSCTRLA	**DFTCODE**	0x50013090
	DFT Unlock Code	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																								B							

#	Bit(s)	Field	Description	Type	Reset
A	31	DFT_LOCK	Set Only bit. Write 1 to this bit to lock DFT related config bits	r/w	0
B	7:0	DFTCODE	Test Mode Unlock Enable Code. 0x1C needs to be written to this register to unlock the DFT_TESTMODE_SEL and DFT_TESTMODE_START registers — read-only, unless 'dft_code access' is enabled	*wo*	0

7.2.16.38 SYSCTRLA->DFT_ACCESS_ENABLED

SYSCTRLA	**DFT_ACCESS_ENABLED**	0x50013094
	DFT access enabled	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															A

#	Bit(s)	Field	Description	Type	Reset
A	0	DFT_ACCESS_ENABLED	A status flag that is set when DFT access is enabled	*ro*	0

7.2.16.39 SYSCTRLA->DFTTESTMODESTART

SYSCTRLA	**DFTTESTMODESTART**	0x50013098

	DFT Mode Start

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															A

#	Bit(s)	Field	Description	Type	Reset
A	0	DFTTESTMODESTART	Puts the ASIC into DFT testmode. Once the start bit is set, the I/O configuration will switch from Application mode to DFT Test Mode. The General Purpose I/Os will be configured as a JTAG interface. Once Test Mode is enabled, the ASIC will be boundary terminated and the processor will lose the ability to communicate with any ASIC peripherals. A chip power cycle is required to get out of the DFT test mode. Test Mode Enable state — only accessible when 'dft access' is enabled	*wo*	0

7.2.16.40 SYSCTRLA->NAME

SYSCTRLA	**NAME**	0x5001309C
	ASIC name	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																															

#	Bit(s)	Field	Description	Type	Reset
A	31:0	NAME	*ASIC name* A read from this register will return the ASIC name	*ro*	n/a

7.2.16.41 SYSCTRLA->REV

SYSCTRLA	**REV**	0x500130A0
	Silicon Revision	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																A															

#	Bit(s)	Field	Description	Type	Reset
A	15:0	REV	*Silicon Revision* A read from this register will return the ASCII silicon revision (e.g. ASCII C0 is 0x4330)	*ro*	n/a

7.2.16.42 SYSCTRLA->BORTESTMODE

SYSCTRLA	**BORTESTMODE**	0x500130A4
	BOR Testmode Enable	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															A

#	Bit(s)	Field	Description	Type	Reset
A	0	ENABORTESTMODE	***BOR Testmode Enable*** 0x**0** — BOR Testmode Disabled: Reference Voltage for BOR is from Band Gap (Functional Mode) 0x**1** — BOR Testmode Enabled: Reference Voltage for BOR is from gpio1_anaOut (Test Mode — read-only, unless 'trim access' is enabled	r/w	0

7.2.17 **GPIO**

ASIC GPIO registers

7.2.17.1 **GPIO->GPADATA**

GPIO	**GPADATA**	0x50014000
	GPIO Port A Data	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																								A							

#	Bit(s)	Field	Description	Type	Reset
A	7:0	GPADATA<1024>	*Port A data* To aid in the efficiency of software, the GPIO ports allow for the modification of individual bits in the GPIO Data by using bits [9:2] of the address bus as a enable. In this manner, software drivers can modify individual GPIO pins in a single instruction without affecting the state of the other pins. This method is more efficient than the conventional method of performing a read-modify-write operation to set or clear an individual GPIO pin. To write all the bits at the same time use address offset of 0x07F	*dual*	0

7.2.17.2 **GPIO->GPBDATA**

GPIO	**GPBDATA**	0x50014400
	GPIO Port B Data	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																													A		

#	Bit(s)	Field	Description	Type	Reset
A	2:0	GPBDATA<1024>	*Port B data*	*dual*	*6*

7.2.17.3 **GPIO->GPENA**

GPIO	**GPENA**	0x50014800
	GPIO Port Enables	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																														A	B

#	Bit(s)	Field	Description	Type	Reset
A	1	GPBENA	Enables the Clock-Gate, can be cleared to save power, if none of the GPIO functionality is required	r/w	*1*
B	0	GPAENA	Enables the Clock-Gate, can be cleared to save power, if none of the GPIO functionality is required	r/w	*1*

7.2.17.4 **GPIO->GPAP03**

GPIO	**GPAP03**	0x50014804
	GPIO Port A Pin 0-3 Control	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		A	B	C	D	E	F			G	H	I	J	K	L			M	N	O	P	Q	R			S	T	U	V	W	X

#	Bit(s)	Field	Description	Type	Reset
A	29	GPAACTDET[3]	*Pin 3 activity interrupt*	*ro*	n/a
B	28	GPACLR[3]	*Pin 3 interrupt clear*	*wo*	0

#	Bit(s)	Field	Description	Type	Reset
			– cleared automatically after each write		
C	27	GPAFE[3]	*Pin 3 falling edge enable*	r/w	0
D	26	GPARE[3]	*Pin 3 rising edge enable*	r/w	0
E	25	GPAIE[3]	*Pin 3 interrupt enable*	r/w	0
F	24	GPADIR[3]	*Pin 3 output enable*	r/w	0
G	21	GPAACTDET[2]	*Pin 2 activity interrupt*	*ro*	n/a
H	20	GPACLR[2]	*Pin 2 interrupt clear* – cleared automatically after each write	*wo*	0
I	19	GPAFE[2]	*Pin 2 falling edge enable*	r/w	0
J	18	GPARE[2]	*Pin 2 rising edge enable*	r/w	0
K	17	GPAIE[2]	*Pin 2 interrupt enable*	r/w	0
L	16	GPADIR[2]	*Pin 2 output enable*	r/w	0
M	13	GPAACTDET[1]	*Pin 1 activity interrupt*	*ro*	n/a
N	12	GPACLR[1]	*Pin 1 interrupt clear* – cleared automatically after each write	*wo*	0
O	11	GPAFE[1]	*Pin 1 falling edge enable*	r/w	0
P	10	GPARE[1]	*Pin 1 rising edge enable*	r/w	0
Q	9	GPAIE[1]	*Pin 1 interrupt enable*	r/w	0
R	8	GPADIR[1]	*Pin 1 output enable*	r/w	0

#	Bit(s)	Field	Description	Type	Reset
S	5	GPAACTDET[0]	*Pin 0 activity interrupt*	*ro*	n/a
T	4	GPACLR[0]	*Pin 0 interrupt clear* – cleared automatically after each write	*wo*	0
U	3	GPAFE[0]	*Pin 0 falling edge enable*	r/w	0
V	2	GPARE[0]	*Pin 0 rising edge enable*	r/w	0
W	1	GPAIE[0]	*Pin 0 interrupt enable*	r/w	0
X	0	GPADIR[0]	*Pin 0 output enable*	r/w	0

7.2.17.5 **GPIO->GPAP47**

GPIO	**GPAP47**	0x50014808
	GPIO Port A Pin 4-7 Control	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		A	B	C	D	E	F			G	H	I	J	K	L			M	N	O	P	Q	R			S	T	U	V	W	X

#	Bit(s)	Field	Description	Type	Reset
A	29	GPAACTDET[7]	*Pin 7 activity interrupt*	*ro*	n/a
B	28	GPACLR[7]	*Pin 7 interrupt clear* – cleared automatically after each write	*wo*	0
C	27	GPAFE[7]	*Pin 7 falling edge enable*	r/w	0
D	26	GPARE[7]	*Pin 7 rising edge enable*	r/w	0
E	25	GPAIE[7]	*Pin 7 interrupt enable*	r/w	0

#	Bit(s)	Field	Description	Type	Reset
F	24	GPADIR[7]	*Pin 7 output enable*	r/w	0
G	21	GPAACTDET[6]	*Pin 6 activity interrupt*	*ro*	n/a
H	20	GPACLR[6]	*Pin 6 interrupt clear* – cleared automatically after each write	*wo*	0
I	19	GPAFE[6]	*Pin 6 falling edge enable*	r/w	0
J	18	GPARE[6]	*Pin 6 rising edge enable*	r/w	0
K	17	GPAIE[6]	*Pin 6 interrupt enable*	r/w	0
L	16	GPADIR[6]	*Pin 6 output enable*	r/w	0
M	13	GPAACTDET[5]	*Pin 5 activity interrupt*	*ro*	n/a
N	12	GPACLR[5]	*Pin 5 interrupt clear* – cleared automatically after each write	*wo*	0
O	11	GPAFE[5]	*Pin 5 falling edge enable*	r/w	0
P	10	GPARE[5]	*Pin 5 rising edge enable*	r/w	0
Q	9	GPAIE[5]	*Pin 5 interrupt enable*	r/w	0
R	8	GPADIR[5]	*Pin 5 output enable*	r/w	0
S	5	GPAACTDET[4]	*Pin 4 activity interrupt*	*ro*	n/a
T	4	GPACLR[4]	*Pin 4 interrupt clear* – cleared automatically after each write	*wo*	0
U	3	GPAFE[4]	*Pin 4 falling edge enable*	r/w	0

#	Bit(s)	Field	Description	Type	Reset
V	2	GPARE[4]	*Pin 4 rising edge enable*	r/w	0
W	1	GPAIE[4]	*Pin 4 interrupt enable*	r/w	0
X	0	GPADIR[4]	*Pin 4 output enable*	r/w	0

7.2.17.6 **GPIO->GPBP02**

GPIO	**GPBP02**	0x5001480C
	GPIO Port B Pin 0-2 Control	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										A	B	C	D	E	F			G	H	I	J	K	L			M	N	O	P	Q	R

#	Bit(s)	Field	Description	Type	Reset
A	21	GPBACTDET[2]	*Pin 2 activity interrupt*	*ro*	n/a
B	20	GPBCLR[2]	*Pin 2 interrupt clear* – cleared automatically after each write	*wo*	0
C	19	GPBFE[2]	*Pin 2 falling edge enable*	r/w	0
D	18	GPBRE[2]	*Pin 2 rising edge enable*	r/w	0
E	17	GPBIE[2]	*Pin 2 interrupt enable*	r/w	0
F	16	GPBDIR[2]	*Pin 2 output enable*	r/w	0
G	13	GPBACTDET[1]	*Pin 1 activity interrupt*	*ro*	n/a
H	12	GPBCLR[1]	*Pin 1 interrupt clear* – cleared automatically after each write	*wo*	0

#	Bit(s)	Field	Description	Type	Reset
I	11	GPBFE[1]	*Pin 1 falling edge enable*	r/w	0
J	10	GPBRE[1]	*Pin 1 rising edge enable*	r/w	0
K	9	GPBIE[1]	*Pin 1 interrupt enable*	r/w	0
L	8	GPBDIR[1]	*Pin 1 output enable*	r/w	0
M	5	GPBACTDET[0]	*Pin 0 activity interrupt*	*ro*	n/a
N	4	GPBCLR[0]	*Pin 0 interrupt clear* – cleared automatically after each write	*wo*	0
O	3	GPBFE[0]	*Pin 0 falling edge enable*	r/w	0
P	2	GPBRE[0]	*Pin 0 rising edge enable*	r/w	0
Q	1	GPBIE[0]	*Pin 0 interrupt enable*	r/w	0
R	0	GPBDIR[0]	*Pin 0 output enable*	r/w	0

7.2.18 **BUCK_TRIM**

BUCK Trim Memory

7.2.18.1 **BUCK_TRIM->DATA0**

BUCK_TRIM	**DATA0**	0x50018000
	BUCK Trim Memory Data0 Register contains trim register bits for the 5V BUCK Regulator. The TRIMKEY in SYSCTRLA needs to be set to 0x1D155AFE before any access(Read & write) to these trim registers	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A	B	C	D	E	F	G	H	I								J	K	L				M	N	O	P	Q	R	S	T	U	V

#	Bit(s)	Field	Description	Type	Reset
A	31	FRCDPWM	*forced_pwm* forcedpwm	r/w	*1*
B	30	MNGO2PWMMASK	*db_main_go2pwm_masked* mask go2pwm events	r/w	0
C	29	MNGO2PWMFRCD	*db_main_go2pwm_forced* force a go2pwm event	r/w	0
D	28	MNGO2PFMMASK	*db_main_go2pfm_masked* mask go2pfm events	r/w	0
E	27	MNGO2PFMFRCD	*db_main_go2pfm_forced* force a go2pfm event	r/w	0
F	26	MNGATECLKHF	*db_main_gate_clk_hf* gate main clk	r/w	0
G	25	MNBYPFSTOP	*db_main_bypass_force_stop* ignore counter bypass or comp bypas (not in sawchuk)	r/w	*1*
H	24	CLKOVERRIGHT	*db_clk_overright*	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			clk will control hs_on and ls_on bypassing the carousel		
I	23:16	CRDLYTMOUTIGN	*db_carou_dly_timeout_ignored* Delay Timeout Ignored	r/w	0
J	15	FMAGRQST	*db_force_mag_rqst* force mag_rqst HIGH	r/w	0
K	14	FDMGRQST	*db_force_demag_rqst* force demag_rqst HIGH	r/w	0
L	13:10	CRPAUSE	*db_carou_pause* use bits 3:0 to select where the carou will pause	r/w	0
M	9	CRLSONMSK	*db_carou_ls_on_mask* force ls_on low	r/w	0
N	8	CRMAGMASK	*db_carou_mag_masked* mag_rqst masked for carou	r/w	0
O	7	CRLSONFRCD	*db_carou_ls_on_forced* force ls_on HIGH	r/w	0
P	6	CRHSONMSK	*db_carou_hs_on_mask*	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			force hs_on low		
Q	5	CRHSONFRCD	*db_carou_hs_on_forced* force hs_on HIGH, ONLY take effect in DEBUG MODE	r/w	0
R	4	CRFLGLSONDNTWT	*db_carou_flag_ls_on_dont_wait* do not wait for flag to proceed	r/w	0
S	3	CRFLGLSOFFDNTWT	*db_carou_flag_ls_off_dont_wait* do not wait for flag to proceed	r/w	*1*
T	2	CRFLGHSONDNTWT	*db_carou_flag_hs_on_dont_wait* do not wait for flag to proceed	r/w	0
U	1	CRFLGHSOFFDNTWT	*db_carou_flag_hs_off_dont_wait* do not wait for flag to proceed	r/w	*1*
V	0	CRDMGMASK	*db_carou_demag_masked* demag_rqst masked for carou	r/w	0

7.2.18.2 **BUCK_TRIM->DATA1**

BUCK_TRIM	**DATA1**	0x50018004
	BUCK Trim Memory Data1	

	Register contains trim register bits for the 5V BUCK Regulator. The TRIMKEY in SYSCTRLA needs to be set to 0x1D155AFE before any access(Read & write) to these trim registers

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A	B	C	D					E	F	G	H					I	J	K	L					M	N	O	P	Q			

#	Bit(s)	Field	Description	Type	Reset
A	31	SPAREU0W7B7	*spareu0_w7_b7* spare	r/w	0
B	30	SPAREU0W7B6	*spareu0_w7_b6* spare	r/w	*1*
C	29	SPAREU0W7B5	*spareu0_w7_b5* spare	r/w	0
D	28:24	DTB03SEL	***dtb03_sel*** 0x**0** — main_statemachine_pfm 0x**1** — state_eventctrl_mag2off 0x**2** — state_eventctrl_invalid01 0x**3** — state_eventctrl_wait1 0x**4** — state_eventctrl_wait2off 0x**5** — state_eventctrl_wait2 0x**6** — buck_en 0x**7** — buck_bypass	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			0x**8** — buck_start 0x**9** — cycle_ready 0x**A** — event_ctrl_demag_start 0x**B** — event_ctrl_demag_stop 0x**C** — event_ctrl_mag_start 0x**D** — state_carou_start_demag_sensors 0x**E** — state_carou_start_mag_sensors 0x**F** — state_carou_stop_demag 0x**10** — dly_timeout_01 0x**11** — dly_timeout_02 0x**12** — dly_timeout_03 0x**13** — dly_timeout_04 0x**14** — vfly_valid 0x**15** — flag_hs_on 0x**16** — flag_ls_on 0x**17** — hs_on_int_01 0x**18** — ls_on_int_01 0x**19** — pwm_start_demag_event_03 0x**1A** — flag_hs_on 0x**1B** — flag_ls_on 0x**1C** — mag_rqst		

#	Bit(s)	Field	Description	Type	Reset
			0x**1D** — demag_rqst 0x**1E** — tie_lo 0x**1F** — pwm_start_demag_event_02		
E	23	SPAREU0W6B7	*spareu0_w6_b7* spare	r/w	0
F	22	ATBON02DIRECT	*atb_on_02_direct* activate atb direct path 02	r/w	0
G	21	ATBON02BUF	*atb_on_02_buf* activate atb buffered path 02	r/w	0
H	20:16	DTB02SEL	***dtb02_sel*** 0x**0** — overcurrent_detection 0x**1** — main_statemachine_pwm2pfm 0x**2** — main_statemachine_pfm2pwm 0x**3** — state_eventctrl_offstate 0x**4** — state_eventctrl_demag_rqst 0x**5** — rst_n 0x**6** — buck_clk_500k 0x**7** — buck_forced_pfm 0x**8** — buck_forced_pwm 0x**9** — mag_rqst	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			0x**A** — dly_start_01 0x**B** — dly_start_02 0x**C** — dly_start_03 0x**D** — dly_start_04 0x**E** — clk_out_hf_gated 0x**F** — pwm_start_demag_event_unmask_01 0x**10** — pwm_start_mag_event_unmask_01 0x**11** — pwm_start_demag_event_unmask_01 0x**12** — pfm_start_mag_event_unmask_01 0x**13** — pfm_stop_demag_event_unmask_01 0x**14** — state_carou_demag 0x**15** — state_carou_mag 0x**16** — state_carou_stop_mag 0x**17** — state_carou_stop_demag_sensors 0x**18** — demag_rqst 0x**19** — flag_hs_on 0x**1A** — flag_ls_on 0x**1B** — tie_hi		

#	Bit(s)	Field	Description	Type	Reset
			0x**1C** — tie_lo 0x**1D** — tie_hi 0x**1E** — tie_lo 0x**1F** — tie_hi		
I	15	ATBON01DIRECT	*atb_on_01_direct* activate atb direct path 01	r/w	0
J	14	ATBON01BUF	*atb_on_01_buf* activate atb buffered path 01	r/w	0
K	13	DTBON	*dtb_on* activate dtb output	r/w	0
L	12:8	DTB01SEL	***dtb01_sel*** 0x**0** — main_statemachine_off 0x**1** — main_statemachine_off2pwm_bypass 0x**2** — main_statemachine_invalid02 0x**3** — main_statemachine_invalid03 0x**4** — main_statemachine_bypass2off 0x**5** — main_statemachine_pwm 0x**6** — main_statemachine_pwm2off2 0x**7** — main_statemachine_pfm2off2	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			0x**8** — main_statemachine_invalid7 0x**9** — main_statemachine_invalid12 0x**A** — main_statemachine_pwm2off 0x**B** — main_statemachine_pfm2off 0x**C** — main_statemachine_invalid15 0x**D** — state_eventctrl_mag_rqst 0x**E** — clk_out_hf 0x**F** — pwm_start_mag_event_01 0x**10** — pwm_start_demag_event_01 0x**11** — pfm_start_mag_event_01 0x**12** — pfm_start_demag_event_01 0x**13** — pfm_stop_demag_event_01 0x**14** — overvoltage_event 0x**15** — state_carou_off 0x**16** — state_carou_invalid_01 0x**17** — state_carou_invalid_02 0x**18** — state_carou_invalid_03 0x**19** — state_carou_invalid_04 0x**1A** — state_carou_invalid_05 0x**1B** — state_carou_start_demag 0x**1C** — state_carou_start_mag		

#	Bit(s)	Field	Description	Type	Reset
			0x**1D** — state_carou_stop_mag_sensors 0x**1E** — hs_on_01 0x**1F** — ls_on_01		
M	7	SPAREU0W4B7	*spareu0_w4_b7* spare	r/w	*1*
N	6	SPAREU0W4B6	*spareu0_w4_b6* spare	r/w	0
O	5	SPAREU0W4B5	*spareu0_w4_b5* spare	r/w	0
P	4	FRCDPFM	*forced_pfm* forcedpfm	r/w	0
Q	3:0	MNDLYTMOUTIGN	*db_main_dly_timeout_ignored* not in use	r/w	0

7.2.18.3 BUCK_TRIM->DATA2

BUCK_TRIM	**DATA2**	0x50018008
	BUCK Trim Memory Data2	

	Register contains trim register bits for the 5V BUCK Regulator. The TRIMKEY in SYSCTRLA needs to be set to 0x1D155AFE before any access(Read & write) to these trim registers

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	a	b	c	d	e	f

#	Bit(s)	Field	Description	Type	Reset
A	31	FOENIRAMPCOMP	*db_force_off_en_iramp_compensation*	r/w	0
B	30	ENATB2IRAMPCOMP	*en_atb_02_iramp_comp*	r/w	0
C	29	ENATB1IRAMPCOMP	*en_atb_01_iramp_comp*	r/w	0
D	28	FOENOC	*db_force_off_en_overcurrent*	r/w	0
E	27	ENATB2ILIM	*en_atb_02_ilim*	r/w	0
F	26	ENATB1ILIM	*en_atb_01_ilim*	r/w	0
G	25	ENATB2ILOOP	*en_atb_02_current_loop*	r/w	0
H	24	ENATB1ILOOP	*en_atb_01_current_loop*	r/w	0
I	23	PHCOMP4	*placeholder_comp04*	r/w	0
J	22	PHCOMP3	*placeholder_comp03*	r/w	0
K	21	PHCOMP2	*placeholder_comp02*	r/w	0
L	20	PHCOMP1	*placeholder_comp01*	r/w	0
M	19	ENATB2V2I	*en_atb_02_v2i*	r/w	0
N	18	ENATB1V2I	*en_atb_01_v2i*	r/w	0
O	17	ENATB2IBIAS	*en_atb_02_ibias*	r/w	0
P	16	ENATB1IBIAS	*en_atb_01_ibias*	r/w	0

#	Bit(s)	Field	Description	Type	Reset
Q	15	SPAREU1W1B7	*spareU1_W1_b7*	r/w	0
R	14	PWMSTDMGEVTAIU1	*db_pwm_start_demag_event_alwaysinuse_01*	r/w	0
S	13	PWMSTDMGEVTIGN2	*db_pwm_start_demag_event_ignore_02*	r/w	0
T	12	PWMSTDMGEVTIGN1	*db_pwm_start_demag_event_ignore_01*	r/w	0
U	11	PWMSTMAGEVTIGN2	*db_pwm_start_mag_event_ignore_02*	r/w	*1*
V	10	PWMSTMAGEVTIGN1	*db_pwm_start_mag_event_ignore_01*	r/w	0
W	9	PWMSTMAGEVTAIU2	*db_pwm_start_mag_event_alwaysinuse_02*	r/w	0
X	8	PWMSTMAGEVTAIU1	*db_pwm_start_mag_event_alwaysinuse_01*	r/w	0
Y	7	PFMSPDMGEVTIGN1	*db_pfm_stop_demag_event_ignore_01*	r/w	0
Z	6	PFMSPDMGEVTAIU1	*db_pfm_stop_demag_event_alwaysinuse_01*	r/w	0
a	5	PFMSTDMGEVTIGN1	*db_pfm_start_demag_event_ignore_01*	r/w	0
b	4	PFMSTDMGEVTAIU1	*db_pfm_start_demag_event_alwaysinuse_01*	r/w	0
c	3	PFMSTMAGEVTIGN2	*db_pfm_start_mag_event_ignore_02*	r/w	*1*
d	2	PFMSTMAGEVTIGN1	*db_pfm_start_mag_event_ignore_01*	r/w	0
e	1	PFMSTMAGEVTAIU2	*db_pfm_start_mag_event_alwaysinuse_02*	r/w	0
f	0	PFMSTMAGEVTAIU1	*db_pfm_start_mag_event_alwaysinuse_01*	r/w	0

7.2.18.4 **BUCK_TRIM->DATA3**

BUCK_TRIM	**DATA3**	0x5001800C
	BUCK Trim Memory Data3	

	Register contains trim register bits for the 5V BUCK Regulator. The TRIMKEY in SYSCTRLA needs to be set to 0x1D155AFE before any access(Read & write) to these trim registers

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A						B	C	D	E	F	G	H		I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X		

#	Bit(s)	Field	Description	Type	Reset
A	31:26	SPAREU1W7	*spareU1_W7*	r/w	0
B	25	DBENDEMAGCOMP02	*db_en_demag_comp02*	r/w	0
C	24	DBENDEMAGCOMP01	*db_en_demag_comp01*	r/w	0
D	23	SPAREU1W6B7	*spareU1_W6_b7*	r/w	0
E	22	FOENPFM	*db_force_off_en_pfm*	r/w	0
F	21	FOIOSISENSE	*db_force_off_ioffset_isense*	r/w	0
G	20	FOENISENSE	*db_force_off_en_isense*	r/w	0
H	19:18	SELILOADOSISENSE	*sel_iload_os_isense*	r/w	*2*
I	17	ENATB2FGND	*en_atb_02_fgnd*	r/w	*1*
J	16	ENATB1FGND	*en_atb_01_fgnd*	r/w	0
K	15	SPAREU1W5B7	*spareU1_W5_b7*	r/w	0
L	14	RELPDFRMLDO5V	*db_release_pd_from_ldo5V*	r/w	0
M	13	ENATB2ANALV	*en_atb_02_analog_lv*	r/w	0
N	12	ENATB1ANALV	*en_atb_01_analog_lv*	r/w	0
O	11	FENVEACLMPUP	*db_force_en_vea_clamp_up*	r/w	0
P	10	FENVEACLMPDN	*db_force_en_vea_clamp_down*	r/w	0

#	Bit(s)	Field	Description	Type	Reset
Q	9	FOEA	*db_force_off_ea*	r/w	0
R	8	FORBIDRLDRPC	*db_forbid_rladder_precharge*	r/w	0
S	7	FOVDAC	*db_force_off_vdac*	r/w	0
T	6	ENATB2VDAC	*en_atb_02_vdac*	r/w	0
U	5	ENATB1VDAC	*en_atb_01_vdac*	r/w	0
V	4	ENATB2VLOOP	*en_atb_02_voltage_loop*	r/w	0
W	3	ENATB1VLOOP	*en_atb_01_voltage_loop*	r/w	0
X	2:0	ENATBISENS	*en_atb_isense*	r/w	0

7.2.18.5 **BUCK_TRIM->DATA4**

BUCK_TRIM	**DATA4**	0x50018010
	BUCK Trim Memory Data4 Register contains trim register bits for the 5V BUCK Regulator. The TRIMKEY in SYSCTRLA needs to be set to 0x1D155AFE before any access(Read & write) to these trim registers	

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
A	B		C			D										E				F				G	H	I	J	K		L	

#	Bit(s)	Field	Description	Type	Reset
A	31	SPAREU2W3B7	*spareU2_W3_b7* spare	r/w	0
B	30:29	SETCOMP	*set_compensation*	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			note in use		
C	28:26	SETIRAMPCOMP	*set_iramp_compensation* If Ksense=10800 then Iramp=Sa/Ksense and 0x**0** — 11.57 A/s = 11.57 µA/us 0x**1** — 23.14 A/s = 23.14 µA/us 0x**2** — 46.29 A/s = 46.29 µA/us 0x**3** — 69.44 A/s = 69.44 µA/us 0x**4** — bit 2 is not in use 0x**5** — bit 2 is not in use. 0x**6** — bit 2 is not in use. 0x**7** — bit 2 is not in use.	r/w	*3*
D	25:24	SETIOS	*set_ioffset* **TBA** 0x**0** — 88.62uA 0x**1** — 105.5uA 0x**2** — 122.4uA 0x**3** — 139.3uA	r/w	*3*
E	15:12	SETILIM	*set_ilim* **TBA** 0x**0** — TBA 0x**1** — TBA 0x**2** — TBA	r/w	*0xA*

#	Bit(s)	Field	Description	Type	Reset
			0x**3** — TBA 0x**4** — TBA 0x**5** — TBA 0x**6** — TBA 0x**7** — TBA 0x**8** — TBA 0x**9** — TBA 0x**A** — TBA 0x**B** — TBA 0x**C** — TBA 0x**D** — TBA 0x**E** — TBA 0x**F** — TBA		
F	11:8	TRIMILIM	*trim_ilim* **TBA** 0x**0** — TBA 0x**1** — TBA 0x**2** — TBA 0x**3** — TBA 0x**4** — TBA 0x**5** — TBA	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			0x**6** — TBA 0x**7** — TBA 0x**8** — TBA 0x**9** — TBA 0x**A** — TBA 0x**B** — TBA 0x**C** — TBA 0x**D** — TBA 0x**E** — TBA 0x**F** — TBA		
G	7	SPAREU2W0B7	*spareU2_W0_b7* spare	r/w	0
H	6	SPAREU2W0B6	*spareU2_W0_b6* spare	r/w	0
I	5	SPAREU2W0B5	*spareU2_W0_b5* spare	r/w	0
J	4	SPAREU2W0B4	*spareU2_W0_b4* spare	r/w	0
K	3:2	SETVEACLMPUP	*set_vea_clamp_up* **Trig point when iout=1uA**	r/w	*2*

#	Bit(s)	Field	Description	Type	Reset
			0x**0** — 2.4V 0x**1** — 2.3V 0x**2** — 2.2V 0x**3** — 2.1V		
L	1:0	SETVEACLMPDN	*set_vea_clamp_down* **Trig point when iout=1uA** 0x**0** — 400mV 0x**1** — 500mV 0x**2** — 600mV 0x**3** — 700mV	r/w	*1*

7.2.18.6 **BUCK_TRIM->DATA5**

BUCK_TRIM	**DATA5**	0x50018014
	BUCK Trim Memory Data5 Register contains trim register bits for the 5V BUCK Regulator. The TRIMKEY in SYSCTRLA needs to be set to 0x1D155AFE before any access(Read & write) to these trim registers	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A		B		C		D		E		F		G		H		I				J		K	L	M	N		O	P		Q	

#	Bit(s)	Field	Description	Type	Reset
A	31:30	CRDLYSET8	*carou_dly_set_08* **min_off_time**	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			0x**0** — 40ns 0x**1** — 60ns 0x**2** — 80ns 0x**3** — 100ns		
B	29:28	CRDLYSET7	*carou_dly_set_07* **start_iramp_compensation** 0x**0** — 600ns 0x**1** — 700ns 0x**2** — 800ns 0x**3** — 900ns	r/w	0
C	27:26	CRDLYSET6	*carou_dly_set_06* **start_mag -> unmask_event_comp** 0x**0** — 80ns 0x**1** — 100ns 0x**2** — 120ns 0x**3** — 140ns	r/w	0
D	25:24	CRDLYSET5	*carou_dly_set_05* pfm only (stop_demag -> start_mag) - not in use for now. one code - 200ns	r/w	0
E	23:22	CRDLYSET4	*carou_dly_set_04*	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			stop_mag_sensors -> stop_mag one code - 20ns		
F	21:20	CRDLYSET3	*carou_dly_set_03* **mag -> start_mag_sensors** 0x**0** — 20ns 0x**1** — 40ns 0x**2** — 60ns 0x**3** — 80ns	r/w	0
G	19:18	CRDLYSET2	*carou_dly_set_02* stop_demag_sensors -> stop_demag one code - 20ns	r/w	0
H	17:16	CRDLYSET1	*carou_dly_set_01* demag -> demag_sensors one code - 20ns	r/w	0
I	15:12	SPAREU2W5	*spareU2_W5* spare	r/w	0
J	11:10	PFMIPEAK	*pfm_ipeak* **TBA** 0x**0** — 650mA 0x**1** — 700mA 0x**2** — 750mA	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			0x**3** — 800mA		
K	9	ENPDGCOMP	*en_pgooddown_comparator* TBA	r/w	0
L	8	ENPUGCOMP	*en_pgoodup_comparator* TBA	r/w	0
M	7	FOENIOS	*db_force_off_en_ioffset* spare	r/w	0
N	6:5	SETFGNDIDAC	*set_fgnd_idac*	r/w	*1*
O	4	TRIMISENSEIOS	*trim_isense_ioffset*	r/w	0
P	3:2	SELISENSEOSISENSE	*sel_isense_os_isense*	r/w	*3*
Q	1:0	SETRI2V	*set_ri2v*	r/w	*1*

7.2.19 **TIMER0**

7.2.19.1 **TIMER0->COUNT**

TIMER0	**COUNT**	0x50020000
	Timer Counter Register	

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
A																															

#	Bit(s)	Field	Description	Type	Reset
A	31:0	COUNT	Initial counter value. The timer will count from this value to 0xFFFFFFFF and roll over to 0x00000000. At this point it will generate an interrupt if enabled. The interrupt routine is responsible for reloading the value if needed as this timer does not auto-reload the original content	r/w	0

7.2.19.2 TIMER0->CFG

TIMER0	**CFG**	0x50020004
	Timer Control Register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															A

#	Bit(s)	Field	Description	Type	Reset
A	0	ENA	*Enable* This bit starts/stops the timer: 1 = Timer Running 0 = Timer Inactive	r/w	0

7.2.20 TIMER1

7.2.20.1 TIMER1->COUNT

TIMER1	**COUNT**	0x50020008
	Timer Counter Register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																															

#	Bit(s)	Field	Description	Type	Reset
A	31:0	COUNT	Initial counter value. The timer will count from this value to 0xFFFFFFFF and roll over to 0x00000000. At this point it will generate an interrupt if enabled. The interrupt routine is responsible for reloading the value if needed as this timer does not auto-reload the original content	r/w	0

7.2.20.2 TIMER1->CFG

TIMER1	**CFG**	0x5002000C
	Timer Control Register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															A

#	Bit(s)	Field	Description	Type	Reset
A	0	ENA	*Enable* This bit starts/stops the timer: 1 = Timer Running 0 = Timer Inactive	r/w	0

7.2.21 TIMER2

7.2.21.1 TIMER2->COUNT

TIMER2	**COUNT**	0x50020010

	Timer Counter Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																															

#	Bit(s)	Field	Description	Type	Reset
A	31:0	COUNT	Initial counter value. The timer will count from this value to 0xFFFFFFFF and roll over to 0x00000000. At this point it will generate an interrupt if enabled. The interrupt routine is responsible for reloading the value if needed as this timer does not auto-reload the original content	r/w	0

7.2.21.2 TIMER2->CFG

TIMER2	**CFG**	0x50020014
	Timer Control Register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																															A

#	Bit(s)	Field	Description	Type	Reset
A	0	ENA	*Enable* This bit starts/stops the timer: 1 = Timer Running 0 = Timer Inactive	r/w	0

7.2.22 WDT1

Watchdog timer

7.2.22.1 **WDT1->CFG**

WDT1	**CFG**	0x50020018
	Config	

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
																											A		B	C	D

#	**Bit(s)**	**Field**	**Description**	**Type**	**Reset**
A	4:3	PRESET	Defines the watchdog timeout period. It means that the WDT internal counter will count from 0 to the prescaler value at the system clock speed and trigger if not cleared. For instance, a system running from a 30MHz Crystal with WDTPRES[1I0] = 10 will trigger the WDT after approximately 0.14 seconds if not cleared properly and in time by the application 0x**0** — 2^{13} / System Clock 0x**1** — 2^{19} / System Clock 0x**2** — 2^{22} / System Clock 0x**3** — 2^{32} / System Clock	r/w	0
B	2	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 cleared by the application	r/w	0
C	1	RSTEN	*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	r/w	0

#	Bit(s)	Field	Description	Type	Reset
D	0	ENA	*WDT Enable* This bit can be asserted but it cannot be de-asserted. It means that once the WDT is enabled it cannot be turned off until a Reset or Power-On Reset occurs	r/w	0

7.2.22.2 **WDT1->KEY**

WDT1	**KEY**	0x5002001C
	Writing the sequence CLEAR, KEY0, KEY1 to this register will clear (pet) the watchdog - preventing it from timing out and resetting the system.	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																															

#	Bit(s)	Field	Description	Type	Reset
A	31:0	KEY	To clear the WDT counting the following words must be written in this order and without any other instruction between then: 0x3C570001 0x007F4AD6	r/w	0

7.2.23 FLASH

7.2.23.1 **FLASH->FLADDR**

FLASH	**FLADDR**	0x50020020
	Destination address for flash write / erase operation	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															A																

#	Bit(s)	Field	Description	Type	Reset
A	16:0	ADDR	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 sector to be erased. This register must be written in the correct sequence or the operation will fail	r/w	*0xFFFF*

7.2.23.2 **FLASH->FLWRDT**

FLASH	**FLWRDT**	0x50020024
	Flash data to be written	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																															

#	Bit(s)	Field	Description	Type	Reset
A	31:0	DATA	Content to be written into the targeted address. This register must be written in the correct sequence or the operation will fail	r/w	0

7.2.23.3 **FLASH->UNLBWR**

FLASH	**UNLBWR**	0x50020028
	Flash data unlock register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																															

#	Bit(s)	Field	Description	Type	Reset
A	31:0	UNLOCK_WRITE	Control register to unlock write. A value of 0x55555555 must be written to this address at the	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			correct point in the write sequence or the operation will fail		

7.2.23.4 **FLASH->BWRSTRT**

FLASH	**BWRSTRT**	0x5002002C
	Flash write start register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																															

#	Bit(s)	Field	Description	Type	Reset
A	31:0	WRITE_START	Control register to start a write. A value of 0xAAAAAAAA must be written to this address at the correct point in the write sequence or the operation will fail	r/w	0

7.2.23.5 **FLASH->UNLSER**

FLASH	**UNLSER**	0x50020030
	Flash sector erase unlock register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																															

#	Bit(s)	Field	Description	Type	Reset
A	31:0	UNLOCK_ERASE	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	r/w	0

7.2.23.6 **FLASH->SERSTRT**

FLASH	**SERSTRT**	0x50020034

	Flash sector erase start register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																															

#	Bit(s)	Field	Description	Type	Reset
A	31:0	ERASE_START	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	r/w	0

7.2.23.7 **FLASH->FLSCTRL**

FLASH	**FLSCTRL**	0x50020040
	Flash control register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																														A	

#	Bit(s)	Field	Description	Type	Reset
A	1:0	CTRL	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	r/w	*1*

7.2.23.8 **FLASH->FLSCP**

FLASH	**FLSCP**	0x50020044
	Flash code protection register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																															

#	Bit(s)	Field	Description	Type	Reset
A	31:0	CODE_PROT	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. NOTE1: This register does not lock the Flash Memory against read/write/erase by the applications program. Instead what it does is to disable 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 the system 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 enable the debug communication	r/w	0

7.2.23.9 FLASH->FLS_UNLOCK_CTRL_OP

FLASH	**FLS_UNLOCK_CTRL_OP**	0x50020050
	Flash Unlock Control Operation Register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A																															

#	Bit(s)	Field	Description	Type	Reset
A	31:0	UNLOCK_CTRL_OP	*Flash Control Operation Register Unlock value*	r/w	0

#	Bit(s)	Field	Description	Type	Reset
			0xACDC_1972 needs to be written in this register to unlock the Control Operation Register access. When this register is read, it returns the state of the lock: 0: The Control Operation Register is locked. The Control Operation Register (FLASH_CTRL_OP) cannot be written. 1: The Control Operation Register is unlocked. The Control Operation Register (FLASH_CTRL_OP) can be written. Note: After each write to the FLASH_CTRL_OP register, the state of the lock is cleared and the pattern needs to be written again to allow a new configuration of the register		

7.2.23.10 **FLASH->CTRL_OP**

FLASH	**CTRL_OP**	0x50020054
	Flash Control Operation Register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																													A		B

#	Bit(s)	Field	Description	Type	Reset
A	2:1	SIZE	*SIZE of the write operation* Refer to data sheet for more information of the use of this field	r/w	0

#	Bit(s)	Field	Description	Type	Reset
B	0	CHIP	*CHIP bit* This bit is only used during the Erase operation. It allows the system to erase more than one sector. 0: The Erase operation will only erase the sector selected by the FLASH_ADDR register value. 1: The Erase operation will erase the full main array of the flash	r/w	0

7.2.23.11 **FLASH->TRIM**

FLASH	**TRIM**	0x50020058
	Flash Trim Register	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														A	B	C															

#	Bit(s)	Field	Description	Type	Reset
A	17	SLEEPDEEP_CFG	*Deep Sleep VDD_IO configuration* This register will be automatically populated with the value stored in the NVR sector 1 (@0001_0000). When set, the system will NOT be reset if VDD_IO is going away during Deep Sleep mode. Otherwise (0), the system is reset if VDD_IO is removed	r/w	0
B	16	SDIO_TIMING_CFG	*SDIO interface timing configuration* This register will be automatically populated with the value stored in the NVR sector 1	r/w	*1*

#	Bit(s)	Field	Description	Type	Reset
			(@0001_0000). When set, the SDIO/INT signals are captured on the rising edge of CLK. When cleared, these data are captured on the falling edge of CLK		
C	15:0	OSC_TRIM	*Oscillator Trim Value* This register will be automatically populated with the value stored in the NVR sector 1 (@0001_0000)	r/w	*0x86*

8 Electrical Characteristics

8.1 Absolute Maximum Ratings

Table 4 Absolute Maximum Ratings, Voltages Referenced to ground

Names	Conditions	Min.	Max.	Unit
VBAT/VBAT PO	No damage, t<500ms	-0.3	+45	V
VBAT/VBAT PO	No damage, t<5min	-0.3	+28	V
VBAT/VBAT PO	No damage, t<5ms	-1.1		V
VBAT/VBAT PO	No damage, t<20ns	-4.0		V
VBAT/VBAT PO	No damage, ISO 7637-2 pulse 1, VBAT=13.5V, TA=23°+/-5C, test pulse applied to VBAT via reverse polarity diode and more than 4.7uF capacitor	-100		V
VBAT/VBAT PO	No damage, ISO 7637-2 pulse 2 VBAT=13.5V, TA=23°+/-5C, test pulse applied to VBAT via reverse polarity diode and more than 4.7uF capacitor		+50	V
VBAT/VBAT PO	No damage, accept ISO 7637-2 pulses 3A, 3B, VBAT=13.5V, TA=(23+/-5)°C, test pulse applied to VBAT via reverse polarity diode and more than 4.7uF capacitor	-150	+100	V
VBAT/VBAT PO	No damage, ISO 7637-2 pulses 5b VBAT=13.5V, TA=(23+/-5)°C, test pulse applied to VBAT via reverse polarity diode and more than 4.7uF capacitor		+45	V
LIN	No damage, t<500ms	-40	+40	V
LIN	No damage, ISO 7637-3 pulse 1 VBAT=13.5V, TA=23°+/-5C, test pulse applied to LIN via 1nF capacitor	-100		V

LIN	No damage, ISO 7637-2 pulse 2 VBAT=13.5V, TA=23°+/-5C, test pulse applied via 1nF capacitor		+50	V
LIN	No damage, ISO 7637-2 pulses 3A, 3B VBAT=13.5V, TA= (23+/-5) °C, test pulse applied via 1nF capacitor	-150	+100	V
LEDx	No damage	-0.3	6.0	V
VDD5P0	No Damage	-0.3	6.0	V
BULX, BUBS	No Damage	-0.3	45	V
GPIO1~9, SWDCLK, SWDIO	No Damage	-0.3	3.6	V
VBAT/VBATPO/LIN_IN/LIN_OUT/GND	ESD HBM	-4	+4	kV
All pins except VBAT/VBATPO and LIN	ESD HBM	-2	+2	kV
All pins	ESD CDM	-750	+750	V
Storage Temp		-55	+150	°C

Note: Stresses beyond those listed under absolute maximum ratings may cause permanent damage to the device. The absolute maximum ratings provided in the table above are limiting values that do not lead to a permanent damage of the part. But functional operation of the device at these or any other conditions beyond those indicated under recommended operating conditions is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

8.2 ELECTRICAL CHARACTERISTICS

Table 5 Electrical Characteristics

Parameter	Conditions	Min.	Typ.	Max.	Unit
Operating ambient Temperature		-40	25	125	°C
Operating Junction Temperature		-40	25	125	°C
Package Thermal Resistance(Ambient Temp 125°C)	Junction to Ambient (ThetaJA)		27		°C /W
	Junction to Board (ThetaJB)		10		°C /W
VBAT		6	13.5	19	V
IO Supply (VDD3P3)		3.0	3.3	3.6	V
ASIC Core Supply (VDD3P3)		3.0	3.3	3.6	V
MCU Core Supply	MCU Core Supply including SRAM and Flash	1.35	1.5	1.65	V
Flash Memory					
Sector Endurance		20k			cycles
Data Retention	@25degC	100			Years
Data Retention	@85degC	25			Years
5V Buck Converter					
Buck Input Supply	VBAT input supply (Maximum Range)	6	13.5	19	V
	VBAT input supply recommended for maximum output capability	8	13.5	19	V
Buck Output Voltage VDD5P0	V_{BAT}=13.5V, configurable range of VDD5P0.		5	6	V
Buck Load current	8V<=$V_{BAT<}$=19V, V_{VDD5P0}=5.0V.			750	mA
	V_{BAT}=6V, V_{VDD5P0}=5.0V.			500	mA

Parameter	Conditions	Min.	Typ.	Max.	Unit
Buck Peak Efficiency	At I_{LOAD}=500mA, VBAT from 8V to 19V, V_{VDD5P0}=5V, Fsw=500kHz.	85	88	91	%
Buck Quiescent current	Normal Switching mode		1	5	mA
Buck Switching Frequency (Fsw)	Forced PWM mode with integrated FETs	290	500	760	kHz
Buck High Side switch RDSon	V_{BAT}=12V, Iout=0.5A		0.65		Ω
Buck Low Side switch RDSon	V_{BAT}=12V, Iout=0.5A		0.85		Ω
Buck Ripple	V_{BAT}=13.5V, V_{VDD5P0}=5.0V, I_{Load}=300mA			60	mV
Buck Load regulation	V_{BAT}=13.5V, V_{VDD5P0}=5.0V, load current from 1uA to 500mA			20	mV
Buck Line regulation	I_{Load}=500mA, Input voltage VBAT from 8V to 19V.			20	mV
Buck Soft Start	VBAT=13.5V, I_{load}<200mA, VDD5P0 from 0 to 5V			10	ms
Buck Inductor	L1 nominal value		10		uH
Buck Output Capacitor	C_{OUT} nominal value		44		uF
Buck Input Capacitor	C_{IN} nominal value, place near VBATPO as close as possible.		10		uF
Buck Monitor Over Voltage Threshold				6.78	V
Buck Monitor Under Voltage Threshold		1.47		4.43	V
Clocks					

Parameter	Conditions	Min.	Typ.	Max.	Unit
System RC Oscillator Frequency		8		16	MHz
System RC Oscillator Accuracy	16MHz	-5		5	%
System RC Oscillator start up time			10		us
Auxiliary clock	Used in SLEEP mode		16		kHz
Auxiliary clock accuracy		-5		5	%
POR/BOR					
POR (VDD3P3)		1.8		2.1	V
BOR VDD3P3	200mV window, steps every 100mV (2.3 to 3.2V)	2.3		3.2	V
BOR VDD1P5	Max Value at which system resume operation	1.31			V
Battery Monitor					
Under Voltage Threshold	Analog Comparator Generates interrupt to MCU except in SLEEP mode (disabled feature)	4.5	5.0	5.5	V
		5.5	6.0	6.5	V
		6.5	7.0	7.5	V
		7.5	8.0	8.5	V
		8.5	9.0	9.5	V
Under Voltage hysteresis	Includes digital debounce ~50ms	0.1	0.3	0.4	V
Under Voltage Digital debounce time	Programmable for signal rise and fall, 62.5ns or 62.5us steps			16	ms
Over Voltage Threshold	Analog Comparator, generates interrupt to MCU except in SLEEP mode (disabled feature)	14		19	V

Parameter	Conditions	Min.	Typ.	Max.	Unit
Over Voltage Hysteresis	Programmable	0.4	1.1	2.4	V
Over Voltage Digital debounce time	Programmable for signal rise and fall, 62.5ns or 62.5us steps			16	ms
Current Source LED(Low Voltage Pins)					
LED voltage	minimum voltage to allow current regulation			0.5	V
	Recommended minimum voltage to ensure sink current error < 7%			1.0	V
Sink Current	VBAT>6V	0.1		45	mA
Sink Current step size			100		uA
Sink Current Error	Ta=25degC, LED voltage >1.0V	-7		+7	%
Temperature Drift			-0.025		%/K
Over Temperature Monitor					
Overtemp Threshold	Analog Comparator, generates interrupt or reset to MCU.	90		165	degC
Overtemp hysteresis		-10			degC
Temperature Sensor					
Temperature range	The MCU is in charge to pull the ADC related data from temperature sensor.	-40		150	degC
Temperature Accuracy		-10		+10	degC
Active current			20		uA
Wake Up					

Parameter	**Conditions**	**Min.**	**Typ.**	**Max.**	**Unit**
TWAKEUP	LIN_IN/LIN_OUT, programmable	30	150	200	us
SAR ADC					
Resolution			10		bits
Conversion Speed	17 cycles per conversion (4 cycles for S/H and 13 cycles for conversion)			200	ks/s
ADC Clock	16MHz RC clock divided by 4			4	MHz
INL	Guaranteed by design	-2		2	LSB
DNL	Guaranteed by design	-1		1	LSB
TCURR	Minimum time to wait after the positive edge of the sync input before starting the 1st conversion of the sequence to cover ADC input buffer transient time. Programmable Value (TCURR+1)x250ns, TCURR=7 to 15	2		4	us
TGUARD	Minimum guard time during which there is no channel input selected. (TGUARD+1)x250ns Programmable Value, default 1us	0.25	1	4	ns
TCHNL	Minimum time to wait after TGUARD time to start a new ADC conversion. Programmable Value (TCHNL+1)x250ns, TCHNL=7 to 15	2		4	us
Reference voltage		1.10	1.20	1.30	V

Parameter	Conditions	Min.	Typ.	Max.	Unit
LIN EC specified with VBAT=8V to 16V – refer to LIN 2.x specification, VBUS=LIN pin/line					
Supply Voltage	supply voltage range	6	13.5	19	V
IBUS_LIM	Current Limitation for Driver dominant state driver on VBUS = VBAT=16V	40		200	mA
Rslave	Lin Slave Pullup	20	30	60	k
Rmaster	Lin Master Pullup	900	1000	1100	
IBUS_PAS_dom	Input Leakage Current at the Receiver including Pull-Up Resistor driver off VBUS = 0V VBAT= 12V	-1			mA
IBUS_PAS_rec	Driver off, VBUS>VBAT 8V<VBAT<16V 8V<VBUS<16V			20	uA
IBUS_no_GND	Control unit disconnected from ground GND Device = VSUP 0V<VBUS<16V VBAT = 12V Loss of local ground must not affect communication in the residual network. **LIN 2.2A**	-1		+1	mA
Device Bus Leakage Current Ground Disconnected	VBAT= VGND=12V, 0V<VBUS<18V **J2602**	-100		100	uA
IBUS_no_BAT	VBAT disconnected 0<VBUS<16V, VBAT=0V **LIN 2.2A**			100	uA

Parameter	Conditions	Min.	Typ.	Max.	Unit
	Node has to sustain the current that can flow under this condition. Bus must remain operational under this condition.				
Device Bus Leakage current VBAT disconnected	0V<VBUS<18V, VBAT=VGND=0V **J2602**	-23		23	uA
BUS_VOL Transmitter dominant voltage	Load 500Ohms, driver open drain active	0.0		0.2	VSUP
BUS_VOH Transmitter recessive voltage	Driver open drain high impedance	0.8		1.0	VSUP
CSLAVE	LIN pin input capacitance Note that LIN 2.2A spec 220pF typ, 250pF max as total node capacitance at the connector including the physical bus driver and all other components including C_{LIN}			35	pF
VBUSdom	Receiver dominant state			0.4	VSUP
VBUSrec	Receiver recessive state	0.6			VSUP
VBUS_CNT	Center point Receiver VBUS_CNT =(Vth_dom+ Vth_rec)/2	0.475	0.5	0.525	VSUP
Vhys	Receiver hysteresis VHYS = Vth_rec -Vth_dom			0.175	VSUP
Trx_pd	propagation delay of receiver C_{RXD} load 20pF (RX output of transceiver, internal node, access in test mode) minimum slew rate for the LIN rising and falling edges is 50V/us			6	us

Parameter	Conditions	Min.	Typ.	Max.	Unit
Trx_sym	symmetry of receiver propagation delay rising edge w.r.t. falling edge C_{RXD} load 20pF C_{RXD} load 20pF (RX output of transceiver, internal node, access in test mode)	-2		+2	us
LIN Timing parameters (CBUS ; RBUS): (1nF; 1k / (6.8nF;660 / (10nF;500					
D1 Duty Cycle (20kbits/s)	THRec(max) = 0.744 x VSUP; THDom(max) = 0.581 x VSUP; VSUP = 7.0V...16V; tBit = 50µs; D1 = tBus_rec(min) / (2 x tBit)	0.39 6			-
D2 Duty Cycle (20kbits/s)	THRec(min) = 0.422 x VSUP; THDom(min) = 0.284 x VSUP; VSUP = 7.6V...16V; tBit = 50µs; D2 = tBus_rec(max) / (2 x tBit)			0.581	-
D3 Duty Cycle (10.4kbits/s)	THRec(max) = 0.778 x VSUP; THDom(max) = 0.616 x VSUP; VSUP = 7.0V...16V; tBit = 96µs; D3 = tBus_rec(min) / (2 x tBit)	0.41 7			-
D4 Duty Cycle (10.4kbits/s)	THRec(min) = 0.389 x VSUP; THDom(min) = 0.251 x VSUP; VSUP = 7.6V...16V; tBit = 96µs;			0.590	-

Parameter	Conditions	Min.	Typ.	Max.	Unit
	D4 = tBus_rec(max) / (2 x tBit)				
tBus_rec(max)-tBus_dom(min)	Δt3, 10.4kbs operation, low speed mode, **J2602**			15.9	us
tBus_dom(max)-tBus_rec(min)	Δt4, 10.4kbs operation, low speed mode, **J2602**			17.28	us
GPIOs					
GPIOVIL	Input Low Voltage			0.3* VDD3 P3	V
GPIOVIH	Input High Voltage	0.7* VDD D3P3			V
GPIOIOL	Max load current with output voltage=VOL			10	mA
GPIOIOH	Max load current with output voltage=VOLH			10	mA
GPIOVOL	Output Low Voltage			0.4	V
GPIOVOH	Output High Voltage	2.4			V
GPIOPU	Pull Up Resistance			110	kOhm
GPIOPD	Pull Down Resistance			110	kOhm
SWDCLK, SWDIO					
SWDVIL				0.8	V
SWDVIH		2			V
SWDCLKIOL	SWCLK, Max load current with output voltage=VOL			4	mA
SWDCLKIOH	SWCLK, Max load current with output voltage=VOH			4	mA
SWDIOIOL	SWDIO, Max load current with output voltage=VOL			8	mA

Parameter	Conditions	Min.	Typ.	Max.	Unit
SWDIOIOH	SWDIO, Max load current with output voltage=VOH			8	mA
SWDVOL				0.4	V
SWDVOH		2.4			V
SWDPU (SWDIO IO)	Pull Up Resistance	22		110	kOhm
SWDPD (SWDCLK IO)	Pull Down Resistance	22		110	kOhm
SWDVIL				0.8	V

Electrical Characteristics are valid over the full temperature range of Tj = -40°C to +125°C and a supply range of 6V <= VBAT <= 19V unless otherwise noted.

The figure below shows the relation between the propagation delay, the TX thresholds and associated receiver duty cycles. Refer to D1 to D4 duty cycles in the table above for THRec and THDom threshold levels.

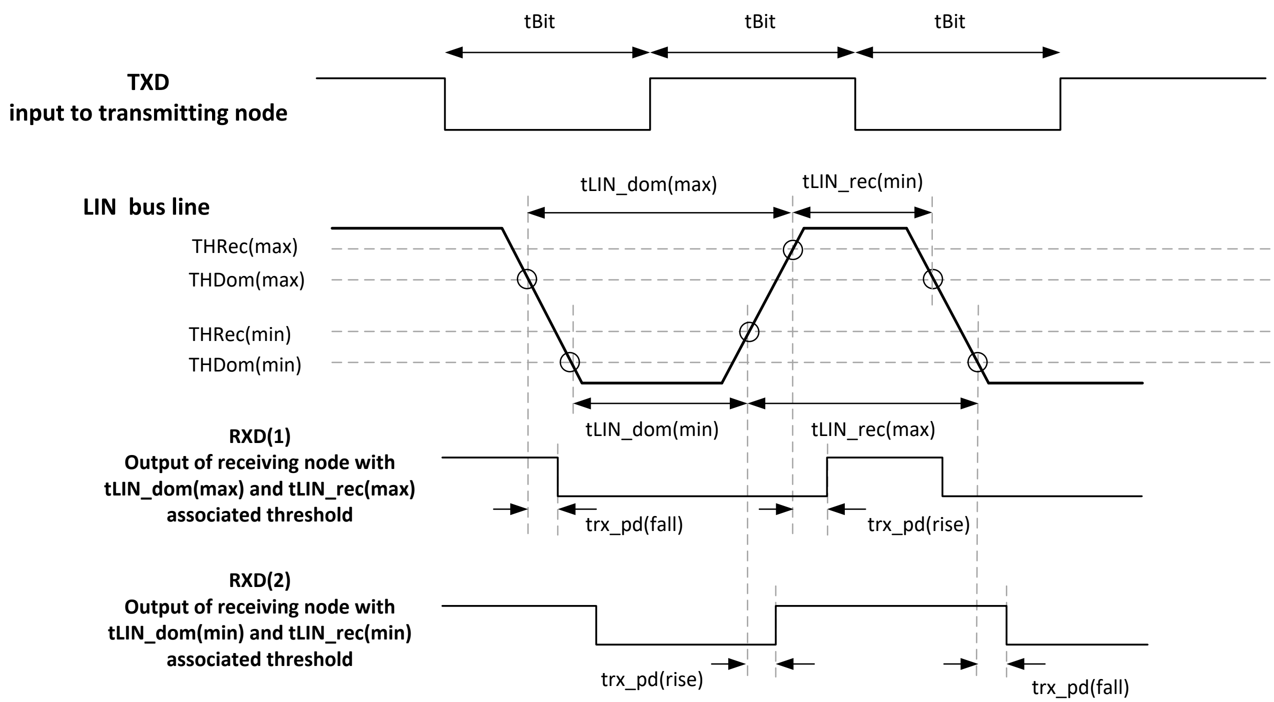


Figure 3 LIN timing Diagram

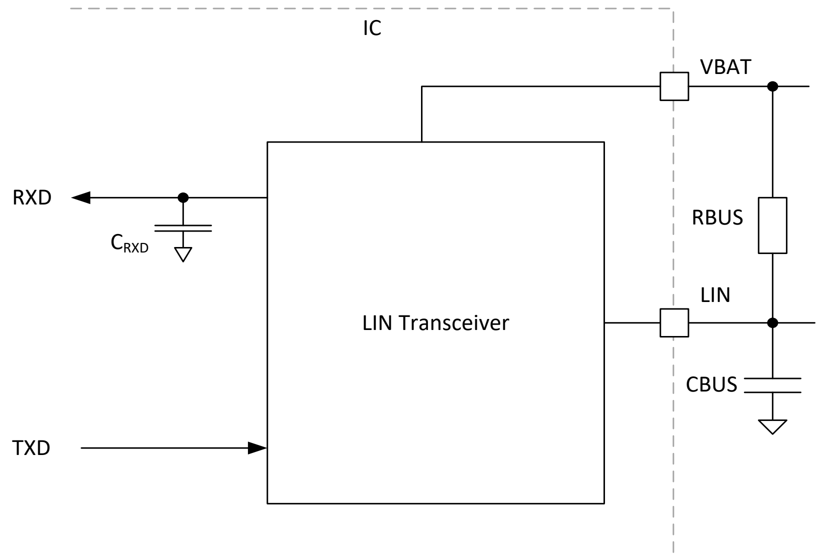


Figure 4 LIN AC Test Circuit

8.3 CURRENT CONSUMPTION

Table 6 Current Consumption

Mode	**Conditions**	**Min.**	**Typ.**	**Max.**	**Unit**
Normal	Ta=85C, VBAT=18V, RCO=16MHz, full functionality: MCU running, no flash write, LED OFF, ADC ON, VBAT and TEMP monitor ON, WDT ON.			10	mA
Sleep Mode1	Main regulator (3.3V) ON, Load dump protection active. Ta=85degC max, VBAT=13.5V Overvoltage/Undervoltage detection, PWM, LED driver, Tempsensor and ADC are OFF except LIN Switch on and GPIO toggling and wake up timer.	70	85	100	uA
Sleep Mode2	Main regulator (3.3V) ON, Load dump protection active. Ta=85degC max, VBAT=13.5V Overvoltage/Undervoltage detection,	60	75	90	uA

Mode	Conditions	Min.	Typ.	Max.	Unit
	PWM, LED driver, Tempsensor and ADC are OFF except one LIN RX on and GPIO toggling and wake up timer.				

9 DEVICE FUNCTIONAL DESCRIPTION

9.1 MCU FEATURES

9.1.1 MCU Core

The chip implements one ARM Cortex M0 core.

Additional documentation on ARM Cortex-M0 32 bits microcontroller can be found at

http://www.arm.com/products/processors/cortex-m/cortex-m0.php

9.1.2 System Memory (SRAM)

MCU core implements 16kbytes of SRAM. MCU can execute codes from the SRAM memories.

9.1.3 Flash Non Volatile Memory

MCU implements a Programmable Flash Memory with x32 configuration, sector and chip erase and byte program capability. It integrates five 512bytes nonvolatile registers (NVR) sectors.

In normal operation the ARM core fetches instructions (or data permanently stored) from the Flash memory but it is also possible for a program to alter the content of the flash memory. The following operations can be performed in the Flash Memory:

- Byte Write

- Sector Erase
- Block Erase
- Code Protect

For a description of the flash memory registers, please refer to the product register map. Here is a simple description of the basic features supported:

- Registers support to write/erase data to a byte, sector address
- Support programmable read wait states*
- Support system clock divider for write/erase functions
- Protection mechanism to unlock flash memory write and start flash memory byte-write
- Protection mechanism to unlock flash memory sector erase

* The design is implemented such that the timings associated with the flash macro meet the maximum speed of the system clock requirements.

9.1.4 **Interrupt vectors**

The first 148 bytes of Flash Memory are organized following the standard created by ARM. In this standard the Address 0x00000 contains the top-of-stack address (four bytes). The following addresses contain interrupt vectors used by the microcontroller:

Table 7 Interrupt Vector

Vector Name	Address	Comments
STACK_VALUE	0x00000	Typically set to 0x20000FFFF (Top of SRAM)
Reset_Handler	0x00004	Reset routine entry
Reserved	0x00008	Reserved. No NMI implemented.
HardFault Handler	0x0000C	

Reserved	0x00010 to 0x00028	
SVC_Handler	0x0002C	
Reserved	0x00030-0x00034	
PendSV_Handler	0x00038	
SysTick_Handler	0x0003C	
WULIN_Handler,	0x00040	Wake Up LIN Slave
PWMAUX_Handler,	0x00044	PWM Aux
WUTIMER_Handler,	0x00048	Wake Up Timer
BOR_Handler,	0x0004C	Brown out event
VBAT_Handler,	0x00050	Over/Under Voltage event
OVTEMP_Handler,	0x00054	Over Temperature event
WatchdogA_Handler,	0x00058	ASIC watchdog timeout
GPIO_Handler,	0x0005C	GPIO Interrupts
LINS_Handler,	0x00060	LIN Slave bus event
LINM_Handler,	0x00064	LIN Master bus event
UART_Handler,	0x00068	UART event
ADC_Handler,	0x0006C	ADC data ready
PWM_Handler,	0x00070	PWM event
BUCK_Handler	0x00074	Buck event
SPI_Handler	0x00078	SPI Master event
Lullaby_Handler,	0x0007C	Software Interrupt

Timer0_Handler	0x00080	
Timer1_Handler	0x00084	
Timer2_Handler	0x00088	
Watchdog_Handler	0x0008C	
BTE_Handler	0x00090	Block Transfer – Contact indie to get more information.
Reserved	0x00094	

All other addresses in the flash memory can be used for the user's program.

The meanings of the standard interrupt vectors (Provided with the ARM Cortex M0 core) are defined in ARM's documentation. One of the sources of information is:

http://infocenter.arm.com/help/topic/com.arm.doc.dui0497a/DUI0497A_cortex_m0_r0p0_generic_ug.pdf

9.1.5 Interrupt Enabling/Disabling Process

Cortex-M0 implements a NVIC (**N**ested **V**ector **I**nterrupt **C**ontroller) peripheral capable of handling up to 16 peripheral's interrupts. Upon reset the microcontroller can answer only to Reset, NMI (Non-Maskable Interrupt) and Hard-Fault interrupts/exceptions. All other interrupts must be enabled. To enable and disable the interrupts the user must use access the ISER (Interrupt Set Enable Register) and ICER (Interrupt Clear Enable Interrupt) registers associated with the desired interrupt.

NOTE: Both inline functions and all parameters are defined in the product_file.h file, which must be included in the source files. Besides that the product_file.h file contains a list of available interrupts. The format of this list is as follows:

```
typedef enum IRQn
{
  //******  Cortex-M0 Processor Exceptions Numbers ***************************
  NonMaskableInt_IRQn       = -14,   // Non Maskable Interrupt
```

```
HardFault_IRQn        = -13,   // Hard Fault Interrupt
SVCall_IRQn           = -5,    // SV Call Interrupt
PendSV_IRQn           = -2,    // Pend SV Interrupt
SysTick_IRQn          = -1,    // System Tick Interrupt

//*****  CM0IKMCU Cortex-M0 specific Interrupt Numbers **********************
IRQ04_IRQn            = 0,     // Product specific
IRQ04_IRQn            = 1,     // Product specific
IRQ04_IRQn            = 2,     // Product specific
IRQ04_IRQn            = 3,     // Product specific
IRQ04_IRQn            = 4,     // Product specific
IRQ05_IRQn            = 5,     // Product Specific
IRQ06_IRQn            = 6,     // Product Specific
IRQ07_IRQn            = 7,     // Product Specific
IRQ08_IRQn            = 8,     // Product Specific
IRQ09_IRQn            = 9,     // Product Specific
IRQ10_IRQn            = 10,    // Product Specific
IRQ11_IRQn            = 11,    // Product Specific
IRQ12_IRQn            = 12,    // Product Specific
IRQ13_IRQn            = 13,    // Product Specific
IRQ14_IRQn            = 14,    // Product Specific
IRQ15_IRQn            = 15,    // Product Specific
TIMER0_IRQn           = 16,    // Timer 0
TIMER1_IRQn           = 17,    // Timer 1
TIMER2_IRQn           = 18,    // Timer 2
WATCHDOG_IRQn         = 19     // Watchdog timer
} IRQn_Type;
```

9.1.6 **Flash Code protection**

The controlled access to the flash content is based on disabling all communications with the debug interface, therefore preventing any external attack. Hence, the application code is still able to modify the Flash content.

Upon Power-On Reset or Normal Reset, MCU core disables the communication with the debug interface for a small time interval (8192 system clock cycles). If the application needs to be protected it is mandatory to set the protection register with the appropriate code in the beginning of the initialization process and before the internal hardware enable the debug communication. In other words, if during this time interval the protection register is loaded with a specific pattern, then the communication remains disabled after the end of this interval and stays disabled until this register is loaded with a different pattern. To allow for debug communication the application has only to write a different value in the lock register.

If a part is protected, the emulator can still erase and program the part, but first it will be required to erase the Flash content, therefore protecting it.

9.1.7 **Systick Timer**

This timer is an optional peripheral created by ARM and implemented in the Cortex M0 160/8. It is fully described in the Cortex-M0 Devices Generic User Guide (Chapter 4.4 Optional System Timer, Systick) found at:

http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dui0497a/Babieigh.html

9.1.8 **Timers (0, 1 and 2)**

The MCU implements three identical timers: Timer0, Timer1 and Timer2. All three timers operate using the system clock as clock source. They increment at the system clock rate starting from the loaded value in the counter until they roll over from 0xFFFFFFFF to 0x00000000. At this point an interrupt is generated if enabled. The interrupt routine is responsible for reloading the value if needed as this timer does not auto-reload the original content.

9.1.9 **Watch Dog Timer**

The MCU implements a WDT (**W**atch **D**og **T**imer) that can operate in one of two ways:

- Interrupt Mode: In the event of a WDT rollover an interrupt will be generated.
- Reset Mode: In the event of a WDT rollover the microcontroller will reset.

The WDT supports Reset, Enable, status/flag and clear functions. It integrates a pre-scaler that can divide the system clock by 2^{13}, 2^{19}, 2^{22} or 2^{32}. It means that the WDT internal counter will count from 0 to the pre-scaler value at the system clock speed and trigger if not cleared.

For instance, a system running from a 30MHz system clock and 2^{22} pre-scaler value will trigger the WDT after approximately 0.14 seconds if not cleared properly and in time by the application.

9.1.10 **MCU Core to ASIC interface**

The ASIC die will be communicating to the indie Cortex M0 through a proprietary interface. The interface used with mcu should be fully kept as is to enable any swap between ASIC die and MCU die.

9.2 **ASIC FEATURES**

9.2.1 **Power on sequence**

The figure below shows the power on sequence , VDD3V3/ VDD1V5 connect to the external 4.7uF capacitor.

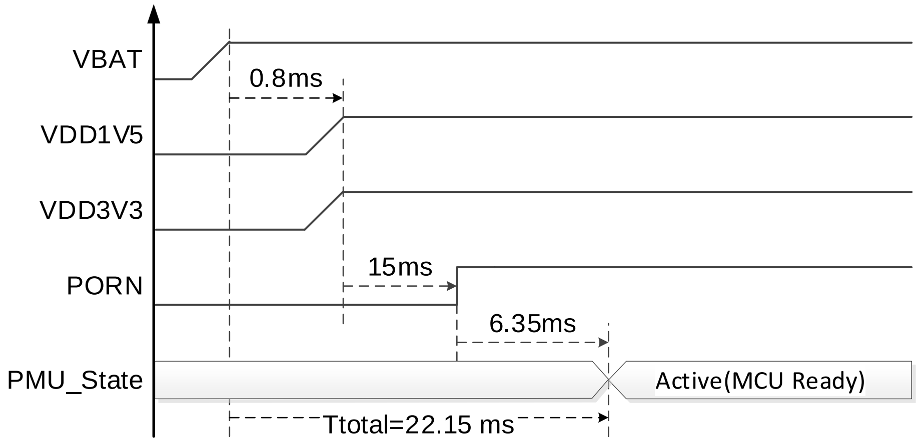


Figure 5 power on sequence

ASIC has an On-chip Brown-out Detection (BOD) circuit for monitoring the VCC level during operation by comparing it to a fixed trigger level. The trigger level for the BOD can be selected by the BOR3V3THRESH / BOR1V5THRESH. The trigger level has a hysteresis to ensure spike free Brown-out Detection.

The BOD circuit can be enabled/disabled by the fuse BOR_EN. In the figure below, when the BOD is enabled and BOR ACTION is configured as Hard reset, and VDD_3V3/ VDD_1V5 decreases to a value below the trigger level (BOR_3V3_thre/ BOR_1V5_thre), the Brown-out Reset(BORN) is activated. When VCC increases above the trigger level (BOR_3V3_thre/ BOR_1V5_thre), the BORN is released.

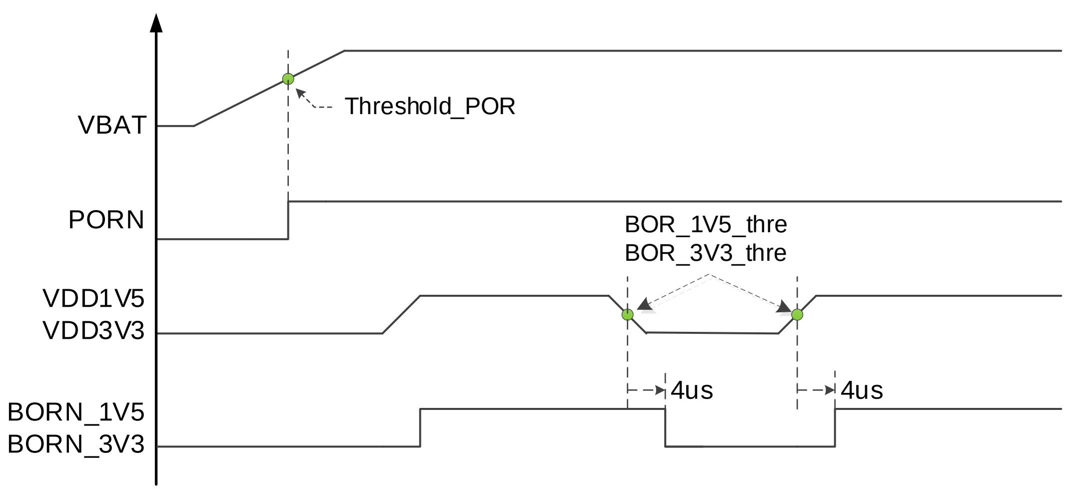


Figure 6 BORN Generation

9.2.2 Clock Generation

Two clock sources are integrated in the device. The system clock is based on a RC network and will be trimmed to meet the accuracy requirements specified in the EC Table. Additionally an auxiliary clock will be used during sleep.

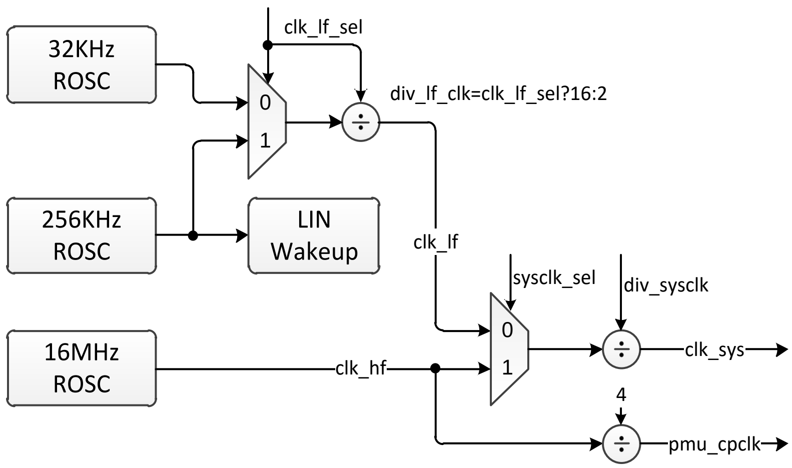


Figure 7 Clock Generation

9.2.3 Reset

Both ASIC and MCU integrates Power on Reset (POR) circuitry: MCU POR monitors its input supply and generate a reset every time the MCU supply is recycled. ASIC POR monitors the main 3V3 LDO supply and generate a reset every time the LDO supply is recycled. Both POR maintain their output reset active as long as the monitored supply voltage is not above the minimum supply level to ensure safe logic operation of the Power Management Unit including the necessary analog features such as clock generator, bandgap, etc…This level is hardcoded and process technology dependent.

Additionally, the ASIC integrates Brown Out (BOR) circuit detectors that are configurable by SW (enable/disable as well as threshold programmable for the main ASIC core supply) and are actively monitored by the ASIC power management unit (PMU). In case any BOR is triggered, the PMU can be configured to either do the following (per BOR blocks).

- trigger a system reset
- generate an interrupt to the MCU for further actions.
- do nothing

The table below show the BOR levels settings for both VDD3P3 (ASIC Core supply) and VDD1P5 (MCU Core Supply). The * are the default settings values after reset and do not need to be changed as the thresholds are guaranteed to provide safe margin for full operation across PVT.

Table 8 BOR Trigger Level [* reset default]

BOR Setting <3:0>	BOR Level VDD3P3		BOR Level VDD1P5	
	Threshold [V]	VBAT [V]	Threshold [V]	VBAT [V]
0	2.10	4.40	1.376*	5.1
1	2.10	4.40	1.344	5.03
2	2.10	4.40	1.315	4.95
3	2.32	4.47	1.285	4.88
4	2.39	4.55	1.413	5.19
5	2.46	4.63	1.449	5.28
6	2.56*	4.73	1.484	5.37
7	2.65	4.83	1.523	5.47
8	2.74	4.94	1.568	5.59
9	2.84	5.05	1.608	5.71
A	2.96	5.18		
B	3.08	5.32		
C	3.21	5.48		
D				
E				
F				

Finally, the MCU watchdog timer (if configured to do so) or a reset instruction can reset the MCU logic while on the ASIC side the watchdog timer (if configured to do so) or a reset register write can reset the ASIC system.

The block diagram below illustrates the possible triggers of a reset on both side of the design: ASIC and MCU.

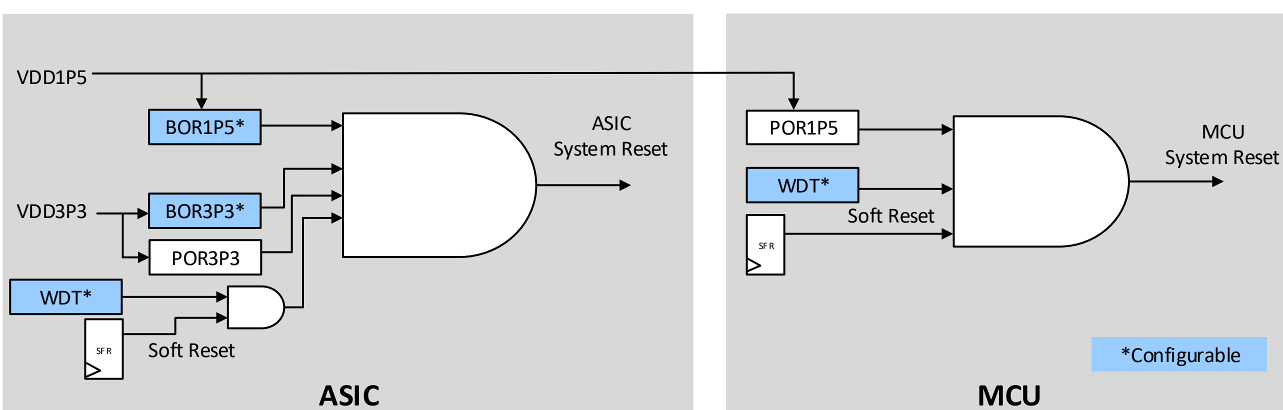


Figure 8 System Reset sources (Reset active LOW)

9.2.4 **PMU and Load Dump Protect circuits**

ASIC integrates battery monitoring functionality that will detect load dump events occurring at the battery supply pin. An analog comparator will detect over voltage transients after going through a load dump limitation circuit. The protection circuit can handle DC voltage up to 45V and will limit these to maximum of *Over Voltage Threshold* nominal, therefore the comparator will detect any over voltage transients at the battery pin beyond that threshold. The circuit will generate a digital output signal to be filtered and sent to the interrupt controller. The circuit will be active during normal operation mode – not during sleep condition.

The voltage regulators themselves are supplied by VBAT directly and sustain DC level of load dump voltages.

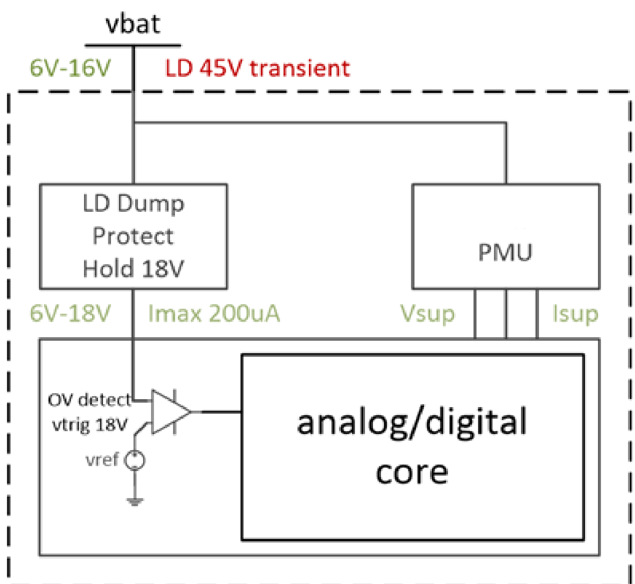


Figure 9 Load Dump Protect

9.2.5 **LIN Interface**

9.2.5.1 **LIN Transceiver**

Supports LIN Protocol Controller according to LIN 2.x and SAE J2602 (rev J2602-1_201211). The IC contains two integrated PHYs (One for Slave and the other for Master) for low speed vehicle serial data network communication using the LIN protocol.

Notice that LIN Master PHY is disconnected from LIN Master controller is the internal LIN Switch is enabled.

9.2.5.1.1 LIN RxD Debounce

For preventing RxD spikes in case of RF interferences and automotive pulses, two digital glitch filters are integrated in the data pathes of LIN RxD.

Debounce thresholds for low to high and high to low can be programmable independently (See IOCTRLA->LINSGFCONF,IOCTRLA->LINSGFCONF1&IOCTRLA->LINMGFCONF,IOCTRLA->LINMGFCONF1).

$$T_{thres} = \left(T_{hfosc}\right) \times LINDBNCTHRESx$$

For instance, the default DBNCTHRES value is 0x30 and HFOSC freq is 16MHz.Thus the default debounce threshold is 62.5ns * 48 = 3 us.

9.2.5.1.2 LIN TxD Timeout Monitor

Two TxD Timout monitors are integrated to prevent dominant bus due to internal malfunction. LIN TX is controlled by TxD signal from LIN controller or GPA(Selected by bit LINS_HWMODE in IOCTRLA->LIN). If TxD is stuck at low over a specified TxD timeout time due to a crash of MCU/GPA/LIN Controller, the integrated TxD monitor will switch off the LIN TX output automatically until a low to high transition of TxD.

9.2.5.1.3 Short LIN Bus to Ground

If the bus idle timeout monitor detects the bus is shorted to ground (See bit BUS_IDLE_TIMEOUT_DOMINANT in LINS->STATUSEXT), LIN Slave's 30K pullup will be automatically switched off to prevent a fast discharge of the car battery.

If enabled, the feature is always on even the chip is in hibernate mode.

9.2.5.1.4 External LIN Transceiver Mode

In this mode, the internal LIN transceiver is bypassed and LIN Slave controller's RxD/TxD can be connected with an external transceiver through GPIO3/4 or GPIO5/6(See HWMODE bits from IOCTRLA->GPIO3 to IOCTRLA->GPIO6); LIN Master controller's RxD/TxD can be connected with an external transceiver through GPIO1/2(See HWMODE bits from IOCTRLA->GPIO1 to IOCTRLA->GPIO2).

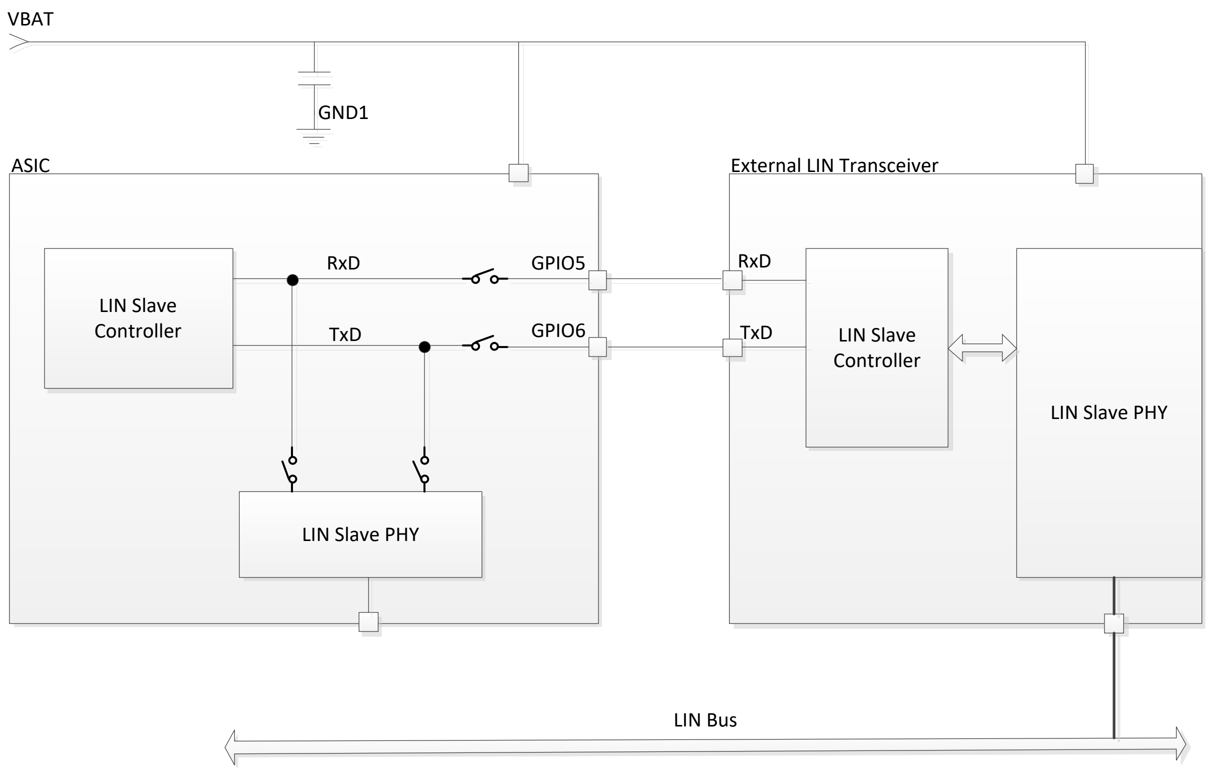


Figure 10 External LIN Transceiver Connection

9.2.5.2 **LIN controller**

The IC contains two LIN cores. One is for LIN Slave and the other is for LIN Master. The LIN core is a communication controller that performs serial communication. It implements the datalink layer of the LIN Protocol Specification. LIN uses a single master / multiple slave concept for the message transfer between nodes of the LIN network. The LIN controller core comprises an interface to connect a micro controller that accesses the LIN core registers to control the transmission and reception of message frames.

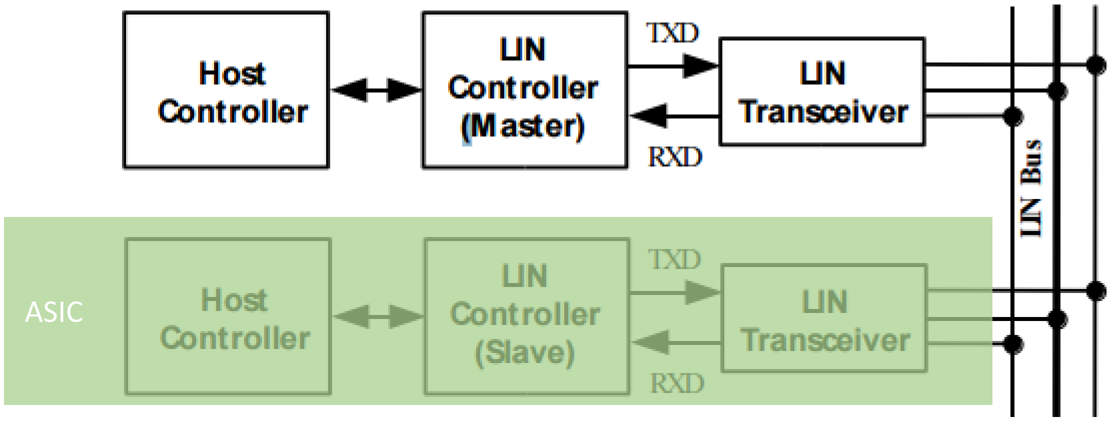


Figure 11 Lin System

Features:

- One LIN Slave Controller
  - Support of LIN specification 2.2A/SAE J2602
  - Backward compatibility to LIN 1.3
  - Supports LIN auto-addressing through an internal LIN switch.
- One LIN Master Controller
  - Only available when the internal LIN switch is not used.
  - Only supports auto-baudrate LIN Slaves.
- Programmable data rate between 1 Kbit/s and 20 Kbit/s (for master)
- Automatic bit rate detection (for slave)
- 8-byte data buffer
- 8-bit host controller interface
- Fully synchronous design, available in VHDL or Verilog, completely synthesizable
- Support auto addressing
  - Note: Node configuration and diagnostics implemented by the host controller

9.2.5.2.1 Data Length Register and Enhanced Checksum

The host controller has to define the length of the data field of the current LIN frame by adjusting the DATA LENGTH register. If the data length bits[3:0] are loaded with the value "1111b" the length of the data field is decoded from Bit 5 and 4 of the identifier register (ID) according to table below (e.g. compatibility to LIN specification 1.1). Otherwise the amount of data bytes can be written directly to the DATA LENGTH register (supported values are 0...8).

Table 9 ID bits and number of bits

ID (Bit 5)	ID (Bit 4)	Number of Bytes in the data field
0	0	2
0	1	2
1	0	4
1	1	8

The LIN core supports classic checksum (Spec 1.3, inverted eight bit sum with carry over all data bytes) and enhanced checksum (Spec. 2.0, inverted eight bit sum with carry over all data bytes and protected identifier). The host controller has to set the checksum type used in the current frame by adjusting Bit ENCHK in the data length register ('1' for enhanced checksum, '0' for classic checksum).

9.2.5.2.2 Timing Settings for "Wake up Repeat Time" and "Bus Inactivity Time"

The time for repeating of wake up because of no reaction on the bus and for go to sleep because of inactivity on the bus can be optionally written by the host controller to registers WUPREPEAT and BUSINACTIVE (address 0x0F).

Table 10 LIN Inactivity Time

BUSINACTIVE [1:0]	LIN Inactivity Time (sec.)
00	4*
01	6
10	8
11	10

Table 11 LIN Wake-Up Repeat Time

WUPREPEAT [1:0]	LIN Wake-Up Repeat Time (msec.)
00	180*
01	200
10	220
11	240

9.2.5.2.3 Bit Time Settings

The Bit rate of the LIN system has to be defined in the bit timing registers.

Table 12 Bit Timing Related Registers

Name	**Description**	**Width(bits)**
BTDIV	Bit time divider integer value	9
PRESCL	Clock Prescaler	2

The LIN bit rate $f$bit can be calculated from system clock $f$clk and bit timing registers according to the following general equation:

$$f_{bit} = \frac{f_{clock}}{2^{prescl} * bt\_div * (bt\_mul + 1)}$$

Note that the procedure of adjusting the bit timing registers is different between master and slave. For the slave controller, the Bit timing register adjustment of slave is the following:

The steps for adjusting the bit timing registers of the LIN slave are explained below.

Note: Register bt_mul does not exist in the slave. The LIN core slave synchronizes to any bit rate between 1 Kbit/s and 20 Kbit/s. Nevertheless, the bit timing registers have to be adjusted to adapt the LIN core to the used system clock frequency.

- Setting up the pre-scaler register depending on system clock according to the following equation; the value has to be rounded down to the next integer value:

$$prescl = \ln\left(\frac{f_{clock}}{20KHz * 200}\right) * \frac{1}{\ln 2}$$

- Adjusting the bit time divider depending on system clock and pre-scaler according to the following equation; the value has to be rounded down to the next integer value:

$$bt\_div = \frac{f_{clock}}{2^{prescl} * 20KHz}$$

System Clock	**PRESCL**	**BTDIV**

8MHz	1	200
12MHz	1	300
16MHz	2	200

Table 13 Sample value for setting up bit timing registers

9.2.5.2.4 Controlling the LIN core (slave) by a host controller

The first step before transmitting messages with the LIN core is setting up the bit rate of the LIN system. For that, the host controller has to load the bit time registers. After that, the message transfer can be started. The LIN core slave detects the header of the message frame sent by the LIN master and synchronizes its internal bit time to the master bit time. An interrupt to the host controller is requested after the reception of the IDENTIFIER FIELD, after the reception of a wakeup signal (if the slave is in sleep mode), when an error is detected or when the message transfer is completed.

The following steps have to be done by the host controller when an interrupt is requested.

1) Check bit DATAREQ in the status register (it is 1 when the IDENTIFIER FIELD has been received). Proceed with the following if DATAREQ is set else proceed with step 2.

    a. Load the identifier from the ID register and process it.
    b. Adjust the bit TRANSMIT in the control register ("1" - if the current frame is a transmit operation for the slave, "0" – if the current frame is a receive operation for the slave).
    c. Load the data length in the data LENGHT register (number of data bytes or value "1111b" if the data length should be decoded from the identifier) and set the checksum type (enhanced checksum (Bit ENHCHK = '1') or classic checksum (Bit ENHCHK = '0')).
    d. Load the data to transmit into the data buffer (for transmit operation only).
    e. Set the bit DATAACK in the control register.

Note 1: Steps a..e have to be done during the IN-FRAME RESPONSE SPACE, if the current frame is a transmit operation for the slave; otherwise a timeout will be detected by the master. If the current frame is a receive operation for the slave, steps a..e have to be finished until the reception of the first byte after the IDENTIFIER FIELD. Otherwise, the internal receive buffer of the slave core will be overwritten and a timeout error will be detected in the slave core.

Note 2: If the host controller of the slave detects an unknown identifier (e.g. extended identifier) it has to write a '1' to bit "stop" in control register instead of setting bit DATAACK (steps b .. e can be skipped). In that case the LIN core slave stops the processing of the LIN communication until the next SYNC BREAK is received.

2) Check bit ERROR in the status register. Perform error handling and proceed with step 6 if bit ERROR is set else proceed with step 3.

Note 3: Bit TIMEOUT in the error register and bit WAKEUP in the status register are set if the slave has sent a wakeup signal but the master did not respond within 150ms.

3) Check bit BUSIDLETIMEOUT in the status register and activate the sleep mode by setting bit SLEEP in the control register if BUSIDLETIMEOUT is set.
4) Check bit WAKEUP in the status register (it is set if the slave has received a wakeup signal). If WAKEUP is set proceed with step 6 else proceed with step 4.

Note 4: Bit COMPLETE in the status register is not changed when a wakeup signal is transmitted or received. Therefore, bit WAKEUP has to be checked before bit COMPLETE

5) Check bit COMPLETE in the status register (it is set if the transmission was successful). If COMPLETE is set and the current frame was a receive operation for the slave, load the received data bytes from the data buffer.
6) Set the bits RSTINT (reset interrupt) and RSTERR (reset error) in the control register to reset the interrupt request and the error flags.

9.2.5.2.5 Sleep Mode and Wakeup

To reduce the systems power consumption the LIN Protocol Specification defines a Sleep Mode. The message used to broadcast a Sleep Mode request has to be started by the host controller of the LIN core master in the same way as a normal transmit message. The host controller of the LIN core slave has to decode the Sleep Mode Frame from Identifier and data bytes. After that, it has to put the LIN slave node into the Sleep Mode by setting bit SLEEP in the control register. If bit SLEEP in the control register of the LIN core slave is not set and there is no bus activity for 4 s to 10 s (specified bus idle timeout), bit SLEEP & BUSIDLETIMOUT are set and an interrupt request is generated. After that application has to understand that the LIN bus is in Sleep Mode. The bus inactivity time which should be defined as bus idle timeout for the slave can optionally set to values 4s, 6s, 8s or 10s as possible accordingly with specification 2.2A.

After receiving a Wakeup signal from the master or any slave node a wakeup request is generated, the host controller terminates the Sleep Mode of the LIN bus by clearing bit SLEEP in the control register.

Notice that don’t enter hibernate mode when bit SLEEP = 0, because in this state dominant signal will not be taken as a wakeup request.

To send a Wakeup signal, the host controller of the LIN core has to set the bit WAKEUPREQ in the control register. After successful transmission of the wakeup signal with the LIN core master the WAKEUP bit in the status register of the sending LIN core master is set and an interrupt request is generated. The LIN core slave does not generate an interrupt request after successful transmission of the Wakeup signal but it generates an interrupt request if the master does not respond to the Wakeup signal within 150ms to 250ms. This value can be set optionally to 180ms, 200ms, 220ms or 240ms as it is possible accordingly with specification 2.2A. In that case, bit ERROR and bit TIMOUT are set**.** The host controller has to decide whether to transmit another Wakeup signal or not.

9.2.5.2.6 Error Detection and Handling

The LIN core generates an interrupt request and stops the processing of the current frame if it detects the following errors：

- BITMON: The bit value monitored on the bus is different from the sent bit value.
- Timeout caused by wakeup repeat timeout.
- BUS IDLE Timeout.

The errors detected in Break/Sync field will be ignored by LIN controller. The other errors happen in ID field/Data field will be recorded. In DataReq interrupt routine, the application has to check the type of error by processing the ERROR register. After that, it has to reset the ERROR register and the ERROR bit in status register by writing a 1 to bit RSTERR in control register. Starting a new message with the LIN core master or sending a Wakeup signal with master or slave is possible only if bit ERROR in status register is 0.

9.2.5.3 **Auto Addressing (Lin Switch Mode):**

While iND83211 integrates a lin switch and two LIN interface pins, LIN_IN and LIN_OUT, connecting to LIN transceiver. Software is in charge to control the lin switch through IOCTRLA LIN control registers. When SWON is set to 1, the lin switch will connect LIN_IN and LIN_OUT. At the same time, the lin master controller will lose the connection with the lin master transceiver.

With this lin switch, iND83211 can support auto addressing function. BCM connects iND83211 in a chain by connecting LIN_IN to upstream LIN bus, and connecting LIN_OUT to downstream LIN bus, as the figure below shows. At the first boot of the system, every switch is on. The following is a SNPD sequence for instance:

1. Lin Master sends a diagnostic frame (ID=3C, 8 bytes data frame) to inform the slaves that SNPD sequence is started. Then the slaves disconnect downstream LIN bus by opening the internal switch. Thus, only the first one, iND83211 1, can receive message from BCM via LIN_IN pin.
2. Lin Master sends 1st NAD configure frame with NAD=”01”. Thus “01” address is assigned to iND83211 1 at first. And then iND83211 1 close its internal LIN switch, thus iND83211 2 can receive message from LIN bus.
3. Then system can assign the second address for iND83211 2 accordingly. By analogy, all iND83211s on the chain can be assigned address.

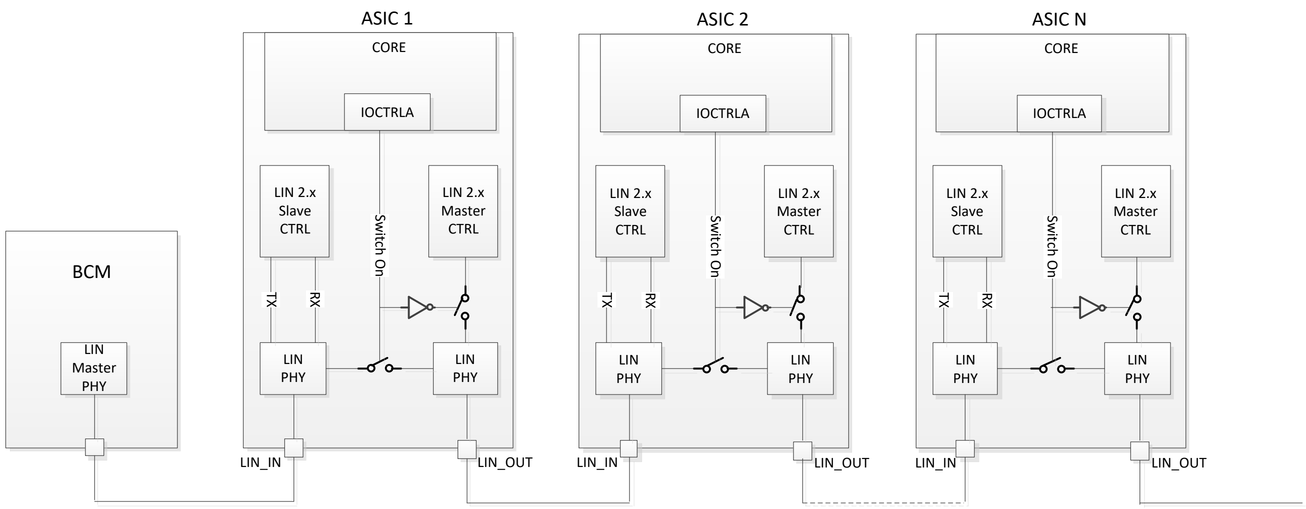


Figure 12 LIN auto addressing

Cautions:

1. LIN protocol requires inter-frame space not negative. That means any space > = 0 is acceptable. Since the internal switch is controlled by software, there are a big latency from frame complete interrupt to control the switch by interrupt routine. So a big inter-frame space is required during SNPD sequence, while this is determined by LIN master.

9.2.6 Buck Regulator

The IC integrates a switching synchronous buck converter with integrated high and low side switches to allow for small external components and reduced output ripple. The buck operates in forced PWM mode at a fixed frequency. Additionally, the buck can work with a clock with spread spectrum frequency content that can reduce the peak radiated and conducting noise on both the input and output supplies, making it easier to comply with electromagnetic interference (EMI) standards. The buck should also support soft start feature to prevent high in rush current after enabling the regulator. The maximum in-rush current shall not exceed 10 times the nominal buck load (so about 3A).

9.2.7 LED Driver Stage

ASIC integrates a high precision open drain LED Driver Stage that allows for LED currents in the range of 100uA to 45mA in 100uA increments. The LED bias circuit uses a precise bandgap referenced current(**CurrentV2I**) which is multiplied over stages to sink current via the LED which is connected to the HVIO(LED) pin. After factory test, the trim value for **CurrentV2I** = 25uA will be recorded and boot program is in charge of initializing the V2I trim bits correctly(refer to register description). The mirror stages consist of 100uA unit cells which are weighted linear to provide 100uA ~ 45mA current （CurrentLED）. The desired current is provided according the formula below:

$$\mathrm{CurrentLED(mA)} = \mathrm{TRIM}[8:0] * CurrentV2I * 4 = TRIM[8:0] * 100uA$$

Notice that if LED current is set to > 45mA, the accuracy is not guaranteed.

After delicate calibration by LED trim bits(refer to register description), the combination of stages allows for high accurate LED current in 100uA steps that are combined at the HVIO(LED) pad on chip (refer to the figure below).

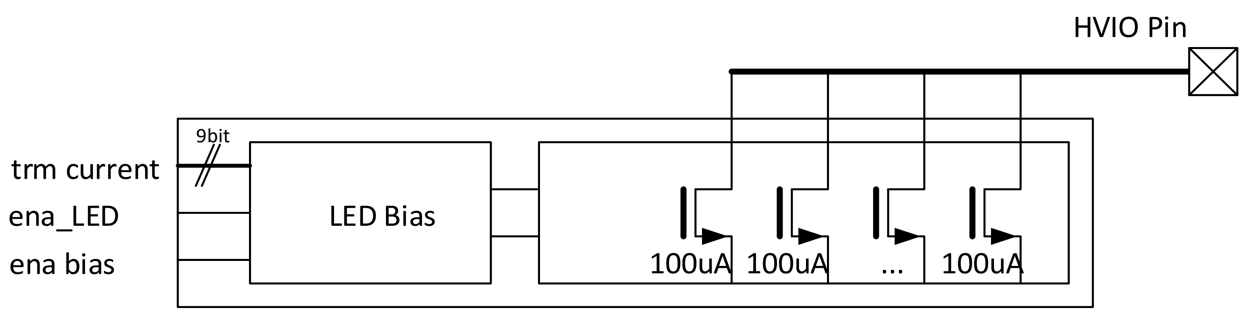


Figure 13 LED Driver Concept

The operation of the LED Driver IP requires two actions. At first the LED bias circuit needs to be enabled (ena_bias) (refer to), followed by enabling the LED (ena_LED). Latter control signal is driven by the pulse width modulator that will drive the current from the battery via the LED. The PWM signal utilizes an input signal at maximum of 250Hz and modulates its pulse width (refer to section 9.2.15 LED PWM).

9.2.8 House Keeping SAR ADC

- 10-bit resolution, single ended input
- Calibrated 1.22V Bandgap Voltage reference
- Global ADC attenuator (1:1 or 1:3)
- ADC is used for monitoring
    - CH0: Band gap reference
    - CH1/CH3:
        - Accurate VBAT(1/32): Supply Voltage (limited to max voltage limited by the load dump protection circuit)
        - VDD5V(1/8)
        - Internal Temperature Sensor

- CH2: Forward voltage of the external LEDs
- CH4: GPIO analog inputs
- CH5: VDD1V5

- ADC system capable of being configured for single or automatic multiple channel conversions (VBAT, HVIO and Temperature Sensor).
- Interrupt on conversion complete regardless of digital comparator configuration

There is a global ADC attenuator for measuring an ADC channel which voltage will exceed the bandgap reference voltage, such as VDD1V5 channel, GPIO analog input channels. The feature will take effect on all ADC channels.

ADC automatic sequencer:

The following diagram shows how the VBAT (CH1), LED Forward voltage-VFW (CH2) voltages are measured along with the junction temperature monitor (CH3). VFW voltages are converted from differential voltage to single-end voltage and shifted to ADC range(1/4 attenuation). This topology guarantees excellent (VBAT-HVIO) differential voltage measurement. The DC Voltage of VBAT can also be taken but with accuracy. While both PWM and ADC sequencer run from the same system clock (System RC Oscillator), the PWM signal is further downscaled while the ADC sampling clock can be adjusted to meet sample rate requirements. In order to reduce SW overhead, the PWM block provides a SYNC signal to the ADC sequencer which can then use this information to start the programmed conversions. In order to optimize the time taken to convert all channels (CH1,2,3) sequentially, the ADC sequencer can be configured to automatically start CHx conversion after TCURR and follow by the other channels after a duration set by TGUARD+TCHNL. TGUARD is the guard time where there is no channel selected, while switching from one channel in the sequence to the other to avoid any overlap. TCHNL is the time to wait after the guard time TGUARD for the conversions of 2nd or 3rd channel in the sequence, to allow settling of the channel before start of the conversion. The sequencer also provides a register that defines which channel need to be converted (1 to 3) and in which sequence. The sync feature is enabled by setting the SYNCENA bit and SYNCEDGE bit in ADC CNTRL Register. This enables the ADC conversion to be synchronized with the **positive edge or negative edge** of Sync input which is coming from PWM output. In short, ADC conversion is synchronized with the edge of the PWM output, if SYNCENA is set, and ADC is asked for conversion.

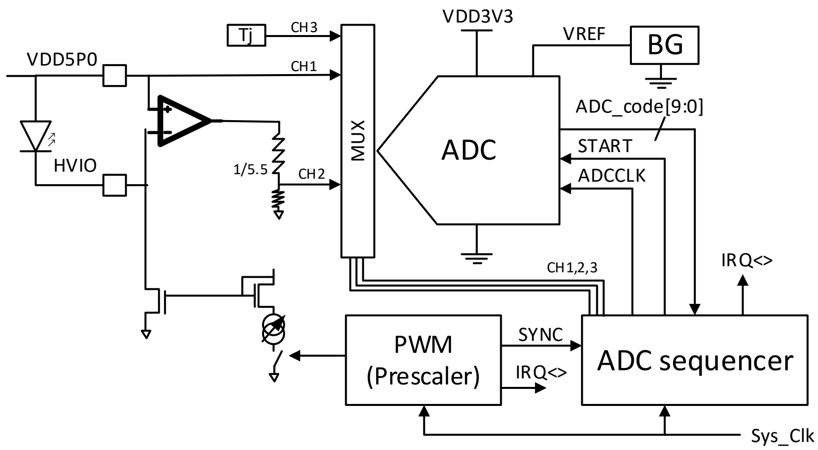


Figure 14 ADC synchronization with PWM

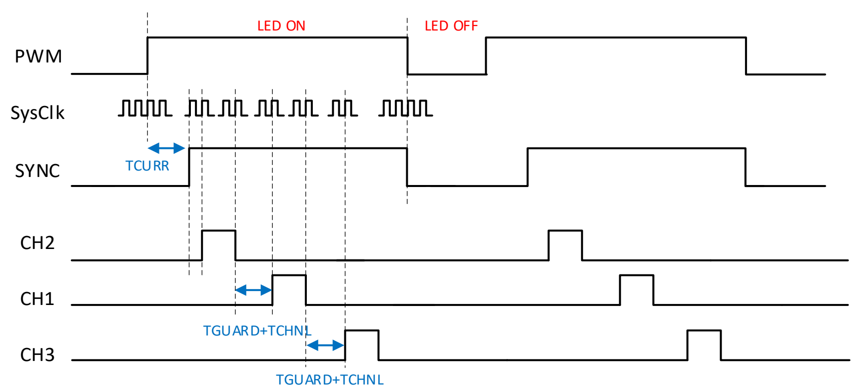


Figure 15 ADC read channels sequence triggered by PWM posedge

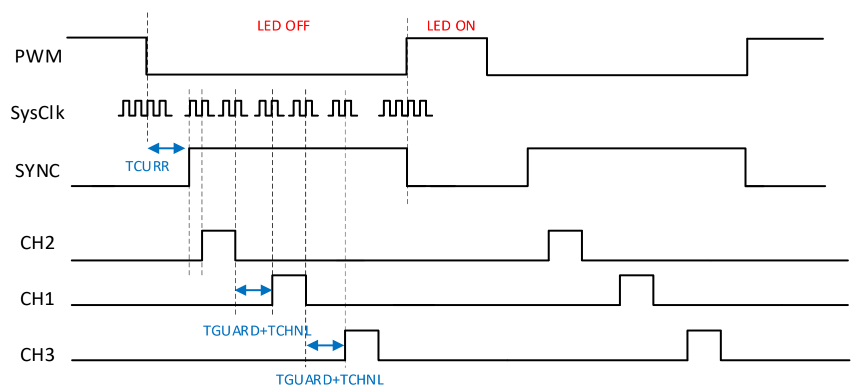


Figure 16 ADC read channels sequence triggered by PWM negedge

9.2.9 Over and Under Voltage detection (VBAT)

The over and under voltage comparators are based on comparing a divided voltage from the Load Dump limiter output feeding an analog comparator with hysteresis (refer to EC Table for electrical parameters and PMU description). The Over and Under voltage events generate Interrupts to the interrupt controller.

9.2.10 Temperature monitor

The MCU is in charge to pull the ADC related data from temperature sensor and in case the measured data is too high, MCU will reduce power profile to reduce heat. The table below shows the tempsensor output voltage corresponding to Tj from analog simulation for reference. Calibration is required due to different offset per chip.

Table 14 Tempsensor Output Voltage vs Junction Temp

Junction Temp (°C)	Tempsensor Output Voltage(V) @VBAT=13.5V
-40	0.5125
-30	0.5345
-20	0.5565
-10	0.5785
0	0.6005
10	0.6226
20	0.6446
30	0.6667
40	0.6887
50	0.7108
60	0.7328
70	0.7549
80	0.7770
90	0.7991
100	0.8212
110	0.8433
120	0.8654
130	0.8875
140	0.9097
150	0.9319

9.2.11 **Over Temperature detection**

The over temperature comparator monitors the junction temperature with hysteresis. The Over temperature event generates reset or interrupt to the interrupt controller.

9.2.12 **UART**

The UART circuit includes a 'baud' rate generator with software-programmable divider ratios for rates from 100kHz to 2MHz. The UART features a 32-bytes-deep receive/transmit FIFO that minimizes the microcontroller overhead command and provides a set of maskable interrupt support to monitor and manage TX and RX data paths. The UART does not include any HW to filter out received messages that are not matching the LIN ID address. In other words, the address associated with the master command will have to be decoded and executed if the ID matches. It is expected that given the clock source, the embedded FW may have to compensate for clock frequency drift over temperature to keep a reliable communication path between nodes.

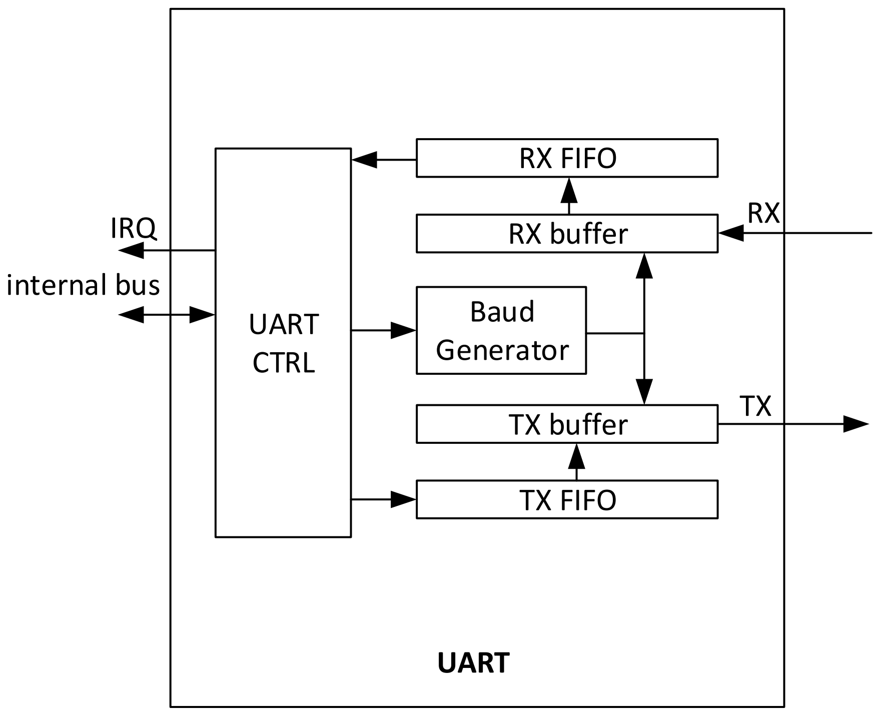


Figure 17 UART Interface

9.2.13 **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, 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.
- 32-byte deep reception FIFO, shared with UART
- 32-byte deep transmission FIFO, shared with UART
- Interrupt upon events related to transmission, reception, and error:
    - Write Collision
    - Transmission FIFO full and empty
    - Reception FIFO full and empty

9.2.14 **ASIC Watchdog Timer**

While it exists a watchdog timer in the MCU, the ASIC integrates another watch-dog timer that is intended to be used to recover from a situation where there may have been a software fault or other system failure where the software has ceased to operate correctly. If the timer is not reset by software periodically, it will time out, and this event can be used to reset the system.

Features:

- Programmable timeout period (Refer to EC table) with instantaneous access to the value of watchdog timeout counter
- Status flag, stop and clear/reset control registers
- The Watchdog timer is by default active after power up and set to its maximum duration setting (EC table)

9.2.15 **LED PWM**

LED PWMs are used to control accurately the light intensity.

Features:

- 24x16bit PWM channels with shared period length, independent pulse rise and independent pulse fall timestamps.
- Frequency and duty cycle of PWM waveform support up to a maximum Resolution of 16 bits. The 16-bit resolution is only achievable if the pre-scaler from system clock is correctly chosen to have enough clock cycles depth to count up to 2^16.
- Programmable pre-scaler: system clock division (PWM_DIV)

- Programmable PWM Period (PWM_PER)

$$\text{Period} = \frac{1 + (\text{PWM\_PER} \times \text{PWM\_DIV})}{SystemClock}$$

- Programmable duty cycle 0 to 100% (PWM_PW)

$$\text{PulseWidth} = \frac{1 + ((\text{PWM}_{\text{PFALL}} - \text{PWM}_{\text{PRISE}}) \times \text{PWM\_DIV}))}{SystemClock}$$

- Pre-scaler, period, pulse rise & pulse fall configurations will be updated at the end of the current output period.
- Support interrupt generation when new programmed PWM control data become active. After new pulse parameters (Period, pulse rise & pulse fall) have been loaded into their respective registers, an UPDATE bit can be set to 1 that will trigger the activation of the new parameters at the end of the current pulse as not to affect the pulse shape. Basically the UPDATE bit clear is the interrupt.
- PWM Frequency range 80 - 250 Hz
- Pulse rise -> fall cases, listed in priority:
    - PRISE = 0, PFALL = PERIOD: 100% On
    - PRISE = 0, PFALL = 0: equivalent to PRISE = 0 & PFALL = 1
    - PRISE > PFALL: 100% off
    - PRISE = PFALL: 100% off
    - PRISE < PFALL: Normal case. On at PRISE, Off at PFALL.

9.2.16 **Hibernate Mode**

IC must be able to enter SLEEP mode through SW request. The device should be able to come out of SLEEP with either the slow auxiliary or the system clock. It's up to the SW to select which clock shall be used after a wake event is detected. The SW should not have to request to go to sleep with the same clock selected for wake up.

9.2.17 **Wake up Mode**

Coming out from SLEEP mode can happen through the following events:

- After a low pulse on the LIN pin such that a dominant (low) voltage level is applied for longer than TWAKEUP time.
- GPIOs pin toggling either from high to low or low to high levels (VIL/VIH) and maintaining the active state for more than TWAKEUP time.

- Wake up timer. Programmable range, refer the EC Table. Wake up timer should have the option to be disabled.

MCU should be able to check which wake up events triggered the system through a status register read. GPIOs trigger should be consolidated into 1 status register. MCU to clear the register after status check.

9.2.18 **ASIC Watchdog Timer**

While it exists a watchdog timer in the MCU, the ASIC integrates another watch-dog timer that is intended to be used to recover from a situation where there may have been a software fault or other system failure where the software has ceased to operate correctly. If the timer is not reset by software periodically, it will time out, and this event can be used to reset the system or generate an interrupt .

Features:

- Programmable timeout period with instantaneous access to the value of watchdog timeout counter
- Works on a 16KHz clock independent of system clock
- Status flag, stop and clear/reset control registers
- The Watchdog timer is by default active after power up and set to its maximum duration setting
- Window mode supported: If enabled, WDTA can only be cleared after the watchdog window opened and before time-out period; Otherwise, WDTA will issue a system reset or an interrupt.

10 BOM

#	Description	Quantity	Comment
CBATx	2.2uF 50V X7R	1x	
CV1p5	4.7uF 10V	1x	
CV3p3	10uF 10V	1x	
CLIN	220pF 0603 X7R	2x	one CLIN in master and one CLIN in slave. Only one CLIN in slave if no auto addressing is required.
C1	20uF	2x	Buck converter output capacitor
CBST	100nF	1x	Boostrap cap
L1	Size (AxBxHeight?) L=10uH DCR<100mOhm	1x	Buck power inductor
C3		1x	Buck input capacitor

11 ERRATA

11.1 ADC CONTINUOUS MODE IRQ HANDLING

The hardware implementation of continuous mode can't be trusted as it's depending on the irq subroutine execution time vs the time required to do a set sequence of conversion. For a typical interrupt handler, clear interrupt flag is required in order to allow new interrupt to be fired. But for ADC in continuous mode, the interrupts occur very intensively, new interrupt can be fired while current interrupt handling is still not yet completed, in such use cases the IRQCLR is ineffective if an new interrupt event occurs at the same time, potentially it could block any new ADC interrupt event to be fired. To avoid of such conflicts, firmware needs do additional check in the interrupt handler to make sure ADC interrupt flag is cleared successfully.

Here is an example implementation of ISR where after IRQCLR, do a check to see if there is new ADC has been done, if yes, set IRQCLR to again.

```
void ADC_Handler(void)
{
   ADC_SFRS->CNTRL.IRQCLR = 1
   while(ADC_SFRS->STATUS.CONVDONE)
      ADC_SFRS->CNTRL.IRQCLR = 1;
}
```

Alternatively, user can choose not to use ADC in continuous mode and use single conversion and setting the CONVERT every time before leaving the IRQ handler.