iND83231 “Krankl”

indie’s highly integrated 400MHz transceiver with integrated MCU and high power supply

12/23/2015 **Preliminary Data sheet**

1 Document Revision History

Table 1 Revision History

Rev #	Date	Description	By
0.1	June 17th 2014	Initial version (collected from AyDeeKay documentation)	GKo
0.2	Aug. 24th 2014	1st release	GKo
1.1	Apr. 15th 2015	Updated: - Ported doc to product spec type with indie format - Adjusted reference to 12MHz RC - Pin list – naming updated – created pin mux table - Added summary reg tables for GIO, SIO, etc - Electrical table centralized	LF
1.2	12/23/15	Format change, modified table	CR
1.21	5/16/15	Product number change for consistency	PH

2 Table of Content

1 Document Revision History 2

2 Table of Content 3

3 General Description 5

3.1 CPU Architecture: 6

3.2 Memory: 6

3.3 ISM band 433.92MHz ASK/OOK Radio: 6

3.4 Peripherals: 6

3.5 Clocks 7

3.6 Reliability 7

3.7 Device overview 8

3.8 Pin Description 9

3.8.1 Pin Types 9
3.8.2 Pin List 10
3.8.3 Pin State upon Power-On Reset 11
3.8.4 Alternative pin functions 12

4 Absolute Maximum Ratings 14

5 Functional blocks 15

5.1 ARM Cortex M0 Microcontroller Subsystem 15

5.1.1 Memory Map 15
5.1.2 Timers 18
5.1.3 Watch Dog Timer (WDT) 21
5.1.4 Interrupt Vectors 23

5.2 RF Receiver Subsystem 25

5.2.1 RF Receiver Registers 27

5.3 RF Transmitter Subsystem 42

5.3.1 Fractional PLL 44
5.3.2 Transmission State Machine Operation 44
5.3.3 Chip Rate Selection 45
5.3.4 Transmitter registers 50

5.4 RF Switch 64

5.5 General Purpose ADC 8 Bit 65

5.5.1 ADC Description 65
5.5.2 ADC Clock and Sampling Period 66
5.5.3 Configuration of Reference Voltages for the ADC 67
5.5.4 ADC Start and Status 68

5.5.5 ADC Registers 69

5.6 Pulse Width Modulators (PWM) *77*

5.6.1 PWM Usage Description 77

5.6.2 PWM Registers 79

5.7 Zero Crossing Sense (ZCS) *86*

5.7.1 ZCS Usage Description 86

5.8 Digital Triac Timer *88*

5.8.1 Digital Triac Timer Usage Description 88

5.8.2 DTT Control Register Map 89

5.9 Serial Interfaces *96*

5.9.1 UART 96

5.9.2 SPI Interface 104

5.10 GPIOS *111*

5.10.1 General Purpose IO (GIO) 111

5.10.2 High Current Pull down IO (SIO) 112

5.10.3 GIO and SIO Connections to ADC 114

5.10.4 Port Configuration Registers 114

5.10.5 GIO PA Configuration Registers 121

5.10.6 SIO PC Configuration Registers 129

5.10.7 GPIO PD Configuration Registers 132

5.10.8 GPIO PE Registers 143

5.10.9 GPIO PF Registers 152

5.11 Clock sources *155*

5.11.1 Clock Source Operation and Description 155

5.11.2 Clock Related Registers 156

5.12 Power Management Unit *158*

5.12.1 PMU control of Reset 158

5.12.2 PMU control of sleep and deep sleep (halt) modes 159

5.12.3 PMU control of Brownout 159

5.12.4 PMU Registers 160

5.13 Wake Up Timer *163*

5.13.1 Wake-Up Timer Operation and Description 163

5.13.2 Wake-Up Timer Register 163

6 Electrical Characteristics 165

7 Bill of Material 170

8 Package Outline 171

9 References 172

10 Contacts 173

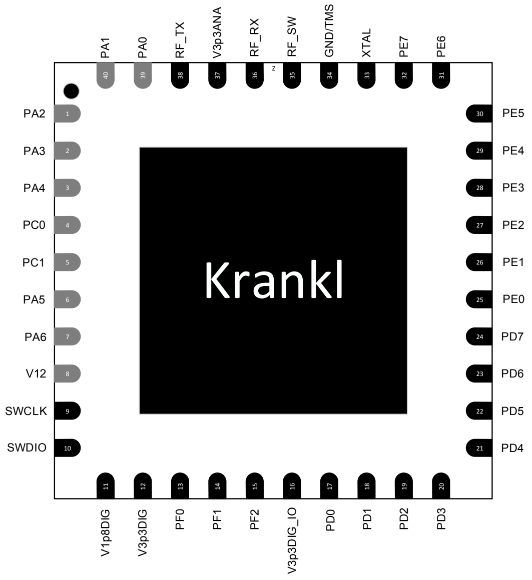
3 General Description

Main Features:

- **433.92 MHz ASK Transceiver with physical layer decoder**
- **ARM Cortex M0 32-bit µC - Low Power**
- **160kB embedded Flash Memory and 8kB SRAM**
- **Supports UART and SPI peripheral interfaces**
- **Single 30MHz Crystal clock oscillator**
- **Fully integrated Power management subsystem.**
- **Runs directly from high operating voltage 9V to 32V, with a maximum of 45V**
- **Nineteen 3.3V Low Voltage GPIOs**
- **Seven 12V GPIOs, sink up to 25mA to GND**
- **Two 12V high current sink GPIOs, sink up to 200mA to GND (Relay Driver)**
- **Protected high current pull downs**

Package:

QFN 6mm x 6mm 40-Pins



Top View

This ASIC has an integrated super-heterodyne ISM band, 433.92MHz, ASK transceiver. The oscillator frequency for the RF is generated by fractional-N type frequency synthesizers from a crystal oscillator. This ASIC integrates the following features:

3.1 CPU Architecture:

- ARM Cortex-M0 processor
- System Tick Timer (SysTick – 24 bits, interruptible)
- 3 additional 32-bit timers
- Serial Wire Debugger
- Built-in Nested Vectored Interrupt Controller (NVIC)
- Programmable Watch-Dog Timer

3.2 Memory:

- 160kByte of Flash Program Memory
- 8kByte of SRAM

3.3 ISM band 433.92MHz ASK/OOK Radio:

- A super-heterodyne receiver, with up to -110dBm of sensitivity
- A Transmitter supporting with a maximum output power of +13dBm
- Integrated fractional-N phase locked loop (PLL) referenced to crystal oscillator

3.4 Peripherals:

- Two 8 bits SAR ADC, with a total of 29 input channels connected to the GPIOs and 12V supply along with selectable input voltage references for increased resolution.
- 3 PWM channels
- 28 general purpose IO ports
  - 7 high voltage (9-35V) general purpose IO ports which can source 5mA or sink 25mA (PA[0-6])
  - 2 high voltage (9-35V) general purpose IO ports, which can sink 200mA in order to directly drive a relay coil (PC[0/1])
  - 19 low voltage (3.3V nominal) general purpose IO ports (PD[0-7],PE[0-7],PF[0-2])
- 2 Dedicated IOs for Zero Crossing and Power Sense Detection used for Light Timing and 3 Way Switch input.
- Integrated Power Management
- Dedicated Wake-up Timer
- Zero crossing Detector and Power Sense Functionality

3.5 Clocks

Krankl integrates three oscillators which may be used to generate a time base

- a 30MHz high accuracy crystal oscillator, which is used as a reference for the RF
- a 12MHz, 5% accurate RC oscillator, for current saving operation
- a 10kHz, low power oscillator permanently activated for house keeping

3.6 Reliability

Krankl is designed to withstand load dump events of up to 45V on its supply pin and on every high voltage IO. It is also designed to withstand electrical discharges to its 12V IOs according to ISO10605.

3.7 Device overview

Figure 1 depicts a high-level block diagram of the Krankl device. The application subsystems can be grouped into several types: RF receiver, RF transmitter, power management, general purpose IO ports, clock generation, zero crossing detect, power on reset, brown out reset, and micro controller unit.

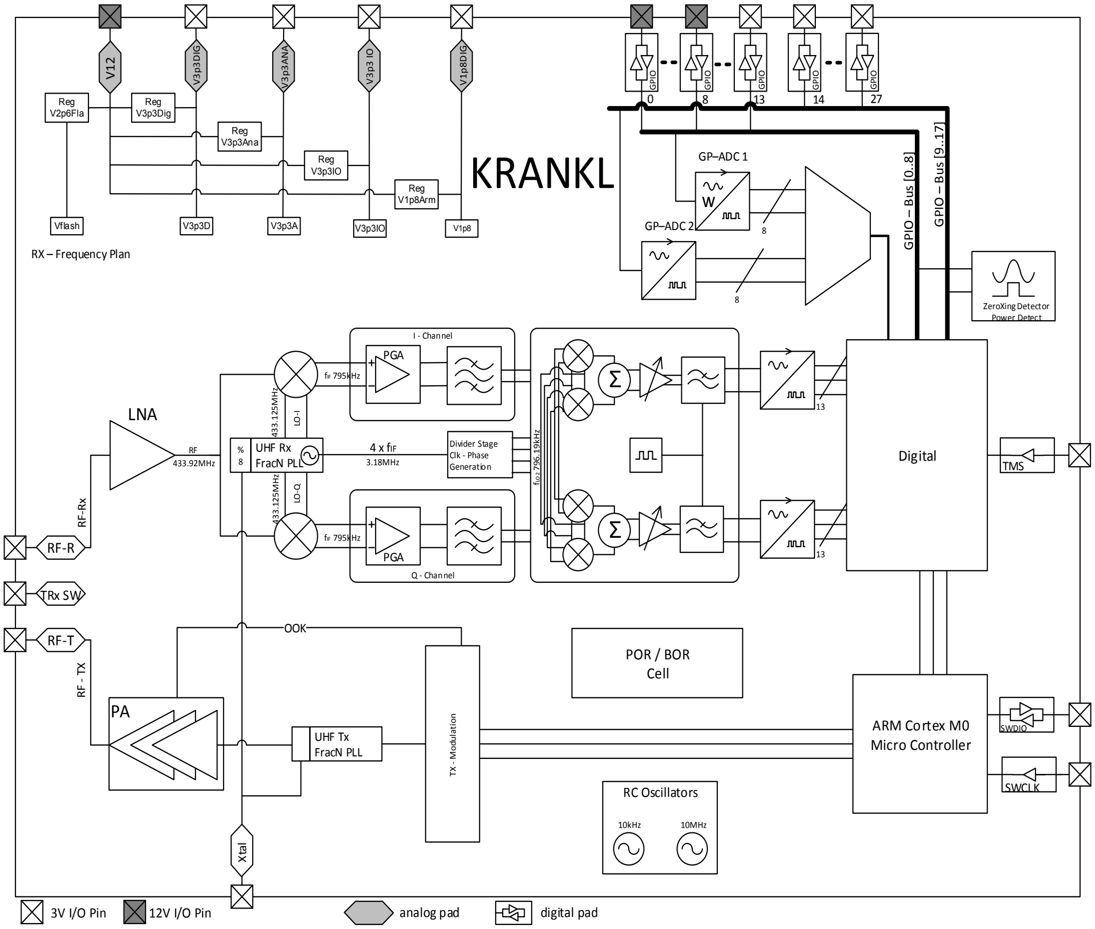


Figure 1 Functional Block

The microcontroller debug and access port consists of clock (SWCLK) and data (SWDIO) signals. The interface operates at a typical 3.3V CMOS level and is intended to handle communication between an external debugger and Krankl IC.

The general-purpose ports handle basic IO functions with any 12-24V system, such as sensing switch states and driving relays. Each driver is capable of 25mA pull-down and 5mA pull-up to Supply/ground levels when configured as outputs. A subset of the pins (SIO type) are capable of 200mA pull down with 5mA pull-up capability. The high current ports are intended to be used to drive relays, and will be protected against over-current, over-temperature.

The high-current ports (PC0/1) have an auto shutdown capability to place the port into tristate mode if a short-circuit condition is detected in the current path they control. The short-circuit is detected using a circuit which monitors the voltage drop across a PCB trace in-line with the current path, which should be connected to the appropriate differential sense pins on Krankl. Any time the current sensed is above a threshold (differential 1V across the PCB trace), the port will automatically enter a latched tristate mode, which can be recycled by software.

The clock generation subsystem consists of a single ended crystal oscillator. The crystal oscillator will be configured to operate with the crystal connected between a single pin and ground. This pin can be used alternately as an external clock source just by driving it with a digital level. When operating with a crystal, a 100nF series dc block capacitor is required between crystal and this pin in order to avoid premature aging of crystal and/or unstable oscillation.

The power management portion consists of a bandgap voltage reference and several linear voltage regulators. The voltage reference is used to provide the reference voltage for the voltage regulators, as well as for other circuits internal, such as, for example, the A/D converters. A voltage regulator provides power for the internal 3.3V digital logic (V3P3DIG) and the bus interface, and is available at a pin to allow placement of a bypass capacitor and to allow it to be used as the power source for the microcontroller. Another 3.3V regulator supplies power to the majority of on-chip analog functions (V3P3ANA). The ARM Cortex M0 micro controller unit requires a 2.6V power supply for the embedded flash memory and a 1.8V supply for the micro controller core unit, each is sourced from a separate voltage regulator. Krankl utilizes a dedicated regulator for its 3.3V GPIO circuitry.

The RF receiver is configured to receive and decode ASK (OOK) modulated messages with nominal carrier frequency of 433.92MHz ±100KHz. Messages are fully decoded and buffered without microcontroller intervention. The transmitter is capable to transmit ASK coded signal at 433.92MHz.

3.8 Pin Description

3.8.1 Pin Types

- SUPPLY=Supply Pin or Decoupling Pin for internal Supply Regulator
- ANAIN=Analog Input, ANAOUT=Analog Output, ANAIO=Analog Input/Output.
- DIGIO=Digital IO
- VREG= Regulated Output, SUPPLY=Supply input Pin
- GIO = General Purpose IO, 25mA pull-down and 5mA pull-up
- SIO = General Purpose IO with 200mA pull-Down and 5mA pull-up
- GND= Ground
- Not Connected (NC)

3.8.2 Pin List

Table 2 Pin List

#	Pin Name	Type	Voltage	Description
1	PA2	GIO	12V	High Voltage General Purpose IO
2	PA3	GIO	12V	High Voltage General Purpose IO
3	PA4	GIO	12V	High Voltage General Purpose IO
4	PC0	SIO	12V	High Voltage General Purpose IO with high current sink
5	PC1	SIO	12V	High Voltage General Purpose IO with high current sink
6	PA5	GIO	12V	High Voltage General Purpose IO
7	PA6	GIO	12V	High Voltage General Purpose IO
8	V12	SUPPLY	12V	12V Power supply
9	SWCLK	DIGIO	3.3V	SWD Debug/MCU clock access port
10	SWDIO	DIGIO	3.3V	SWD Debug/MCU data access port
11	V1P8DIG	SUPPLY	3.3V	Voltage regulator output: 1.8V
12	V3P3DIG	SUPPLY	3.3V	Voltage regulator output: 3.3V digital Core Supply
13	PF0	GPIO	3.3V	General Purpose IO
14	PF1	GPIO	3.3V	General Purpose IO
15	PF2	GPIO	3.3V	General Purpose IO
16	V3P3DIG_IO	SUPPLY	3.3V	Voltage regulator output: 3.3V IO supply
17	PD0	GPIO	3.3V	General Purpose IO
18	PD1	GPIO	3.3V	General Purpose IO
19	PD2	GPIO	3.3V	General Purpose IO
20	PD3	GPIO	3.3V	General Purpose IO
21	PD4	GPIO	3.3V	General Purpose IO
22	PD5	GPIO	3.3V	General Purpose IO
23	PD6	GPIO	3.3V	General Purpose IO
24	PD7	GPIO	3.3V	General Purpose IO
25	PE0	GPIO	3.3V	General Purpose IO

#	Pin Name	Type	Voltage	Description
26	PE1	GPIO	3.3V	General Purpose IO
27	PE2	GPIO	3.3V	General Purpose IO
28	PE3	GPIO	3.3V	General Purpose IO
29	PE4	GPIO	3.3V	General Purpose IO
30	PE5	GPIO	3.3V	General Purpose IO
31	PE6	GPIO	3.3V	General Purpose IO
32	PE7	GPIO	3.3V	General Purpose IO
33	XTAL	ANAIO	3.3V	IO Crystal Oscillator, Single clock input
34	GND/TMS	DIGIO	3.3V	Input test mode state, GND in mission mode
35	RF-SW	ANAIO	3.3V	TRX filter switch
36	RF-RX	ANAIN	3.3V	IO RX path 433MHz
37	V3P3ANA	SUPPLY	3.3V	Voltage regulator output, 3.3V analog supply
38	RF-TX	ANAOUT	3.3V	IO TX path 433MHz
39	PA0	GIO	12V	High Voltage General Purpose IO
40	PA1	GIO	12V	High Voltage General Purpose IO

3.8.3 Pin State upon Power-On Reset

- Unless otherwise noted, all pins default to tristate (high impedance) upon power-on reset.
- Pins otherwise noted include:
  - SUPPLY pins.
  - VREG pins: output a regulator voltage upon power-on reset are at specified voltage levels in the presence of 12V supply.
  - RF pins are configurable via SW to enter tristate

3.8.4 Alternative pin functions

Table 3 Pin Muxing

#	Pin Name	F0	F1 (PWX)	F2 (ADC)
1	PA2			ADC1
2	PA3			ADC2
3	PA4			ADC1
4	PC0		PWM2	ADC1
5	PC1			ADC2
6	PA5		PWM1	ADC2
7	PA6			ADC1
8	V12	-	-	-
9	SWCLK	-	-	-
10	SWDIO	-	-	-
11	V1P8DIG	-	-	-
12	V3P3DIG	-	-	-
13	PF0			ADC1
14	PF1			ADC2
15	PF2			ADC1
16	V3P3DIG	-	-	-
17	PD0			ADC1
18	PD1			ADC2
19	PD2		PWM2	ADC1
20	PD3		PWM1	ADC2
21	PD4		PWM3	ADC1
22	PD5		PWM3	ADC2
23	PD6			ADC1
24	PD7			ADC2
25	PE0	SPI_CS_n		ADC1
26	PE1	SPI_MISO		ADC2
27	PE2	SPI_SCK		ADC1

#	Pin Name	F0	F1 (PWX)	F2 (ADC)
28	PE3	SPI_MOSI		ADC2
29	PE4	UART_RX		ADC1
30	PE5	UART_TX		ADC2
31	PE6			ADC1
32	PE7			ADC2
33	XTAL	-	-	-
34	GND/TMS	-	-	-
35	RF-SW	-	-	-
36	RF-RX	-	-	-
37	V3P3ANA	-	-	-
38	RF-TX	-	-	-
39	PA0			ADC1
40	PA1			ADC2

4 Absolute Maximum Ratings

Table 4 Absolute Maximum Ratings

Absolute Maximum Ratings, Voltages Referenced to Ground				
Name	**Conditions**	**Min**	**Max**	**Unit**
12V Supply voltage (12V nominal)	Short duration, no long term damage	-0.3	+45	V
GIO voltage	configured as input, no damage	-0.3	V12+0.3	V
SIO voltage	configured as input, no damage	-0.3	V12+0.3	V
3.3V GPIO	configured as input, no damage	-0.3	V3p3DIG+0.3	V
3V Analog IO	XTAL and RF, no damage	-0.3	V3p3AN+0.3	V
Operating Temp.	de-rated performance, full functionality	-40	+85	°C
ESD/Transient Robustness				
Storage Temperature	de-rated performance, full functionality	-55	+150	°C
HBM (all pins)	any pin to any other pin or ground, loose part	-4	4	kV
HBM (GIO, SIO)	to other pin in group or ground, soldered to application board	-6	6	kV
CDM	no damage	-300	300	V
MM	no damage	-150	150	V

Stresses beyond those listed under absolute maximum ratings may cause permanent damage to the device. These are stress ratings only, and 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.

5 Functional blocks

5.1 ARM Cortex M0 Microcontroller Subsystem

Krankl device includes an embedded microcontroller subsystem, which is based on the ARM Cortex M0 core. It includes a program flash memory of 160kBytes, and an SRAM of 8kBytes. It includes three 32-bit timers, plus a dedicated watchdog timer. Additionally, it includes a **N**ested **V**ector **I**nterrupt **C**ontroller (NVIC) to scheduled hardware interrupts, and a **W**akeup **I**nterrupt **C**ontroller (WIC), which enable the control of the various power modes.

5.1.1 **Memory Map**

Table 5 System Memory Map

System Memory Map			
Address	**Memory**	**Description**	**Reference**
0x00000000 - 0x00027FFF	Flash	160kByte Flash Memory	N/A
0x00028000 - 0x0003FFFF	N/A	Reserved	N/A
0x00040000 - 0x000400FF	Flash	256Byte 1st NVR Sector	N/A
0x00040100 - 0x000401FF	Flash	256Byte 2nd NVR Sector	N/A
0x00040200 - 0x1FFFFFFF	N/A	Reserved	N/A
0x20000000 - 0x20001FFF	SRAM	8kByte SRAM	N/A
0x20002000 - 0x4FFFFFFF	N/A	Reserved	N/A
0x50000000 - 0x5000007F	Peripheral	128Byte peripheral fast access	
0x50000080 - 0x50000085	Peripheral	6Byte Block Transfer control	N/A
0x50000086 - 0x5000FFFF	N/A	Reserved	N/A
0x50010000 - 0x5001FFFF	Peripheral	64kByte peripheral slow access	

System Memory Map			
0x50020000 - 0x5002001F	Peripheral	32Byte timer control	N/A
0x50020020 - 0x50020047	Flash	40Byte Flash program/erase control	N/A
0x50020048 - 0xDFFFFFFF	N/A	Reserved	N/A
0xE0000000 - 0xE00FFFFF	Private peripheral bus	ARM peripherals	N/A
0xE0100000 - 0xEFFFFFFF	N/A	Reserved	N/A
0xF0000000 - 0xF0001FFF	System ROM tables	ARM core IDs	N/A
0xF0002000 - 0xFFFFFFFF	N/A	Reserved	N/A

Table 6 Peripheral Fast Access Memory Map

Peripheral Fast Access Memory Map			
Address	Peripheral	Description	Reference
0x50000000 - 0x50000005	PMU	Power management unit control	
0x50000008 - 0x5000000F	ADC	ADC control	
0x50000010 - 0x50000017	UART	UART control	
0x5000001C - 0x5000001F	SPI	SPI control	
0x50000020 – 0x50000031	RF RX	RF Receiver control	
0x50000032 – 0x5000003F	RF TX	RF Transmitter control	
0x50000040 - 0x5000004F	PWM	Pulse width modulator control	
0x50000050 - 0x5000005F	DTT	Digital Triac Timer control	
0x50000060 - 0x5000007F	GPIO	GPIO Byte control	

Table 7 Peripheral Slow Access Memory Map

Peripheral Slow Access Memory Map			
Address	Peripheral	Description	Reference
0x50010000 - 0x500104FF	GPIO	GPIO bit control	
0x50000008 - 0x5000000C	ADC 1	ADC and miscellaneous control	
0x50010004 - 0x50010007	TX PLL	RF TX Synthesizer Setting	
0x50018004 - 0x50018007	RX PLL	RF RX Synthesizer Setting	
0x50018008 - 0x5001800B	ADC 2	ADC Setting	
0x50000000	CLK Source	RC Oscillator and XTAL Setting	
0x50000060 - 0x5000006A		Port Control	

5.1.2 Timers

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

5.1.2.1 *Timer registers*

TMR0REG: 32-bit Timer initial value register							
TMR0REG		0x50020000			0x00000000		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
T7	T6	T5	T4	T3	T2	T1	T0
T15	T14	T13	T12	T11	T10	T9	T8
T23	T22	T21	T20	T19	T18	T17	T16
T31	T30	T29	T28	T27	T26	T25	T24
MSB							LSB
Bit31-0 **T[31:0]**: Timer Register initial value register.							

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

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

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

TMR2REG: 32-bit Timer initial value register							
TMR2REG		0x50020010			0x00000000		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
T7	T6	T5	T4	T3	T2	T1	T0

TMR2REG: 32-bit Timer initial value register							
T15	T14	T13	T12	T11	T10	T9	T8
T23	T22	T21	T20	T19	T18	T17	T16
T31	T30	T29	T28	T27	T26	T25	T24
MSB							LSB
Bit31-0 **T[31:0]**: Timer Register initial value register.							

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

5.1.2.2 *Timer Operation*

The operation of the timers is quite straightforward: Load the initial counter register, enable the timer and either check (polling mode) the current value of the counter register or enable the interrupt and process it inside the interrupt service routine.
Note: Inside the interrupt the application code must reload the timer counting register.

Code Example1: Enable Timer1 to count from 0xFFFF0000 and to generate interrupt:

```
TMR_Config( 1, TIMERON, 0xFFFF0000);  //Enable timer1 to count up from 0xFFFF0000
NVIC_ClearPendingIRQ( TIMER1_IRQn );        //Clear pending interrupt
NVIC_EnableIRQ( TIMER1_IRQn );        //Enable Timer1 interrupt
```

```
void Timer1_Handler( void )
{
  *TMR1REG = 0xFFFF0000;             //Reload Register
  //**** From this point application code inside ISR****
}
```

5.1.3 Watch Dog Timer (WDT)

Krankl implements a WDT (**W**atch **D**og **T**imer) that can operate in one of two basic 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.

5.1.3.1 *WDT Registers*

The Watch Dog Timer implements two 32-bit registers:

WDTCTRL: WDT Control Register. (32-bit)							
WDTCTRL		0x50020018			0x0000000x		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	WDTPRES 1	WDTPRES 0	RSTFLAG	RESETEN	WDTEN
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
MSB							LSB

WDTCTRL: WDT Control Register. (32-bit)

Bit4-3 **WDTPRES1: WDTPRES0**: WDT Prescaler:

00 = 2^{13}/SystemClock

01 = 2^{19}/SystemClock

10 = 2^{22}/SystemClock

11 = 2^{32}/SystemClock

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

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

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

For instance, a system running from a 30MHz Crystal with WDTPRES[1…0] = 10 will trigger the WDT after approximately 0.14seconds if not cleared properly and in time by the application.

WDTCLR: WDT Clear Register. (32-bit)

WDTCLR		0x5002001C			0x0000000x		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
WCLR7	WCLR6	WCLR5	WCLR4	WCLR3	WCLR2	WCLR1	WCLR0
WCLR15	WCLR14	WCLR13	WCLR12	WCLR11	WCLR10	WCLR9	WCLR8
WCLR23	WCLR22	WCLR21	WCLR20	WCLR19	WCLR18	WCLR17	WCLR16
WCLR31	WCLR30	WCLR29	WCLR28	WCLR27	WCLR26	WCLR25	WCLR24
MSB							LSB

WDTCLR: WDT Clear Register. (32-bit)
Bit31-0 **WCLR[31:0]**: Clear Register. To clear the WDT counting the following words must be written in this order and without any other instruction between then: 0x3C570001 0x007F4AD6 **Warning:** Programming WDTCLR with other values or in the wrong order will cause the watchdog to throw an interrupt or reset the system.

Example Code: Setting and clearing the WDT. (Interrupt mode with a time of 2^22)

```
WDT_Config(WDT_INT, WDT22);   //Enable  WDT  in  interrupt  mode  (2^22  system  clock
cycles)
WDT_Clear();                          //Clear WDT
```

5.1.4 Interrupt Vectors

Krankl implements an interrupt vector defined in the following table:

Table 8 Interrupt Vector table

Cortex M0 Specific Exceptions			
Name	**Number**	**Comments**	**Required Interrupt Handler (Function)**
HardFault_IRQn	-13	HardFault handler*	HardFault_Handler (void)
SVCall_IRQn	-5	Supervisory call*	
PendSV_IRQn	-2	Interrupt-driven request for system level service*	
SysTick_IRQn	-1	SysTick Timer interrupt	void SysTick_Handler(void)

Cortex M0 Specific Exceptions			
Name	**Number**	**Comments**	**Required Interrupt Handler (Function)**
BrownOut_IRQn	0	Brownout detection interrupt	void BrownOut_Handler (void)
ClkMon_IRQn	1	Clock monitor interrupt	void ClkMon_Handler (void)
-	2	*Reserved*	
PIN_IRQn	3	Pin change interrupt	void PIN_Handler (void)
RFRE_IRQn	4	RF: Rising Edge base band signal reception interrupt	void RFRE_Handler (void)
RFFE_IRQn	5	RF: Falling Edge base band signal reception interrupt	void RFFE_Handler (void)
TX_done_IRQn	6	*RF: Burst transmission done interrupt*	void TX_done_IRQ_Handler (void)
TX_reload_IRQn	7	*RF: Transmission FIFO reload interrupt*	void TX_reload_IRQ_Handler (void)
UART_IRQn	8	*UART*	void UART_Handler (void)
-	9	*Reserved*	void Default_IRQ_Handler (void)
SPI_IRQn	10	*SPI*	void SPI_Handler (void)
-	11	*Reserved*	void Default_IRQ_Handler (void)
RFMSG_IRQn	12	RF: Message received interrupt	void RFMSG_Handler (void)
IRQ13_IRQn to IRQ15_IRQn	13-15	*Reserved*	void Default_IRQ_Handler(void)
TIMER0_IRQn	16	Timer0 interrupt	void Timer0_Handler (void)
TIMER1_IRQn	17	Timer1 interrupt	void Timer1_Handler (void)

TIMER2_IRQn	18	Timer2 interrupt	void Timer2_Handler (void)
WATCHDOG_IRQn	19	Watchdog timer interrupt	void Watchdog_Handler (void)

*Note: For more information see *Cortex-M0 Devices – Generic Users Guide (ARM DUI 0497A (ID112109))* at: http://infocenter.arm.com/help/topic/com.arm.doc.dui0497a/DUI0497A_cortex_m0_r0p0_generic_ug.pdf

5.2 RF Receiver Subsystem

Krankl implements a programmable ISM (Industrial, scientific and medical band, 300-450MHz) OOK (on-off keying) low-IF receiver. The local oscillator is generated using a fully integrated fractional-N PLL referenced to an external crystal reference. The received data is digitized using analog to digital converters before being processed by an autonomous digital section.

The receiver uses Weaver architecture for image rejection, primarily to avoid noise imaging. After amplification through an LNA, a RF mixer is used to generate I/Q signals at the IF frequency of 795KHz, where it is filtered to ~500KHz bandwidth. After the second frequency conversion, the I and Q signals are filtered to ~150kHz bandwidth. The wide bandwidth relative to the data symbol rate is necessary to accommodate manufacturing variation in the transmit and receive frequency references.

The frequency generation for the local oscillators is accomplished using a PLL locked to the crystal frequency. The VCO is a low current quadrature ring oscillator. It is expected that 30MHz crystal in combination with a divide by 8 prescaler stage will be utilized for frequency reference. FXO is equivalent to 3.75MHz. In the default frequency plan, the first LO is generated from 115.5*fXO = 433.125MHz using the PLL in frac-N mode. The second LO is generated by dividing the first LO, first by a high speed divide-by-eight prescalar, followed by a programmable divider and a quadrature divide-by-four. The default frequency plan uses divide by 17 for the programmable part, for a second LO of 796.2kHz. Therefore the overall division ratio for the LO2 signals equal 544 (8x17x4), to be verified in the frequency table below.

Table 9

F_RX	F_REF	F_EN	NX	NF	HISD	DIV_LO2	F_LO1	F_IF	F_LO2	Freq_error
MHz	MHz	{0,1}	{1..255}	{0..255}	{0,1}	{0..3}	MHz	kHz	kHz	kHz
433.92	3.75	1	115	128	1	1	433.125	795.0	796.19	-1.186

Due to the slight difference between the LO and IF frequencies, there is a 1.2kHz frequency offset in the baseband data, which appears as a slight additional transmit frequency error to the decoder. For other crystal frequencies, different settings will need to be programmed for the PLL and divider as described below.

After analog filtering, the baseband signal is then digitized at 298kS/s using a 12bit ADC. The digitized signal is dc-offset corrected and AM detected using a CORDIC to produce an AM baseband signal, which is filtered and decimated to an approximately 5kHz bandwidth with 18.6kS/s data rate.

Data is digitally extracted from filtered baseband signal using a digital bit slicer. An integrated decoder may be utilized to decode 1/3-2/3 duty-cycle encoded data. Decoded bits are stored in a bit buffer with capability to store messages as long as 80 bits. Once an entire valid message is stored in the RF bit buffer, an interrupt is generated. The receiver then enters an armed state, but with decoder inactive until the microcontroller re-enables the receiver to receive subsequent messages. The microcontroller should read any data from the bit buffer before re-enabling or else it will be lost.

Alternatively, if an application requires a coding scheme other than 1/3-2/3 coding, the slicer digital output may be made available in real time for the micro to decode the signal by software. The raw signal is guaranteed glitch-free, allowing a simple decode.

The receiver can be run in an autonomous “sniffing” mode with, for example, a 5% on-time duty cycle, in order to save power. Whether in sleep mode or not, intervention by the microcontroller is only required upon reception of an entire message with valid number of bits. The micro can therefore be asleep during normal RF reception, and only needs be awoken by interrupt after an entire message arrives, allowing significant power savings.

Additionally, functional control such as enabling and disabling the receiver, and the received bits are controllable through registers. The register section describes the functions of various registers related to the receiver. All control registers preset when power-on reset or software reset is asserted.

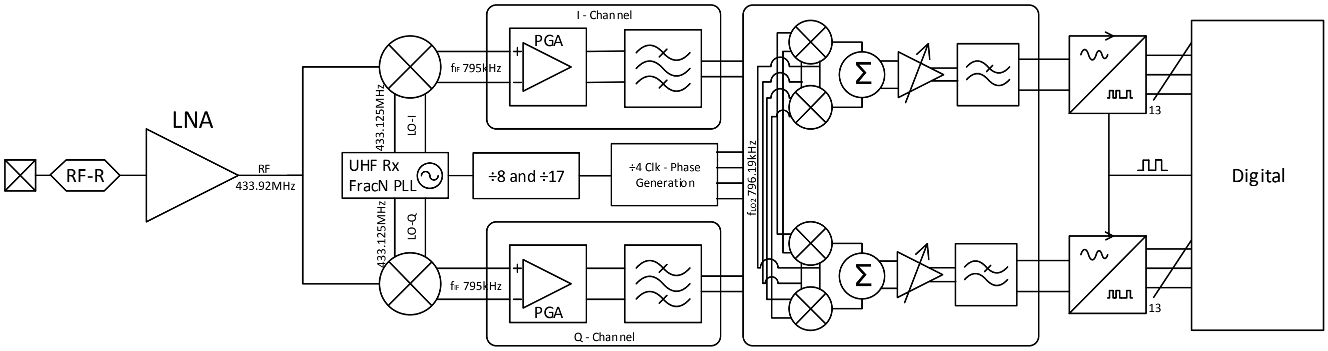


Figure 2 RF Receiver subsystem

5.2.1 RF Receiver Registers

RF Receiver Register Map			
Address	**Register Name**	**Field Name**	**Description**
0x50000020	RF_Buff0-F	RXDATA 0-127	RF Buffer Registers containing received bits.
0x50000030	RF_NUMB	RFNUMB0-7	Returns the number of bits contained in the bit buffer
0x50000031	RF_STATUS	RF_SLEEP	RF sleep mode indicator
		SLC_OUT	Slicer output
		DCDMD0	Determines whether state machine transitions on edge or level
		DCDMD1	Determines what to do if there is a overly-long non-guard band element
		ADC_FLG	Overflow indicator
		MSG_RDY	Message ready indicator
		SNF_EN	Enables sniff/sleep mode in the receiver
		RF_EN	Enables RF blocks
0x50011000	RF_NBMIN	Low_BPS	Indication to interpret all bit timings as 2X (slow transmitter)
		NBMIN[6:0]	Minimum number of bits for a valid transaction
0x50011001	RF_AGCCTRL	AGC_EN	Automatic gain control enable
		AGCTRM[6:0]	Controls gain of analog blocks
0x50011002	RF_SLCCTRL	ALPHA[1:0]	Controls decay time of slicer level
		BETA[1:0]	Controls attack time of slicer level
		FTIME[1:0]	Controls the time to allow the slicer to fast bias
		DR_SYM[1:0]	Sets post-CORDIC decimation rate
0x50011003	RF_SYSTIME	PLLTIME[1:0]	Controls time to wait for PLL to bias

		SLEEPTIME[2:0]	Controls sleep time between sniff cycles
		WAKETIME[2:0]	Controls the time to stay awake after seeing a valid guard band
0x50011004/5	RF_DCDTIME	MIN_GB[5:0]	Minimum number of additional samples after passing MIN_TE and MAX_TE for a low element to be considered a valid guard band length
		MAX_TE[5:0]	Maximum number of additional samples after passing MIN_TE for an edged to be considered short enough to be valid
		MIN_TE[3:0]	Minimum number of samples for a valid element time
0x50011006	RF_SNIFMODE	RT_SEL[1:0]	Selects a source for real time output
		RF_ON	Force analog RF to stay on
		SNIFF_NE[3:0]	Number of edges to check in each sniff cycle before committing to a long wake cycle
0x50011007	RF_AGCMON	AGCDATA[6:0]	RF gain control value
0x50011008/9	RF_SIGI	CAL_I[11:0]	Channel I DC calibration value
0x5001100A/B	RF_SIGQ	CAL_Q[11:0]	Channel Q DC calibration value
0x5001100C/D	RF_SLCHI	PDHI[15:0]	Peak detector high value
0x5001100E/F	RF_SLCLO	PDLO[15:0]	Peak detector low value
0x50018004	RF_NX	NX[7:0]	Integer part of the divider value for PLL divider
0x50018005	RF_NF	NF[7:0]	Fractional part of the divider value for PLL divider
0x50018006	RF_FETRIM0	HI_SD	Controls image rejection mixer to use high side or low side mixing
		FE_EN	Controls fractional mode

		CP_TRIM	Adjusts charge pump current in PLL for optimum settling time
		LF_TRM	Adjust loop filter stabilization resistor to control overshoot
0x50018007	RF_FETRIM1	DIV_LO2	Divider value from PLL frequency to LO2 frequency
		BBGAIN[1:0]	Sets baseband amplifier gain
		LNA_DRES	LNA drain resistor
		LNA_PG[2:0]	Gain selector

The following registers define the behavior of the RX-RF:

RF_BUFF0-F: RF Buffer Registers containing received bits.							
RF_BUFF0-F		0x50000020-2F			0xXX		
R	R	R	R	R	R	R	R
RXDATA7	RXDATA6	RXDATA5	RXDATA4	RXDATA3	RXDATA2	RXDATA1	RXDATA0
RXDATA15	RXDATA14	RXDATA13	RXDATA12	RXDATA11	RXDATA10	RXDATA9	RXDATA8
:	:	:	:	:	:	:	:
RXDATA119	RXDATA118	RXDATA117	RXDATA116	RXDATA115	RXDATA114	RXDATA113	RXDATA112
RXDATA127	RXDATA126	RXDATA125	RXDATA124	RXDATA123	RXDATA122	RXDATA121	RXDATA120
MSB							LSB
Bit127-0 **RXDATA[7:0]**: Received data bits. Most recently received bit is stored in RXDATA0. This register should be read when a complete message is ready (determined by reading MSG_RDY bit or having received an interrupt from the RF system) for repeatable results							

RF_NUMB: Returns the number of bits contained in the bit buffer.

RF_NUMB: Returns the number of bits contained in the bit buffer.							
RF_NUMB		0x50000030			0xXX		
R	R	R	R	R	R	R	R
RFNUMB7	RFNUMB6	RFNUMB5	RFNUMB4	RFNUMB3	RFNUMB2	RFNUMB1	RFNUMB0
MSB							LSB
Bit7-0 **RFNUMB [7:0]:** Returns the number of received data bits contained in the bit buffer. For repeatable results, it is recommended to only read this register when a complete message is ready							

RF_STATUS: RF Status Register.							
RF_STATUS		0x50000031			0xXX		
R/W	R/W	R	R	Reserved	Reserved	R	R
RF_EN	SNF_EN	MSG_RDY	AGC_FLG	DCDMD1	DCDMD0	SLC_OUT	RF_SLEEP
MSB							LSB

RF_STATUS: RF Status Register.

Bit7 **RF_EN**: Enables RF block.

0 = Disable the receiver

1 = Enable the receiver

Bit6 **SNF_EN**: Enables sniff/sleep mode in the receiver

0 = Continuous mode

1 = Sniff/Sleep mode

Bit5 **MSG_RDY**: Message ready indicator. Reading returns one if a complete message is available and the receiver is in armed mode, otherwise returns zero. Write is ignored unless in armed mode, where writing a zero causes the receiver to leave the armed state and start decoding message again

Bit4 **AGC_FLG**: AGC overflow indicator. Returns one if an overflow (signal too large) has occurred in receiver. Intended for use in continuous mode, where the micro may want to reduce the gain setting

Bit3 **DCDMD1**: Determines what to do if there is an overly-long non-guard band element

0 = reset the decoder and start looking for new message

1 = wait until a guard band is received before resetting state machine

Bit2 **DCDMD0**: Determines whether state machine transitions on edge or level

0 = transition on edge

1 = transition on level

Bit1 **SLC_OUT**: Slicer Output

Bit0 **RF_SLEEP**: RF sleep mode indicator

0 = RF RX is active

1 = RF RX is sleeping

RF_NBMIN: RF NB MIN Register.

RF_NBMIN		0x50011000			0x28		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
LOW_BPS	NBMIN6	NBMIN5	NBMIN4	NBMIN3	NBMIN2	NBMIN1	NBMIN0

RF_NBMIN: RF NB MIN Register.							
MSB							LSB

Bit7 **LOW_BPS**: Indication to interpret all bit timings as 2X (for slow transmitters)

0 = normal rate

1 = count bit timings at half rate

Bit6-0 **NBMIN[6:0]**: Minimum number of bits for a valid message

RF_AGCCTRL: RF Automatic Gain Control Register.							
RF_AGCCTRL		0x50011001			0x98		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
AGC_EN	AGCTRM6	AGCTRM5	AGCTRM4	AGCTRM3	AGCTRM2	AGCTRM1	AGCTRM0
MSB							LSB

RF_AGCCTRL: RF Automatic Gain Control Register.

Bit7 **AGC_EN**: Automatic Gain Control Enable

0 = Fixed gain mode

1 = AGC enabled

Bit6-0 **AGCTRM[6:0]**: Controls gains of analog blocks

If AGC_EN is zero, then the bits in the control registers are used to directly control the AGC trim of the analog blocks. If AGE_EN is one, then the bits sets the default gain and reduction step

Bit4 (0 = gain steps down the table in increment of 1; 1= gain steps down the table in increment of 2)

Bit3-0	st3	st2	LNA	gain	delta
0000	00	00	000	-4.7dB	
0001	00	00	010	-1.0dB	3.7dB
0010	00	00	011	4.5dB	5.5dB
0011	00	00	100	9.7dB	5.2dB
0100	00	00	101	15.4dB	5.7dB
0101	00	00	110	21.0dB	5.6dB
0110	00	00	111	26.8dB	5.8dB
0111	00	10	111	32.4dB	5.6dB
1000	10	11	111	38.6dB	6.2dB (default)
1001	11	11	111	45.5dB	6.9dB
1010	11	11	111	52.6dB	7.1dB
1011	not allowed				
11xx	not allowed				

RF_SLCCTRL: RF Slicer Control Register

RF_SLCCTRL		0x50011002			0xAB		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ALPHA1	ALPHA0	BETA1	BETA0	FTIME1	FTIME0	DR_SYM1	DR_SYM0

RF_SLCCTRL: RF Slicer Control Register

MSB							LSB

Bit7-6 **ALPHA[1:0]**: Controls decay time of slicer level. When input is inside slicer levels, slicer decays according to equation (clocked at decimated data rate):

y[n] = (1-ALPHA)*y[n-1] + ALPHA*x[n]

00 = 1/256 (fastest decay rate)

01 = 1/512

10 = 1/1024

11 = 1/2048 (slowest decay rate)

Bit5-4 **BETA[1:0]**: Controls attack time of slicer level. When input is outside slicer levels, slicer grows according to equation (clocked at decimated data rate):

y(n) = (1-BETA)*y[n-1] + BETA*x[n]

00 = 1/2 (fastest attack rate)

01 = 1/4

10 = 1/8

11 = 1/16 (slowest attack rate)

Bit3-2 **FTIME[1:0]**: Controls the time to allow the slicer to fast bias, measured in clock cycles of fxo/64

00 = 1 cycle (17us for 3.75MHz clock (30MHz crystal / 8))

01 = 32 cycles (546us for 3.75MHz clock)

10 = 48 cycles (819us for 3.75MHz clock)

11 = 64 cycles (1.09ms for 3.75MHz clock)

Bit1-0 **DR_SYM[1:0]**: Sets post-CORDIC decimation rate.

00 = 13X

01 = 14X

10 = 15X

11 = 16X

RF_SYSTIME: RF System Time Register.

RF_SYSTIME: RF System Time Register.

RF_SYSTIME		0x50011003			0x5A		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PLLTIME1	PLLTIME0	SLPTIME2	SLPTIME1	SLPTIME0	WAKETIME2	WAKETIME1	WAKETIME0
MSB							LSB

RF_SYSTIME: RF System Time Register.

Bit7-6 **PLLTIME[1:0]**: Controls the time to wait for PLL to bias.

Measured in clock cycles of the fxo/64

00 = 32 cycles (546us for 3.75MHz clock (30MHz crystal / 8))

01 = 48 cycles (819us for 3.75MHz clock)

10 = 64 cycles (1.09ms for 3.75MHz clock)

11 = 128 cycles (2.18ms for 3.75MHz clock)

Bit5-3 **SLPTIME[2:0]**: Controls the sleep time between sniff cycles.

Measured in clock cycles of the fxo/64.

000 = 4*1024 cycles (70ms for 3.75MHz clock (30MHz crystal / 8))

001 = 6*1024 cycles (105ms for 3.75MHz clock)

010 = 8*1024 cycles (140ms for 3.75MHz clock)

011 = 10*1024 cycles (175ms for 3.75MHz clock)

100 = 12*1024 cycles (209ms for 3.75MHz clock)

101 = 14*1024 cycles (245ms for 3.75MHz clock)

110 = 32*1024 cycles (559ms for 3.75MHz clock)

111 = 128*1024 cycles (2.27s for 3.75MHz clock)

Bit2-0 **WAKETIME[2:0]**: Controls the time to stay awake after seeing a valid guard band.

Measured in clock cycles of the fxo/64.

000 = 8*1024 cycles (140ms for 3.75MHz clock (30MHz crystal / 8))

001 = 10*1024 cycles (175ms for 3.75MHz clock)

010 = 12*1024 cycles (209ms for 3.75MHz clock)

011 = 14*1024 cycles (245ms for 3.75MHz clock)

100 = 16*1024 cycles (280ms for 3.75MHz clock)

101 = 18*1024 cycles (315ms for 3.75MHz clock)

110 = 24*1024 cycles (419ms for 3.75MHz clock)

111 = 32*1024 cycles (586ms for 3.75zMHz clock)

RF_DCDTIME: RF Decode Time Control Register.

RF_DCDTIME: RF Decode Time Control Register.							
RF_DCDTIME		0x50011004/5			0x3614		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MAX_TE1	MAX_TE0	MIN_GB5	MIN_GB4	MIN_GB3	MIN_GB2	MIN_GB1	MIN_GB0
MIN_TE3	MIN_TE2	MIN_TE1	MIN_TE0	MAX_TE5	MAX_TE4	MAX_TE3	MAX_TE2
MSB							LSB

Bit5-0 **MIN_GB[5:0]**: Minimum number of ADDITIONAL samples after passing MIN_TE and MAX_TE for a low element to be considered a valid guard band length. Default setting with 3.75MHz clock (30MHz crystal / 8). Corresponds to minimum guard band time of 3.4ms

Bit11-6 **MAX_TE[5:0]**: Maximum number of ADDITIONAL samples after passing MIN_TE for an edge to be considered short enough to be valid. Default setting with 3.75MHz clock (30MHz crystal / 8). Corresponds to maximum element time of 1.4ms

Bit15-12**MIN_TE[3:0]**: Minimum number of samples for a valid element time. Counted in decimated data rate (fxo/(12*DR_SYM)). Default setting with 3.75MHz clock (30MHz crystal / 8). Corresponds to minimum element time of 153us

RF_SNIFMODE: RF Sniff Mode Register.							
RF_SNIFMODE		0x50011006			0x0B		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RT_SEL1	RT_SEL0	RF_ON	Reserved	SNIFF_NE3	SNIFF_NE2	SNIFF_NE1	SNIFF_NE0
MSB							LSB

RF_SNIFMODE: RF Sniff Mode Register.

Bit7-6 **RT_SEL[1:0]**: Selects a source for real-time output.

00 = supervisor clock

01 = decimator output (serialized stream)

10 = slicer output

11 = PLL_EN (high if analog blocks enabled)

Bit5 **RF_ON**: Force analog RF to stay on (for test).

0 = Normal

1 = RF stays on for test

Bit3-0 **SNIFF_NE[3:0]**: Number of edges to check in each sniff cycle before committing to a long Wake cycle. When in sniff mode, the first SNIFF_NE edges are tested for valid timing. If any one of these first edges is badly timed, then the receiver will go to sleep

RF_AGCMON: AGC Monitor Register

RF_AGCMON		0x50011007			0xXX		
R	R	R	R	R	R	R	R
Reserved	AGCDATA 6	AGCDATA 5	AGCDATA 4	AGCDATA 3	AGCDATA 2	AGCDATA 1	AGCDATA 0
MSB							LSB

Bit6-0 **AGCDATA[6:0]**: RF gain control value

RF_SIGI: I Channel Calibration Monitor Register

RF_SIGI		0x50011008/9			0xXXXX		
R	R	R	R	R	R	R	R
CAL_I7	CAL_I6	CAL_I5	CAL_I4	CAL_I3	CAL_I2	CAL_I1	CAL_I0
Reserved	Reserved	Reserved	Reserved	CAL_I11	CAL_I10	CAL_I9	CAL_I8

RF_SIGI: I Channel Calibration Monitor Register							
MSB							LSB
Bit11-0**CAL_I[11:0]**: Channel I DC calibration value							

RF_SIGQ: Q Channel Calibration Monitor Register							
RF_SIGQ		0x5001100A/B			0xXXXX		
R	R	R	R	R	R	R	R
CAL_Q7	CAL_Q6	CAL_Q5	CAL_Q4	CAL_Q3	CAL_Q2	CAL_Q1	CAL_Q0
Reserved	Reserved	Reserved	Reserved	CAL_Q11	CAL_Q10	CAL_Q9	CAL_Q8
MSB							LSB
Bit11-0**CAL_Q[11:0]**: Channel Q DC calibration value							

RF_SLCHI: Peak Detector High Value Monitor Register							
RF_SLCHI		0x5001100C/D			0xXXXX		
R	R	R	R	R	R	R	R
PDHI7	PDHI6	PDHI5	PDHI4	PDHI3	PDHI2	PDHI1	PDHI0
PDHI15	PDHI14	PDHI13	PDHI12	PDHI11	PDHI10	PDHI9	PDHI8
MSB							LSB
Bit15-0**PDHI[15:0]**: Peak detector high value							

RF_SLCLO: Peak Detector Low Value Monitor Register							
RF_SLCLO		0x5001100E/F			0xXXXX		

RF_SLCLO: Peak Detector Low Value Monitor Register							
R	R	R	R	R	R	R	R
PDLO7	PDLO6	PDLO5	PDLO4	PDLO3	PDLO2	PDLO1	PDLO0
PDLO15	PDLO14	PDLO13	PDLO12	PDLO11	PDLO10	PDLO9	PDLO8
MSB							LSB
Bit15-0**PDLO[15:0]**: Peak detector low value							

RF_NX: Controls the integer portion of the PLL feedback divider.							
RF_NX		0x50018004			0x79		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
NX7	NX6	NX5	NX4	NX3	NX2	NX1	NX0
MSB							LSB
Bit7-0 **NX[7:0]**: Integer part of the divider value for PLL divider. If (F_EN = 0) $f_{LO} = f_{xo} * NX$ If (F_EN = 1) $f_{LO} = f_{xo} * (NX + NF/256)$							

RF_NF: Controls the fractional portion of the PLL feedback divider							
RF_NF		0x50018005			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
NF7	NF6	NF5	NF4	NF3	NF2	NF1	NF0
MSB							LSB
Bit7-0 **NF[7:0]**: Fractional part of the divider value for PLL divider. If (F_EN = 0) $f_{LO} = f_{xo} * NX$ If (F_EN = 1) $f_{LO} = f_{xo} * (NX + NF/256)$							

RF_FETRIM0: RF Front-end Trim0 Register.

RF_FETRIM0		0x50018006			0x9A		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
HI_SD	F_EN	CP_TRM1	CP_TRM0	LF_TRM3	LF_TRM2	LF_TRM1	LF_TRM0
MSB							LSB

Bit7 **HI_SD**: Controls image reject mixer to control whether to use high-side or low-side mixing

0 = low-side

1 = high-side

Bit6 **F_EN**: Controls fractional mode

0 = integer-N mode

1 = fractional-N mode

Bit5-4 **CP_TRM[1:0]**: Adjust charge pump current in PLL for optimum settling time

Bit3-0 **LF_TRM[3:0]**: Adjust loop filter stabilization resistor in PLL to control overshoot

RF_FETRIM1: RF Front-end Trim1 Register.

RF_FETRIM1		0x50018007			0x4A		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DIV_LO21	DIV_LO20	BBGAIN1	BBGAIN0	LNA_DRES	LNAB_PG2	LNAB_PG1	LNAB_PG0
MSB							LSB

RF_FETRIM1: RF Front-end Trim1 Register.

Bit7-6 **DIV_LO2[1:0]**: Divide value from PLL frequency to LO2 frequency ($f_{LO2} = f_{LO1}/(32*DIV\_LO2)$)

00 = 8*15*4

01 = 8*16*4

10 = 8*17*4

11 = 8*18*4

Bit5-4 **BBGAIN[1:0]**: Sets baseband amplifier gain

00 = 0.6 dB

01 = 6.0 dB

10 = 11.3 dB

11 = 15.8 dB

Bit3 **LNA_DRES**: LNA drain resistor

0 = 2kW

1 = 1kW

Bit2-0 **LNAB_PG[2:0]**: Adjust loop filter stabilization resistor in PLL

I = 200uA*Bit0 + 500uA*Bit1 + 1mA*Bit2

5.3 RF Transmitter Subsystem

Krankl implements a powerful RF transmitter operating in the ISM (**I**ndustrial, **S**cientific and **M**edical) band and capable of transmitting ASK/OOK modulations. Its main characteristics are:

- Precise fractional-N PLL referenced to a 30MHz crystal oscillator, which drives into a modulation control circuit, and then into a high power output stage
- High power output stage
- Integrated power control
- Autonomous state machine which controls transmit bursts

The following block diagram details the main structures used by the transmitter.

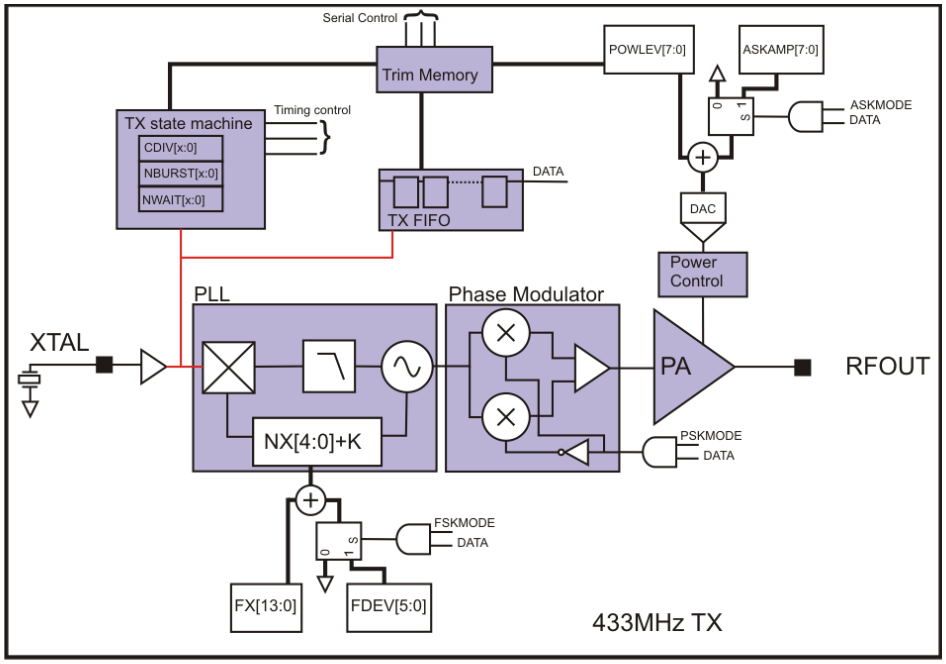


Figure 3 ASK Transmitter

The output stage amplifies modulated data up to a maximum power level of +15dBm. The device has a single output pin, which requires external matching. The PA output power is regulated by an integrated power control stage. This power control circuit takes an internal measurement, which correlates with the output power of the high power output stage, and compares it to an internal reference, which is set by a digital control register.

The RF transmit frequency is generated by a fractional-N PLL. The PLL may be operated in integer mode by setting bit F_EN to logic 0, or in fractional mode by setting bit F_EN to logic 1. The frequency generated is calculated according to the following formula:

In integer mode, $f_{VCO}=f_{XTAL}*(NX[4:0]+K)$, where K=8

or in fractional mode, $f_{VCO}=f_{XTAL}*(NX[4:0]+K+(NF[13:0]/2^{14}))$, where K=9

5.3.1 Fractional PLL

The RF transmitter frequency is generated by a fractional-N PLL circuit. The default reference frequency is 30 MHz. The PLL may be operated in integer mode by clearing the FRACEN bit (**TXCTRL0** register) or in fractional mode by setting FRACEN bit (**TXCTRL0** register).
The frequency generated is calculated depending on the PLL setup of NX and NF bits (located in **PLLCTRL0**, **PLLCTRL1** and **PLLCTRL2** registers):
PLL in integer mode:

$$Fvco = Fxtal \times (NX[4...0] + 8)$$

PLL in fractional mode:

$$Fvco = Fxtal \times (NX[4...0] + 9 + \frac{NF[13...0]}{2^{14}})$$

In the ISM band reaches from 433.05 to 434.79 MHz. Krankl can generate nominal frequencies from 433.049927 MHz up to 434.789429 MHz in steps of ~1831Hz.

Code Example: Setting the RF Transmitter to transmit at 433.92 MHz in fractional mode from a 30 MHz clock:

```
  test = RF_Set_Frequency( 30000000, 433920000, PLL_FRACTIONAL);
```

5.3.2 Transmission State Machine Operation

The Krankl transmitter operates as a Finite State Machine (FSM) implemented in hardware. This FSM provides a group of registers that allow for a very flexible and powerful yet simple transmission of data. The FSM provides automatic transmission of up to 16 bits without reloading transmission registers. The FSM may interact with the microcontroller via 2 interrupts. The interrupts are:

- Single Byte Left: when the buffer has only a single byte left to transmit (so that the micro can choose to refill both bytes as they are double-buffered).
- End of Burst: After sending all bits the FIFO is cleared and the transmitter waits NWAIT samples. At this moment an interrupt is generated. The micro can refill the data buffer with samples and reprogram the clock division rate for the next burst if required or, if no new data is to be sent then it can disable the system.

The steps to properly set this FSM are as follows:

- Select the modulation scheme ASK/OOK, using the MODUL[1…0] bits (TXCTRL0 register).
- Set the PLL for the required frequency.
- Select the bit rate using TXCDIV[10…0]. (TXCLKDIV and TXCTRL3)

- Define the total number of bits per burst with NTXBITS[9…0] bits (NBURST and TXCTRL2).
- Define the number of bits between bursts using NWAIT[5…0] (TXCTRL2).
- Select the appropriate values for the different timing elements: TDET1, TDET2, TDET3, TDET4, TDET5 and TPATCH. (TXCTRL3, TXCTRL4, TXCTRL5)
- If required, enable the interrupts associated with the transmission (end of burst and end of first byte transmission).
- Load the first two bytes of data to be transmitted in TXDATA[15…0] bits (TXDATL and TXDATH).
- Set the STARTX bit (TXCTRL0 register) to initiate transmission.

5.3.2.1 *ASK Modulation Selection*

Select ASK/OOK by simply loading the MODUL[1…0] bits in the TXCTRL0 register :

Code example 1:
(To set the ASK modulation now using access functions)

```
        ret = RF_Set_Modulation( ASK );
```

5.3.3 Chip Rate Selection

The required chip rate can be calculated applying following equation:

$$chiprate = \frac{1.25MHz}{\mathrm{TXCDIV}}$$

For instance, to obtain a chip rate of 19200bits/sec:

$$chiprate = \frac{1.25MHz}{\mathrm{TXCDIV}} = 19200 \Rightarrow \mathrm{TXCDIV} = \frac{1.25MHz}{19200} \cong 65$$

(This in fact will generate a chip rate of ~19231bits/sec, an error of 0.16%)
Code example: (From the calculation above)

```
        TXTIMING->CLKDIV.HOWRD = 0xF800; // Clear CLKDIV
        TXTIMING->CLKDIV.BYTE[0] = 65;   // Load the proper value in the lower 8 bits
```

Note:
Loading TXDIV with ‘0’ will result in the maximum divider (1024), resulting in the minimum chip rate of ~1220bits/sec.
Baud rate will be fraction of above chip rate and depends on the format of coding used. For example if the coding used is Manchester, baud rate would be chip rate divided by two. If coding used is 1/3 2/3

coding, baud rate will be chip rate divided by three.

5.3.3.1 *Burst Size Selection*

The number of bits to be transmitted per burst is defined in NTXBITS[9…0]. The following code example selects a 700 bits/burst transmission: (700 = 0x2BC)

```
      *TXCTRL2 &= 0xFC;              //Clear the upper 2 bits of the burst size
      *TXCTRL2 |= 0x02;              //Load 0x02 into them
      *NBURST = 0xBC;                //Load 0xBC into lower 8 bits (700d = 0x2BC)
```

Example Code using access function:

```
      RF_Set_BurstSize( 700 );
```

5.3.3.2 *Inter-Bursts Bit-Number Selection*

The time waited between bursts is defined in number of bits at the current bit-rate. This time is defined as a function of NWAIT and the timing elements. The calculation of the total time between bursts is explained in the following.

Code Example: To create a 7 bits NWAIT time:

```
      *TXCTRL2 &= ~0xFC;             //Clear the 5 bits of NWAIT
      *TXCTRL2 |= (0x07<<2);         //Load 7d (0x07) into them
```

5.3.3.3 *Timing Elements Selection*

The RF transmitter is made of several subsystems. In order to operate properly they have to be activated and the proper settling time defined for each one of them. The figure below shows the several timing elements:

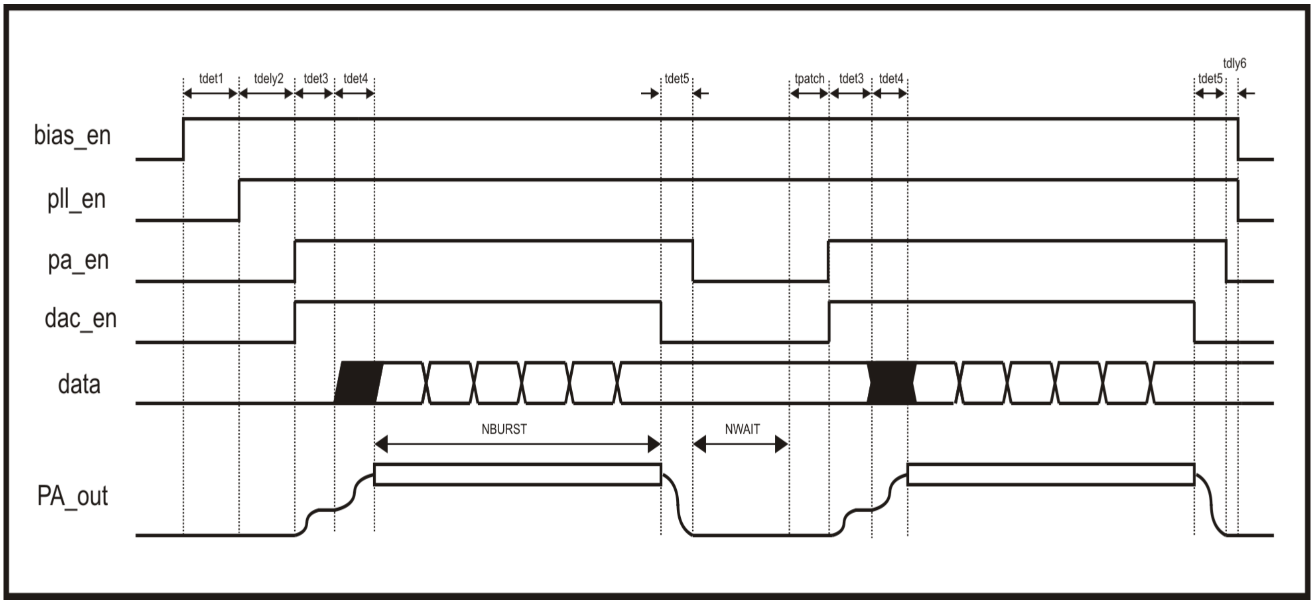


Figure 4 Transmitter Timing

Once the transmission is started several subsystems are automatically activated:
PA (**P**ower **A**mplifier)
PLL (**P**hase **L**ocked **L**oop)
DAC (**D**igital-**A**nalog **C**onverter)

Figure 4 Transmitter Timing**Error! Reference source not found.** shows different delay times each of this subsystems require to reach their proper operational condition. These delay times are defined by timing elements as follows:

Tdet1 – This time element allows for the PA bias to reach the proper levels. The minimum recommended time for this delay is 1usec, therefore selecting TDET1 = ‘b0 (1.6usec) is adequate.

Tdet2 – This time element allows for the PLL to be enabled and stabilize properly. The minimum time for it to do so is 1msec. and the recommended value for this field is TDET2 = ‘b010 (1.6msec).

Tdet3 – This time element defines the “Pedestal” period. This time is necessary to ensure the PA reached a midway power level prior to being enabled into full power. The recommended time for this parameter is ‘b101 (4usec).

Tdet4 and Tdet5* – These time elements are responsible for allowing for the minimum delay required to bring the PA to full power and minimum power. Both time elements should in principle be given the same value. The recommended value is ‘b10100 (32usec).
*TDET5 is not directly accessible in Krankl and it reflects the value of TDET4.

Tpatch – This time element is responsible for adjusting the wait time between bursts to create an integer multiple of bit times interval.

Therefor the interval between bursts is equal to:

$$T = t\det 5 + NWAIT + tpatch + t\det 3 + t\det 4 = NWAIT + tpatch + t\det 3 + 2\times t\det 4$$

Assuming the recommended values for Tdet3 and 4 we have then:

$$T = NWAIT + tpatch + 68u\sec$$

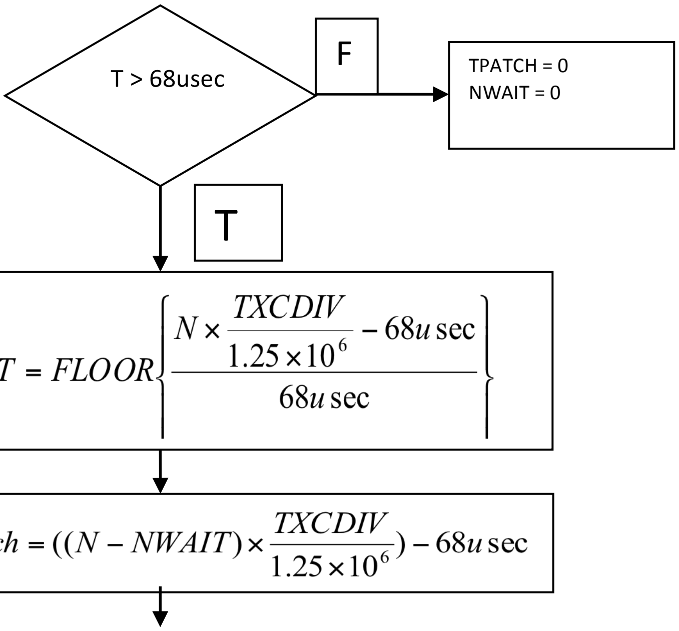
For a given number N of bit-times required between bursts the calculation becomes:

$$T = N\times\frac{TXCDIV}{1.25\times 10^{6}} = NWAIT + tpatch + 68u\sec$$

And:

$$Tpatch = 781\times 10^{-9}\times(TPATCH[10...0]) + 0.8\times 10^{-6}$$

The following flowchart defines the procedure to adjust the parameters:


T > 68usec
F
TPATCH = 0
NWAIT = 0
T
$NWAIT = FLOOR\left\{\frac{N\times\frac{TXCDIV}{1.25\times 10^{6}} - 68u\sec}{68u\sec}\right\}$
$tpatch = ((N - NWAIT)\times\frac{TXCDIV}{1.25\times 10^{6}}) - 68u\sec$

For instance, if the user wants a baud rate of 19200bits/sec and 7 bit-time of interval between bursts assuming the recommended values described above the calculation becomes:

$$T = (7)\times\frac{65}{1.25\times 10^{6}} = 364u\sec$$

T is larger than 68usec, therefore we can go ahead:

$$NWAIT = FLOOR\frac{\left[7\times\frac{65}{1.25\times10^{6}} - 68u\sec\right]}{68u\sec} = 4$$

We have then 4 bit-time waits. Now calculating the tpatch time:

$$tpatch = \left[(7-4)\times\frac{65}{1.25\times10^{6}}\right] - 68u\sec = 88u\sec$$

$$TPATCH = ROUND\left[\frac{88u\sec - 0.8u\sec}{781n\sec}\right] = 112$$

This gives us the following:

$$T = 68u\sec + 4\times\frac{65}{1.25\times10^{6}} + 781\times10^{-9}\times112 + 0.8u\sec = 364.3u\sec$$

Code Example:

```
*TXCTRL3 &= 0x07;            //Clear upper 5 bits
*TXCTRL3 |= (0x10<<3);       //Set TPATCH = 112d = 0x70 (Low 5 bits)
*TXCTRL1 &= 0xC0;            //Clear TPATCH[10…5] bits
*TXCTRL1 |= 0x03;            //Load 0x03 (High 6 bits)
*TXCTRL4 &= 0xF0;            //Clear lower 4bits
*TXCTRL4 |= (0x02<<1)+0x00;  //Set TDET2='b010  and TDET1=0
*TXCTRL5 |= (0x05<<5)+0x14;  //Set TDET3='b101  and TDET4=0x14
```

5.3.4 Transmitter registers

The transmitter provides several registers that provide a flexible and simple to use interface:

Transmitter Register Map			
Address	**Register Name**	**Field Name**	**Description**
0x50000032	TXCTRL0	STARTX	Single bit which starts the transmission
		NXTBURST	Next Burst Flag
		MODUL	Modulation Selection
		FRACEN	PLL Fractional Enable bit
		[2…0]	Reserved
0x50000033	TXCTRL1	PEDES	Pedestal Time
		TPATCH	Delay Time between disabling the PLL and turning off the bias circuit
0x50000034	POWLEV	POWLEV	Power Level
0x50000035	POWAP	POWAP	Power level for ASK/OOK transmission of '1' (High Level)
0x50000036	TXDATL	TXDATA	Low Byte of Transmission Data
0x50000037	TXDATH	TXDATA	High Byte of Transmission Data
0x50000038	NBURST	NTXBITS	Number of bits of transmission in the next burst low byte
0x50000039	TXCTRL2	NWAIT	Number of bits between bursts
		NTXBITS	Number of bits of transmission in the next burst higher 2 bits
0x5000003A	TXCLKDIV	TXCDIV	Low byte of clock divider
0x5000003B	TXCTRL3	TPATCH	Lower 5 bits of the delay between disabling the PLL and turning off the bias

Transmitter Register Map			
Address	**Register Name**	**Field Name**	**Description**
		TXCDIV	Upper three bits of the Transmission Clock Divider. (data rate is the 1.25Mbps/CDIV)
0x5000003C	TXCTRL4	TRAMP	Prevents the PA output power ramp up/down when the data bit changes.
		TDET2	Delay between enabling the PLL and enabling the power control DAC & power amplifier
		TDET1	Delay between enabling the bias and enabling the PLL
0x5000003D	TXCTRL5	TDET3	Delay between enabling the power control DAC and enabling the power amplifier
		TDET4	PA output power rising and falling time
0x50018000	PLLCTRL0	NF[7:0]	Fractional N portion of feedback divider lower bits
0x50018001	PLLCTRL1	NX[1:0]	Lower two bits of the Integer portion of feedback divider
		NF[13:8]	Six higher bits of the Fractional Portion of Feedback Divider
0x50018002	PLLCTRL2	LPFTRIM[1:0]	Low pass filter trim bits
		CPTRIM[2:0]	Charge pump trim bits
		NX[4:2]	Integer portion of feedback divider high bits
0x50018003	PLLCTRL3	DF[5:0]	Frequency deviation control
		LPFTRIM[3:2]	Low pass filter trim bits
0x5001800E	PACTRL0	LPFCAL[3:0]	PLL loop filter calibration
		PctrlCcal[3:0]	Power control C calibration

Transmitter Register Map			
Address	**Register Name**	**Field Name**	**Description**
0x5001800F	PACTRL1	Rcal[1:0]	Power control R calibration
		PACoreITrim	Current trimming of PA core
		PAdrv2ITrim	Current trimming of PA driver stage
		PACoreCcal[3:0]	PA core C calibration

A more detailed description of each register follows below:

TXCTRL0: First Transmitter Control Register							
TXCTRL0		0x50000032			0x08		
R/W	R/W	R/W	R/W	R/W	Reserved	Reserved	Reserved
STARTX	NXTBURST	MODUL1	MODUL0	FRAC_EN	RF_SW	-	TX_ON
MSB							LSB

TXCTRL0: First Transmitter Control Register

Bit7 **STARTX**: Start Transmission.

0 = Do not start transmit

1 = Start Transmitting

Bit6 **NXTBURST**: Set to indicate there is a next burst.

0 = No next burst

1 = There will be a next burst

Bit5-4 **MODUL[1:0]:** Modulation selected.

00 = Reserved

11 = ASK/OOK

Bit3 **FRAC_EN**: Fractional PLL enabled.

0 = Fractional PLL disabled

1 = Fractional PLL enabled

Bit2 **RF_SW**: Toggles RF Switch between RX and TX Mode

0 = RX Mode

1 = TX Mode

Bit1 **Reserved**

Bit0 **TX_ON**: TX on for constant transmission

0 = Constant transmission disabled

1 = Constant transmission enabled

TXCTRL1: Second Transmitter Control Register

TXCTRL1		0x50000033			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PEDES1	PEDES0	TPATCH10	TPATCH9	TPATCH8	TPATCH7	TPATCH6	TPATCH5
MSB							LSB

TXCTRL1: Second Transmitter Control Register

Bit7-6 **PEDES[1: 0]**: Pedestal Time:

00 = 12.8usec.

01 = 14.4usec

10 = 16usec.

11 = 17.6usec.

Bit5-0 **TPATCH[10:5]**: Delay between disabling the PLL and turning off the bias:

0x000 = 0.8usec

:

0x7FF = 1.6msec

POWLEV: Power Level of ASK/OOK Transmission in Level "0"							
POWLEV		0x50000034			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POWLEV 7	POWLEV 6	POWLEV5	POWLEV4	POWLEV3	POWLEV2	POWLEV 1	POWLEV 0
MSB							LSB

POWLEV: Power Level of ASK/OOK Transmission in Level “0”

Bit7-0 **POWLEV[7:0]**: Power Level, defined as follows:

POWLEV[7:0] (HEX)	**Pout (dBm)**
06-00	<-11
0A-07	-9.5
0B-08	-8
10-09	-6.5
12-0A	-5
12-11	-3.5
17-12	-2
18-14	-0.5
1B-18	+1
1F-1B	+2.5
24-1F	+4
28-26	+5.5
36-31	+7
45-37	+8.5
55-39	+10
70-54	+11.5

POWAP: Power Level of ASK/OOK transmission in level “1”

POWAP		0x50000035			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POWAP7	POWAP6	POWAP5	POWAP4	POWAP3	POWAP2	POWAP1	POWAP0
MSB							LSB

Bit7-0 **POWAP[7:0]**: Refer to **POWLEV** register.

TX TXDATL: TX data lower byte register							
TXDATL		0x50000036			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
TXDATA7	TXDATA6	TXDATA5	TXDATA4	TXDATA3	TXDATA2	TXDATA1	TXDATA0
MSB							LSB
Bit7-0 **TXDATA[7:0]**: Lower byte of data to be transmitted.							

TXDATU: TX data upper byte register							
TXDATU		0x50000037			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
TXDATA15	TXDATA14	TXDATA13	TXDATA12	TXDATA11	TXDATA10	TXDATA9	TXDATA8
MSB							LSB
Bit7-0 **TXDATA[15:8]**: Upper byte of data to be transmitted.							

NBURST: Number of bits in the next burst							
NBURST		0x50000038			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
NTXBITS7	NTXBITS6	NTXBITS5	NTXBITS4	NTXBITS3	NTXBITS2	NTXBITS1	NTXBITS0
MSB							LSB
Bit7-0 **NTXBITS[7:0]**: Lower byte of the counter of number of bits to be transmitted in the next burst.							

TXCTRL2: Third TX Control Register for RF Transmission

TXCTRL2: Third TX Control Register for RF Transmission							
TXCTRL2		0x50000039			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
NWAIT5	NWAIT4	NWAIT3	NWAIT2	NWAIT1	NWAIT0	NTXBITS9	NTXBITS8
MSB							LSB
Bit7-2 **NWAIT[5:0]**: Number of bits (time) between bursts. Bit1-0 **NTXBITS[9:8]**: Upper two bits of the counter of number of bits to be transmitted in the next burst.							

TXCLKDIV: Transmission Clock Divider							
TXCLKDIV		0x5000003A			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
TXCDIV7	TXCDIV6	TXCDIV5	TXCDIV4	TXCDIV3	TXCDIV2	TXCDIV1	TXCDIV0
MSB							LSB
Bit7-0 **TXCDIV[7:0]**: Low byte of clock divider. (data rate is defined as 1.25Mbps/ TXCLKDIV)							

TXCTRL3: Fourth TX Control Register for RF Transmission							
TXCTRL3		0x5000003B			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
TPATCH4	TPATCH3	TPATCH2	TPATCH1	TPATCH0	TXCDIV10	TXCDIV9	TXCDIV8
MSB							LSB

TXCTRL3: Fourth TX Control Register for RF Transmission

Bit7-3 **TPATCH[4:0]**: Lower 5 bits of the delay between disabling the PLL and turning off the bias.

0x000 = 0.8usec

:

0x7FF = 1.6msec

Bit2-0 **TXCDIV[10:8]**: Upper three bits of the Transmission Clock Divider. (data rate is 1.25Mbps/CDIV)

Example: If we want to transmit data at a rate of 19200bits/sec we should load **TXCDIV** with:

$$baud = \frac{1.25MHz}{\mathrm{TXCDIV}} = 19200 \Rightarrow \mathrm{TXCDIV} = \frac{1.25MHz}{19200} \cong 65$$

This in fact will generate a baud rate of ~19231bits/sec. (0.16% error)

NOTE: The equation for TPATCH is:

$$Tpatch = 78 \times 10^{-9} \times (TPATCH[10...0]) + 0.8 \times 10^{-6}$$

TXCTRL4: Fifth TX Control Register for RF Transmission							
TXCTRL4		0x5000003C			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
TRAMP3	TRAMP2	TRAMP1	TRAMP0	TDET2_2	TDET2_1	TDET2_0	TDET1
MSB							LSB

TXCTRL4: Fifth TX Control Register for RF Transmission

Bit7-4 **TRAMP[3:0]**: Prevents the PA output power ramp up/down when the data bit changes. This delay is calculated as:

$$Delay = \frac{TRAMP}{SysClk}$$

This leads to the following values: (Assuming System Clock = 30 MHz)

0000 = 0usec

:

1000 = 0.27usec

:

1111 = 0.5usec

Bit3-1 **TDET2[2:0]**: Delay between enabling the PLL and enabling the power control DAC & power amplifier. This delay is calculated as:

$$Delay = \frac{TDET2}{1.25MHz} \times 1024$$

This leads to the following values:

000 = 0msec

:

100 = 3.28usec

:

111 = 5.73usec

Bit0 **TDET1**: Delay between enabling the bias and enabling the PLL:

0 = 1.6usec

1 = 8usec

TXCTRL5: Sixth TX Control Register for RF Transmission

TXCTRL5		0x5000003D			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
TDET3_2	TDET3_1	TDET3_0	TDET4_4	TDET4_3	TDET4_2	TDET4_1	TDET4_0

TXCTRL5: Sixth TX Control Register for RF Transmission							
MSB							LSB

Bit7-5 **TDET3[2:0]**: Delay between enabling the power control DAC and enabling the power amplifier:

$$Delay = \frac{TDET3}{1.25MHz}$$

This leads to the following values:

000 = 0usec

:

100 = 3.2usec

:

111 = 5.6usec

Bit4-0 **TDET4[4:0]**: PA output power rising delay time:

$$RiseFall = \frac{TDET4}{1.25MHz} \times 2$$

PLLCTRL0: PLL Control Register 0							
PLLCTRL0		0x50018000			0x06		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
NF7	NF6	NF5	NF4	NF3	NF2	NF1	NF0
MSB							LSB
Bit7-0 **NF[7:0]**: Fractional Portion of Feedback Divider lower bits.							

PLLCTRL1: PLL Control Register 1							
PLLCTRL1		0x50018001			0x2D		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
NX1	NX0	NF13	NF12	NF11	NF10	NF9	NF8
MSB							LSB
Bit7-6 **NX[1:0]**: Integer portion of feedback divider low bits. Bit5-0 **NF[13:8]**: Fractional Portion of feedback divider high bits.							

PLLCTRL2: PLL Control Register 2							
PLLCTRL2		0x50018002			0xCB		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
LPFTRIM1	LPFTRIM0	CPTRIM2	CPTRIM1	CPTRIM0	NX4	NX3	NX2
MSB							LSB
Bit7-6 **LPFTRIM[1:0]**: Low pass filter trim bits. Bit5-3 **CPTRIM[2:0]**: Charge pump trim bits. Bit2-0 **NX[4:2]**: Integer portion of feedback divider high bits.							

PLLCTRL3: PLL Control Register 3							
PLLCTRL3		0x50018003			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DF5	DF4	DF3	DF2	DF1	DF0	LPFTRIM3	LPFTRIM2
MSB							LSB

Bit7-2 **DF[5:0]**: Frequency deviation control

Bit1-0 **LPFTRIM[3:2]**: Low pass filter trim bits.

PACTRL0: Power Amplifier Control Register 0							
PACTRL0		0x5001800E			0x44		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PctrlCcal0	PctrlCcal0	PctrlCcal0	PctrlCcal0	LPFCcal3	LPFCcal2	LPFCcal1	LPFCcal0
MSB							LSB

Bit7-4 **PCtrlCcal[3:0]**: Power control C calibration

Bit3-0 **LPFCcal[3:2]**: Loop filter C calibration

PACTRL1: Power Amplifier Control Register 1							
PACTRL1		0x5001800F			0x82		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PACoreCal[3]	PACoreCal[2]	PACoreCal[1]	PACoreCal[0]	PAdrv2Itrim	PACoreItrim	Rcal[1]	Rcal[0]
MSB							LSB

Bit7-4 **PCtrlCcal[3:0]**: PA core C calibration

Bit3 **PAdrv2Itrim**: Current trim PA driver stage

Bit2 **PACoreItrim**: Current trim PA core stage

Bit 1-0 **Rcal**[1:0]: Pa core R calibration

5.4 RF Switch

Krankl implements an RF Switch to allow for time duplexed reception and transmission sharing a single antenna.

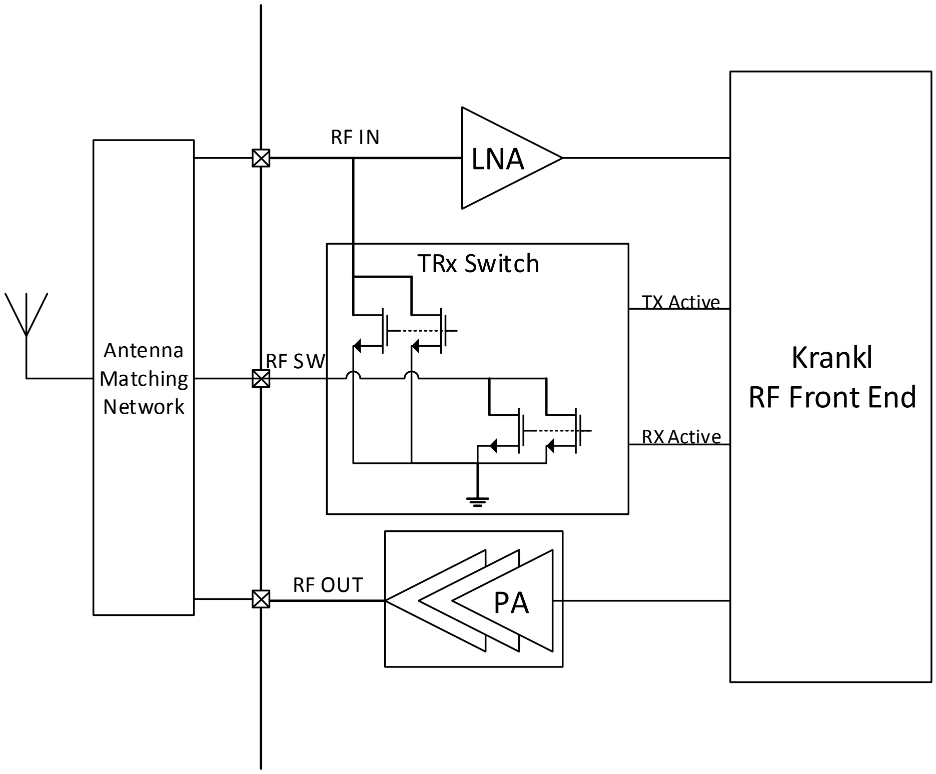


Figure 5 RF Switching Circuit Concept

During transmission the TX Active signal will be enabled allowing for accurate matching of the antenna impedance with the PA output impedance. For proper reception the RX Active signal will be switched on which results in a matched impedance network for the RX bath.

5.5 General Purpose ADC 8 Bit

Krankl includes two analog to digital converter (ADC). The ADC is an 8-bit analog to digital converter with single ended input. The main features are described below:

- 8-bit resolution
- Single ended input
- Configurable reference (VREF = VREFHI-VREFLO)
    - Either based on the bandgap voltage (VBG) or ADC block supply voltage.
    - Reference may be scaled in order to provide more resolution around a smaller input voltage range
- Maximum ADC input range is from 0V to its supply voltage
- It may read from a total of 29 channels (28 GPIOs and Supply Voltage)
- Using VBG as the reference, supply voltage can be measured in calibration mode

5.5.1 **ADC Description**

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

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

- Step 1: Select the input channel to be measured
- Step 2: Configure ADC settings.
    - Set ADC clock frequency.
    - Configure references by programming the ADCREFHI, ADCREFLO, ADCPGN, and ADCREFS bits.
- Step 3: Start the ADC conversion.
- Step 4: Check the ADC status bit and read the data.

The following section will describe each configuration step in detail.

All GPIOs (PA[6:0], PC[1:0], PD[7:0], PE[7:0] and PF[2:0]) and the Supply Voltage reference are available as inputs to the ADC. The user can control which input is connected for the conversion by programming the control bits, **ADDCH[7:0]**, in the **ADDCHANNEL** register. ADC1 has 15 input channels connected to P ports while ADC2 has 13 inputs channel connected to P port and an supply monitor.

Code Example: Selecting PA0 for ADC input channel

```
ADC_Select_Channel( ADC_PA0 );
```

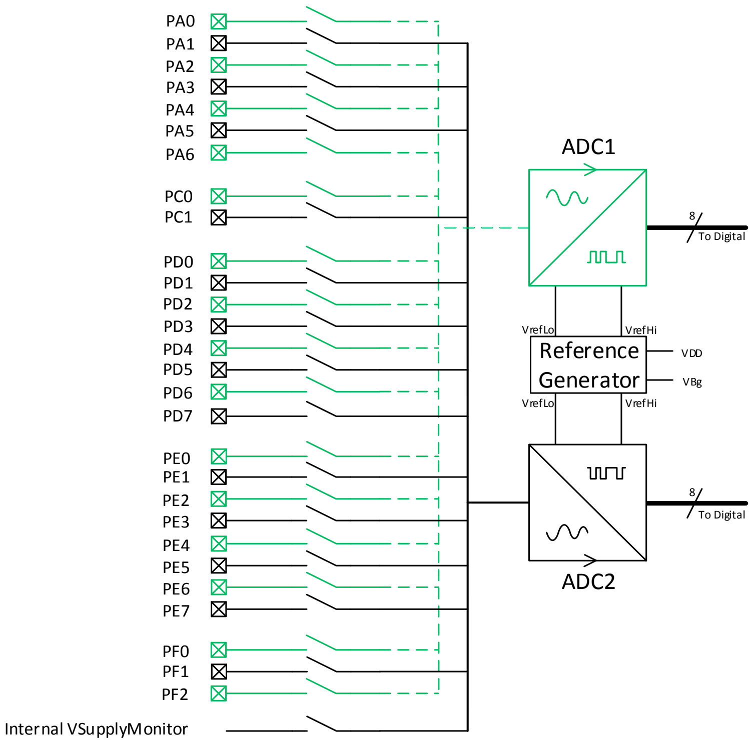


Figure 6 ADC input multiplexing

5.5.2 ADC Clock and Sampling Period

The conversion algorithm has a basic period of 9 cycles (one for sampling, one for each bit). There is a two-cycle latency from the last bit measurement until the data becomes available to the user. Additionally, there is a single idle cycle to allow biasing before any conversion is initiated. Thus a single conversion will take 13 cycles.

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

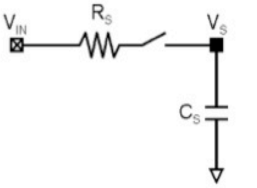
The ADC clock frequency can be programmed through the **ADCCLKDIV** register. As an example, if ADC clock is derived from a 12MHz oscillator divided by 16, the input has 1.3μs to settle. Since the maximum value of the internal sample/hold capacitor, C_s is 10pF, the maximum source resistance of the signal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

$$R_s = \frac{1.3\ \ s}{10\,pF \times 5.5} = 24.2k$$

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

Code Example: Configure ADC clock to 12MHz RC oscillator frequency divided by 16.

```
ADC_ClkDiv (16);
```



$$V_S = V_{IN}(1 - e^{-\frac{t}{\tau}}) \quad , where\ \tau = R_S C_S$$

$$V_S\ is\ within\ 1\ LSB\ of\ V_{IN}\ after\ t = 5.5\tau$$

Figure 7 ADC Input Settling Time

5.5.3 Configuration of Reference Voltages for the ADC

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

ADCREFS=0*

$$VREFHI = \frac{ADCREFHI}{ADCPGN} \times VBG$$

$$VREFLO = \frac{ADCREFLO}{ADCPGN} \times VBG$$

ADCREFS=1

$$VREFHI = \frac{ADCREFHI}{15} \times VDD$$

$$VREFLO = \frac{ADCREFLO}{15} \times VDD$$

* If $\left(\frac{15}{ADCPGN} \times VBG\right) < \left(VDD - 0.1V\right)$

Figure 8 ADC Reference Voltage

Once the reference voltages are established, the ADC conversion equation for input voltage (VIN) can be defined as:

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

Here are a few examples:

- Example 1: In a system operating with VDD=3V, there is a signal that moves between 0V and 2.94V. In this case it is recommended that VDD be selected as the reference source and the following setting be made, **ADCREFH**=15, **ADCREFL**=0, and **ADCPGN**=15. This selection would allow for the maximum range of measurement (0V to VDD).
    - Code Sample `ADC_Reference_Config( ADC1, 15, 0, 15, ADCREFVDD);`
- Example 2: In a system operating with VDD=3V and VBG= 1.21V, there is a signal that moves between 0V and 2.5V. In this case VBG can be selected as the reference source with settings of **ADCREFH**=13, **ADCREFL**=0 and **ADCPGN**=7. This configuration would allow the signal range of measurement to be 0V to 2.6V:
    - Code Sample `ADC_Reference_Config( 13, 0, 7, ADCREFBG);`
- Example 3: In a system operating with VDD=3V and VBG=1.21V, there is a signal that moves between 1.71V and 2.2V. In this case selecting VBG as the reference source and settings of **ADCREFH**=15, **ADCREFL**=11, and **ADCPGN**=8 we can achieve higher resolution:
    - Code Sample `ADC_Reference_Config( 15, 11, 8, ADCREFBG);`

The resolution in Example 3 can be calculated as follows:

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

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

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

5.5.4 ADC Start and Status

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

Code Example: Enable the converter and start conversion

```
ADC_Enable(ADCEN);
ADC_Start();                              //ADC enabled and start
while ( ADC_ConversionComplete() == 1 );  //Wait until completed
```

5.5.5 ADC Registers

The following registers control the behavior of the ADC:

ADC Control Register Map				
Address	Register Name	Description	Reset Value	Reference
0x5000000C	ADCCHANNELS	ADC Channel Select Register	0x00	
0x5000000D	ADCSTART	ADC Start Register (Start conversion)	0x00	
0x5000000E	ADC1DATA	ADC1 Conversion Result Register	0x00	
0x5000000F	ADC2DATA	ADC2 Conversion Result Register	0x00	
0x5000000B	ADCCLKDIV	ADC Clock Divider Control Register	0x6F	
0x50018008	ADCTRIM0	ADC1 Reference High and Low Setting	0xF0	
0x50018009	ADCTRIM1		0xF0	
0x5001800A	ADCTRIM2	ADC2 Reference High and Low Setting	0xF0	
0x5001800B	ADCTRIM3		0x9F	

A more detailed description of each register follows below:

ADCCHANNELS: ADC Channel Select Register							
ADCCHANNELS		0x5000000C			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADC2CH7	ADC2CH6	ADC2CH5	ADC2CH4	ADC1CH3	ADC1CH2	ADC1CH1	ADC1CH0
MSB							LSB

Bit7-4 **ADC2CH[3:0]**: Channel Selection for ADC2: 14 inputs

Bit3-0 **ADC1CH[3:0]**: Channel Selection for ADC1: 15 inputs

ADC2CH[3:0]	ADC1CH[3:0]
0000 = PD1	0000 = PD0
0001 = PD3	0001 = PD2
0010 = PD5	0010 = PD4
0011 = PD7	0011 = PD6
0100 = PE1	0100 = PE0
0101 = PE3	0101 = PE2
0110 = PE5	0110 = PE4
0111 = PE7	0111 = PE6
1000 = PA1	1000 = PA0
1001 = PA3	1001 = PA2
1010 = PA5	1010 = PA4
1011 = V12 with 1/10 divider	1011 = PA6
1100 = PC1	1100 = PC0
1101 = PF1	1101 = PF0
1110 = Reserved	1110 = PF2
1111 = NC	1111 = NC

ADCSTART: ADC Start Register							
ADCSTART		0x5000000D			0x00		
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	R/W
–	–	–	–	–	–	–	START
MSB							LSB
Bit0 **START**: Writing one starts the conversion. Reading returns the status of conversion; ‘0’ means conversion is finished and ‘1’ means the conversion is ether pending or in progress							

ADC1DATA: ADC1 Result Register							
ADC1DATA		0x5000000E			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADC1DT7	ADC1DT6	ADC1DT5	ADC1DT4	ADC1DT3	ADC1DT2	ADC1DT1	ADC1DT0
MSB							LSB
Bit7-0 **ADC1DT[7:0]**: ADC Result							

ADC2DATA: ADC2 Result Register							
ADC2DATA		0x5000000F			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADC2DT7	ADC2DT6	ADC2DT5	ADC2DT4	ADC2DT3	ADC2DT2	ADC2DT1	ADC2DT0
MSB							LSB
Bit7-0 **ADC2DT[7:0]**: ADC Result							

ADCCLKDIV: ADC Clock Divider Control Register							
ADCCLKDIV		0x5000000B			0x6F		
R/W	Reserved	R/W	R/W	R/W	R/W	R/W	R/W
SWMODE	-	ADCDIV0	ADCDIV0	ADCDIV0	ADCDIV2	ADCDIV1	ADCDIV0
MSB							LSB
Bit5-0 **ADCDIV[5:0]:** ADC Clock divider 00000 = System Clock/2 00001 = System Clock/4 11111 = System Clock/64 Bit7 **SWMODE:** ADC Mux Switch Mode 0 = ADC Mux switch is open after the conversion 1 = ADC Mux switch is closed to the selected input after the conversion							

ADCTRIM0: ADC Trim0 Register							
ADCTRIM0		0x50018008			0xF0		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PHI1[3]	PHI1[2]	PHI1[1]	PHI1[0]	PLO1[3]	PLO1[2]	PLO1[1	PLO1[0]
MSB							LSB

Bit7-4 **PHI1[3:0]:** ADC1 Reference High Setting

Bit3-0 **PLO1[3:0]:** ADC1 Reference Low Setting

ADC Trim1 Register							
ADCTRIM1		0x50018009			0xF0		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADC_SW[1]	ADC_SW[0]	ADC_CAL	ADC1_FulRg	PGN1[3]	PGN1[2]	PGN1[1]	PGN1[0]
MSB							LSB

Bit7-6 **ADCSW[1:0]:** ADC Enable Bit

00 = Correlated double sampling off

01 = Input offset Calibration on

1x = Correlated doubling sampling on (default)

Bit5 **ADCCAL:** ADC Calibration

0 = Normal

1 = Calibration mode

Bit4 **ADC1_FulRg:** ADC1 Internal Reference Source Selection

0 = Band Gap

1 = VDD

Bit3-0 **PGN1[3:0]**: ADC1 Reference Gain

ADCTRIM2: ADC Trim2 Register							
ADCTRIM2		0x5001800A			0xF0		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PHI2[3]	PHI2[2]	PHI2[1]	PHI2[0]	PLO2[3]	PLO2[2]	PLO2[1]	PLO2[0]
MSB							LSB
Bit7-4 **PHI2[3:0]:** ADC2 Reference High Setting Bit3-0 **PLO2[3:0]:** ADC2 Reference Low Setting							

ADCTRIM3: ADC Trim3 Register

ADCTRIM3		0x5001800B			0x9F		
Reserved	R/W	R/W	R/W	R/W	R/W	R/W	R/W
-	ADC_ENABLE	ADC_GndOff	ADC2_FulRg	PGN2[3]	PGN2[2]	PGN2[1]	PGN2[0]
MSB							LSB

Bit6 **ADC_ENABLE:** Enable OPamp

Bit5 **ADC_GndOff:** Ground offset correction

0 = bandgap and ADC reference have common ground

1 = difference between bandgap and ADC reference is compensated by switched capacitor circuit

Bit4 **ADC2_FulRg:** ADC2 Internal Reference Source Selection

0 = Band Gap

1 = ADC supply voltage

Bit3-0 **PGN2[3:0]**: ADC2 Reference Gain

5.6 Pulse Width Modulators (PWM)

Krankl includes three PWM generators. Their main characteristics are:

- Twelve bit resolution – Both period and pulse width.
- Independent Prescalers
- Programmable active level
- Programmable outputs
    - PWM1 connected to PA5 and PD3
    - PWM2 connected to PC0 and PD2
    - PWM3 connected to PD4 and PD5

5.6.1 **PWM Usage Description**

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

The waveform is controlled by 12-bit period word (PWMnPER and PWMnEXT) and 12-bit pulse width word (PWMnPW and PWMnEXT) are used to determine the output waveform.

The programming of the PWM is double buffered, thus preventing any flickering to occur when modifying the settings. The new value in the buffer shall be used at the start of the next PWM period after it is written. PWMnEXT and PWMnPW should be written before PWMnPER to ensure that all new settings are updated and applied at the start of the same PWM period. A write to PWMnPER triggers the update of the buffered register.

The entire waveform can be scaled by adjusting the Prescaler value in PWMnCTRL. The Prescaler can be set to one of eight different settings shown below:

Table 5.10 – PWM Prescaler Divider Values

PRESC[3:0]	Divide Value (f_{XO}/f_{PWM})
000	1
001	2
010	4
011	8
100	32
101	256
110	8,192

Table 5.10 – PWM Prescaler Divider Values	
PRESC[3:0]	**Divide Value (f_{XO}/f_{PWM})**
111	262,144

The output period is calculated as follows taking PWMPER times the prescaler divider:

$$Period = \frac{1 + \left(PWMPER \times DIVIDE\_VALUE\right)}{SystemClock}$$

The PWM pulse width is calculated as follows:

$$Pulse\_Wdith = \frac{1 + (PWMPW \times DIVIDE\_VALUE)}{SystemClock}$$

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

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

Code Example: Setting the PWM1, enabled, with period and width separated, not inverted, with a PRESC[2:0] = 100:

```
PWM_Setup( PWM1, PWMEN, PWMNINV, PRESC4);
```

Code Example: Selecting the period and pulse width of the same PWM1.

```
      PWM_PW (PWM1,100);
      PWM_Period (PWM1,200);
```

5.6.2 PWM Registers

The following registers are provided to control the PWMs:

Pulse Width Modulator Control Register Map				
Address	Register Name	Description	Reset Value	Reference
0x50000040	PWM1_CTRL	PWM 1 control	0x00	
0x50000041	PWM1_PER	PWM 1 period	0x00	
0x50000042	PWM1_PW	PWM 1 pulse width	0x00	
0x50000043	PWM1_EXT	PWM 1 period and pulse width	0x00	
0x50000044	PWM2_CTRL	PWM 2 control	0x00	
0x50000045	PWM2_PER	PWM 2 period	0x00	
0x50000046	PWM2_PW	PWM 2 pulse width	0x00	
0x50000047	PWM2_EXT	PWM 2 period and pulse width	0x00	
0x50000048	PWM3_CTRL	PWM 3 control	0x00	
0x50000049	PWM3_PER	PWM 3 period	0x00	
0x5000004A	PWM3_PW	PWM 3 pulse width	0x00	
0x5000004B	PWM3_EXT	PWM 3 period and pulse width	0x00	

5.6.2.1 *PWM 1 Registers*

PWM1CTRL: PWM1 Control Register							
PWM1CTRL		0x50000040			0x00		
R/W	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
PWM_EN	-	-	PWM_INV	-	PRESC2	PRESC1	PRESC0
MSB							LSB

Bit7 **PWM_EN:** PWM1 enable bit.

0 = PWM Disabled

1 = PWM Enabled

Bit4 **PWM_INV:** PWM output signal direction

0 = normal logic

1 = inverted logic (active low)

Bit2-0 **PRESC[2:0]:** PWM 1 Prescaler

000 = System Clock/1

001 = System Clock/2

010 = System Clock/4

011 = System Clock/8

100 = System Clock/32

101 = System Clock/256

110 = System Clock/8192

111 = System Clock/262144 (2^{18})

PWM1PER: PWM1 Period Low Byte Register							
PWM1PER		0x50000041			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PWM1PER7	PWM1PER6	PWM1PER5	PWM1PER4	PWM1PER3	PWM1PER2	PWM1PER1	PWM1PER0

PWM1PER: PWM1 Period Low Byte Register							
MSB							LSB
Bit7-0 **PWM1PER[7:0]:** PWM1 period register							

PWM1PW: PWM1 Pulse Width Low Byte Register							
PWM1PW		0x50000042			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PWM1PW 7	PWM1PW 6	PWM1PW 5	PWM1PW 4	PWM1PW 3	PWM1PW 2	PWM1PW 1	PWM1PW 0
MSB							LSB
Bit7-0 **PWM1PW[7:0]:** PWM1 width register							

PWM1EXT: PWM1 extension with high nibble of period and pulse width							
PWM1EXT		0x50000043			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PWM1PER 11	PWM1PER 10	PWM1PER 9	PWM1PER 8	PWM1PW 11	PWM1PW 10	PWM1PW 9	PWM1PW 8
MSB							LSB
Bit7-4 **PWM1PER[11:8]:** PWM1 period (high nibble) Bit3-0 **PWM1PW[11:8]:** PWM1 width (high nibble)							

5.6.2.2 *PWM 2 Registers*

PWM2CTRL: PWM2 Control Register							
PWM2CTRL		0x5000004C			0x00		
R/W	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
PWM_EN	-	-	PWM2_INV	-	PRESC2	PRESC1	PRESC0
MSB							LSB

Bit7 **PWM_EN:** PWM2 enable bit.

0 = PWM Disabled

1 = PWM Enabled

Bit4 **PWM_INV:** PWM output signal direction

0 = normal logic

1 = inverted logic (active low)

Bit2-0 **PRESC[2:0]:** PWM 2 Prescaler

000 = System Clock/1

001 = System Clock/2

010 = System Clock/4

011 = System Clock/8

100 = System Clock/32

101 = System Clock/256

110 = System Clock/8192

111 = System Clock/262144 (2^{18})

PWM2PER: PWM2 Period Low Byte Register							
PWM2PER		0x5000004D			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PWM2PER7	PWM2PER6	PWM2PER5	PWM2PER4	PWM2PER3	PWM2PER2	PWM2PER1	PWM2PER0

PWM2PER: PWM2 Period Low Byte Register							
MSB							LSB
Bit7-0 **PWM2PER[7:0]:** PWM2 period register							

PWM2PW: PWM2 Pulse Width Low Byte Register							
PWM2PW		0x5000004E			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PWM2PW 7	PWM2PW 6	PWM2PW 5	PWM2PW 4	PWM2PW 3	PWM2PW 2	PWM2PW 1	PWM2PW 0
MSB							LSB
Bit7-0 **PWM2PW[7:0]:** PWM2 width register							

PWM2EXT: PWM2 extension with high nibble of period and pulse width							
PWM2EXT		0x5000004F			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PWM2PE R11	PWM2PE R10	PWM2PE R9	PWM2PE R8	PWM2PW 11	PWM2PW 10	PWM2PW 9	PWM2PW 8
MSB							LSB
Bit7-4 **PWM2PER[11:8]:** PWM2 period (high nibble) Bit3-0 **PWM2PW[11:8]:** PWM2 width (high nibble)							

5.6.2.3 *PWM 3 Registers*

PWM3CTRL: PWM3 Control Register							
PWM3CTRL		0x50000048			0x00		
R/W	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
PWM_EN	-	-	PWM_INV	-	PRESC2	PRESC1	PRESC0
MSB							LSB

Bit7 **PWM_EN:** PWM3 enable bit.

0 = PWM Disabled

1 = PWM Enabled

Bit4 **PWM_INV:** PWM output signal direction

0 = normal logic

1 = inverted logic (active low)

Bit2-0 **PRESC[2:0]:** PWM 3 Prescaler

000 = System Clock/1

001 = System Clock/2

010 = System Clock/4

011 = System Clock/8

100 = System Clock/32

101 = System Clock/256

110 = System Clock/8192

111 = System Clock/262144 (2^{18})

PWM3PER: PWM3 Period Low Byte Register							
PWM3PER		0x50000049			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PWM3PER7	PWM3PER6	PWM3PER5	PWM3PER4	PWM3PER3	PWM3PER2	PWM3PER1	PWM3PER0

PWM3PER: PWM3 Period Low Byte Register							
MSB							LSB
Bit7-0 **PWM3PER[7:0]:** PWM3 period register							

PWM3PW: PWM3 Pulse Width Low Byte Register							
PWM3PW		0x5000004A			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PWM3PW 7	PWM3PW 6	PWM3PW 5	PWM3PW 4	PWM3PW 3	PWM3PW 2	PWM3PW 1	PWM3PW 0
MSB							LSB
Bit7-0 **PWM3PW[7:0]:** PWM3 width register							

PWM3EXT: PWM3 extension with high nibble of period and pulse width							
PWM3EXT		0x5000004B			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PWM3PER11	PWM3PER10	PWM3PER9	PWM3PER8	PWM3PW 11	PWM3PW 10	PWM3PW 9	PWM3PW 8
MSB							LSB
Bit7-4 **PWM3PER[11:8]:** PWM3 period (high nibble) Bit3-0 **PWM3PW[11:8]:** PWM3 width (high nibble)							

5.7 Zero Crossing Sense (ZCS)

Krankl implements a circuit for sensing zero crossing events (positive or negative) from the AC supply. This feature is used to provide an accurate flicker free dimming experience.

The main characteristics of the ZCS are:

- Determine the Zero Crossing of an AC Signal for the following uses:
  - Accurate Zero Crossing used as a timing reference for turning on Light triacs and accurately setting the illumination level of the attached lamp bulb.
  - Must provide polarity of event: Negative to Positive crossing and Positive to Negative crossing.
- Determine if the AC signal is present or absent
  - When the AC signal is removed from the circuit, an alert is sent to the Processor. The interruption of power will be used for housekeeping such as storing power off state to Flash.

5.7.1 ZCS Usage Description

The Zero Crossing Sense Circuit will provide an accurate measurement for both the rising and falling portions of the AC input wave at either 50 or 60Hz.

One output of the ZCS circuit is to be used as the timing for the Hardware Triac Timer.

Zero Crossing Detector Performance Specification					
name	**conditions**	**min**	**typ**	**max**	**unit**
Line Voltage Range 60Hz	V3p3ANA=3.3V, T_A=27°C	90	110	140	Vrms
Line Voltage Range 50Hz	V3p3ANA=3.3V, T_A=27°C	200	220	240	Vrms

5.7.1.1 *Zero Crossing Detector (ZCS)*

The Zero Crossing Detector monitors a sinusoidal input voltage, e.g. the line voltage. Based upon the event and its gradient (rising or falling edge) it generates a differential square output waveform

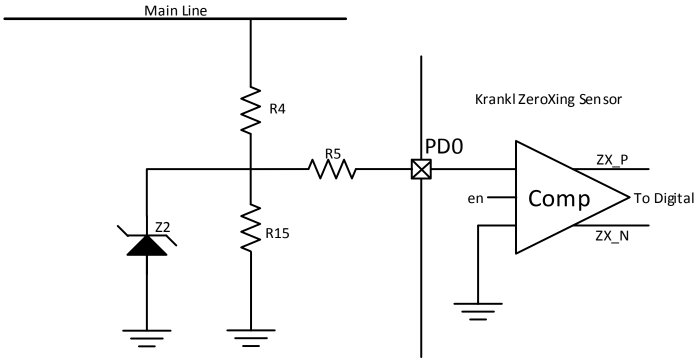


Figure 9 Zero Crossing Detector Implementation

name	conditions	R4	R15	unit
Line Voltage Range 60Hz, 110V	V3p3ANA=3.3V, T_A=27°C	220k	3.3k	Ohm
Line Voltage Range 50Hz, 220V	V3p3ANA=3.3V, T_A=27°C	200K	1.4K	Ohm

5.8 Digital Triac Timer

Krankl implements two identical digital triac timing blocks used to activate the triacs for consistent illumination without any flicker of the emitted light.
The main characteristics of this timer are:

- Timing is based on the Zero Crossing Sense Circuit
- SW will control the illuminate and dim rate of each lamp during turn on / off
- Light shall not flicker, including during modification of the illumination level
- Triac Pulse methods are configurable

5.8.1 **Digital Triac Timer Usage Description**

In order to reduce the burden on the processor, control of the Lamp Triacs has been shifted to a digital block.
The Zero Crossing Circuit provides the reference for the circuit to assert or de-assert the Triac GPIO.
The default state of the output is ‘High’. This corresponds to the Triac being off, which keeps the Lamp in an off state.
In order to control the triacs to meet the assortment of devices connected to the Lights, the DTT has 4 programmable Activate and Deactivate times; the Primary reference for each assert and de-assert time shall be configurable.

Example (Referenced to positive and negative crossing):

- Positive crossing reference registers
  - Activate0
  - Deactivate0
- Negative crossing reference registers
  - Activate1
  - Deactivate1

The programming of the Digital Triac Timer is double buffered, thus preventing any flickering occurring when modifying the illumination state. The new value in the buffer shall be used on the 50/60 Hz Cycle after it is written.

The DTT has a 12-bit pre-scalar which, combined with the 8-bit programmable activate/deactivate counter, allows a maximum countable period of 34.952ms (2^20 * 30MHz clock period). At 50Hz the maximum required is 20ms and 16.67ms at 60Hz. The resolution is 34.952ms/2^8, which is 0.1365ms. Step is 0.84% step for 60Hz and 1.36% step for 50Hz.

If the Short Circuit Protection Criteria fails, the lamp shall not be activated on the next cycle.
The figure below provides an example of how the zero crossing combined with the timers are used to dimmer the light.

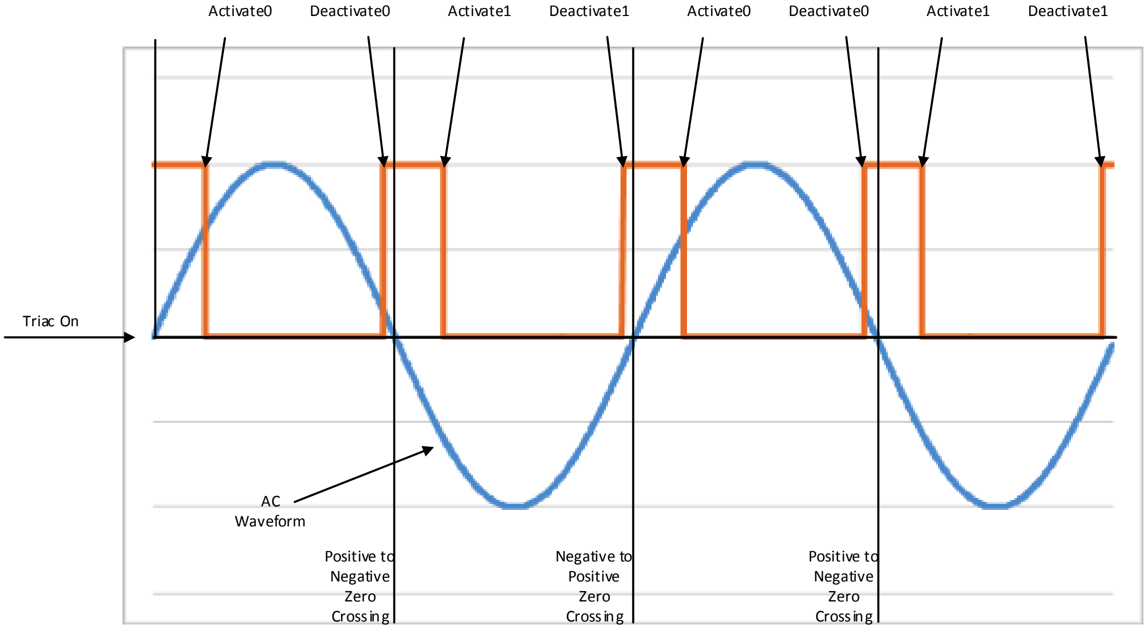

Activate0
Deactivate0
Activate1
Deactivate1
Activate0
Deactivate0
Activate1
Deactivate1
Triac On
AC Waveform
Positive to Negative Zero Crossing
Negative to Positive Zero Crossing
Positive to Negative Zero Crossing

Figure 10 Single Digital Triac Timer example

5.8.2 DTT Control Register Map

The following registers are provided to control the DTT:

Digital Triac Timer Control Register Map				
Address	Register Name	Description	Reset Value	Reference
0x50000050	DTT_CTRL	DTT General control	0x00	
0x50000051	DTT0_ACT0	DTT 0 TIMER 0 assert register	0x00	
0x50000052	DTT0_DACT0	DTT 0 TIMER 0 deassert register	0x00	
0x50000053	DTT0_ACT1	DTT 0 TIMER 1 assert register	0x00	
0x50000054	DTT0_DACT1	DTT 0 TIMER 1 deassert register	0x00	
0x50000055	DTT0_PRESCALE_L	DTT 0 prescaler low byte	0x00	

Digital Triac Timer Control Register Map				
Address	Register Name	Description	Reset Value	Reference
0x50000056	DTT0_PRESCALE_M	DTT 0 prescaler high nibble	0x00	
0x50000057	Reserved		0x00	
0x50000058	Reserved		0x00	
0x50000059	DTT1_ACT0	DTT 1 TIMER 0 assert register	0x00	
0x5000005A	DTT1_DACT0	DTT 1 TIMER 0 deassert register	0x00	
0x5000005B	DTT1_ACT1	DTT 1 TIMER 1 assert register	0x00	
0x5000005C	DTT1_DACT1	DTT 1 TIMER 1 deassert register	0x00	
0x5000005D	DTT1_PRESCALE_L	DTT 1 prescaler low byte	0x00	
0x5000005E	DTT1_PRESCALE_M	DTT 1 prescaler high nibble	0x00	
0x5000005F			0x00	

DTT_CTRL: DTT General Control							
DTTCTRL		0x50000050			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DTT1_REF 1	DTT1_REF 0	DTT1_INV	DTT1_EN	DTT0_REF 1	DTT0_REF 0	DTT0_INV	DTT0_EN
MSB							LSB

DTT_CTRL: DTT General Control

Bit7 **DTT1_REF1:** DTT1 Reference bit for DTT1_TIMER1

0 = DTT1 TIMER1 Reference positive edge zero crossing, 1st half of semicycle

1 = DTT1 TIMER1 Reference negative edge zero crossing, 2nd half of semicycle

Bit6 **DTT1_REF0:** DTT1 Reference bit for DTT1_TIMER0

0 = DTT1 TIMER0 Reference positive edge zero crossing, 1st half of semicycle

1 = DTT1 TIMER0 Reference negative edge zero crossing, 2nd half of semicycle

Bit5 **DTT1_INV:** DTT1 inversion bit.

0 = Normal logic

1 = Inverted logic (DTT1 output will still be high when disabled)

Bit4 **DTT1_EN:** DTT1 enable bit.

0 = DTT1 Disabled

1 = DTT1 Enabled

Bit3 **DTT0_REF1:** DTT0 Reference bit for DTT0_TIMER1

0 = DTT0 TIMER1 Reference positive edge zero crossing, 1st half of semicycle

1 = DTT0 TIMER1 Reference negative edge zero crossing, 2nd half of semicycle

Bit2 **DTT0_REF0:** DTT0 Reference bit for DTT0_TIMER0

0 = DTT0 TIMER0 Reference positive edge zero crossing, 1st half of semicycle

1 = DTT0 TIMER0 Reference negative edge zero crossing, 2nd half of semicycle

Bit1 **DTT0_INV:** DTT0 inversion bit.

0 = Normal logic

1 = Inverted logic (DTT0 output will still be high when disabled)

Bit0 **DTT0_EN:** DTT0 enable bit.

0 = DTT0 Disabled

1 = DV3p3TT0 Enabled

DTT0_ACT0: DTT0 TIMER 0 activation register

DTT0_ACT0	0x50000051	0x00

DTT0_ACT0: DTT0 TIMER 0 activation register							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DTT0ACT0_7	DTT0ACT0_6	DTT0ACT0_5	DTT0ACT0_4	DTT0ACT0_3	DTT0ACT0_2	DTT0ACT0_1	DTT0ACT0_0
MSB							LSB
Bit7-0 **DTT0ACT0[7:0]:** DTT0 TIMER0 assert value							

DTT0_DACT0: DTT0 TIMER 0 deactivation register							
DTT0_DACT0		0x50000052			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DTT0DCT0_7	DTT0DCT0_6	DTT0DCT0_5	DTT0DCT0_4	DTT0DCT0_3	DTT0DCT0_2	DTT0DCT0_1	DTT0DCT0_0
MSB							LSB
Bit7-0 **DTT0DCT0[7:0]:** DTT0 TIMER0 deassert value							

DTT0_ACT1: DTT0 TIMER 1 activation register							
DTT0_ACT1		0x50000053			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DTT0ACT1_7	DTT0ACT1_6	DTT0ACT1_5	DTT0ACT1_4	DTT0ACT1_3	DTT0ACT1_2	DTT0ACT1_1	DTT0ACT1_0
MSB							LSB
Bit7-0 **DTT0ACT1[7:0]:** DTT0 TIMER1 assert value							

DTT0_DACT1: DTT0 TIMER 1 deactivation register							
DTT0_DACT1		0x50000054			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DTT0_DACT1: DTT0 TIMER 1 deactivation register							
DTT0DCT1_7	DTT0DCT1_6	DTT0DCT1_5	DTT0DCT1_4	DTT0DCT1_3	DTT0DCT1_2	DTT0DCT1_1	DTT0DCT1_0
MSB							LSB
Bit7-0 **DTT0DACT1 [7:0]:** DTT0 TIMER1 deassert value							

DTT0_PRESCALE_L: DTT0_PRESCALE register							
DTT0_PRE_L		0x50000055			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DTT0PRE_7	DTT0PRE_6	DTT0PRE_5	DTT0PRE_4	DTT0PRE_3	DTT0PRE_2	DTT0PRE_1	DTT0PRE_0
MSB							LSB
Bit7-0 **DTT0_PRESCALER[7:0]:** DTT0 prescaler value LSB							

DTT0_PRESCALE_M: DTT0_PRESCALE register							
DTT0_PRE_M		0x50000056			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
				DTT0PRE_11	DTT0PRE_10	DTT0PRE_9	DTT0PRE_8
MSB							LSB
Bit3-0 **DTT0_PRESCALER[11:8]:** DTT0 prescaler value MSB							

DTT1_ACT0: DTT1 TIMER 0 activation register							
DTT1_ACT0		0x50000059			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DTT1ACT0_7	DTT1ACT0_6	DTT1ACT0_5	DTT1ACT0_4	DTT1ACT0_3	DTT1ACT0_2	DTT1ACT0_1	DTT1ACT0_0

DTT1_ACT0: DTT1 TIMER 0 activation register							
MSB							LSB
Bit7-0 **DTT1ACT0[7:0]:** DTT1 TIMER0 assert value							

DTT1_ACT1: DTT1 TIMER 1 activation register							
DTTTMR5_PER		0x5000005A			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DTT1DCT0_7	DTT1DCT0_6	DTT1DCT0_5	DTT1DCT0_4	DTT1DCT0_3	DTT1DCT0_2	DTT1DCT0_1	DTT1DCT0_0
MSB							LSB
Bit7-0 **DTT1DACT0[7:0]:** DTT1 TIMER0 deassert value							

DTT1_DACT1: DTT1 TIMER 1 deactivation register							
DTT1_DACT1		0x5000005C			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DTT1DCT1_7	DTT1DCT1_6	DTT1DCT1_5	DTT1DCT1_4	DTT1DCT1_3	DTT1DCT1_2	DTT1DCT1_1	DTT1DCT1_0
MSB							LSB
Bit7-0 **DTT1DACT1[7:0]:** DTT1 TIMER1 deassert value							

DTT1_PRESCALE_L: DTT1_PRESCALE register							
DTT1_PRE_L		0x5000005D			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DTT1PRE_7	DTT1PRE_6	DTT1PRE_5	DTT1PRE_4	DTT1PRE_3	DTT1PRE_2	DTT1PRE_1	DTT1PRE_0
MSB							LSB

DTT1_PRESCALE_L: DTT1_PRESCALE register
Bit7-0 **DTT1_PRESCALER[7:0]:** DTT1 prescaler value LSB

DTT1_PRESCALE_M: DTT1_PRESCALE register							
DTT1_PRE_M		0x5000005E			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
				DTT1PRE_11	DTT1PRE_10	DTT1PRE_9	DTT1PRE_8
MSB							LSB
Bit3-0 **DTT1_PRESCALER[11:8]:** DTT1 prescaler value MSB							

5.9 Serial Interfaces

5.9.1 UART

Krankl includes a UART (**U**niversal **A**synchronous **R**eceiver **T**ransmitter) module. The main characteristics are defined below:

- Interrupt available for transmission, reception and error events
- Reception timeout timer
- Programmable break reception and transmission
- Programmable parity with "sticky" parity option
- Selectable number of bits from 5 to 8
- Selectable number of stop-bits: 1, 1 ½, 2
- Programmable loop-back
- Swappable TXD and RXD (PE[4] and PE[5])
- Transmitter Polarity selection

5.9.1.1 *UART Operation*

The UART protocol requires two wires (UTXD and URXD). PE[4] and PE[5] are configured as UTXD and URXD when the MDUART bit is set in the pin configuration register, PCONF. In order to use the UART, the following steps must be followed:

- Step 1: Select the pins position (normal or swapped) of the interface and also its polarity. The normal position (not swapped) is TX= PE[4] and RX = PE[5].

Code Example: Selecting UART with normal polarity and swapped: (UART pins swapped: TX=PE[5] and RX = PE[4])

```
UART_Setup(UARTSWAP_EN, UARTPOL_NORMAL, UART_MODE_EN);
```

- Step 2: Define the following parameters:
  - Loop back: Used mainly in tests, internally connects the output to the input.
  - Break enable: Pulls the output down while asserted, raising the output once de-asserted.
  - Sticky parity: Forces the parity to stay stable in one direction.
  - Even/Odd parity selection and enable: Selection and enable of Even or Odd parity bits.
  - Number of stop bits: Selection of 1 (default), 1½ (5-bit communications only) or 2 stop bits.
  - Data size in bits (5,6,7,8): Selection of the number of bits used in the communication

Code example: Setting the above parameters: (no loop-back, no break signal, no sticky parity, no parity, one stop-bit, 8-bit communications)

```
      UART_Ctrl(UART_LBDIS,  UART_BREAKDIS,  UART_STPDIS,UART_ODDEN,  UART_PARDIS,  \
                        UART_10STOP, UART_8BITS);
```

Define the baud rate. The baud rate is calculated as follows: (UARTDIV is a 16-bit register)

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

The following table provides some register values, baud rates and related errors:

UART baud rates, divider values and errors with 30MHz input clock			
Baud	**UARTDIV**	**Real Baud**	**Error (%)**
300	6249	300	0.00%
600	3124	600	0.00%
1200	1562	1200	0.03%
2400	780	2401	0.03%
4800	390	4795	0.10%
9600	194	9615	0.16%
19200	97	19133	0.35%
38400	48	38265	0.35%
57600	32	56818	1.36%
115200	15	117188	1.73%

Code Example: Setting the UART to operate at 9600 baud:

```
      UART_BaudRateDivider(194);
```

- Step 3: Enable the UART and its interrupt: The UART may generate an interrupt for events related to:
  - Transmission completed.
  - Reception: Timeout of ~40 bit-times without reception, and data received.
  - Errors detected: Framing error, parity error, and overrun error.
  - Break signal detected (received).

Code example: Enabling the UART with no timeout interrupts, errors, transmission and reception interrupts with interrupts once 1 byte is received.

```
//Timeout interrupt disabled, Error enabled, TX enabled, RX enabled

      UART_Interrupt_Control(UART_TOUTDIS,UART_ERREN,UART_TXEN,UART_RXEN);

//UART enabled
```

```
      UART_Ctrl1( UARTEN);

//Enabling interrupt from UART at the microcontroller

      NVIC_EnableIRQ(UART_IRQn);         //Enable UART interrupt
```

Processing of the UART interrupt:

```
void UART_Handler( void )           // IRQ 8 UART
{
      switch(UART_Interrupt_Status())
      {
            case  UART_ERROR:
                  //Process reception error here
                  SWITCH(UART_CheckError())
                  {
                        case UART_FRAMING_ERROR: // Process this error here
                              break;
                        case UART_PARITY_ERROR: // Process this error here
                              break;
                        case UART_OVERRUN_ERROR: // Process this error here
                              break;
                        case 0: // No error, exit
                        default:
                              break;
                  }
                  break;
            case  UART_RXRDY:
                  //Process data received
                  mydata = *UARTDATA;     //Read data received by uart into mydata
                  break;
            case  UART_TIMEOUT:
                  //Process reception timeout
                  break;
            case  UART_TXDONE:
                  //Process transmission complete
                  break;
            case  UART_NOINT: //No int.
            default:
            //If no interrupt asserted or something else happened nothing to do
                        break;
      }
}
```

5.9.1.2 *UART Registers*

The following registers for the UART are defined in Krankl:

UART Control Register Map				
Address	Register Name	Description	Reset Value	Reference
0x50000010	UARTDATA	UART Data Register	0x00	
0x50000011	UARTICTRL	UART Interrupt Control Register	0x00	
0x50000012	UARTLCTRL	UART Line Control Register	0x00	
0x50000013	UARTEN	UART Enable Register	0x00	
0x50000014	UARTLSTAT	UART Line Status	0x00	
0x50000016-7	UARTDIV	UART Baud Rate Divider	0x0000	

UARTDATA: UART Data Register							
UARTDATA		0x50000010			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
UARTD7	UARTD6	UARTD5	UARTD4	UARTD3	UARTD2	UARTD1	UARTD0
MSB							LSB
Bit7-0 UARTD [7:0]: UART data, both received and to be transmitted.							

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

UARTICTCL: UART Interrupt Control Register

Bit7-4 **UISTTS [3:0]:** UART Interrupt status:

0001 = No Interrupt asserted

0010 = Transmission completed

0100 = Data received

0110 = Reception error

1100 = Reception timeout (~40 bit-time)

Bit3 **UTOUTIEN**: UART time-out interrupt enable bit:

0 = Time-out interrupt disabled

1 = Time-out interrupt enabled

Bit2 **URXERREN**: UART reception error interrupt enable bit:

0 = Reception error interrupt disabled

1 = Reception error interrupt enabled

Bit1 **UTXIEN**: UART transmission completed interrupt enable bit:

0 = Transmission completed interrupt disabled

1 = Transmission completed interrupt enabled

Bit0 **URXIEN**: UART reception interrupt enable bit:

0 = Reception interrupt disabled

1 = Reception interrupt enabled

UARTCTRL: UART Line Control Register

UARTLCTRL		0x50000012			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ULOOPEN	UBREAKEN	USTICKEN	UPARITY	UPAREN	USTOP	USTOP	USIZE
MSB							LSB

UARTCTRL: UART Line Control Register

Bit7 **ULOOPEN**: UART loop back enable:

0 = UART loop back disabled

1 = UART loop back enabled

Bit6 **UBREAKEN:** UART break enable:

0 = UART break disabled

1 = UART break enabled

Bit5 **USTICKEN**: UART sticky parity enable bit:

0 = Sticky parity disabled

1 = Sticky parity enabled

Bit4 **UPARITY**: UART parity bit:

0 = Odd parity

1 = Even parity

Bit3 **UPAREN**: UART parity enable bit:

0 = Parity disabled

1 = Parity enabled

Bit2 **USTOP**: UART stop bit:

0 = One stop bit

1 = If a 5-bit transmission it selects1.5 stop bits, otherwise 2 stop bits (6, 7 and 8 bits)

Bit1-0 **USIZE**: UART transmission size:

00 = 5-bit data

01 = 6-bit data

10 = 7-bit data

11 = 8-bit data

UARTCTRL1: UART Enable Register

UARTCTRL1		0x50000013			0x00		
Reserved	Reserved	Reserved	Reserved	R/W	Reserved	Reserved	Reserved

UARTCTRL1: UART Enable Register							
-	-	-	-	UARTEN	-	-	-
MSB							LSB
Bit3 **UARTEN**: UART enable: 0 = UART disabled 1 = UART enabled							

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

UARTSTATUS: UART Line Status Register

Bit7 **UERR**: UART error:

0 = No error

1 = Error in UART

Bit6 **UTXEMPTY:** UART transmission empty:

0 = UART transmitter not empty

1 = UART transmitter empty

Bit5 **UTXFFEMPTY**: UART transmission buffer empty:

0 = TX buffer not empty

1 = TX buffer empty

Bit4 **UBREAKINT**: UART break interrupt:

0 = No break interrupt

1 = Break interrupt

Bit3 **UFRMERR**: UART framing error:

0 = UART no framing error

1 = UART framing error

Bit2 **UPRTYERR**: UART parity error:

0 = No parity error

1 = Parity error

Bit1 **UOVRUNERR**: UART overrun error:

0 = No overrun error

1 = Overrun error

Bit0 **UDTRDY**: UART data ready:

0 = No data ready (reception)

1 = Data ready (reception)

UARTDIV: UART Baud Rate Divider Register (16-bit)

UARTDIV		0x50000016			0x0000		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

UARTDIV: UART Baud Rate Divider Register (16-bit)							
UDIV7	UDIV6	UDIV5	UDIV4	UDIV3	UDIV2	UDIV1	UDIV0
UDIV15	UDIV14	UDIV13	UDIV12	UDIV11	UDIV10	UDIV9	UDIV8
MSB							LSB
Bit15-0UDIV [15:0]: UART clock divider							

5.9.2 SPI Interface

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. In Krankl, the SPI is implemented as a master. The module is compatible with an industry standard SPI interface. There are many references available, but a simple overview can be found at:
http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus

Krankl SPI module's main features are defined below:

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

The SPI protocol requires four wires (SCK, MISO, MOSI, and SS). The pins PE[3:0] are configured as the SPI bus when the MDSPI bit is set in the pin configuration register, PCONF. The following table describes how each pin is connected:

SPI Interface Signals			
Name	**Pin Number**	**Pin Name**	**Comments**
MISO	26	PE1	Mater In Slave Out
MOSI	28	PE3	Master Out Slave In
SCLK	27	PE2	Serial Clock

SPI Interface Signals			
Name	**Pin Number**	**Pin Name**	**Comments**
SS	25	PE0	Slave Select

5.9.2.1 *SPI Operation*

Only the master mode is implemented in Krankl. Krankl configures the clock frequency and generates the serial clock (SCK) for the interface. The data transfer is synchronous through SCK. The SPI is a full-duplex system; data is transmitted and received simultaneously. Krankl sends the information to the slave device through the MOSI line and receives the data through MISO line. The CPOL and CPHA bits in the SPI control register determine when to sample the data.

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

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

The timing diagram is shown below.

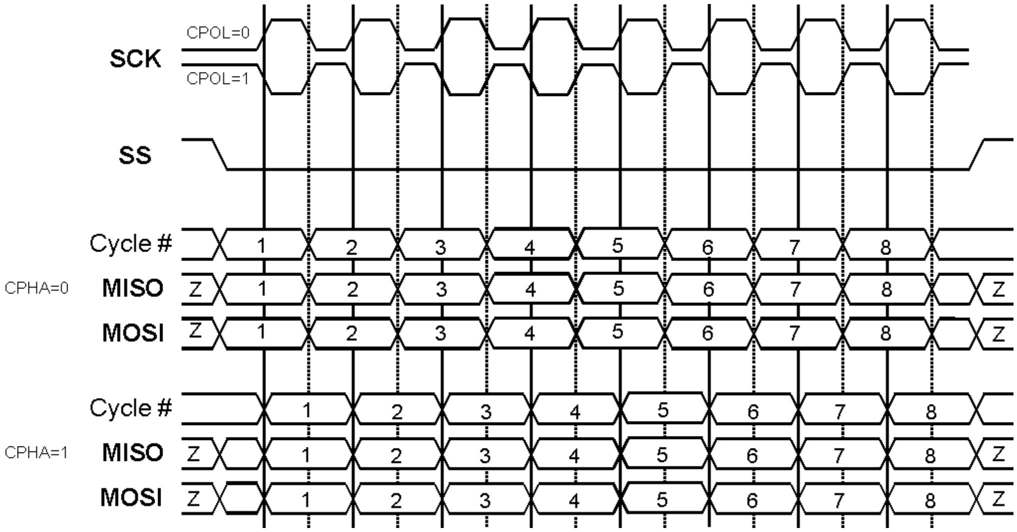


Figure 11 SPI Timing Diagram

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

To operate it properly the following steps must be performed:

- Step 1: Configure and enable the SPI: Select if the interrupt is enabled, set the polarity, phase and the clock divider.
  Code Example: Enabling the SPI interface with interrupt enabled, clock polarity negative (base value = 0), phase1 (data captured on the clock's falling edge and propagated on the rising edge), and divider = 2.

```
SPI_Config(SPI_INT_EN,SPI_EN,SPI_CPOL_NEG,SPI_PHASE1,2);
```

- Step 2: Enable the interrupt (if required):
  Code Example:

```
//Enabling interrupt from SPI at the microcontroller

NVIC_EnableIRQ(SPI_IRQn);         //Enable SPI interrupt
```

- Step 3: Process the Interrupt (if required) and detect the reason for the interrupt (error, transmission or reception related and act accordingly):
  Code Example: Processing SPI interrupt:

```
void SPI_Handler( void )            // IRQ A SPI
{
      uint8_t status;
      if ( (status = SPI_ReadStatus() ) & SPI_INT_FLAG )
      {
            if (status == SPI_WCOL)
            {
            //Process write collision (data is written to the SPI data
            //register while a SPI data transfer is in progress)

            }
            if (status == SPI_TX_FIFO_FULL)
            {
            //Process Transmission FIFO full
            }
            if (status == SPI_TX_FIFO_EMPTY)
            {
            //Process end of transmission of data previously
            //in transmission FIFO
                  *SPIDATA = outdata[j++];
                  *SPIDATA = outdata[j++];
                  *SPIDATA = outdata[j++];
            }
            if (status == SPI_RX_FIFO_FULL)
            {
            //Process reception FIFO full, normally by reading
            //all bytes of data
                  for ( I = SPI_ReadRxFifoSize(); I > 0; i--)
                        mydata[i++] = *SPIDATA;
            //Other processing here
            }
            if (status == SPI_RX_FIFO_EMPTY)
            {
            //Process when no more information is available (received)
            }
      }
}
```

5.9.2.2 *SPI Registers*

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

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

SPCR: SPI Control Register

Bit7 **SINTE**: SPI Interrupt enable

0 = Interrupt is disabled

1 = Interrupt is enabled

Bit4 **MSTR:** Master Mode Select Bit

SPI is always in master mode in Krankl, and therefore, it is always set to logic '1'.

Bit3 **CPOL** SPI clock polarity

0 = The base value of the clock is zero

1 = The base value of the clock is one

Bit2 **CPHA**: SPI clock phase

0 = data is captured on clock transition from base and data is propagated on the clock transiti base

1 = data is captured on clock transition to base and data is propagated on the clock transition base

Bit1-0 **SCKSTD[1:0]**: SPI standard clock divider selection

Please refer to SPER register for system clock

SPSR: SPI Status Register

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

SPSR: SPI Status Register

Bit7 **SINTF**: SPI interrupt flag

0 = Interrupt not asserted

1 = Interrupt asserted

Bit6 **SWCOL:** SPI write collision is set when the SPDR register is written to while the transmit FIFO i

0 = No collision

1 = collision

Bit3 **STXFF**: SPI transmit FIFO full

0 = transmit FIFO not full

1 = transmit FIFO full

Bit2 **STXFE**: SPI transmit FIFO empty

0 = transmit FIFO not empty

1 = transmit FIFO empty

Bit1 **SRXFF**: SPI reception FIFO full

0 = reception FIFO not full

1 = reception FIFO full

Bit0 **SRXFE**: SPI reception FIFO empty

0 = reception FIFO not empty

1 = reception FIFO empty

SPDR: SPI Data Register

SPDR		0x5000001E			0xXX		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
SPID7	SPID6	SPID5	SPID4	SPID3	SPID2	SPID1	SPID0
MSB							LSB

Bit7-0 **SPID[7:0]**: SPI data, used in both transmission and reception

SPER: SPI Extension Register

SPER: SPI Extension Register

SPER		0x5000001F			0x00		
R/W	R/W	Reserved	Reserved	R/W	R/W	Reserved	Reserve d
SICNT1	SINCT0	-	-		SPE	SCKEXT1	SCKEXT0
MSB							LSB

Bit7-6 **SICNT[1:0]**: SPI Interrupt Counter Bits

00 = SINTF is set after every completed transfer

01 = SINTF is set after two completed transfers

10 = SINTF is set after three completed transfers

11 = SINTF is set after four completed transfers

Bit2 **SPE**: SPI Enable

0 = SPI module is disabled

1 = SPI module is enabled

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

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

5.10 GPIOS

Krankl provides 28 general-purpose IO pins. Krankl's IO pins are implemented with several different capabilities divided into the following groups:

- GIO is a general purpose IO referred to V12. When used as an output it is capable of sinking (to ground) or sourcing from the V12 supply. When used as an input the GIO can be programmed to be high impedance or have pull up or pull down with various strenghts. The state of the pin can be read while in output mode, therefore allowing for a software-based over current protection.
- SIO has the same functions as GIO, but with a much higher sink capability, which may be protected against over-current through software.
- 3V3IO are 3.3V digital IOs, which are referenced to an internal regulator.

5.10.1 **General Purpose IO (GIO)**

The GIO interface pins are intended to operate as reconfigurable general-purpose inputs or outputs referenced to V12. Additionally, they can be used for open-drain pull-down on systems with voltage equal to or lower than their supply voltage, e.g. for 3.3V systems. The strength of the pull up and pull down are programmable. Additionally, outputs are self-protected against potentially damaging loads. When the high-level output is activated, the output current is limited by an internal circuit to a current which can be sustained continuously.

When the low-level output is activated, protection against thermal damage caused by a short circuit must be done by user software by comparing the voltage level on the pad with the intended driven level shortly after activation. If the level is too high (VIL threshold not triggered), the user software must tristate or activate a high level output within 200 milliseconds to avoid potential damage to the chip.

This protection is not intended to protect these pins against voltage overshoot from driving strongly inductive loads, and so GIO pins should not be used for inductive loads without additional protection on the PCB (**P**rinted **C**ircuit **B**oard).

Pin configuration is accomplished using special function registers, PAnCFG, PBnCFG, and PCnCFG.

The receiver is active at all times, and any read from the port will always return the data read from the pin, even if the pin is set as an output.

GIO and SIO Pin Functional Configuration			
DD	**PUP**	**PDN**	**Pin Function**
0	0	0	high-Z input
0	0	1	input with pull down (current level set by STR)

0	1	0	input with pull up (current level set by STR)
0	1	1	reserved
1	X	0	push/pull (to Supply levels) output, with simple load protection
1	0	X	
1	1	1	open-drain output, with simple load protection

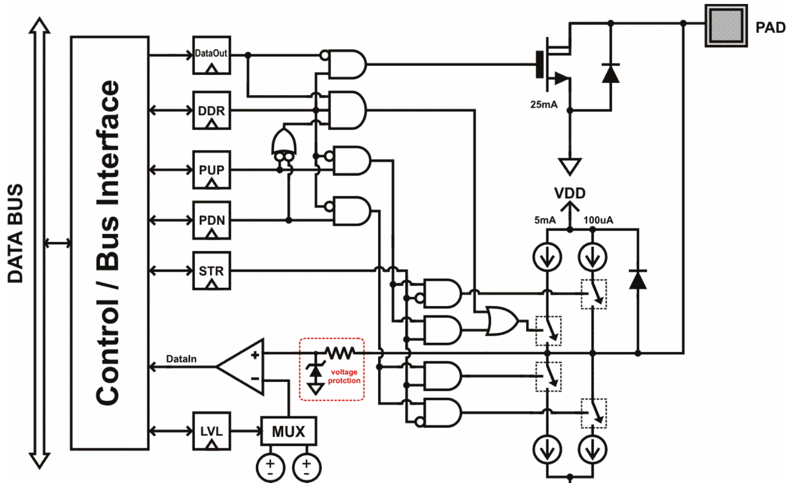


Figure 12GIO Implementation

5.10.2 **High Current Pull down IO (SIO)**

SIO interface pins are intended to operate as reconfigurable inputs or outputs referenced to V12, optimized for use as high-current pull-downs. They can be used for open-drain pull-down on systems with voltage equal to or lower than their supply voltage, e.g. 3.3V systems. High current and low current pull ups can be selected in receive mode. Internal circuits are protected against sustained high voltage up to 45V applied to the pad. A pull-down mode may be activated when using the IO as an output. Pull-down output mode may be protected against potentially damaging loads by comparing the voltage level on the pad with a maximum level corresponding to a safe margin for thermal damage. If the power dissipated in the transistor is too high, which happens when output voltage is above VIL with the pull down on, then the user software must turn the pull down off within approximately 200 milliseconds to avoid thermal damage. When the high-level output is activated, the output current is limited by an internal circuit to a current which can be sustained continuously. The receiver is active at all times, and any read from the port will always return the data read from the pin, even if the pin is set as an output.

It is recommended that the pin be configured as push/pull when driving inductive loads to assist the freewheel function and reduce strain on the ESD diode which is responsible for conducting the free wheel current by allowing some current to pass through the PMOS transistor.
If the load is resistive, then the open drain mode of protection is preferred.

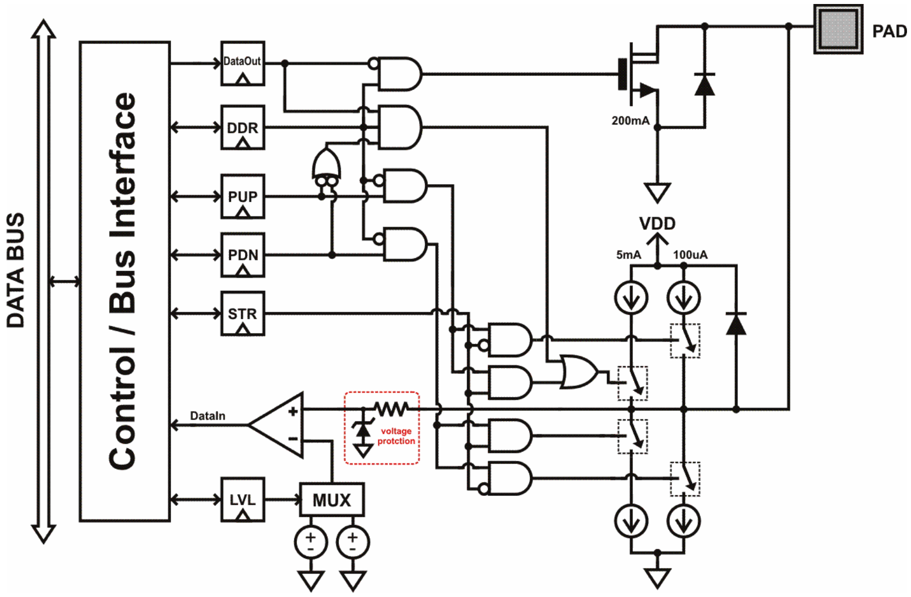


Figure 13 SIO Implementation

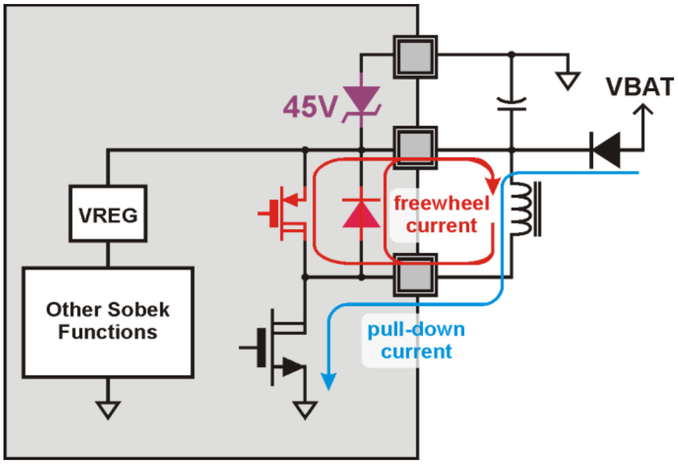


Figure 14 Free wheel action

5.10.3 GIO and SIO Connections to ADC

All GIOs and SIOs are connected to the ADC input channel selector (refer to 5.5). The signals applied to these pins can either directly goes through the multiplexor or is attenuated by factor of 8 and then goes through the multiplexor depending on the LVL bit.

5.10.4 Port Configuration Registers

Port configuration registers				
Address	Register Name	Description	Reset Value	Reference
0x50000060	PORTA	Port A input and output register		
0x50000061	PORTC	Port C input and output register		
0x50000062	PORTD	Port D input and output register		
0x50000063	PORTE	Port E input and output register		
0x50000064	PORTF	Port F input and output register		
0x50000065	PMODE	Pin Mode register		
0x50000066	PAINT	Port A Interrupt Enable Register		
0x50000067	PCINT	Port C Interrupt Enable Register		
0x50000068	PDINT	Port D Interrupt Enable Register		
0x50000069	PEINT	Port E Interrupt Enable Register		
0x5000006A	PFINT	Port F Interrupt Enable Register		

The following registers control the behavior of the GPIO pins:

<u>PORTA:</u> Port A input and output register. When a bit is selected as output the corresponding pin will reflect the register's bit. When a bit is selected as input the bit will reflect the condition of the pin.

PORTA		0x50000060			0x00		
Reserved	R/W	R/W	R/W	R/W	R/W	R/W	R/W
-	PA6	PA5	PA4	PA3	PA2	PA1	PA0
MSB							LSB

Bit3-0 **PA[6:0]**: Port A register bits.

0 = Pin state is '0'

1 = Pin state is '1'

<u>PORTC:</u> Port C input and output register. When a bit is selected as output the corresponding pin will reflect the register's bit. When a bit is selected as input the bit will reflect the condition of the pin.

PORTC		0x50000061			0x00		
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	R/W	R/W
-	-	-	-	-	-	PC1	PC0
MSB							LSB

Bit3-0 **PC[1:0]**: Port C register bits.

0 = Pin state is '0'

1 = Pin state is '1'

<u>PORTD:</u> Port D input and output register. When a bit is selected as output the corresponding pin will reflect the register’s bit. When a bit is selected as input the bit will reflect the condition of the pin.

PORTD		0x50000062			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
MSB							LSB

Bit7-0 **PD[7:0]**: Port D register bits.

0 = Pin state is ‘0’

1 = Pin state is ‘1’

<u>PORTE:</u> Port E input and output register. When a bit is selected as output the corresponding pin will reflect the register’s bit. When a bit is selected as input the bit will reflect the condition of the pin.

PORTE		0x50000063			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
MSB							LSB

Bit3-0 **PE[7:0]**: Port E register bits.

0 = Pin state is ‘0’

1 = Pin state is ‘1’

<u>PORTF:</u> Port F input and output register. When a bit is selected as output the corresponding pin will reflect the register's bit. When a bit is selected as input the bit will reflect the condition of the pin.

PORTF		0x50000064			0x00		
Reserved	Reserved	Reserved	Reserved	Reserved	R/W	R/W	R/W
-	-	-	-	-	PF2	P**F**1	PF0
MSB							LSB

Bit3-0 **PE[2:0]**: Port E register bits.

0 = Pin state is '0'

1 = Pin state is '1'

<u>PMODE:</u> Pin Mode register.

PMODE		0x50000065			0x00		
Reserved	Reserved	Reserved	Reserved	Reserved	R/W	R/W	R/W
-	-	-	upswap	modeDTT	modeSPI	modeUART	utxpol
MSB							LSB

PMODE: Pin Mode register.

Bit4 **upswap**: UART Pin Swap

0 = PE4 as UART_RX and PE5 as UART_TX

1 = PE5 as UART_RX and PE4 as UART_TX

Bit3 **modeDTT**: pin configuration

0 = PC0 and PA5 as GOIOs

1 = PC0 and PA5 for DTT

Bit2 **modeSPI**: pin configuration

0 = PE0-3 as GPIOs

1 = PE0-3 for SPI

Bit1 **modeUART**: pin configuration

0 = PE4 and PE5 as GPIOs

1 = PE4 and PE5 for UART

Bit0 **utxpol**: UART TX polarity

0 = non inverted

1 = inverted

PAINT: Port A Interrupt Enable Register.

PAINT		0x50000066			0x00		
Reserved	Reserved	R/W	R/W	R/W	R/W	R/W	R/W
-	-	PAINTM5	PAINTM4	PAINTM3	PAINTM2	PAINTM1	PAINTM0
MSB							LSB

Bit6-0 **PAINTM[6:0]**: Port A Interrupt Enable bit.

0 = Pin interrupt disabled

1 = Pin interrupt enabled

PCINT: **Port C Interrupt Enable Register.**							
PCINT		0x50000067			0x00		
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	R/W	R/W
-	-	-	-	-	-	PCINTM1	PCINTM0
MSB							LSB

Bit1-0 **PCINTM[1:0]**: Port A Interrupt Enable bit.

0 = Pin interrupt disabled

1 = Pin interrupt enabled

PDINT: **Port D Interrupt Enable Register.**							
PDINT		0x50000068			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PDINTM7	PDINTM6	PDINTM5	PDINTM4	PDINTM3	PDINTM2	PDINTM1	PDINTM0
MSB							LSB

Bit7-0 **PDINTM[7:0]**: Port D Interrupt Enable bit.

0 = Pin interrupt disabled

1 = Pin interrupt enabled

PEINT: Port E Interrupt Enable Register.							
PORTE		0x50000069			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PEINTM7	PEINTM6	PEINTM5	PEINTM4	PEINTM3	PEINTM2	PEINTM1	PEINTM0
MSB							LSB

Bit7-0 **PEINTM[7:0]**: Port E Interrupt Enable bit.

0 = Pin interrupt disabled

1 = Pin interrupt enabled

PFINT: Port E Interrupt Enable Register.							
PORTF		0x5000006A			0x00		
Reserved	Reserved	Reserved	Reserved	Reserved	R/W	R/W	R/W
-	-	-	-	-	PFINTM2	PFINTM1	PFINTM0
MSB							LSB

Bit2-0 **PFINTM[2:0]**: Port F Interrupt Enable bit.

0 = Pin interrupt disabled

1 = Pin interrupt enabled

5.10.5 GIO PA Configuration Registers

GIO configuration registers				
Address	Register Name	Description	Reset Value	Reference
0x50018014	PA0CFG	PA0 GIO configuration register		
0x50018015	PA1CFG	PA1 GIO configuration register		
0x50018016	PA2CFG	PA2 GIO configuration register		
0x50018017	PA3CFG	PA3 GIO configuration register		
0x50018018	PA4CFG	PA4 GIO configuration register		
0x50018019	PA5CFG	PA5 GIO configuration register		
0x5001801A	PA6CFG	PA6 GIO configuration register		

<table>
<tr><td colspan="8">PA0CFG: PA0 GIO configuration register.</td></tr>
<tr><td colspan="2"><b>PA0CFG</b></td><td colspan="3">0x50018014</td><td colspan="3">0x00</td></tr>
<tr><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>-</td><td>-</td><td>-</td><td>DDR_PA0</td><td>STR_PA0</td><td>PUP_PA0</td><td>PDN_PA0</td><td>LVL_PA0</td></tr>
<tr><td>MSB</td><td></td><td></td><td></td><td></td><td></td><td></td><td>LSB</td></tr>
</table>

PA0CFG: PA0 GIO configuration register.

Bit4 **DDR**: Data Direction

0 = Input

1 = Output

Bit3 **STR**: Pull Up/Down Strength Control

0 = Low Strength (100uA)

1 = High Strength (5mA)

Bit2 **PUP**: Pull up enable.

0 = Disabled

1 = Enabled

Bit1 **PDN**: Pull Down enable.

0 = Disabled

1 = Enabled

Bit0 **LVL**: Input Threshold level

0 = Low Threshold

1 = High Threshold

PA1CFG: PA1 GIO configuration register.

PA1CFG		0x50018015			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	DDR_PA1	STR_PA1	PUP_PA1	PDN_PA1	LVL_PA1
MSB							LSB

PA1CFG: PA1 GIO configuration register.

Bit4 **DDR**: Data Direction

0 = Input

1 = Output

Bit3 **STR**: Pull Up/Down Strength Control

0 = Low Strength (100uA)

1 = High Strength (5mA)

Bit2 **PUP**: Pull up enable.

0 = Disabled

1 = Enabled

Bit1 **PDN**: Pull Down enable.

0 = Disabled

1 = Enabled

Bit0 **LVL**: Input Threshold level

0 = Low Threshold

1 = High Threshold

PA2CFG: PA2 GIO configuration register.

PA2CFG		0x50018016			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	DDR_PA2	STR_PA2	PUP_PA2	PDN_PA2	LVL_PA2
MSB							LSB

PA2CFG: PA2 GIO configuration register.

Bit4 **DDR**: Data Direction

0 = Input

1 = Output

Bit3 **STR**: Pull Up/Down Strength Control

0 = Low Strength (100uA)

1 = High Strength (5mA)

Bit2 **PUP**: Pull up enable.

0 = Disabled

1 = Enabled

Bit1 **PDN**: Pull Down enable.

0 = Disabled

1 = Enabled

Bit0 **LVL**: Input Threshold level

0 = Low Threshold

1 = High Threshold

PA3CFG: PA3 GIO configuration register.

PA3CFG		0x50018017			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	DDR_PA3	STR_PA3	PUP_PA3	PDN_PA3	LVL_PA3
MSB							LSB

PA3CFG: PA3 GIO configuration register.

Bit4 **DDR**: Data Direction

0 = Input

1 = Output

Bit3 **STR**: Pull Up/Down Strength Control

0 = Low Strength (100uA)

1 = High Strength (5mA)

Bit2 **PUP**: Pull up enable.

0 = Disabled

1 = Enabled

Bit1 **PDN**: Pull Down enable.

0 = Disabled

1 = Enabled

Bit0 **LVL**: Input Threshold level

0 = Low Threshold

1 = High Threshold

PA4CFG: PA4 GIO configuration register.

PA4CFG		0x50018018			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	DDR_PA4	STR_PA4	PUP_PA4	PDN_PA4	LVL_PA4
MSB							LSB

PA4CFG: PA4 GIO configuration register.

Bit4 **DDR**: Data Direction

0 = Input

1 = Output

Bit3 **STR**: Pull Up/Down Strength Control

0 = Low Strength (100uA)

1 = High Strength (5mA)

Bit2 **PUP**: Pull up enable.

0 = Disabled

1 = Enabled

Bit1 **PDN**: Pull Down enable.

0 = Disabled

1 = Enabled

Bit0 **LVL**: Input Threshold level

0 = Low Threshold

1 = High Threshold

PA5CFG: PA5 GIO configuration register.

PA5CFG		0x50018019			0x00		
Reserved	R/W	Reserved	R/W	R/W	R/W	R/W	R/W
-	SelPWM1_PA5	-	DDR_PA5	STR_PA5	PUP_PA5	PDN_PA5	LVL_PA5
MSB							LSB

PA5CFG: PA5 GIO configuration register.	
Bit6	**SelPWM**: Data MUX 0 = Data output selected 1 = PWM output selected
Bit4	**DDR**: Data Direction 0 = Input 1 = Output
Bit3	**STR**: Pull Up/Down Strength Control 0 = Low Strength (100uA) 1 = High Strength (5mA)
Bit2	**PUP**: Pull up enable. 0 = Disabled 1 = Enabled
Bit1	**PDN**: Pull Down enable. 0 = Disabled 1 = Enabled
Bit0	**LVL**: Input Threshold level 0 = Low Threshold 1 = High Threshold

PA6CFG: PA6 GIO configuration register.

PA6CFG		0x5001801A			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	DDR_PA6	STR_PA6	PUP_PA6	PDN_PA6	LVL_PA6
MSB							LSB

PA6CFG: PA6 GIO configuration register.	
Bit4	**DDR**: Data Direction
	0 = Input
	1 = Output
Bit3	**STR**: Pull Up/Down Strength Control
	0 = Low Strength (100uA)
	1 = High Strength (5mA)
Bit2	**PUP**: Pull up enable.
	0 = Disabled
	1 = Enabled
Bit1	**PDN**: Pull Down enable.
	0 = Disabled
	1 = Enabled
Bit0	**LVL**: Input Threshold level
	0 = Low Threshold
	1 = High Threshold

5.10.6 **SIO PC Configuration Registers**

SIO configuration registers				
Address	Register Name	Description	Reset Value	Reference
0x5001801C	PC0CFG	PC0 SIO configuration register		
0x5001801D	PC1CFG	PC1 SIO configuration register		

PC0CFG: PC0 SIO configuration register.							
PC0CFG		0x5001801C			0x00		
Reserved	R/W	Reserved	R/W	R/W	R/W	R/W	R/W
-	SelPWM2_PC0	-	DDR_PC0	STR_PC0	PUP_PC0	PDN_PC0	LVL_PC0
MSB							LSB

PC0CFG: PC0 SIO configuration register.

Bit6 **SelPWM**: Data MUX

0 = Data output selected

1 = PWM output selected

Bit4 **DDR**: Data Direction

0 = Input

1 = OutputZ

Bit3 **STR**: Pull Up/Down Strength Control

0 = Low Strength (100uA)

1 = High Strength (5mA)

Bit2 **PUP**: Pull up enable.

0 = Disabled

1 = Enabled

Bit1 **PDN**: Pull Down enable.

0 = Disabled

1 = Enabled

Bit0 **LVL**: Input Threshold level

0 = Low Threshold

1 = High Threshold

PC1CFG: PC1 SIO configuration register.

PC1CFG		0x5001801D			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	DDR_PA1	STR_PA1	PUP_PA1	PDN_PA1	LVL_PA1
MSB							LSB

PC1CFG: PC1 SIO configuration register.	
Bit4	**DDR**: Data Direction 0 = Input 1 = Output
Bit3	**STR**: Pull Up/Down Strength Control 0 = Low Strength (100uA) 1 = High Strength (5mA)
Bit2	**PUP**: Pull up enable. 0 = Disabled 1 = Enabled
Bit1	**PDN**: Pull Down enable. 0 = Disabled 1 = Enabled
Bit0	**LVL**: Input Threshold level 0 = Low Threshold 1 = High Threshold

5.10.7 GPIO PD Configuration Registers

GIO PD configuration registers				
Address	Register Name	Description	Reset Value	Comment
0x50018020	PD0CFG	PD0 GIO configuration register		AC power (APS)
0x50018021	PD1CFG	PD1 GIO configuration register		AC power (APS)
0x50018022	PD2CFG	PD2 GIO configuration register		
0x50018023	PD3CFG	PD3 GIO configuration register		
0x50018024	PD4CFG	PD4 GIO configuration register		
0x50018025	PD5CFG	PD5 GIO configuration register		
0x50018026	PD6CFG	PD6 GIO configuration register		
0x50018027	PD07CFG	PD7 GIO configuration register		

PD0CFG: PD0 GPIO configuration register.							
PD0CFG		0x50018020			0x00		
R/W	Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W
en_ZeroXing	-	-	-	OE_PD0	RE_PD0	PU_n_PD0	PD_PD0
MSB							LSB

PD0CFG: PD0 GPIO configuration register.
Bit7 **en_ZeroXing** 0 = ZeroXing Detector inactive 1 = ZeroXing Detector active Bit3 **OE**: Output Select 0 = High impedance 1 = Output Bit2 **RE**: GPIO read enable bit. 0 = Read Disabled 1 = Read Enabled Bit1 **PU_n**: IO pin Pull-Up control bit. 0 = Pull-Up Active 1 = Pull-Up Inactive Bit0 **PD**: GPIO read enable bit. 0 = Pull Down inactive 1 = Pull Down active

Krankl uses the Schmitt trigger in Pad PD0 to sense that the AC power (APS) is present. The intended use of this will be to sense Line Voltage to ensure that the Zero Crossing signal is valid.
The main characteristics are:

- Determine if the AC signal is present or absent
- For Line Detection, when the AC signal is removed from the circuit, an alert is sent to the Processor. The interruption of power will be used for housekeeping such as storing power off state to Flash.

The APS will provide the processor with the state of the AC Power accurate measurement for both the rising and falling portions of the AC input wave. This shall work at both 50Hz and 60Hz.
The APS is an important part of the Zero Crossing Sense Circuitry that prevents false crossings if the APS requirements are not met.

Line Detector Performance Specification – PD0			
name	**conditions**	**Line Limit - Vrms**	**Line Active Signal**

Line Detector Performance Specification – PD0			
name	**conditions**	**Line Limit - Vrms**	**Line Active Signal**
Line Voltage = 110V Range	V3p3ANA=3.3V, T_A=27°C	< 90	low
Line Voltage = 220V Range	V3p3ANA=3.3V, T_A=27°C	< 200	low

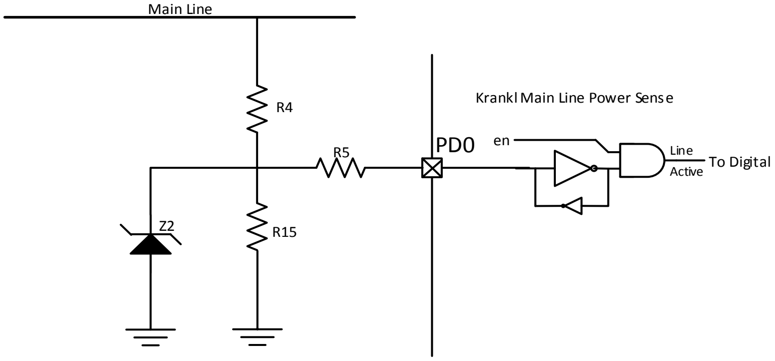


Figure 15 Line Detector Main Line at PD0

PD1CFG: PD1 GPIO configuration register.

PD1CFG		0x50018021			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	-	OE_PD1	RE_PD1	PU_n_PD1	PD_PD1
MSB							LSB

PD1CFG: PD1 GPIO configuration register.	
Bit3	**OE**: Output Select
	0 = High impedance
	1 = Output
Bit2	**RE**: GPIO read enable bit.
	0 = Read Disabled
	1 = Read Enabled
Bit1	**PU_n**: IO pin Pull-Up control bit.
	0 = Pull-Up Active
	1 = Pull-Up Inactive
Bit0	**PD**: GPIO read enable bit.
	0 = Pull Down inactive
	1 = Pull Down active

Krankl uses the Schmitt trigger in Pad PD1 to sense that the AC power (APS) is present. The intended use of this will be to sense Line Voltage to ensure that the Zero Crossing signal is valid.
The main characteristics are:

- Determine if the AC signal is present or absent
- For Line Detection, when the AC signal is removed from the circuit, an alert is sent to the Processor. The interruption of power will be used for housekeeping such as storing power off state to Flash.

The APS will provide the processor with the state of the AC Power accurate measurement for both the rising and falling portions of the AC input wave. This shall work at both 50Hz and 60Hz.
The APS is an important part of the Zero Crossing Sense Circuitry that prevents false crossings if the APS requirements are not met.

Line Detector Performance Specification – PD1			
name	**conditions**	**Line Limit - Vrms**	**Line Active Signal**
Line Voltage = 110V Range	V3p3ANA=3.3V, T_A=27°C	< 90	low
Line Voltage = 220V Range	V3p3ANA=3.3V, T_A=27°C	< 200	low

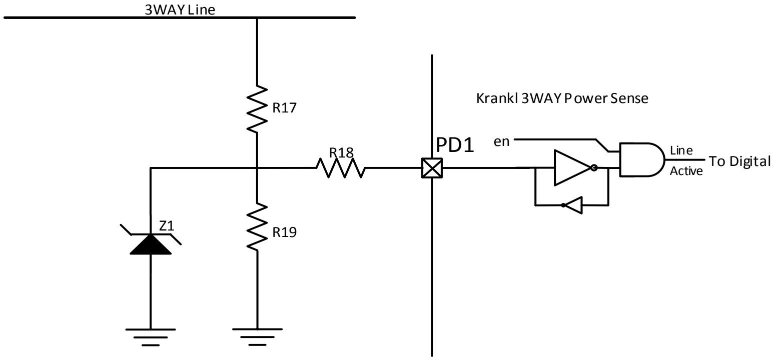


Figure 16 Line Detector Main Line at PD1

PD2CFG: PD2 GPIO configuration register.							
PD2CFG		0x50018022			0x00		
Reserved	R/W	Reserved	R/W	R/W	R/W	R/W	R/W
-	SelPWM_ PD2	-	-	OE_PD2	RE_PD2	PU_n_PD2	PD_PD2
MSB							LSB

PD2CFG: PD2 GPIO configuration register.

Bit6 **SelPWM**: Data MUX

0 = Data output selected

1 = PWM output selected

Bit3 **OE**: Output Select

0 = High impedance

1 = Output

Bit2 **RE**: GPIO read enable bit.

0 = Read Disabled

1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.

0 = Pull-Up Active

1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.

0 = Pull Down inactive

1 = Pull Down active

PD3CFG: PD3 GPIO configuration register.

PD3CFG		0x50018023			0x00		
Reserved	R/W	Reserved	R/W	R/W	R/W	R/W	R/W
-	SelPWM_ PD3	-	-	OE_PD3	RE_PD3	PU_n_PD3	PD_PD3
MSB							LSB

PD3CFG: PD3 GPIO configuration register.

Bit6 **SelPWM**: Data MUX

0 = Data output selected

1 = PWM output selected

Bit3 **OE**: Output Select

0 = High impedance

1 = Output

Bit2 **RE**: GPIO read enable bit.

0 = Read Disabled

1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.

0 = Pull-Up Active

1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.

0 = Pull Down inactive

1 = Pull Down active

PD4CFG: PD4 GPIO configuration register.

PD4CFG		0x50018024			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	SelPWM_ PD4	-	-	OE_PD4	RE_PD4	PU_n_PD4	PD_PD4
MSB							LSB

PD4CFG: PD4 GPIO configuration register.

Bit6 **SelPWM**: Data MUX

0 = Data output selected

1 = PWM output selected

Bit3 **OE**: Output Select

0 = High impedance

1 = Output

Bit2 **RE**: GPIO read enable bit.

0 = Read Disabled

1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.

0 = Pull-Up Active

1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.

0 = Pull Down inactive

1 = Pull Down active

PD5CFG: PD5 GPIO configuration register.

PD5CFG		0x50018025			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	SelPWM_ PD5	-	-	OE_PD5	RE_PD5	PU_n_PD5	PD_PD5
MSB							LSB

PD5CFG: PD5 GPIO configuration register.

Bit6 **SelPWM**: Data MUX
0 = Data output selected
1 = PWM output selected

Bit3 **OE**: Output Select
0 = High impedance
1 = Output

Bit2 **RE**: GPIO read enable bit.
0 = Read Disabled
1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.
0 = Pull-Up Active
1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.
0 = Pull Down inactive
1 = Pull Down active

PD6CFG: PD6 GPIO configuration register.

PD6CFG		0x50018026			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	-	OE_PD6	RE_PD6	PU_n_PD6	PD_PD6
MSB							LSB

PD6CFG: PD6 GPIO configuration register.

Bit3 **OE**: Output Select

0 = High impedance

1 = Output

Bit2 **RE**: GPIO read enable bit.

0 = Read Disabled

1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.

0 = Pull-Up Active

1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.

0 = Pull Down inactive

1 = Pull Down active

PD7CFG: PD7 GPIO configuration register							
PD7CFG		0x50018027			0x00		
R/W	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
SelZX	-	-	-	OE_PD7	RE_PD7	PU_n_PD7	PD_PD7
MSB							LSB

PD7CFG: PD7 GPIO configuration register

Bit3 **SelZX**: Select Zero Xing Signal

0 = From Port register

1 = Zero Crossing Detect output

Bit3 **OE**: Output Select

0 = High impedance

1 = Output

Bit2 **RE**: GPIO read enable bit.

0 = Read Disabled

1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.

0 = Pull-Up Active

1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.

0 = Pull Down inactive

1 = Pull Down active

5.10.8 **GPIO PE Registers**

GIO PE configuration registers				
Address	Register Name	Description	Reset Value	Comment
0x50018028	PE0CFG	PE0 GIO configuration register		
0x50018029	PE1CFG	PE1 GIO configuration register		
0x5001802A	PE2CFG	PE2 GIO configuration register		
0x5001802B	PE3CFG	PE3 GIO configuration register		
0x5001802C	PE4CFG	PE4 GIO configuration register		
0x5001802D	PE5CFG	PE5 GIO configuration register		
0x5001802E	PE6CFG	PE6 GIO configuration register		
0x5001802F	PE7CFG	PE7 GIO configuration register		

PE0CFG: PE0 GPIO configuration register.							
PE0CFG		0x50018028			0x00		
Reserved	Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W
-	-	-	-	OE_PE0	RE_PE0	PU_n_PE0	PD_PE0
MSB							LSB

PE0CFG: PE0 GPIO configuration register.

Bit3 **OE**: Output Select

0 = High impedance

1 = Output

Bit2 **RE**: GPIO read enable bit.

0 = Read Disabled

1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.

0 = Pull-Up Active

1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.

0 = Pull Down inactive

1 = Pull Down active

PE1CFG: PE1 GPIO configuration register.

PD1CFG		0x50018029			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	-	OE_PE1	RE_PE1	PU_n_PE1	PD_PE1
MSB							LSB

PE1CFG: PE1 GPIO configuration register.

Bit3 **OE**: Output Select

0 = High impedance

1 = Output

Bit2 **RE**: GPIO read enable bit.

0 = Read Disabled

1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.

0 = Pull-Up Active

1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.

0 = Pull Down inactive

1 = Pull Down active

PE2CFG: PE2 GPIO configuration register.

PE2CFG		0x5001802A			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	-	OE_PE2	RE_PE2	PU_n_PE2	PD_PE2
MSB							LSB

PE2CFG: PE2 GPIO configuration register.

Bit3 **OE**: Output Select

0 = High impedance

1 = Output

Bit2 **RE**: GPIO read enable bit.

0 = Read Disabled

1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.

0 = Pull-Up Active

1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.

0 = Pull Down inactive

1 = Pull Down active

PE3CFG: PE3 GPIO configuration register.

PE3CFG		0x5001802B			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	-	OE_PE3	RE_PE3	PU_n_PE3	PD_PE3
MSB							LSB

PE3CFG: PE3 GPIO configuration register.

Bit3 **OE**: Output Select

0 = High impedance

1 = Output

Bit2 **RE**: GPIO read enable bit.

0 = Read Disabled

1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.

0 = Pull-Up Active

1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.

0 = Pull Down inactive

1 = Pull Down active

PE4CFG: PE4 GPIO configuration register.

PE4CFG		0x5001802C			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	-	OE_PE4	RE_PE4	PU_n_PE4	PD_PE4
MSB							LSB

PE4CFG: PE4 GPIO configuration register.

Bit3 **OE**: Output Select

0 = High impedance

1 = Output

Bit2 **RE**: GPIO read enable bit.

0 = Read Disabled

1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.

0 = Pull-Up Active

1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.

0 = Pull Down inactive

1 = Pull Down active

PE5CFG: PE5 GPIO configuration register.

PE5CFG		0x5001802D			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	-	OE_PE5	RE_PE5	PU_n_PE5	PD_PE5
MSB							LSB

PE5CFG: PE5 GPIO configuration register.

Bit3 **OE**: Output Select

0 = High impedance

1 = Output

Bit2 **RE**: GPIO read enable bit.

0 = Read Disabled

1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.

0 = Pull-Up Active

1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.

0 = Pull Down inactive

1 = Pull Down active

PE6CFG: PE6 GPIO configuration register.

PE6CFG		0x5001802E			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	-	OE_PE6	RE_PE6	PU_n_PE6	PD_PE6
MSB							LSB

PE6CFG: PE6 GPIO configuration register.

Bit3 **OE**: Output Select

0 = High impedance

1 = Output

Bit2 **RE**: GPIO read enable bit.

0 = Read Disabled

1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.

0 = Pull-Up Active

1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.

0 = Pull Down inactive

1 = Pull Down active

PE7CFG: PE7 GPIO configuration register.

PE7CFG		0x5001802F			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	-	OE_PE7	RE_PE7	PU_n_PE7	PD_PE7
MSB							LSB

PE7CFG: PE7 GPIO configuration register.

Bit3 **OE**: Output Select

0 = High impedance

1 = Output

Bit2 **RE**: GPIO read enable bit.

0 = Read Disabled

1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.

0 = Pull-Up Active

1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.

0 = Pull Down inactive

1 = Pull Down active

5.10.9 GPIO PF Registers

GIO PE configuration registers				
Address	Register Name	Description	Reset Value	Comment
0x50018030	PF0CFG	PF0 GIO configuration register		
0x50018031	PF1CFG	PF1 GIO configuration register		
0x50018032	PF2CFG	PF2 GIO configuration register		

PF0CFG: PF0 GPIO configuration register.							
PF0CFG		0x50018030			0x00		
Reserved	Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W
-	-	-	-	OE_PF0	RE_PF0	PU_n_PF0	PD_PF0
MSB							LSB

Bit3 **OE**: Output Select

0 = High impedance

1 = Output

Bit2 **RE**: GPIO read enable bit.

0 = Read Disabled

1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.

0 = Pull-Up Active

1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.

0 = Pull Down inactive

1 = Pull Down active

PF1CFG: PF1 GPIO configuration register.							
PF1CFG		0x50018031			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	-	OE_PF1	RE_PF1	PU_n_PF1	PD_PF1
MSB							LSB

Bit3 **OE**: Output Select

0 = High impedance

1 = Output

Bit2 **RE**: GPIO read enable bit.

0 = Read Disabled

1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.

0 = Pull-Up Active

1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.

0 = Pull Down inactive

1 = Pull Down active

PF2CFG: PF2 GPIO configuration register.							
PF2CFG		0x50018032			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	-	OE_PF2	RE_PF2	PU_n_PF2	PD_PE2
MSB							LSB

PF2CFG: PF2 GPIO configuration register.

Bit3 **OE**: Output Select

0 = High impedance

1 = Output

Bit2 **RE**: GPIO read enable bit.

0 = Read Disabled

1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.

0 = Pull-Up Active

1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.

0 = Pull Down inactive

1 = Pull Down active

5.11 Clock sources

Krankl provides three clock sources:

- Internal auxiliary oscillator running at 10kHz. (This oscillator is always running, even when the device is in sleep mode, but its power consumption is negligible)
- Internal RC oscillator running at 12MHz.
- Crystal oscillator running at 30MHz

Krankl starts from power-on reset using the internal 12MHz RC oscillator. From this point on the user may select the crystal or the auxiliary oscillators.
The crystal oscillator is required for RF reception. The 10kHz oscillator may be used in power saving modes.

5.11.1 Clock Source Operation and Description

Upon Reset or Power-On Reset the system starts using the internal RC 12MHz oscillator.
Depending on the application requirements the designer can:

- Enable or disable the internal RC oscillator
- Enable or disable the external crystal
- Select the system clock source: RC or Crystal

Example Code: Enable the Crystal oscillator.

```
uint8_t XTCLK_Init ( void )
{
  int i,j;
  // Initialize the Crystal Clock to run the system, divided by one (default)
  j=0;
  while ((*PMUCLK & (XTCLOCK|RC12MHZ)) != XTCLOCK )  //Wait for XT monitor ON
  {
      CLK_CrystalControl( XTON );          // Enable crystal oscillator
      for ( i = 0; i < 20000; i++);
      CLK_SelectClockSource( XTCLOCK );    // Selects crystal as clock source
      j++;
        if (j >= 10) return (FALSE);
  }
  return (TRUE);
}
```

5.11.2 Clock Related Registers

The following registers are used to control the behavior of the clock sources.

Clocks configuration registers				
Address	Register Name	Description	Reset Value	Comment
0x50000000	PMUCLK	Processor Control Register		
0x50000005	RTOUT	Real Time Output Register		

PMUCLK: Processor Control Register.							
PMUCLK		0x50000000			0x15		
R/W	R/W	R	R	R/W	R/W	R/W	R/W
CKD1	CKD0	XOMON	RCMON	XO_CK_ENB	RC_CK_ENB	CKSEL1	CKSEL0
MSB							LSB

PMUCLK: Processor Control Register.

Bit7-6 **CKD[1:0]**: Clock Frequency Divider

00 = Clock Divided by 1

01 = Clock Divided by 2

10 = Clock Divided by 4

11 = Clock Divided by 8

Bit5 **XOMON**: Crystal Oscillator Monitor

0 = Crystal Oscillator Inactive

1 = Crystal Oscillator Active

Bit4 **RCMON**: RC Oscillator Monitor

0 = RC Oscillator Inactive

1 = RC Oscillator Active

Bit3 **XO_CK_ENB**: Crystal Oscillator Control

0 = Crystal Oscillator Disable

1 = Crystal Oscillator Enable

Bit2 **RC_CK_ENB**: RC Oscillator Control

0 = RC Oscillator Disable

1 = RC Oscillator Enable

Bit1-0 **CKSEL[1:0]**: Clock Select

00 = 10 KHz Auxiliary Clock

01 = 12MHz RC Oscillator Clock*

10 = Crystal Oscillator Clock

11 = not used

*Note: This is the clock selected after Power-On-Reset and for clock fault condition.

RTOUT: Real Time Output Register

RTOUT		0x50000005			0x00		
R/W	Reserved	R/W	R/W	R/W	R/W	R/W	R/W
SER_FAST	-	RTSELA2	RTSELA1	RTSELA0	RTSELB2	RTSELB1	RTSELB0

RTOUT: Real Time Output Register							
MSB							LSB

Bit2-0 **RTSELB[2:0]**: Real Time Output at PD1

001 : clk_full (system clock)

010 : clk_div

011 : clk_ser

100 : rto_adc

101 : rto_tx

Bit5-3 **RTSELA[2:0]**: Real Time Output at PD0

001 : clk_AUX (10 kHz clock)

010 : clk_RC (10 MHz clock)

011 : clk_XO (30 MHz clock)

110 : bor_n_mask

111 : rto_rx

5.12 Power Management Unit

Krankl implements a power management unit. Its main characteristics are:

- HW reset - Affects all aspects of Krankl
- SW reset - Does not affect clock nor brownout setup
- Selectable Sleep mode and Halt Mode
- Programmable brownout detector

The PMU module allows for the control of reset, deep sleep (halt), sleep, and brownout.

5.12.1 **PMU control of Reset**

There are two forms of reset that can be issued:

- Hardware reset: In this reset all peripherals are reset, the 10 KHz clock is selected and all other clock sources are disabled, but the brownout selection is kept.
- Software reset: In this reset all peripherals are reset but the clock setup is kept unchanged along with the brownout selection.

Code Examples: HW reset and SW reset:

```
PMU_HwReset();
```

```
PMU_SwReset();
```

5.12.2 PMU control of sleep and deep sleep (halt) modes

The PMU can set the system into sleep or deep sleep (halt) modes.
In the **deep sleep** mode, the following will happen:

1. the CPU is halted
2. Any enabled clock source will continue to operate
3. The three timers (Timer0, Timer1 and Timer2) and the SysTick Timer will stop operating
4. All other peripherals will keep running (if enabled and fed by their required source clock)

The system will leave the deep sleep (halt) mode only through a reset or POR (**P**ower-**O**n **R**eset). The sources of a reset can be the wakeup timer or any peripheral that generates an interrupt independently of the interrupt being enabled by the NVIC module (**N**ested **V**ector **I**nterrupt **C**ontroller).

Note: For those peripherals that have in their registers a bit that locally enables the interrupt, this register has to be enabled in order to reset the system.
Example: *For the GPIOs ports PORTA, PORTB and PORTC the INTE bits of the IO pins selected to reset the part upon change must be set.*
Example: Enabling the reset by enabling an interrupt from PORTB[4] upon change. (Port B, bit4)

```
//PortB.4 interrupt enabled, pull up enabled
Port_Config(PTB, 0, 0x10, 0x00, 0x00, 0x10, 0x00, 0x00);
PMU_Deep_Sleep(); //System in deep sleep (halt)
```

In the **sleep** mode,

1. the CPU is halted
2. Any enabled clock source will continue to operate
3. All timers (Timer0, Timer1 and Timer2) and the SysTick Timer will continue to operate
4. All other peripherals will keep running (if enabled and fed by their required source clock)

Besides a POR and/or reset, the system will leave the sleep mode also through an interrupt; the sources of an interrupt can be any peripheral generating an interrupt.
Note: The interrupt must be enabled by the NVIC module (**N**ested **V**ector **I**nterrupt **C**ontroller) and for those peripherals that have in their registers a bit that locally enables the interrupt this register also has to be enabled in order to generate an interrupt and wakeup the system from sleep.
Example: *For the GPIOs ports PORTA, PORTB and PORTC the INTE bits of the IO pins selected to reset the part upon change must be set.*

5.12.3 PMU control of Brownout

The PMU controls the brownout by

- Enabling or disabling the brownout circuit
- Selecting the behavior when a brownout is detected:
  - Generate an interrupt
  - Reset the system
- Selecting the brownout voltage level

Code Example: Enable the BOR, with interrupt but no reset and with 2.4V level.

```
BOR_ResetControl(BORRSTDIS);  //Brownout reset disabled
BOR_IntControl(BORINTEN);     //Brownout interrupt enabled
BOR_Level(BOR24V);            //Brownout level = 2.4V
BOR_Control(BOREN);           //Brownout enabled
```

5.12.4 PMU Registers

Krankl implements the following PMU registers:

PMU configuration registers				
Address	Register Name	Description	Reset Value	Comment
0x50000001	PMURST	Processor Control Register		
0x50000002	PMUBOR	BOR Control		

PMURST: Processor control register.							
PMURST		0x50000001			0x01		
W	W	R/W	Reserved	Reserved	Reserved	R	R/W
HWRST	SWRST	DLEEP	-	-	-	BROUT	PORF
MSB							LSB

PMURST: Processor control register.

Bit7 **HWRST**: Hardware reset

0 = Idle

1 = Hardware reset (automatically cleared after reset process completed)

Bit6 **SWRST**: Software reset

0 = Idle

1 = Software reset (automatically cleared after reset process completed)

Bit5 **DLEEP**: Deep sleep (HALT) mode

Writing:

- 0 = Clear deep sleep flag
- 1 = Put the system in deep sleep - Halt

Reading:

- 0 = Flag cleared
- 1 = system in deep sleep mode

Bit1 **BROUT**: Brownout indicator

0 = No brownout

1 = Brownout

Bit0 **PORF**: Power-On Reset flag

Writing:

- 0 = Clear POR
- 1 = No effect

Reading:

- 0 = POR flag already cleared by application
- 1 = The system just came out of POR or HW Reset

PMUBOR: BOR Control							
PMUBOR		0x50000002			0x00		
R/W	Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W
BORENB	-	-	-	BORRSTB	BORINT	BOUTVALUE1	BOUTVALUE0
MSB							LSB

Bit7 **BORENB**: Brownout enable (active low)

0 = Brownout enabled

1 = Brownout disabled

Bit3 **BORRSTB**: Brownout Reset enable (active low)

0 = Enable Brownout based reset

1 = Disable Brownout based reset

Bit2 **BORINT**: Brownout interrupt

1 = Brownout interrupt enabled

0 = Brownout interrupt disabled

Bit1-0 **BOUTVALUE [1:0]**: Brownout threshold value

00 = 2.0V

01 = 2.2V

10 = 2.4V

11 = 2.6V

5.13 Wake Up Timer

In addition to the Timer0/1/2, Krankl implements a timer capable of waking-up the microcontroller from a sleep state. The wake up timer is a timer used to allow for recovery from deep sleep, including when the microcontroller is disconnected from its power supply.

5.13.1 Wake-Up Timer Operation and Description

For instance, a value of 0x54 would give a time of: (Assuming the application is running from the 10 kHz internal oscillator)

WakeupPeriod=5*2^5/10kHz=16msec

```
WKP_Timing(5,4)         //Select the mantissa=5 and exponent=4
PMU_Deep_Sleep(SLEEPON);      //Put the part in deep sleep mode, it will
//Reset in 16msec.
```

5.13.2 Wake-Up Timer Register

The following register controls the wake-up timer:

Wake up configuration registers				
Address	Register Name	Description	Reset Value	Comment
0x50000004	WKPTIME	Wakeup timer control		

<table>
<tr><th colspan="8">WKPTIME: Wakeup timer control</th></tr>
<tr><td colspan="2"><b>WKPTIME</b></td><td colspan="3">0x50000004</td><td colspan="3">0x00</td></tr>
<tr><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>MANT3</td><td>MANT2</td><td>MANT1</td><td>MANT0</td><td>EXP3</td><td>EXP2</td><td>EXP1</td><td>EXP0</td></tr>
<tr><td>MSB</td><td></td><td></td><td></td><td></td><td></td><td></td><td>LSB</td></tr>
<tr><td colspan="8">Bit7-4 <b>MANT [3:0]</b>: Mantissa of the wakeup timer
Bit3-0 <b>EXP [3:0]</b>: Exponent of the wakeup timer (range: 0...12)
WakeupPeriod = Mantissa*$2^{(Exponent+1)}$/SystemClock</td></tr>
</table>

6 Electrical Characteristics

Recommended operating conditions unless otherwise specified.

Table 11 EC Supply and Temperature

Electrical Characteristics Over junction temperature range 0°C to 125°C and recommended supply voltage (unless otherwise noted)					
Parameter	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
High supply voltage		9	24	32	V
Operating Ambient Temperature		-20	25	65	°C
V3P3DIG, V3P3DIG_IO Supply	V3P3 Digital and IOs		3.3		V
V3P3DIG LDO maximum load				50	mA
V3P3ANA Supply	V3P3 analog		3.3		V
VP3ANA LDO maximum load				50	mA
V1P8DIG Supply	MCU supply		1.8		V

Table 12 EC RF Receiver

RF Receiver Electrical specification					
Parameter	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
LNA input impedance			10-129j		Ω
Sensitivity			-110		dBm
Frequency Range		300		470	MHz
Data Rate				5	kbps
Maximum input signal				-10	dBm
Total Gain	Voltage gain from RF input to I or Q IF outputs at default gain programming	59	71	74	dB

RF Receiver Electrical specification					
Parameter	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
Spurious Emission				-60	dBm

Table 13 EC RF TX

RF Transmitter Electrical Specification					
Parameter	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
Output Power Range	V3p3ANA=3.3V, T_A=27°C	-13.5		+15.1	dBm
Output Power Step	Valid from +11.5 to -9.5dBm output power		1.5		dB
Operating Frequency	Nominal		433.92		MHz
Current Consumption	Pout=max (+13dBm)		18		mA
Sleep Current Consumption	All blocks disabled			100	nA
Data Rate	Output Power below 0dBm for ASK/OOK			50	kbps
Data Rate	Output Power higher than 0dBm for ASK/OOK			25	kbps
Output Noise	fc = 434.07 MHz, frequency offset=1.6 MHz compliant with EN 300 220-1 (2000.09)			-36	dBm
Harmonics, Conducted 2nd Harmonic 433 MHz 3rd Harmonic 433 MHz			-45.1 -49.4		dBm
Frequency stability	Depends on crystal employed			13	ppm

Table 14 EC HK ADC

ADC Performance Specification					
Parameter	**conditions**	**Min**	**Typ**	**Max**	**Unit**
Conversion speed Sampling rate	Signal source resistance less than 20kΩ			80	ksps
Input Clock Frequency	With 12MHz system clock			1.5	MHz
Input voltage range		0		V3P3ANA	V
Resolution			8		bits
INL[1]				1	LSB
DNL[1]				1	LSB
Internal sampling capacitance[1]				10	pF

[1] Guaranteed by design

Table 15 EC GIO/SIO/GPIOs

General Purpose IOs – Electrical Characteristics						
IO Type	**Name**	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
GIO	V_{IL}	Threshold Low (3V3 systems)		1.65		V
		Threshold High (12V systems)		4		V
	V_{IH}	Threshold Low (3V3 systems)		1.65		V
		Threshold High (12V systems)		4		V
	I_{OL}	Sink current		25		mA
	I_{OH}	Source current (with limiter)		5		mA
	Pull-Down	Strength Low		100		mA
		Strength High		5		mA
	Pull-Up	Strength Low		100		mA
		Strength High		5		mA
SIO	V_{IL}	Threshold Low (3V3 systems)		1.65		V

General Purpose IOs – Electrical Characteristics

IO Type	Name	Conditions	Min	Typ	Max	Unit
		Threshold High (12V systems)		4		V
	V_{IH}	Threshold Low (3V3 systems)		1.65		V
		Threshold High (12V systems)		4		V
	I_{OL}	Sink current		200		mA
	I_{OH}	Source current (with limiter)		5		mA
	Pull-Down	Strength Low		100		mA
		Strength High		5		mA
	Pull-Up	Strength Low		100		mA
		Strength High		5		mA
3V3IO	V_{IL}	Threshold Low		1.65		V
		Threshold High		4		V
	V_{IH}	Threshold Low		1.65		V
		Threshold High		1.65		V
	I_{OL}	Sink current		10		mA
	I_{OH}	Source current		10		mA

Table 16 EC Clocks

Clock Electrical Specification

Parameter	conditions	Min	Typ	Max	Unit
Crystal Oscillator frequency			30		MHz
Frequency stability	Using defined crystal			TBD	ppm
Auxiliary Oscillator	(Calibrated Frequency)		10		kHz

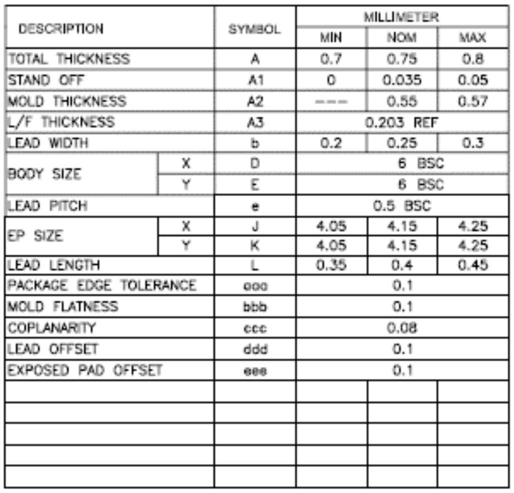
Clock Electrical Specification					
Parameter	**conditions**	**Min**	**Typ**	**Max**	**Unit**
Auxiliary Oscillator accuracy	Post-calibration to 10KHz, T_A=27°C			5	%
RC Oscillator frequency			12		MHz
RC Oscillator accuracy	Post-calibration to 12MHz, T_A=27°C			5	%

7 Bill of Material

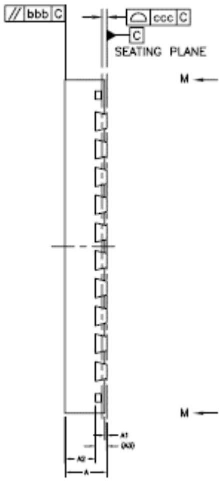
Table 17 BOM

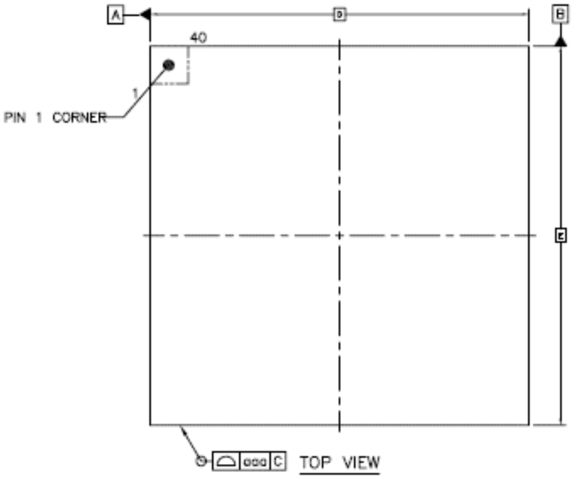
BOM				
Name	**Description**	**Value**	**Rating**	Comment

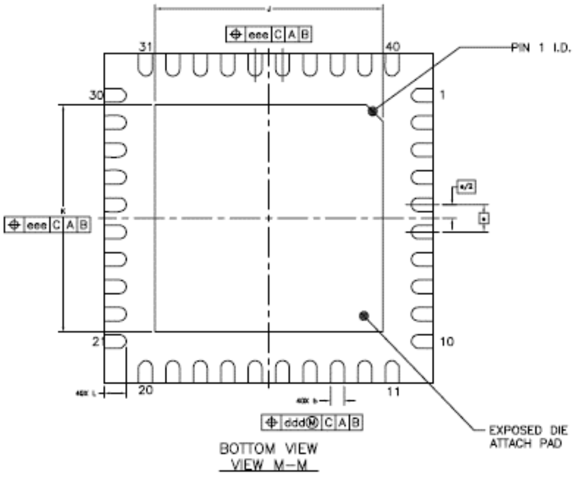
8 Package Outline



DESCRIPTION		SYMBOL	MILLIMETER		
			MIN	NOM	MAX
TOTAL THICKNESS		A	0.7	0.75	0.8
STAND OFF		A1	0	0.035	0.05
MOLD THICKNESS		A2	---	0.55	0.57
L/F THICKNESS		A3	0.203 REF		
LEAD WIDTH		b	0.2	0.25	0.3
BODY SIZE	X	D	6 BSC		
	Y	E	6 BSC		
LEAD PITCH		e	0.5 BSC		
EP SIZE	X	J	4.05	4.15	4.25
	Y	K	4.05	4.15	4.25
LEAD LENGTH		L	0.35	0.4	0.45
PACKAGE EDGE TOLERANCE		aaa	0.1		
MOLD FLATNESS		bbb	0.1		
COPLANARITY		ccc	0.08		
LEAD OFFSET		ddd	0.1		
EXPOSED PAD OFFSET		eee	0.1		







9 References

- ISM Band references:
- http://en.wikipedia.org/wiki/ISM_band

10 Contacts

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

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

Scotland
5th Floor, The Auction House,
63a George Street
Edinburgh EH2 2JG
Tel: +44 131 718 6378

http://www.indiesemi.com/

Important Notice

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

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

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

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

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