True System-on-Chip with Low Power RF Transceiver and 8051 MCU

Applications

- *315/433/868 and 915 MHz ISM/SRD band systems*
- *Low power telemetry*
- *Home and building automation*
- *AMR – Automatic Meter Reading*
- *Wireless alarm and security systems*
- *RKE – Two-way Remote Keyless Entry*
- *Consumer Electronics*
- *Wireless sensor networks*
- *Industrial monitoring and control*

Product Description

The ***CC1110*** is a low-cost true System-on-Chip (SoC) device designed for low-power and low-voltage wireless communication applications.

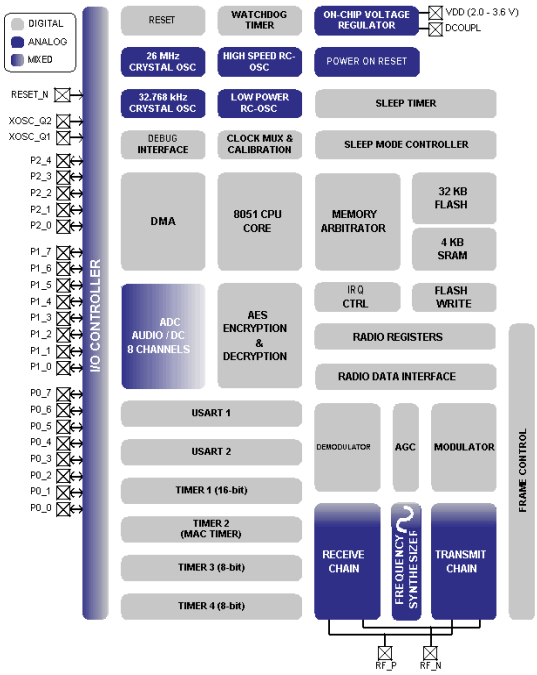
The ***CC1110*** combines the excellent performance of the state-of-the-art CC1100 RF transceiver with an industry-standard enhanced 8051 MCU, 32 KB of in-system programmable flash memory, 4 KB of RAM and many other powerful features.

The ***CC1110*** is highly suited for systems where very low power consumption is required. This is ensured by several advanced low-power operating modes.

The ***CC1110*** provides a tight integration between the MCU core, software and the RF transceiver, making this System-on-Chip solution easy to use as well as improving performance.

CC1110 is supported by a powerful integrated development environment (IDE). In-circuit interactive debugging is supported for the industry standard IAR C-SPY IDE via a simple two-wire serial interface.

CC1110 is based on Chipcon's SmartRF04® technology platform in 0.18 µm CMOS.



The data sheet contains preliminary data, and supplementary data will be published at a later date. Chipcon reserves the right to make changes at any time without notice in order to improve design and supply the best possible product. The product at this point is not fully qualified.

Key Features

- High performance and low power 8051 microcontroller core, typically with 8x the performance per MHz of a standard 8051
- High-performance RF transceiver core – same as in the CC1100
- Frequency bands: 300-348 MHz, 400-464 MHz and 800-928 MHz
- Programmable data rate up to 500 kbps
- 32 KB in-system programmable flash
- 4 KB RAM
- 8-14 bits ADC with up to eight inputs
- 128-bit AES security coprocessor
- Powerful DMA functionality
- Two USARTs / SPI interfaces
- 16-bit timer + three 8-bit timers
- Hardware debug support
- 21 general I/O pins
- Wide supply voltage range (2.0V – 3.6V)
- High sensitivity (–110 dBm at 1.2 kbps)
- Programmable data rate up to 500 kbps

Key Features (continued)

- Low current consumption (RX: 22 mA, TX: 31 mA, with the microcontroller running at 26 MHz)
- Programmable output power up to 10 dBm for all supported frequencies
- Digital RSSI / LQI support
- Excellent receiver selectivity and blocking performance
- 0.6 µA consumption in lowest power mode
- RoHS compliant 6x6mm QLP 36 package

Table of Contents

1 Abbreviations

2-FSK	Binary Frequency Shift Keying
AC	Alternating Current
ADC	Analog to Digital Converter
AES	Advanced Encryption Standard
AFC	Automatic Frequency Offset Compensation
AGC	Automatic Gain Control
AMR	Automatic Meter Reading
ARIB	Association of Radio Industries and Businesses
ASK	Amplitude Shift Keying
Balun	Balanced to unbalanced
BCD	Binary Coded Decimal
BER	Bit Error Rate
CBC	Cipher Block Chaining
CBC-MAC	Cipher Block Chaining Message Authentication Code
CCA	Clear Channel Assessment
CCM	Counter mode + CBC-MAC
CFB	Cipher Feedback
CFR	Code of Federal Regulations
CLK	Clock
CMOS	Complementary Metal Oxide Semiconductor
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CS	Carrier Sense
CTR	Counter mode (encryption)
CW	Continuous Wave
DAC	Digital to Analogue Converter
DC	Direct Current
DMA	Direct Memory Access
DNL	Differential Non-Linearity
DSM	Delta Sigma Modulator
ECB	Electronic Code Book
EM	Evaluation Module
ENOB	Effective Number Of Bits
ESD	Electro Static Discharge
ESR	Equivalent Series Resistance
ETSI	European Telecommunications Standards Institute
FCC	Federal Communications Commission
FEC	Forward Error Correction
FIFO	First In First Out
GFSK	Gaussian Frequency Shift Keying
HSSD	High Speed Serial Debug
I/O	Input / Output
I/Q	In-phase / Quadrature-phase
IDE	Integrated Development Environment
IF	Intermediate Frequency
INL	Integral Non-Linearity
IOC	I/O Controller
IRQ	Interrupt Request
ISM	Industrial, Scientific and Medical
ITU-T	International Telecommunication Union – Telecommunication Standardization Sector
IV	Initialization Vector
JEDEC	Joint Electron Devices Engineering Council
KB	1024 bytes
kbps	kilo bits per second
LC	Inductor Capacitor
LFSR	Linear Feedback Shift Register
LNA	Low-Noise Amplifier
LO	Local Oscillator
LQI	Link Quality Indication
LSB	Least Significant Bit / Byte
MAC	Message Authentication Code
MAC	Medium Access Control
MCU	Micro Controller Unit
MISO	Master In Slave Out
MODEM	Modulator Demodulator
MOSI	Master Out Slave In
MSB	Most Significant Byte
MSK	Minimum Shift Keying
MUX	Multiplexer
NA	Not Available
NC	Not Connected
OFB	Output Feedback (encryption)

OOK	On-Off Keying
PA	Power Amplifier
PCB	Printed Circuit Board
PD	Power-Down
PER	Packet Error Rate
PLL	Phase Locked Loop
PM{0-3}	Power Mode 0-3
PMC	Power Management Controller
POR	Power On Reset
PPM	Parts Per Million
PQT	Preamble Quality Indicator
PWM	Pulse Width Modulator
QLP	Quad Leadless Package
QPSK	Quadrature Phase Shift Keying
R/W	Read/Write
RAM	Random Access Memory
RBW	Resolution Bandwidth
RC	Resistor Capacitor
RCOSC	RC Oscillator
RF	Radio Frequency
RKE	Remote Keyless Entry
RoHS	Restriction on Hazardous Substances
RSSI	Receive Signal Strength Indicator
RTC	Real-Time Clock
RX	Receive
SCK	Serial Clock
SFD	Start of Frame Delimiter
SFR	Special Function Register
SINAD	Signal-to-noise and distortion ratio
SoC	System-on-Chip
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
SRD	Short Range Device
ST	Sleep Timer
T/R	Transmit / Receive
TBD	To Be Decided / To Be Defined
TX	Transmit
UART	Universal Asynchronous Receiver/Transmitter
UHF	Ultra High Frequency
USART	Universal Synchronous/Asynchronous Receiver/Transmitter
VCO	Voltage Controlled Oscillator
VGA	Variable Gain Amplifier
WDT	Watchdog Timer
XOSC	Crystal Oscillator
XTAL	Crystal

2 References

[1] NIST FIPS Pub 197: Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, US Department of Commerce/N.I.S.T., November 26, 2001. Available from the NIST website.

http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

3 Register conventions

Each SFR register is described in a separate table. The table heading is given in the following format:

REGISTER NAME (SFR Address) - Register Description.

Each RF register is described in a separate table. The table heading is given in the following format:

XDATA Address: REGISTER NAME - Register Description

All register descriptions include for each register bit a symbol denoted R/W describing the accessibility of the bit. The register values are always given in binary notation unless prefixed by '0x' which indicates hexadecimal notation.

Symbol	Access Mode
R/W	Read/write
R	Read only
R0	Read as 0
R1	Read as 1
W	Write only
W0	Write as 0
W1	Write as 1
H0	Hardware clear
H1	Hardware set

Table 1: Register bit conventions

4 Features (continued from front page)

4.1 High-Performance and Low-Power 8051-Compatible Microcontroller

- Optimized 8051 core which typically gives 8x the performance of a standard 8051
- Dual data pointers
- In-circuit interactive debugging is supported for the IAR Embedded Workbench through a simple two-wire serial interface

4.2 32 KB Non-volatile Program Memory and 4 KB Data Memory

- 32 KB of non-volatile Flash memory, in-system programmable through a simple two-wire interface or by the 8051 core
- Worst case flash memory endurance: 1000 write/erase cycles (applies per bit cell).
- Programmable read and write lock of portions of Flash memory for software security
- 4096 bytes of internal SRAM

4.3 Hardware AES Encryption/Decryption

- 128-bit AES supported in hardware coprocessor

4.4 Peripheral Features

- Powerful DMA Controller
- Power On Reset
- Battery monitor
- Eight channel, 8-14 bit ADC
- Programmable watchdog timer
- Real time clock with 32.768 kHz crystal oscillator
- Four timers: one general 16-bit timer, two general 8-bit timers, one MAC timer
- Two programmable USARTs for master/slave SPI or UART operation
- 21 configurable general-purpose digital I/O-pins
- True random number generator

4.5 Low Power

- Four flexible power modes for reduced power consumption
- Only 0.8 µA current consumption in standby mode, where external interrupts or the real-time counter can wake up the system
- 0.6 µA current consumption in power down mode, where external interrupts can wake up the system
- System can wake up on external interrupt or real-time counter event
- Low-power fully static CMOS design
- System clock source can be 13 MHz RC oscillator or 26 MHz crystal oscillator. The 26 MHz oscillator is used when the radio is active.
- Optional clock source for ultra-low power operation can be either a low-power RC oscillator or an optional 32.768 kHz crystal oscillator
- Very fast transition from sleep modes to active enable ultra low average power consumption in low duty-cycle systems

4.6 UHF Radio with baseband modem

- Based on the industry leading CC1100 radio core
- Suited for systems compliant with EN 300 220 (Europe) and FCC CFR Part 15 (US)
- Very few external components: Fully on-chip frequency synthesizer, no external filters or RF switch needed
- Flexible support for packet oriented systems: On chip support for sync word detection, address check, flexible packet length and automatic CRC handling.
- Programmable channel filter bandwidth
- OOK and flexible ASK shaping supported
- 2-FSK, GFSK and MSK supported

- Optional automatic whitening and de-whitening of data
- Support for asynchronous transparent receive/transmit mode for backwards compatibility with existing radio communication protocols
- Programmable Carrier Sense indicator
- Programmable Preamble Quality Indicator for detecting preambles and improved protection against sync word detection in random noise
- Support for automatic Clear Channel Assessment (CCA) before transmitting (for listen-before-talk systems)
- Support for per-package Link Quality Indication

5 Absolute Maximum Ratings

Under no circumstances must the absolute maximum ratings given in Table 2 be violated. Stress exceeding one or more of the limiting values may cause permanent damage to the device.

Parameter	Min	Max	Units	Condition
Supply voltage	–0.3	3.6	V	All supply pins must have the same voltage
Voltage on any digital pin	–0.3	VDD+0.3, max 3.6	V	
Input RF level		+10	dBm	
Storage temperature range	–50	150	°C	Device not programmed
Solder reflow temperature		260	°C	T = 10 s

Table 2: Absolute Maximum Ratings

Caution! *ESD sensitive device. Precaution should be used when handling the device in order to prevent permanent damage.*

6 Operating Conditions

The operating conditions for ***CC1110*** are listed Table 3 in below.

Parameter	Min	Max	Unit	Condition
Operating ambient temperature, T_A	-40	85	°C	
Operating supply voltage	2.0	3.6	V	

Table 3: Operating Conditions

7 Electrical Specifications

T_A=25°C, VDD=3.0V if nothing else stated. Measured on Chipcon's CC1110 EM reference design.

Parameter	Min	Typ	Max	Unit	Condition
Power On Reset Voltage		1.1		V	Monitors the unregulated supply
Current Consumption					
MCU Active Mode, static		500		μA	Digital regulator on, High Speed RCOSC running. No radio, crystals, or peripherals.
MCU Active Mode, dynamic		270		μA/MHz	Digital regulator on, High Speed RCOSC running. No radio, crystals, or peripherals.
MCU Active Mode, highest speed		7.5		mA	MCU running at full speed (26 MHz), XOSC running. No peripherals.
MCU Active and RX Mode		22		mA	MCU running at full speed (26 MHz), XOSC running, radio in RX mode. No peripherals.
MCU Active and TX Mode, 0dBm		31		mA	MCU running at full speed (26 MHz), XOSC running, radio in TX mode. No peripherals.
Power mode 1		300		μA	Digital regulator on, High Speed RCOSC and crystal oscillator off. 32.768kHz XOSC, POR and ST active. RAM retention.
Power mode 2		0.8		μA	Digital regulator off, High Speed RCOSC and crystal oscillator off. 32.768kHz XOSC, POR and ST active. RAM retention.
Power mode 3		0.6		μA	No clocks. RAM retention. Power On Reset (POR) active.
Peripheral Current Consumption					Add to the figures above if the peripheral unit is activated
Timer 1		10		μA/MHz	When enabled
Timer 2		10		μA/MHz	When enabled
Timer 3		10		μA/MHz	When enabled
Timer 4		10		μA/MHz	When enabled
Sleep Timer		0.5		μA	Including low-power RC oscillator or 32.768kHz XOSC
AES		50		μA/MHz	When encrypting/decrypting
ADC		0.9		mA	When converting
USART1 / USART2		12		μA/MHz	For each USART in use. Not including current for driving I/O pins.
DMA		30		μA/MHz	When operating, not including current for memory access
Flash write		3		mA	

Table 4: Electrical Specifications

7.1 General Characteristics

T_A=25°C, VDD=3.0V if nothing else stated.

Parameter	Min	Typ	Max	Unit	Condition/Note
Wake-Up and Timing					
Power mode 1 → power mode 0 (active)		38	77	ns	Digital regulator on, High Speed RCOSC or crystal oscillator running. Entry from PM1 to PM0 takes one clock period.
Power mode 2 or 3 → power mode 0 (active)		50		μs	Digital regulator off, High Speed RCOSC and crystal oscillator off. Start-up of regulator and High Speed RCOSC.
Active → RX 26 MHz XOSC initially OFF		495		μs	Digital regulator on. Crystal oscillator off. Start-up of crystal oscillator and RF TX/RX begins.
Active → TX 26 MHz XOSC initially OFF		495		μs	Digital regulator on. Crystal oscillator off. Start-up of crystal oscillator and RF TX/RX begins.
Active → RX		195		μs	Digital regulator on. Crystal oscillator on. Time from enabling radio until RX begins.
Active → TX		195		μs	Digital regulator on. Crystal oscillator on. Time from enabling radio until TX begins.
Radio part					
Frequency range	300		348	MHz	
	400		464	MHz	
	800		928	MHz	
Data rate	1.2		500	kbps	Modulation formats supported: (Shaped) MSK (also known as differential offset QPSK) up to 500kbps 2-FSK up to 500kbps GFSK and OOK/ASK (up to 250kbps) Optional Manchester encoding (halves the data rate) Data rate must be selected to comply with frequency regulations.

Table 5: General Characteristics

7.2 RF Receive Section

T_A=25°C, VDD=3.0V if nothing else stated. Measured on Chipcon's CC1110 EM reference design.

Parameter	Min	Typ	Max	Unit	Condition/Note
Differential input impedance		TBD		Ω	Follow CC1110EM reference design
Receiver sensitivity 315/433/868/915MHz		–109		dBm	2-FSK, 1.2kbps, 5.2kHz deviation, 1% packet error rate, 20 bytes packet length, 58kHz digital channel filter bandwidth
		–99		dBm	2-FSK, 38.4kbps, 20kHz deviation, 1% packet error rate, 20 bytes packet length, 100kHz digital channel filter bandwidth
		-87		dBm	2-FSK, 250kbps, 127kHz deviation, 1% packet error rate, 20 bytes packet length, 540kHz digital channel filter bandwidth
		-86		dBm	OOK, 250kbps OOK, 1% packet error rate, 20 bytes packet length, 540kHz digital channel filter bandwidth
Saturation		–15		dBm	
Digital channel filter bandwidth	58		812	kHz	User programmable. The bandwidth limits are proportional to crystal frequency (given values assume a 26.0MHz crystal).
Adjacent channel rejection, 868MHz		23		dB	2-FSK, 38.4kbps, 20kHz deviation, 1% packet error rate, 20 bytes packet length, 100kHz digital channel filter, 150kHz channel spacing Desired channel 3dB above the sensitivity limit.
Alternate channel rejection, 868MHz		33		dB	2-FSK, 38.4kbps, 20kHz deviation, 1% packet error rate, 20 bytes packet length, 100kHz digital channel filter, 150kHz channel spacing Desired channel 3dB above the sensitivity limit.
Image channel rejection, 868MHz		29		dB	2-FSK, 38.4kbps, 20kHz deviation, 1% packet error rate, 20 bytes packet length, 100kHz digital channel filter, 150kHz channel spacing, IF frequency 305kHz Desired channel 3dB above the sensitivity limit.
Blocking at 1MHz offset, 868MHz		52		dB	Desired channel 3dB above the sensitivity limit. Compliant to ETSI EN 300 220 class 2 receiver requirement.
Blocking at 2MHz offset, 868MHz		54		dB	Desired channel 3dB above the sensitivity limit. Compliant to ETSI EN 300 220 class 2 receiver requirement.
Blocking at 5MHz offset, 868MHz		61		dB	Desired channel 3dB above the sensitivity limit. Compliant to ETSI EN 300 220 class 2 receiver requirement.
Blocking at 10MHz offset, 868MHz		64		dB	Desired channel 3dB above the sensitivity limit. Compliant to ETSI EN 300 220 class 2 receiver requirement.
Spurious emissions			–57 –47	dBm dBm	25MHz – 1GHz Above 1GHz

Table 6: RF Receive Section

7.3 RF Transmit Section

T_A=25°C, VDD=3.0V if nothing else stated. Measured on Chipcon's CC1110 EM reference design.

Parameter	Min	Typ	Max	Unit	Condition/Note
Differential load impedance		TBD		Ω	Follow CC1110EM reference design
Output power, highest setting		10		dBm	Output power is programmable, and full range is available in all frequency bands. Output power must be selected to comply with frequency regulations. Output power is limited to 0 dBm when ASK is used. Delivered to a 50Ω single-ended load via Chipcon reference RF matching network.
Output power, lowest setting		–30		dBm	Output power is programmable, and full range is available in all frequency bands. Delivered to a 50Ω single-ended load via Chipcon reference RF matching network.
Spurious emissions and harmonics, 433/868MHz			–36	dBm	25MHz – 1GHz
			–54	dBm	47-74, 87.5-118, 174-230, 470-862MHz
			–47	dBm	1800MHz-1900MHz (restricted band in Europe), when the operating frequency is below 900MHz (2^{nd} harmonic can not fall within this band when used in Europe)
			–30	dBm	Otherwise above 1GHz
Spurious emissions, 315/915MHz			-49.2	dBm EIRP	<200μV/m at 3m below 960MHz.
			-41.2	dBm EIRP	<500μV/m at 3m above 960MHz.
Harmonics 315MHz			-20	dBc	2^{nd}, 3^{rd} and 4^{th} harmonic when the output power is maximum 6mV/m at 3m. (-19.6dBm EIRP)
			-41.2	dBm	5^{th} harmonic
Harmonics 915MHz			-20	dBc	2^{nd} harmonic
			-41.2	dBm	3^{rd}, 4^{th} and 5^{th} harmonic

Table 7: RF Transmit Parameters

7.4 26 MHz Crystal Oscillator

T_A=25°C, VDD=3.0V if nothing else is stated.

Parameter	Min	Typ	Max	Unit	Condition/Note
Crystal frequency	26	26	27	MHz	26 MHz and 27 MHz is supported.
Crystal frequency accuracy requirement		±40		ppm	This is the total tolerance including a) initial tolerance, b) ageing and c) temperature dependence. The acceptable crystal tolerance depends on RF frequency and channel spacing / bandwidth.

Parameter	Min	Typ	Max	Unit	Condition/Note
ESR			100	Ω	
Start-up time		300		µs	Measured on Chipcon's CC1110 EM reference design.

Table 8: 26 MHz Crystal Oscillator Parameters

7.5 32.768 kHz Crystal Oscillator

T_A =25°C, VDD=3.0V if nothing else is stated.

Parameter	Min	Typ	Max	Unit	Condition/Note
Crystal frequency		32.768		kHz	
ESR		40	130	Ω	
C_0		0.9	2.0	pF	
C_L		12	16	pF	
Start-up time			450	ms	

Table 9: 32.768 kHz Crystal Oscillator Parameters

7.6 Low Power RC Oscillator

T_A=25°C, VDD=3.0V if nothing else is stated.

Parameter	Min	Typ	Max	Unit	Condition/Note
Calibrated frequency	34.6	34.7	36	kHz	Calibrated RC Oscillator frequency is XTAL frequency divided by 750
Frequency accuracy after calibration			+0.3 -10	%	
Temperature coefficient		+0.4		% / °C	Frequency drift when temperature changes after calibration
Supply voltage coefficient		+3		% / V	Frequency drift when supply voltage changes after calibration
Initial calibration time		2		ms	When the RC Oscillator is enabled, calibration is continuously done in the background as long as the crystal oscillator is running.
Wake-up period	58µ		59650	Seconds	Programmable, dependent on XTAL frequency

Table 10: Low Power RC Oscillator parameters

7.7 High Speed RC Oscillator

T_A=25°C, VDD=3.0V if nothing else is stated.

Parameter	Min	Typ	Max	Unit	Condition/Note
Frequency		13		MHz	Calibrated High Speed RC Oscillator frequency is XTAL frequency multiplied by 1/2
Uncalibrated frequency accuracy		±15		%	
Calibrated frequency accuracy			±1	%	
Start-up time			10	µs	
Temperature coefficient			-325	ppm / °C	Frequency drift when temperature changes after calibration
Supply voltage coefficient			28	ppm / V	Frequency drift when supply voltage changes after calibration
Initial calibration time		50		µs	When the High Speed RC Oscillator is enabled, calibration is continuously done in the background as long as the crystal oscillator is running.

Table 11: High Speed RC Oscillator parameters

7.8 Frequency Synthesizer Characteristics

T_A=25°C, VDD=3.0V if nothing else stated. Measured on Chipcon's CC1110 EM reference design.

Parameter	Min	Typ	Max	Unit	Condition/Note
Programmed frequency resolution	397	$F_{XOSC}/2^{16}$	412	Hz	26MHz-27MHz crystal. The resolution (in Hz) is equal for all frequency bands.
Synthesizer frequency tolerance		±40		ppm	Given by crystal used. Required accuracy (including temperature and aging) depends on frequency band and channel bandwidth / spacing.
PLL turn-on / hop time			80	µs	Time from leaving the IDLE state (see Figure 38) until arriving in the RX, FSTXON or TX state, when not performing calibration. Crystal oscillator running.
PLL RX/TX and TX/RX settling time			10	µs	Settling time for the 1xIF frequency step from RX to TX, and vice versa.
PLL calibration time		18739		XOSC cycles	Calibration can be initiated manually, or automatically before entering or after leaving RX/TX.
	0.69	0.72	0.72	ms	Min/typ/max time is for 27/26/26MHz crystal frequency.

Table 12: Frequency Synthesizer Parameters

7.9 Analog Temperature Sensor

T_A=25°C, VDD=3.0V if nothing else stated. Measured on Chipcon's CC1110 EM reference design.

Parameter	Min	Typ	Max	Unit	Condition/Note
Output voltage at –40°C		0.660		V	
Output voltage at 0°C		0.755		V	
Output voltage at +40°C		0.859		V	
Output voltage at +80°C		0.958		V	
Temperature coefficient		2.54		mV/°C	Fitted from –20°C to +80°C
Error in calculated temperature, calibrated		0		°C	From –20°C to +80°C when using 2.44mV / °C, after 1-point calibration at room temperature
Current consumption increase when enabled		0.3		mA	

Table 13: Analog Temperature Sensor Parameters

7.10 8-14 bit ADC

T_A=25°C, VDD=3.0V if nothing else stated. Measured on Chipcon's CC1110 EM reference design.

Parameter	Min	Typ	Max	Unit	Condition/Note
Input voltage	0		AVDD	V	AVDD is voltage on AVDD pin
External reference voltage	0		AVDD	V	AVDD is voltage on AVDD pin
External reference voltage differential	0		AVDD	V	AVDD is voltage on AVDD pin
Number of bits (ENOB)	8		13	bits	The ADC is a delta-sigma. Effective resolution depends on sample rate used. Differential input signal and reference.
Offset		TBD		LSB	
Conversion time	18		114	μs	Using 26 MHz crystal oscillator
Differential nonlinearity (DNL)		±0.3		LSB	8-bits setting
Integral nonlinearity (INL)		±0.8		LSB	8-bits setting
SINAD (sine input)		45 56 66 75		dB dB dB dB	8-bits setting 10-bits setting 12-bits setting 14-bits setting

Table 14: 8-14 bit ADC Characteristics

7.11 Control AC Characteristics

T_A=85°C, VDD=3.0V if nothing else stated. Measured on Chipcon's CC1110 EM reference design.

Parameter	Min	Typ	Max	Unit	Condition/Note
System clock, f_{SYSCLK} t_{SYSCLK}= 1/ f_{SYSCLK}			26	MHz	Applies when the 26 MHz crystal oscillator is used. Maximum system clock is 13 MHz when high speed RC oscillator is used.
RESET_N low width	2.5			ns	See item 1, Figure 1. This is the shortest pulse that is guaranteed to be recognized as a reset pin request.
Interrupt pulse width	t_{SYSCLK}				See item 2, Figure 1.This is the shortest pulse that is guaranteed to be recognized as an interrupt request. In PM2/3 the internal synchronizers are bypassed so this requirement does not apply in PM2/3.

Table 15: Control Inputs AC Characteristics

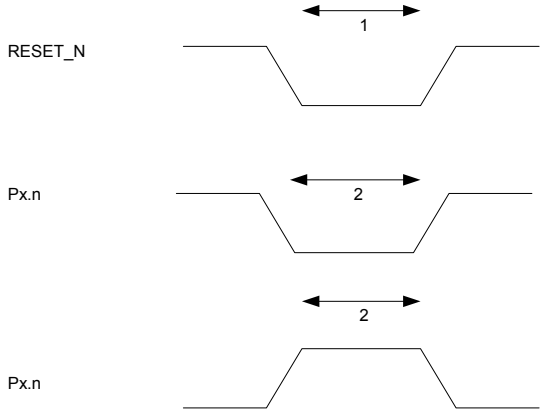


Figure 1: Control Inputs AC Characteristics

7.12 SPI AC Characteristics

T_A=85°C, VDD=3.0V if nothing else stated. Measured on Chipcon's CC1110 EM reference design.

Parameter	Min	Typ	Max	Unit	Condition/Note
SCK period		See section 13.13.3		ns	Master. See item 1 Figure 2
SCK duty cycle		50%			Master.
MISO setup	10			ns	Master. See item 2 Figure 2
MISO hold	10			ns	Master. See item 3 Figure 2
SCK to MOSI			25	ns	Master. See item 4 Figure 2, load = 10 pF
SCK period	100			ns	Slave. See item 1 Figure 2
SCK duty cycle		50%			Slave.
MOSI setup	10			ns	Slave. See item 2 Figure 2
MOSI hold	10			ns	Slave. See item 3 Figure 2
SCK to MISO			25	ns	Slave. See item 4 Figure 2, load = 10 pF

Table 16: SPI AC Characteristics

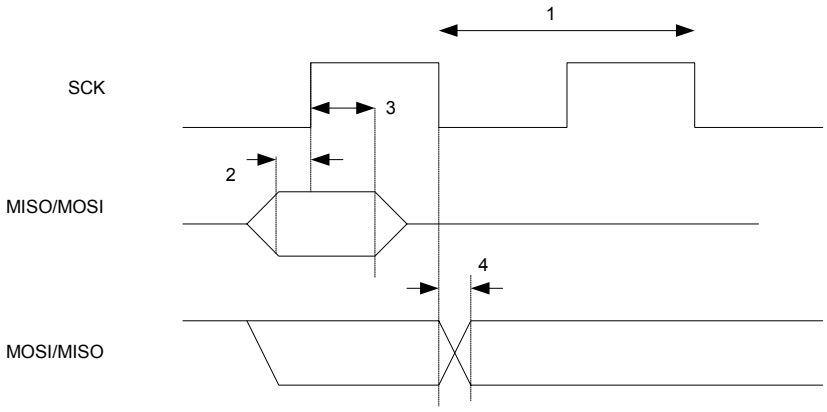


Figure 2: SPI AC Characteristics

7.13 Debug Interface AC Characteristics

T_A=85°C, VDD=3.0V if nothing else stated. Measured on Chipcon's CC1110 EM reference design.

Parameter	Min	Typ	Max	Unit	Condition/Note
Debug clock period	31.25			ns	See item 1 Figure 3
Debug data setup	5				See item 2 Figure 3
Debug data hold	5				See item 3 Figure 3
Clock to data delay			10		See item 4 Figure 3, load = 10 pF
RESET_N inactive after P2_2 rising	10				See item 5 Figure 3

Table 17: Debug Interface AC Characteristics

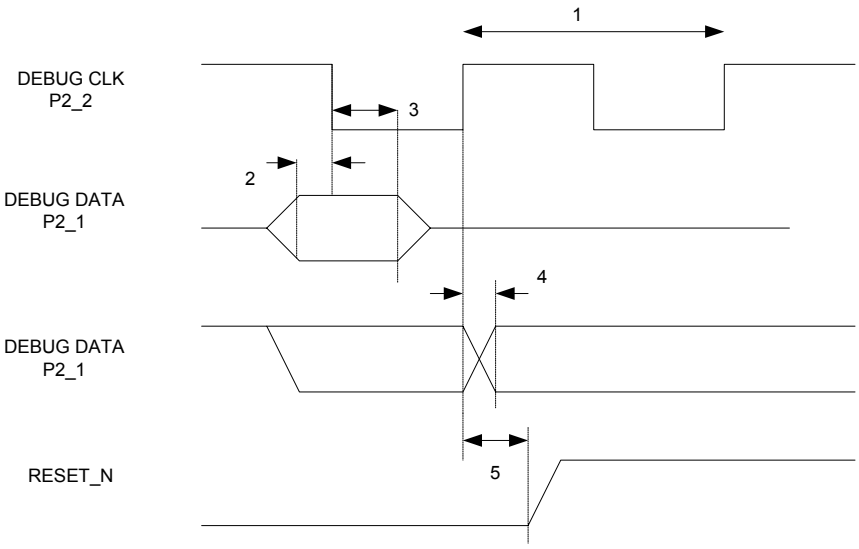


Figure 3: Debug Interface AC Characteristics

7.14 Port Outputs AC Characteristics

T_A=85°C, VDD=3.0V if nothing else stated. Measured on Chipcon's CC1110 EM reference design.

Parameter	Min	Typ	Max	Unit	Condition/Note
P0, P1, P2Port output pins, rise and fall time		10			Load = 10 pF Timing is with respect to 10% VDD and 90% VDD levels.

Table 18: Port Outputs AC Characteristics

7.15 Timer Inputs AC Characteristics

T_A =85°C, VDD=3.0V if nothing else stated. Measured on Chipcon's CC1110 EM reference design.

Parameter	Min	Typ	Max	Unit	Condition/Note
Input capture pulse width	t_{SYSCLK}				Synchronizers determine the shortest input pulse that can be recognized. The synchronizers operate from the current system clock rate

Table 19: Timer Inputs AC Characteristics

7.16 DC Characteristics

The DC Characteristics of CC1110 are listed in Table 20 below.

T_A=25°C, VDD=3.0V if nothing else stated. Measured on Chipcon's CC1110 EM reference design.

Digital Inputs/Outputs	Min	Typ	Max	Unit	Condition
Logic "0" input voltage	0	0.7	0.9	V	
Logic "1" input voltage	VDD-0.25	VDD	VDD	V	
Logic "0" output voltage	0	0	0.25	V	For up to 4mA output current on all pins except P1_0 and P1_1 which are up to 20 mA
Logic "1" output voltage	VDD-0.25	VDD	VDD	V	For up to 4mA output current on all pins except P1_0 and P1_1 which are up to 20 mA
Logic "0" input current	N/A	–1	–1	μA	Input equals 0V
Logic "1" input current	N/A	1	1	μA	Input equals VDD
I/O pin pull-up and pull-down resistor	17	20	23	kΩ	

Table 20: DC Characteristics

8 Pin and I/O Port Configuration

The *CC1110* pinout is shown in Figure 4 and Table 21. See section 13.1 for details on the configuration of digital I/O ports.

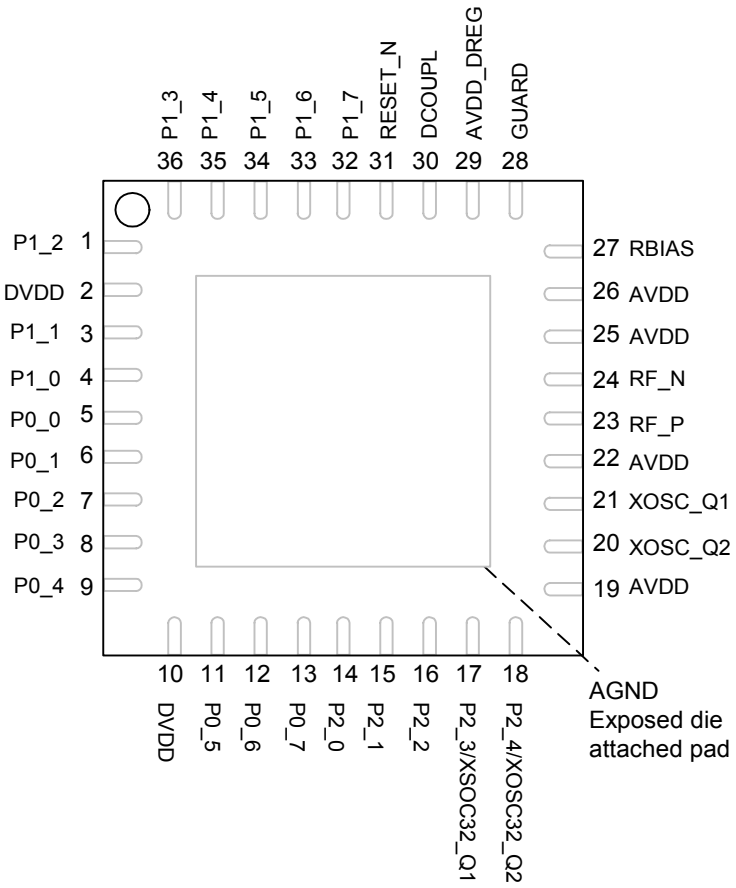


Figure 4: Pinout top view

Note: The exposed die attach pad **must** be connected to a solid ground plane as this is the ground connection for the chip.

Pin	Pin name	Pin type	Description
-	GND	Ground	The exposed die attach pad must be connected to a solid ground plane
1	P1_2	D I/O	Port 1.2
2	DVDD	Power (Digital)	2.0V-3.6V digital power supply for digital I/O
3	P1_1	D I/O	Port 1.1
4	P1_0	D I/O	Port 1.0
5	P0_0	D I/O	Port 0.0
6	P0_1	D I/O	Port 0.1
7	P0_2	D I/O	Port 0.2
8	P0_3	D I/O	Port 0.3
9	P0_4	D I/O	Port 0.4
10	DVDD	Power (Digital)	2.0V-3.6V digital power supply for digital I/O
11	P0_5	D I/O	Port 0.5
12	P0_6	D I/O	Port 0.6
13	P0_7	D I/O	Port 0.7
14	P2_0	D I/O	Port 2.0
15	P2_1	D I/O	Port 2.1
16	P2_2	D I/O	Port 2.2
17	P2_3/XOSC32_Q1	D I/O	Port 2.3/32.768 kHz crystal oscillator pin 1
18	P2_4/XOSC32_Q2	D I/O	Port 2.4/32.768 kHz crystal oscillator pin 2
19	AVDD	Power (Analog)	2.0V-3.6V analog power supply connection
20	XOSC_Q2	Analog I/O	26 MHz crystal oscillator pin 2
21	XOSC_Q1	Analog I/O	26 MHz crystal oscillator pin 1, or external clock input
22	AVDD	Power (Analog)	2.0V-3.6V analog power supply connection
23	RF_P	RF I/O	Positive RF input signal to LNA in receive mode Positive RF output signal from PA in transmit mode
24	RF_N	RF I/O	Negative RF input signal to LNA in receive mode Negative RF output signal from PA in transmit mode
25	AVDD	Power (Analog)	2.0V-3.6V analog power supply connection
26	AVDD	Power (Analog)	2.0V-3.6V analog power supply connection
27	RBIAS	Analog I/O	External precision bias resistor for reference current
28	GUARD	Power (Digital)	Power supply connection for digital noise isolation
29	AVDD_DREG	Power (Digital)	2.0V-3.6V digital power supply for digital core voltage regulator
30	DCOUPL	Power decoupling	1.8V digital power supply decoupling
31	RESET_N	DI	Reset, active low
32	P1_7	D I/O	Port 1.7
33	P1_6	D I/O	Port 1.6
34	P1_5	D I/O	Port 1.5
35	P1_4	D I/O	Port 1.4
36	P1_3	D I/O	Port 1.3

Table 21: Pinout overview

9 Circuit Description

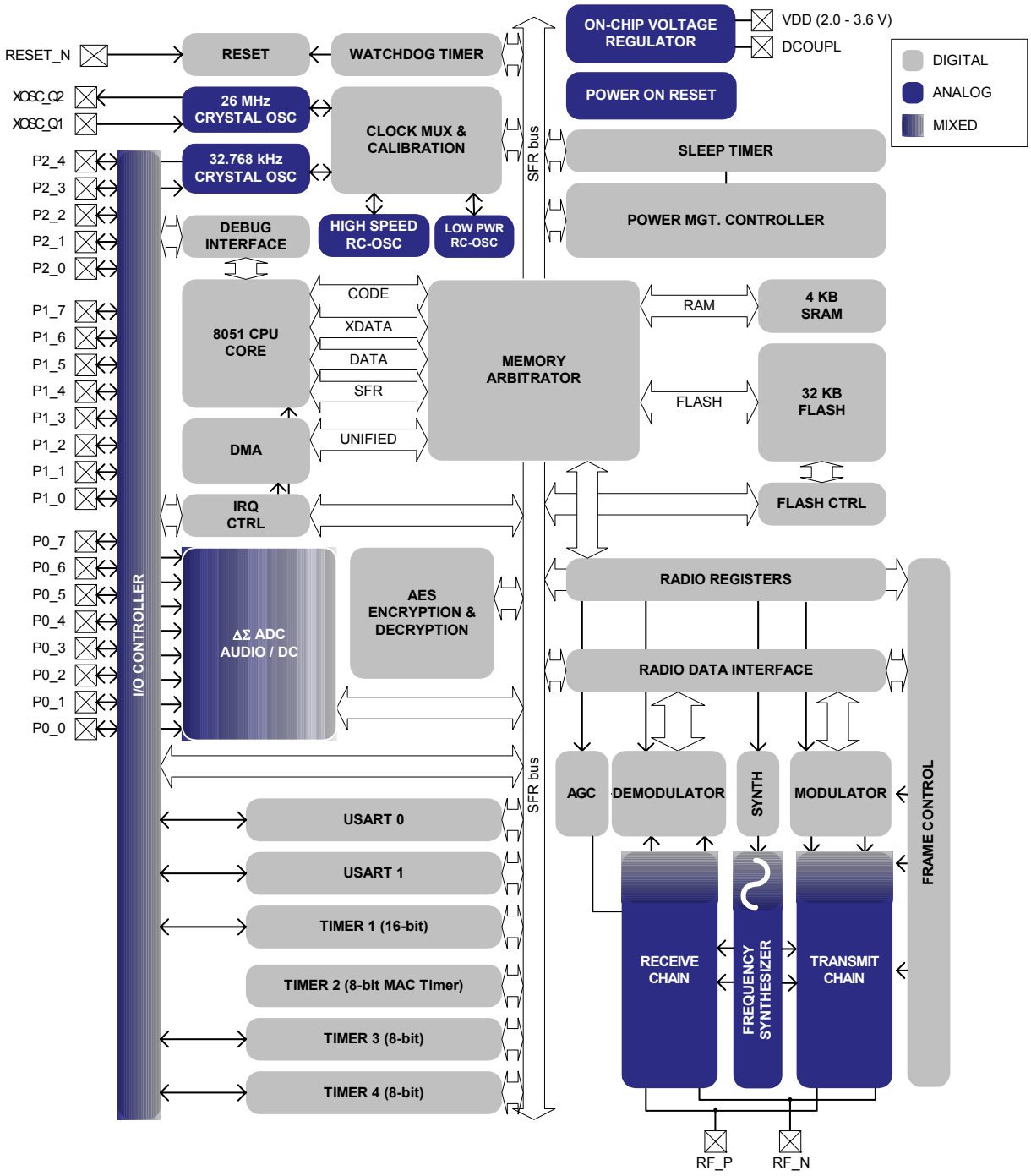


Figure 5: CC1110 Block Diagram

A block diagram of CC1110 is shown in Figure 5. The modules can be roughly divided into one of three categories: CPU-related modules, radio-related modules and modules related to power, test and clock distribution. In the following subsections, a short description of each module that appears in Figure 5 is given.

9.1 CPU and Peripherals

The **8051 CPU core** is a single-cycle 8051-compatible core. It has three different memory access buses (SFR, DATA and CODE/XDATA), a debug interface and an 18-input extended interrupt unit. See section 12 for details.

The **memory crossbar/arbitrator** is at the heart of the system as it connects the CPU and DMA controller with the physical memories and all peripherals through the SFR bus. The memory arbitrator has four memory access points, access at which can map to one of three physical memories: a 4 KB SRAM, 32 KB flash memory or SFR registers. The memory arbitrator is responsible for performing arbitration and sequencing between simultaneous memory accesses to the same physical memory.

The **SFR bus** is drawn conceptually in the block diagram as a common bus that connects all hardware peripherals to the memory arbitrator. The SFR bus also provides access to the radio registers in the radio register bank even though these are indeed mapped into XDATA memory space.

The **4 KB SRAM** maps to the DATA memory space and part of the XDATA/CODE memory spaces. The memory is an ultra-low-power SRAM that retains its contents even when the digital part is powered off (power modes 2 and 3).

The **32 KB flash block** provides in-circuit programmable non-volatile program memory for the device and maps into the CODE and XDATA memory spaces. Writing to the flash block is performed through a **flash controller** that allows page-wise (1024 byte) erasure and byte-wise reprogramming. See section 13.14 for details.

A versatile five-channel **DMA controller** is available in the system and accesses memory using a unified memory space (XDATA) and thus has access to all physical memories. Each channel is configured (trigger, priority, transfer mode, addressing mode, source and destination pointers, and transfer count) with DMA descriptors anywhere in memory. Many of the hardware peripherals rely on the DMA controller for efficient operation (AES core, flash write controller, USARTs, Timers, ADC interface) by performing data transfers between a single SFR address and flash/SRAM. See section 13.2 for details.

The **interrupt controller** services 18 interrupt sources, divided into six *interrupt groups,* each of which is associated with one of four interrupt priorities. An interrupt request is serviced even if the device is in a sleep mode (power modes 1-3) by bringing the CC1110 back to active mode (power mode 0).

The **debug interface** implements a proprietary two-wire serial interface that is used for in-circuit debugging. Through this debug interface it is possible to perform an erasure of the entire flash memory, control which oscillators are enabled, stop and start execution of the user program, execute supplied instructions on the 8051 core, set code breakpoints, and single step through instructions in the code. Using these techniques, it is possible to elegantly perform in-circuit debugging and external flash programming. See section 12.9 for details.

The **I/O-controller** is responsible for all general-purpose I/O pins. The CPU can configure whether peripheral modules control certain pins or whether they are under software control, and if so whether each pin is configured as an input or output and if a pull-up or pull-down resistor in the pad is connected. Each peripheral that connects to the I/O-pins can choose between two different locations to ensure flexibility in various applications. See section 13.1 for details.

The **sleep timer** is an ultra-low power timer that counts 32.768 kHz crystal oscillator or 32.768 kHz RC oscillator periods. The sleep timer runs continuously in all operating modes except power mode 3. It can be configured in one of several resolution modes, to strike the right balance between timer resolution and timeout period. Typical uses for it is as a real-

time counter that runs regardless of operating mode (except power mode 3) or as a wakeup timer to get out of power modes 1 or 2. See section 13.5 for details.

A built-in **watchdog timer** allows the CC1110 to reset itself in case the firmware hangs. When enabled, the watchdog timer must be cleared periodically, otherwise it will reset the device when it times out. See section 13.12 for details.

Timer 1 is a 16-bit timer with timer/counter/PWM functionality. It has a programmable prescaler, a 16-bit period value and three individually programmable counter/capture channels each with a 16-bit compare value. Each of the counter/capture channels can be used as PWM outputs or to capture the timing of edges on input signals. See section 13.3 for details.

Timer 2 (MAC timer) is specifically designed to support time-slotted protocols in software. The timer has a configurable timer period and 18-bit tunable prescaler range. See section 13.4 for details.

Timers 3 and 4 are 8-bit timers with timer/counter/PWM functionality. They have a programmable prescaler, an 8-bit period value and one programmable counter/capture channel with an 8-bit compare value. Each of the counter/capture channels can be used as PWM outputs or to capture the timing of edges on input signals. See section 13.6 for details.

USART 0 and 1 are each configurable as either an SPI master/slave or a UART. They provide double buffering on both RX and TX and hardware flow-control and are thus well suited to high-throughput full-duplex applications. Each has its own high-precision baud-rate generator thus leaving the ordinary timers free for other uses. When configured as an SPI slave they sample the input signal using SCK directly instead of some over-sampling scheme and are thus well suited to high data rates. See section 13.13 for details.

The **AES encryption/decryption core** allows the user to encrypt and decrypt data using the AES algorithm with 128-bit keys. See section 13.9 for details.

The **ADC** supports 8 to 14 bits of resolution in a 30 kHz to 4 kHz bandwidth respectively. DC and audio conversion with up to eight input channels (Port 0) is possible. The inputs can be selected as single ended or differential. The reference voltage can be internal, AVDD, or a single ended or differential external signal. The ADC also has a temperature sensor input channel. The ADC can automate the process of periodic sampling or conversion over a sequence of channels. See Section 13.7 for details.

9.2 Radio

CC1110 features an RF transceiver based on the industry-leading CC1100, requiring very few external components. See Section 15 for details.

10 Power Control

The CC1110 has four power modes, called PM0, PM1, PM2 and PM3. PM0 is the active mode while PM1 to PM3 are low-power modes, where PM3 has the lowest power consumption. The power modes are shown in Table 22 together with voltage regulator and oscillator options.

Power Mode	High speed oscillator	Low-speed oscillator	Voltage regulator (digital)
Configuration	A None B 26 MHz XOSC C HS RCOSC D Both	A None B Low power RCOSC C 32.768 kHz XOSC	A Off B On
PM0	B, C, D	B, C	B
PM1	A	B, C	B
PM2	A	B, C	A
PM3	A	A	A

Table 22: Power modes

PM0: The active mode. The voltage regulator to the digital core is on and either the high speed RC oscillator or the 26 MHz crystal oscillator or both are running. Either the low power RC oscillator or the 32.768 kHz crystal oscillator is running.

PM1: The voltage regulator to the digital part is on. Neither the 26 MHz crystal oscillator nor the high speed RC oscillator is running. Either the low power RC oscillator or the 32.768 kHz crystal oscillator is running. The system will go to PM0 on reset or an external interrupt or when the sleep timer expires.

PM2: The voltage regulator to the digital core is turned off. Neither the 26 MHz crystal oscillator nor the high speed RC oscillator is running. Either the low power RC oscillator or the 32.768 kHz crystal oscillator is running. The system will go to PM0 on reset or an external interrupt or when the sleep timer expires.

PM3: The voltage regulator to the digital core is turned off. None of the oscillators are running. The system will go to PM0 on reset or an external interrupt.

11 Application Circuit

This section describes the recommended application circuit for the RF part of the CC1110, together with crystal oscillator connections.

Only a few external components are required for using the CC1110. RF transceiver. The recommended application circuit is shown in Figure 6. The external components are described in Table 23, and typical values are given in Table 24.

11.1 Bias resistor

The bias resistor R271 is used to set an accurate bias current. It is very important to use the specified tolerance for this resistor.

11.2 Balun and RF matching

C231, C241, L231 and L241 form the recommended balun that converts the differential RF port on CC1110 to a single-ended RF signal (C234 is also needed for DC blocking). Together with an appropriate LC network, the balun components also transform the impedance to match a 50 Ω antenna (or cable). Component values for the RF balun and LC network are easily found using the SmartRF® Studio software. Suggested values are listed in Table 24.

11.3 Crystal

The crystal oscillator uses an external crystal X1, with two loading capacitors (C201 and C211). See section 14 on page 153 for details.

The circuit also shows the connections for the optional 32.768 kHz crystal oscillator, with external crystal X2 and loading capacitors C181 and C171. This crystal oscillator is used by the Sleep Timer providing a real-time clock function and is not required for radio operation.

11.4 Power supply decoupling

The power supply must be properly decoupled close to the supply pins. Note that decoupling capacitors are not shown in the application circuit. The placement and the size of the decoupling capacitors are very important to achieve the optimum performance. Chipcon provides a reference design that should be followed closely.

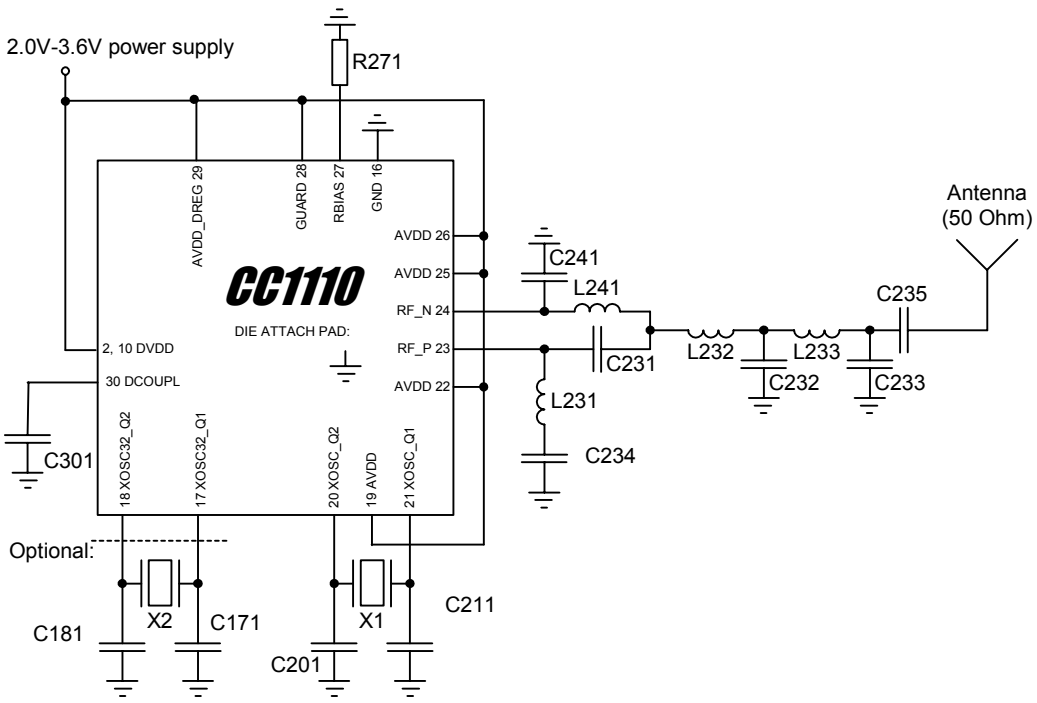


Figure 6: Application Circuit (excluding supply decoupling capacitors and digital I/O)

Component	Description
C301	100 nF decoupling capacitor for on-chip voltage regulator to digital part
C201; C211	Crystal loading capacitors, see section 14 on page 153 for details
C171, C181	Crystal loading capacitors, if X2 is used
C231, C241	RF balun/matching capacitors
C232/C233	RF LC filter/matching capacitors
C234	RF balun DC blocking capacitor
C235	RF LC filter DC blocking capacitor (only needed if there is a DC path in the antenna)
L231/L241	RF balun/matching inductors (inexpensive multi-layer type)
L232/L233	RF LC filter/matching filter inductors (inexpensive multi-layer type)
R271	56 kΩ resistor for internal bias current reference, tolerance 1%
X1	26 MHz-27 MHz crystal, see section 14 on page 153 for details
X2	32.768 kHz crystal, optional

Table 23: Overview of external components (excluding supply decoupling capacitors)

Component	Value at 315 MHz	Value at 433 MHz	Value at 868/915 MHz
C301	100 nF±10%, 0402 X5R		
C201, C211	27 pF±5%, 0402 NP0		
C231, C241	6.8 pF±0.5pF, 0402 NP0	3.9 pF±0.25 pF, 0402 NP0	2.2 pF±0.25 pF, 0402 NP0
C232	12 pF±5%, 0402 NP0	8.2 pF±0.5 pF, 0402 NP0	3.9 pF±0.25 pF, 0402 NP0
C233	6.8 pF±0.5 pF, 0402 NP0	5.6 pF±0.5 pF, 0402 NP0	3.3 pF±0.25 pF, 0402 NP0
C234, C235	220 pF±5%, 0402 NP0	220 pF±5%, 0402 NP0	100 pF±5%, 0402 NP0
L231, L233, L241	33 nH±5%, 0402 monolithic	27 nH±5%, 0402 monolithic	12 nH±5%, 0402 monolithic
L232	18 nH±5%, 0402 monolithic	22 nH±5%, 0402 monolithic	5.6 nH±0.3nH, 0402 monolithic
R271	56 kΩ±1%, 0402		
X1	26.0 MHz surface mount crystal		

Table 24: Bill Of Materials for the CC1110 application circuit (subject to changes)

12 8051 CPU

This section describes the 8051 CPU core, with interrupts, memory and instruction set.

12.1 8051 CPU Introduction

The *CC1110* includes an 8-bit CPU core, which is an enhanced version of the industry standard 8051 core.

The enhanced 8051 core uses the standard 8051 instruction set. Instructions execute faster than the standard 8051 due to the following:

- One clock per instruction cycle is used as opposed to 12 clocks per instruction cycle in the standard 8051.
- Wasted bus states are eliminated.

Since an instruction cycle is aligned with memory fetch when possible, most of the single byte instructions are performed in a single clock cycle. In addition to the speed improvement, the enhanced 8051 core also includes architectural enhancements:

- Dual data pointers
- Extended 18-source interrupt unit

The 8051 core is object code compatible with the industry standard 8051 microcontroller. That is, object code compiled with an industry standard 8051 compiler or assembler executes on the 8051 core and is functionally equivalent. However, because the 8051 core uses a different instruction timing than many other 8051 variants, existing code with timing loops may require modification. Also, because the peripheral units such as timers and serial ports differ from those on other 8051 cores, code which includes instructions using the peripheral units SFRs will not work correctly.

12.2 Reset

The *CC1110* has three reset sources. The following events generate a reset:

- Forcing RESET_N input pin low
- A power-on reset condition
- Watchdog timer reset condition

The initial conditions after a reset are as follows:

- I/O pins are configured as inputs with pull-up
- CPU program counter is loaded with 0x0000 and program execution starts at this address
- All peripheral registers are initialized to their reset values (refer to register descriptions)
- Watchdog timer is disabled

12.3 Memory

The 8051 CPU has four different memory spaces:

CODE. A 16-bit read-only memory space for program memory.

DATA. An 8-bit read/write data memory space, which can be directly or indirectly accessed by a single CPU instruction. The lower 128 bytes of the DATA memory space can be addressed either directly or indirectly, the upper 128 bytes only indirectly.

XDATA. A 16-bit read/write data memory space access to which usually requires 4-5 CPU instruction cycles. Access to XDATA memory is also slower in hardware than DATA access as the CODE and XDATA memory spaces share a common bus on the CPU core and instruction pre-fetch from CODE can thus not be performed in parallel with XDATA accesses.

SFR. A 7-bit read/write register memory space, which can be directly accessed by a single CPU instruction. For SFR registers whose address is divisible by eight, each bit is also individually addressable.

The four different memory spaces are distinct in the 8051 architecture, but are partly overlapping in the *CC1110* to ease DMA transfers and hardware debugger operation.

How the different memory spaces are mapped onto the three physical memories (32 KB flash

program memory, 4 KB SRAM and hardware registers) is described in section 12.3.2.

12.3.1 Memory Map

This section gives an overview of the memory map.

The memory map showing how the different physical memories are mapped into the CPU memory spaces is given in

Figure 7 and Figure 8.

The memory differs from the standard 8051 memory map in two important aspects, as described below.

First, in order to allow the DMA controller access to all physical memory and thus allow DMA transfers between these areas, all the physical memories are mapped into a unified XDATA memory space as shown in

Figure 7.

Secondly, the CODE memory space is mapped so that all physical memory is mapped to CODE space, by using a *unified mapping* of the CODE memory space.

Details about the mapping of all 8051 memory spaces are given in the next section.

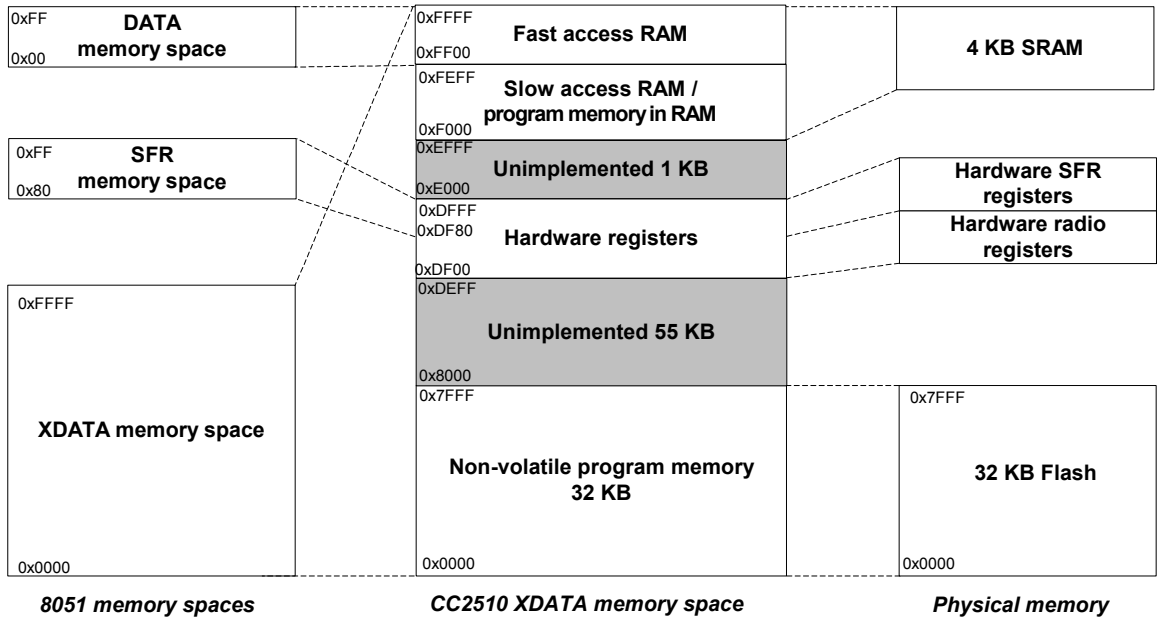


Figure 7: CC1110 XDATA memory space

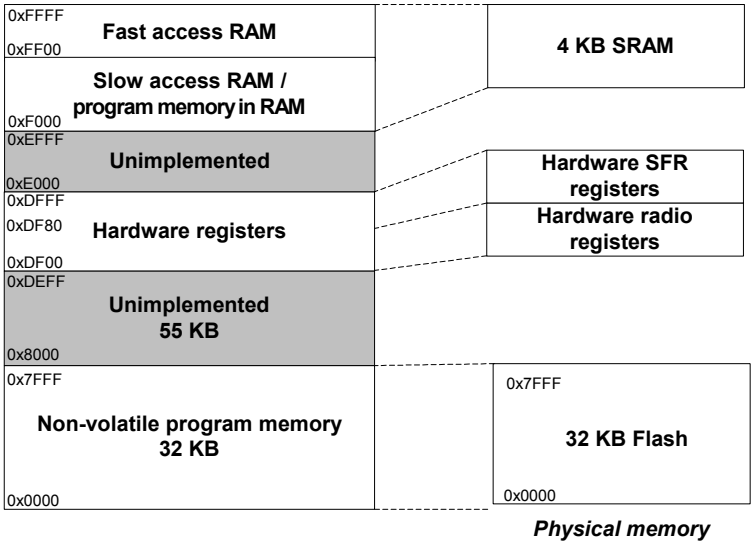


Figure 8: CODE memory space

12.3.2 Memory Space

This section describes the details of each CPU memory space.

XDATA memory space. The 32 KB flash program memory is mapped into the address range 0x0000-0x7FFF, the 4 KB SRAM into address range 0xF000-0xFFFF, radio registers into address range 0xDF00-0xDF3D and the SFR registers into address range 0xDF80-0xDFFF. This allows the DMA controller (and the CPU) access to all the physical memories in a single unified address space. One of the ramifications of this mapping is that the first address of usable SRAM starts at address 0xF000 instead of 0x0000, and therefore compilers/assemblers must take this into consideration.

In low-power modes PM2-3, the 4 KB SRAM at XDATA address range 0xF000-0xFFFF will retain its contents. Refer to section 13.10 on page 130 for a detailed description of power modes and SRAM data retention.

CODE memory space. The mapping of the CODE memory space is similar to the XDATA mapping with the exception that SFR registers internal to the CPU can not be accessed. These SFR registers are shown in grey in Table 25. Since XDATA and CODE memory accesses share a common bus on the CPU, their memory spaces are made similar to simplify memory decoding. The 4 KB SRAM is included in the CODE address space to allow program execution out of the SRAM.

DATA memory space. The 8-bit address range of DATA memory is mapped into the upper 256 bytes of the 4 KB SRAM. This area is also accessible through the CODE and XDATA memory spaces at the address range 0xFF00-0xFFFF.

SFR memory space. The 128-entry hardware register area is accessed through this memory space. The SFR registers are also accessible through the XDATA/DMA address space at the address range 0xDF80-0xDFFF. Some CPU-specific SFR registers reside inside the CPU core and can only be accessed using the SFR memory space and not through the duplicate mapping into XDATA memory space.

12.3.3 Data Pointers

The ***CC1110*** has two data pointers, DPTR0 and DPTR1 to accelerate the movement of data blocks to/from memory. The data pointers are generally used to access CODE or XDATA space e.g.

```
MOVC A,@A+DPTR

MOV A,@DPTR.
```

The data pointer select bit, bit 0 in the Data Pointer Select register DPS, chooses which data pointer shall be the active one during execution of an instruction that uses the data pointer, e.g. in one of the above instructions.

The data pointers are two bytes wide consisting of the following SFRs:

- DPTR0 – DPH0:DPL0
- DPTR1 – DPH1:DPL1

DPH0 (0x83) – Data Pointer 0 High Byte

Bit	Name	Reset	R/W	Description
7:0	DPH0[7:0]	0	R/W	Data pointer 0, high byte

DPL0 (0x82) – Data Pointer 0 Low Byte

Bit	Name	Reset	R/W	Description
7:0	DPL0[7:0]	0	R/W	Data pointer 0, low byte

DPH1 (0x85) – Data Pointer 1 High Byte

Bit	Name	Reset	R/W	Description
7:0	DPH1[7:0]	0	R/W	Data pointer 1, high byte

DPL1 (0x84) – Data Pointer 1 Low Byte

Bit	Name	Reset	R/W	Description
7:0	DPL1[7:0]	0	R/W	Data pointer 1, low byte

DPS (0x92) – Data Pointer Select

Bit	Name	Reset	R/W	Description
7:1	-	0x00	R0	Not used
0	DPS	0	R/W	Data pointer select. Selects active data pointer. 0 DPTR0 1 DPTR1

12.3.4 XDATA Memory Access

The ***CC1110*** provides an additional SFR register MPAGE. This register is used during instructions MOVX A,@Ri and MOVX @Ri,A MPAGE gives the 8 most significant address bits, while the register Ri gives the 8 least significant bits.

In some 8051 implementations, this type of XDATA access is performed using P2 to give the most significant address bits. Existing software may therefore have to be adapted to make use of MPAGE instead of P2.

MPAGE (0x93)– Memory Page Select

Bit	Name	Reset	R/W	Description
7:0	MPAGE[7:0]	0x00	R/W	Memory page, high-order bits of address in MOVX instruction

12.4 SFR Registers

The Special Function Registers (SFRs) control several of the features of the 8051 CPU core and/or peripherals. Many of the 8051 core SFRs are identical to the standard 8051 SFRs. However, there are additional SFRs that control features that are not available in the standard 8051. The additional SFRs are used to interface with the peripheral units and RF transceiver.

Table 25 shows the address to all SFRs in **CC1110**. The 8051 internal SFRs are shown with grey background, while the the other SFRs are specific to **CC1110**.

Note : all internal SFRs (shown with grey background in Table 25), can only be accessed through SFR space as these registers are not mapped into XDATA space.

Table 26 lists the additional SFRs that are not standard 8051 peripheral SFRs or CPU-internal SFRs. The additional SFRs are described in the relevant sections for each peripheral function.

	8 bytes								
80	P0	SP	DPL0	DPH0	DPL1	DPH1	U0CSR	PCON	87
88	TCON	P0IFG	P1IFG	P2IFG	PICTL	P1IEN	-	P0INP	8F
90	P1	RFIM	DPS	MPAGE	-	-	-	-	97
98	S0CON	-	IEN2	S1CON	T2CT	T2PR	T2CTL	-	9F
A0	P2	WORIRQ	WORCTL	WOREVT0	WOREVT1	WORTIME0	WORTIME1	-	A7
A8	IEN0	IP0	-	FWT	FADDRL	FADDRH	FCTL	FWDATA	AF
B0	-	ENCDI	ENCDO	ENCCS	ADCCON1	ADCCON2	ADCCON3	-	B7
B8	IEN1	IP1	ADCL	ADCH	RNDL	RNDH	SLEEP	-	BF
C0	IRCON	U0DBUF	U0BAUD	-	U0UCR	U0GCR	CLKCON	MEMCTR	C7
C8	-	WDCTL	T3CNT	T3CTL	T3CCTL0	T3CC0	T3CCTL1	T3CC1	CF
D0	PSW	DMAIRQ	DMA1CFGL	DMA1CFGH	DMA0CFGL	DMA0CFGH	DMAARM	DMAREQ	D7
D8	TIMIF	RFD	T1CC0L	T1CC0H	T1CC1L	T1CC1H	T1CC2L	T1CC2H	DF
E0	ACC	RFST	T1CNTL	T1CNTH	T1CTL	T1CCTL0	T1CCTL1	T1CCTL2	E7
E8	IRCON2	RFIF	T4CNT	T4CTL	T4CCTL0	T4CC0	T4CCTL1	T4CC1	EF
F0	B	PERCFG	ADCCFG	P0SEL	P1SEL	P2SEL	P1INP	P2INP	F7
F8	U1CSR	U1DBUF	U1BAUD	U1UCR	U1GCR	P0DIR	P1DIR	P2DIR	FF

Table 25: SFR address overview

Table 26: CC1110 specific SFR overview

Register name	SFR Address	Module	Description
ADCCON1	0xB4	ADC	ADC Control 1
ADCCON2	0xB5	ADC	ADC Control 2
ADCCON3	0xB6	ADC	ADC Control 3
ADCL	0xBA	ADC	ADC Data Low

Register name	SFR Address	Module	Description
ADCH	0xBB	ADC	ADC Data High
RNDL	0xBC	ADC	Random Number Generator Data Low
RNDH	0xBD	ADC	Random Number Generator Data High
ENCDI	0xB1	AES	Encryption/Decryption Input Data
ENCDO	0xB2	AES	Encryption/Decryption Output Data
ENCCS	0xB3	AES	Encryption/Decryption Control and Status
DMAIRQ	0xD1	DMA	DMA Interrupt Flag
DMA1CFGL	0xD2	DMA	DMA Channel 1-4 Configuration Address Low
DMA1CFGH	0xD3	DMA	DMA Channel 1-4 Configuration Address High
DMA0CFGL	0xD4	DMA	DMA Channel 0 Configuration Address Low
DMA0CFGH	0xD5	DMA	DMA Channel 0 Configuration Address High
DMAARM	0xD6	DMA	DMA Channel Arm
DMAREQ	0xD7	DMA	DMA Channel Start Request and Status
FWT	0xAB	FLASH	Flash Write Timing
FADDRL	0xAC	FLASH	Flash Address Low
FADDRH	0xAD	FLASH	Flash Address High
FCTL	0xAE	FLASH	Flash Control
FWDATA	0xAF	FLASH	Flash Write Data
P0IFG	0x89	IOC	Port 0 Interrupt Status Flag
P1IFG	0x8A	IOC	Port 1 Interrupt Status Flag
P2IFG	0x8B	IOC	Port 2 Interrupt Status Flag
PICTL	0x8C	IOC	Port Pins Interrupt Mask and Edge
P1IEN	0x8D	IOC	Port 1 Interrupt Mask
P0INP	0x8F	IOC	Port 0 Input Mode
PERCFG	0xF1	IOC	Peripheral I/O Control
ADCCFG	0xF2	IOC	ADC Input Configuration
P0SEL	0xF3	IOC	Port 0 Function Select
P1SEL	0xF4	IOC	Port 1 Function Select
P2SEL	0xF5	IOC	Port 2 Function Select
P1INP	0xF6	IOC	Port 1 Input Mode
P2INP	0xF7	IOC	Port 2 Input Mode
P0DIR	0xFD	IOC	Port 0 Direction
P1DIR	0xFE	IOC	Port 1 Direction
P2DIR	0xFF	IOC	Port 2 Direction
MEMCTR	0xC7	MEMORY	Memory System Control
SLEEP	0xBE	PMC	Sleep Mode Control
CLKCON	0xC6	PMC	Clock Control
RFIM	0x91	RF	RF Interrupt Mask
RFD	0xD9	RF	RF Data
RFIF	0xE9	RF	RF Interrupt flags

Register name	SFR Address	Module	Description
RFST	0xE1	RF	RF Strobe Commands
WORIRQ	0xA1	Sleep Timer	Sleep Timer Interrupts
WORCTRL	0xA2	Sleep Timer	Sleep Timer Control
WOREVT0	0xA3	Sleep Timer	Sleep Timer Event 0
WOREVT1	0xA5	Sleep Timer	Sleep Timer Event 1
WORTIME0	0xA4	Sleep Timer	Sleep Timer Value 0
WORTIME1	0xA6	Sleep Timer	Sleep Timer Value 1
T1CC0L	0xDA	Timer1	Timer 1 Channel 0 Capture/Compare Value Low
T1CC0H	0xDB	Timer1	Timer 1 Channel 0 Capture/Compare Value High
T1CC1L	0xDC	Timer1	Timer 1 Channel 1 Capture/Compare Value Low
T1CC1H	0xDD	Timer1	Timer 1 Channel 1 Capture/Compare Value High
T1CC2L	0xDE	Timer1	Timer 1 Channel 2 Capture/Compare Value Low
T1CC2H	0xDF	Timer1	Timer 1 Channel 2 Capture/Compare Value High
T1CNTL	0xE2	Timer1	Timer 1 Counter Low
T1CNTH	0xE3	Timer1	Timer 1 Counter High
T1CTL	0xE4	Timer1	Timer 1 Control and Status
T1CCTL0	0xE5	Timer1	Timer 1 Channel 0 Capture/Compare Control
T1CCTL1	0xE6	Timer1	Timer 1 Channel 1 Capture/Compare Control
T1CCTL2	0xE7	Timer1	Timer 1 Channel 2 Capture/Compare Control
T2CT	0x9C	Timer2	Timer 2 Timer Count
T2PR	0x9D	Timer2	Timer 2 Prescaler
T2CTL	0x9E	Timer2	Timer 2 Control
T3CNT	0xCA	Timer3	Timer 3 Counter
T3CTL	0xCB	Timer3	Timer 3 Control
T3CCTL0	0xCC	Timer3	Timer 3 Channel 0 Capture/Compare Control
T3CC0	0xCD	Timer3	Timer 3 Channel 0 Capture/Compare Value
T3CCTL1	0xCE	Timer3	Timer 3 Channel 1 Capture/Compare Control
T3CC1	0xCF	Timer3	Timer 3 Channel 1 Capture/Compare Value
T4CNT	0xEA	Timer4	Timer 4 Counter
T4CTL	0xEB	Timer4	Timer 4 Control
T4CCTL0	0xEC	Timer4	Timer 4 Channel 0 Capture/Compare Control
T4CC0	0xED	Timer4	Timer 4 Channel 0 Capture/Compare Value
T4CCTL1	0xEE	Timer4	Timer 4 Channel 1 Capture/Compare Control
T4CC1	0xEF	Timer4	Timer 4 Channel 1 Capture/Compare Value
TIMIF	0xD8	TMINT	Timers 1/3/4 Joint Interrupt Mask/Flags
U0CSR	0x86	USART0	USART 0 Control and Status
U0DBUF	0xC1	USART0	USART 0 Receive/Transmit Data Buffer
U0BAUD	0xC2	USART0	USART 0 Baud Rate Control
U0UCR	0xC4	USART0	USART 0 UART Control
U0GCR	0xC5	USART0	USART 0 Generic Control

Register name	SFR Address	Module	Description
U1CSR	0xF8	USART1	USART 1 Control and Status
U1DBUF	0xF9	USART1	USART 1 Receive/Transmit Data Buffer
U1BAUD	0xFA	USART1	USART 1 Baud Rate Control
U1UCR	0xFB	USART1	USART 1 UART Control
U1GCR	0xFC	USART1	USART 1 Generic Control
WDCTL	0xC9	WDT	Watchdog Timer Control

12.5 CPU Registers

This section describes the internal registers used by the CPU.

12.5.1 Registers R0-R7

The ***CC1110*** provides four register banks of eight registers each. These register banks are mapped in the DATA memory space at addresses 0x00-0x07, 0x08-0x0F, 0x10-0x17 and 0x18-0x1F. Each register bank contains the eight 8-bit register `R0-R7`. The register bank to be used is selected through the Program Status Word `PSW.RS[1:0]`.

12.5.2 Program Status Word

The Program Status Word (PSW) contains several bits that show the current state of the CPU. The Program Status Word is accessible as an SFR and it is bit-addressable. `PSW` contains the Carry flag, Auxiliary Carry flag for BCD operations, Register Select bits, Overflow flag and Parity flag. Two bits in `PSW` are uncommitted and can be used as user-defined status flags

.

PSW (0xD0) – Program Status Word

Bit	Name	Reset	R/W	Description
7	CY	0	R/W	Carry flag. Set to 1 when the last arithmetic operation resulted in a carry (during addition) or borrow (during subtraction), otherwise cleared to 0 by all arithmetic operations.
6	AC	0	R/W	Auxiliary carry flag for BCD operations. Set to 1 when the last arithmetic operation resulted in a carry into (during addition) or borrow from (during subtraction) the high order nibble, otherwise cleared to 0 by all arithmetic operations.
5	F0	0	R/W	User-defined, bit-addressable
4:3	RS[1:0]	00	R/W	Register bank select bits. Selects which set of R7-R0 registers to use from four possible register banks in DATA space. 00 Bank 0, 0x00 – 0x07 01 Bank 1, 0x08 – 0x0F 10 Bank 2, 0x10 – 0x17 11 Bank 3, 0x18 – 0x1F
2	OV	0	R/W	Overflow flag, set by arithmetic operations. Set to 1 when the last arithmetic operation resulted in a carry (addition), borrow (subtraction), or overflow (multiply or divide). Otherwise, the bit is cleared to 0 by all arithmetic operations.
1	F1	0	R/W	User-defined, bit-addressable
0	P	0	R/W	Parity flag, parity of accumulator set by hardware to 1 if it contains an odd number of 1's, otherwise it is cleared to 0

12.5.3 Accumulator

ACC is the accumulator. This is the source and destination of most arithmetic instructions, data transfer and other instructions. The mnemonic for the accumulator (in instructions involving the accumulator) refers to A instead of ACC.

ACC (0xE0) – Accumulator

Bit	Name	Reset	R/W	Description
7:0	ACC[7:0]	0x00	R/W	Accumulator

12.5.4 B Register

The B register is used as the second 8-bit argument during execution of multiply and divide instructions. When not used for these purposes it may be used as a scratch-pad register to hold temporary data.

B (0xF0) – B Register

Bit	Name	Reset	R/W	Description
7:0	B[7:0]	0x00	R/W	B register. Used in MUL/DIV instructions.

12.5.5 Stack Pointer

The stack resides in DATA memory space and grows upwards. The PUSH instruction first increments the Stack Pointer (SP) and then copies the byte into the stack. The Stack

Pointer is initialized to 0x07 after a reset and it is incremented once to start from location 0x08, which is the first register (R0) of the second register bank. Thus, in order to use more than one register bank, the SP should be initialized to a different location not used for data storage.

SP (0x81) – Stack Pointer

Bit	Name	Reset	R/W	Description
7:0	SP[7:0]	0x07	R/W	Stack Pointer

12.6 Instruction Set Summary

The 8051 instruction set is summarized in Table 27. All mnemonics copyrighted © Intel Corporation 1980.

The following conventions are used in the instruction set summary:

- Rn – Register R7-R0 of the currently selected register bank.
- direct – 8-bit internal data location's address. This can be DATA area (0x00 – 0x7F) or SFR area (0x80 – 0xFF).
- @Ri – 8-bit internal data location, DATA area (0x00 – 0xFF) addressed indirectly through register R1 or R0.
- #data – 8-bit constant included in instruction.
- #data16 – 16-bit constant included in instruction.
- addr16 – 16-bit destination address. Used by LCALL and LJMP. A valid branch can be anywhere within the 32 KB CODE program memory space.
- addr11 – 11-bit destination address. Used by ACALL and AJMP. The branch will be within the same 2 KB page of program memory as the first byte of the following instruction.
- rel – Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is –128 to +127 bytes relative to first byte of the following instruction.
- bit – direct addressed bit in DATA area or SFR.

The instructions that affect CPU flag settings located in PSW are listed in Table 28 on page 45. Note that operations on the PSW or bits in PSW will also affect the flag settings.

Table 27. Instruction Set Summary

Mnemonic	Description	Hex Opcode	Bytes	Cycles
Arithmetic operations				
ADD A,Rn	Add register to accumulator	28-2F	1	1
ADD A,direct	Add direct byte to accumulator	25	2	2
ADD A,@Ri	Add indirect RAM to accumulator	26-27	1	2
ADD A,#data	Add immediate data to accumulator	24	2	2
ADDC A,Rn	Add register to accumulator with carry flag	38-3F	1	1
ADDC A,direct	Add direct byte to A with carry flag	35	2	2
ADDC A,@Ri	Add indirect RAM to A with carry flag	36-37	1	2
ADDC A,#data	Add immediate data to A with carry flag	34	2	2
SUBB A,Rn	Subtract register from A with borrow	98-9F	1	1

Mnemonic	Description	Hex Opcode	Bytes	Cycles
SUBB A,direct	Subtract direct byte from A with borrow	95	2	2
SUBB A,@Ri	Subtract indirect RAM from A with borrow	96-97	1	2
SUBB A,#data	Subtract immediate data from A with borrow	94	2	2
INC A	Increment accumulator	04	1	1
INC Rn	Increment register	08-0F	1	2
INC direct	Increment direct byte	05	2	3
INC @Ri	Increment indirect RAM	06-07	1	3
INC DPTR	Increment data pointer	A3	1	1
DEC A	Decrement accumulator	14	1	1
DEC Rn	Decrement register	18-1F	1	2
DEC direct	Decrement direct byte	15	2	3
DEC @Ri	Decrement indirect RAM	16-17	1	3
MUL AB	Multiply A and B	A4	1	5
DIV	Divide A by B	84	1	5
DA A	Decimal adjust accumulator	D4	1	1
Logical operations				
ANL A,Rn	AND register to accumulator	58-5F	1	1
ANL A,direct	AND direct byte to accumulator	55	2	2
ANL A,@Ri	AND indirect RAM to accumulator	56-57	1	2
ANL A,#data	AND immediate data to accumulator	54	2	2
ANL direct,A	AND accumulator to direct byte	52	2	3
ANL direct,#data	AND immediate data to direct byte	53	3	4
ORL A,Rn	OR register to accumulator	48-4F	1	1
ORL A,direct	OR direct byte to accumulator	45	2	2
ORL A,@Ri	OR indirect RAM to accumulator	46-47	1	2
ORL A,#data	OR immediate data to accumulator	44	2	2
ORL direct,A	OR accumulator to direct byte	42	2	3
ORL direct,#data	OR immediate data to direct byte	43	3	4
XRL A,Rn	Exclusive OR register to accumulator	68-6F	1	1
XRL A,direct	Exclusive OR direct byte to accumulator	65	2	2
XRL A,@Ri	Exclusive OR indirect RAM to accumulator	66-67	1	2
XRL A,#data	Exclusive OR immediate data to accumulator	64	2	2
XRL direct,A	Exclusive OR accumulator to direct byte	62	2	3
XRL direct,#data	Exclusive OR immediate data to direct byte	63	3	4
CLR A	Clear accumulator	E4	1	1
CPL A	Complement accumulator	F4	1	1
RL A	Rotate accumulator left	23	1	1
RLC A	Rotate accumulator left through carry	33	1	1
RR A	Rotate accumulator right	03	1	1
RRC A	Rotate accumulator right through carry	13	1	1

Mnemonic	Description	Hex Opcode	Bytes	Cycles
SWAP A	Swap nibbles within the accumulator	C4	1	1
Data transfers				
MOV A,Rn	Move register to accumulator	E8-EF	1	1
MOV A,direct	Move direct byte to accumulator	E5	2	2
MOV A,@Ri	Move indirect RAM to accumulator	E6-E7	1	2
MOV A,#data	Move immediate data to accumulator	74	2	2
MOV Rn,A	Move accumulator to register	F8-FF	1	2
MOV Rn,direct	Move direct byte to register	A8-AF	2	4
MOV Rn,#data	Move immediate data to register	78-7F	2	2
MOV direct,A	Move accumulator to direct byte	F5	2	3
MOV direct,Rn	Move register to direct byte	88-8F	2	3
MOV direct1,direct2	Move direct byte to direct byte	85	3	4
MOV direct,@Ri	Move indirect RAM to direct byte	86-87	2	4
MOV direct,#data	Move immediate data to direct byte	75	3	3
MOV @Ri,A	Move accumulator to indirect RAM	F6-F7	1	3
MOV @Ri,direct	Move direct byte to indirect RAM	A6-A7	2	5
MOV @Ri,#data	Move immediate data to indirect RAM	76-77	2	3
MOV DPTR,#data16	Load data pointer with a 16-bit constant	90	3	3
MOVC A,@A+DPTR	Move code byte relative to DPTR to accumulator	93	1	3
MOVC A,@A+PC	Move code byte relative to PC to accumulator	83	1	3
MOVX A,@Ri	Move external RAM (8-bit address) to A	E2-E3	1	3-10
MOVX A,@DPTR	Move external RAM (16-bit address) to A	E0	1	3-10
MOVX @Ri,A	Move A to external RAM (8-bit address)	F2-F3	1	4-11
MOVX @DPTR,A	Move A to external RAM (16-bit address)	F0	1	4-11
PUSH direct	Push direct byte onto stack	C0	2	4
POP direct	Pop direct byte from stack	D0	2	3
XCH A,Rn	Exchange register with accumulator	C8-CF	1	2
XCH A,direct	Exchange direct byte with accumulator	C5	2	3
XCH A,@Ri	Exchange indirect RAM with accumulator	C6-C7	1	3
XCHD A,@Ri	Exchange low-order nibble indirect. RAM with A	D6-D7	1	3
Program branching				
ACALL addr11	Absolute subroutine call	xxx11	2	6
LCALL addr16	Long subroutine call	12	3	6
RET	Return from subroutine	22	1	4
RETI	Return from interrupt	32	1	4
AJMP addr11	Absolute jump	xxx01	2	3
LJMP addr16	Long jump	02	3	4
SJMP rel	Short jump (relative address)	80	2	3
JMP @A+DPTR	Jump indirect relative to the DPTR	73	1	2
JZ rel	Jump if accumulator is zero	60	2	3

Mnemonic	Description	Hex Opcode	Bytes	Cycles
JNZ rel	Jump if accumulator is not zero	70	2	3
JC rel	Jump if carry flag is set	40	2	3
JNC	Jump if carry flag is not set	50	2	3
JB bit,rel	Jump if direct bit is set	20	3	4
JNB bit,rel	Jump if direct bit is not set	30	3	4
JBC bit,direct rel	Jump if direct bit is set and clear bit	10	3	4
CJNE A,direct rel	Compare direct byte to A and jump if not equal	B5	3	4
CJNE A,#data rel	Compare immediate to A and jump if not equal	B4	3	4
CJNE Rn,#data rel	Compare immediate to reg. and jump if not equal	B8-BF	3	4
CJNE @Ri,#data rel	Compare immediate to indirect and jump if not equal	B6-B7	3	4
DJNZ Rn,rel	Decrement register and jump if not zero	D8-DF	2	3
DJNZ direct,rel	Decrement direct byte and jump if not zero	D5	3	4
NOP	No operation	00	1	1
Boolean variable operations				
CLR C	Clear carry flag	C3	1	1
CLR bit	Clear direct bit	C2	2	3
SETB C	Set carry flag	D3	1	1
SETB bit	Set direct bit	D2	2	3
CPL C	Complement carry flag	B3	1	1
CPL bit	Complement direct bit	B2	2	3
ANL C,bit	AND direct bit to carry flag	82	2	2
ANL C,/bit	AND complement of direct bit to carry	B0	2	2
ORL C,bit	OR direct bit to carry flag	72	2	2
ORL C,/bit	OR complement of direct bit to carry	A0	2	2
MOV C,bit	Move direct bit to carry flag	A2	2	2
MOV bit,C	Move carry flag to direct bit	92	2	3

Instruction	CY	OV	AC
ADD	x	x	x
ADDC	x	x	x
SUBB	x	x	x
MUL	0	x	
DIV	0	x	
DA	x		
RRC	x		
RLC	x		
SETB C	1		
CLR C	x		
CPL C	x		
ANL C,bit	x		
ANL C,/bit	x		
ORL C,bit	x		
ORL C,/bit	x		
MOV C,bit	x		
CJNE	x		

"0"=set to 0, "1"=set to 1, "x"=set to 0/1, "-"=not affected

Table 28: Instructions that affect flag settings

12.7 Interrupts

The CPU has 18 interrupt sources. Each source has its own request flag located in a set of Interrupt Flag SFR registers. Each interrupt requested by the corresponding flag can be individually enabled or disabled. The definitions of the interrupt sources and the interrupt vectors are given in Table 29.

The interrupts are grouped into a set of priority level groups with selectable priority levels.

The interrupt enable registers are described in section 12.7.1 and the interrupt priority settings are described in section 12.7.3 on page 53.

12.7.1 Interrupt Masking

Each interrupt can be individually enabled or disabled by the interrupt enable bits in the Interrupt Enable SFRs IEN0, IEN1 and IEN2. The Interrupt Enable SFRs are described below and summarized in Table 29.

Note that some peripherals have several events that can generate the interrupt request associated with that peripheral. This applies to Port 0, Port 1, Port 2, DMA, Timer 1, Timer 3 , Timer 4 and Radio. These peripherals have interrupt mask bits for each internal interrupt source in the corresponding SFR registers.

In order to use any of the interrupts in the CC1110 the following steps must be taken

1. Enable global interrupt by setting the EA bit in IEN0 to 1
2. Set the corresponding individual interrupt enable bit in the IEN0, IEN1 or IEN2 register to 1.
3. Set individual interrupt enable bit in the peripherals SFR register, if any.
4. Begin the interrupt service routine at the corresponding vector address of that interrupt. See Table 29 for addresses.

Interrupt number	Description	Interrupt name	Interrupt Vector	Interrupt Mask	Interrupt Flag
0	RF TX done / RX ready	RFTXRX	03h	IEN0.RFTXRXIE	TCON.RFTXRXIF
1	ADC end of conversion	ADC	0Bh	IEN0.ADIE	TCON.ADIF
2	USART0 RX complete	URX0	13h	IEN0.URX0IE	TCON.URX0IF
3	USART1 RX complete	URX1	1Bh	IEN0.URX1IE	TCON.URX1IF
4	AES encryption/decryption complete	ENC	23h	IEN0.ENCIE	S0CON.ENCIF
5	Sleep Timer compare	ST	2Bh	IEN0.STIE	IRCON.STIF
6	Port 2 inputs	P2INT	33h	IEN2.P2IE	IRCON2.P2IF
7	USART0 TX complete	UTX0	3Bh	IEN2.UTX0IE	IRCON2.UTX0IF
8	DMA transfer complete	DMA	43h	IEN1.DMAIE	IRCON.DMAIF
9	Timer 1 (16-bit) capture/Compare/overflow	T1	4Bh	IEN1.T1IE	IRCON.T1IF
10	Timer 2 (MAC Timer) overflow	T2	53h	IEN1.T2IE	IRCON.T2IF
11	Timer 3 (8-bit) capture/compare/overflow	T3	5Bh	IEN1.T3IE	IRCON.T3IF
12	Timer 4 (8-bit) capture/compare/overflow	T4	63h	IEN1.T4IE	IRCON.T4IF
13	Port 0 inputs	P0INT	6Bh	IEN1.P0IE	IRCON.P0IF
14	USART1 TX complete	UTX1	73h	IEN2.UTX1IE	IRCON2.UTX1IF
15	Port 1 inputs	P1INT	7Bh	IEN2.P1IE	IRCON2.P1IF
16	RF general interrupts	RF	83h	IEN2.RFIE	S1CON.RFIF
17	Watchdog overflow in timer mode	WDT	8Bh	IEN2.WDTIE	IRCON2.WDTIF

Table 29: Interrupts Overview

IEN0 (0xA8) – Interrupt Enable 0 Register

Bit	Name	Reset	R/W	Description
7	EA	0	R/W	Disables all interrupts. 0 No interrupt will be acknowledged 1 Each interrupt source is individually enabled or disabled by setting its corresponding enable bit
6	–	0	R0	Not used. Read as 0
5	STIE	0	R/W	STIE – Sleep Timer interrupt enable 0 Interrupt disabled 1 Interrupt enabled
4	ENCIE	0	R/W	ENCIE – AES encryption/decryption interrupt enable 0 Interrupt disabled 1 Interrupt enabled
3	URX1IE	0	R/W	URX1IE– USART1 RX interrupt enable 0 Interrupt disabled 1 Interrupt enabled
2	URX0IE	0	R/W	URX0IE - USART0 RX interrupt enable 0 Interrupt disabled 1 Interrupt enabled
1	ADCIE	0	R/W	ADCIE – ADC interrupt enable 0 Interrupt disabled 1 Interrupt enabled
0	RFTXRXIE	0	R/W	RFRXTXIE – RF TX/RX done interrupt enable 0 Interrupt disabled 1 Interrupt enabled

IEN1 (0xB8) – Interrupt Enable 1 Register

Bit	Name	Reset	R/W	Description
7:6	–	00	R0	Not used. Read as 0
5	P0IE	0	R/W	P0IE – Port 0 interrupt enable 0 Interrupt disabled 1 Interrupt enabled
4	T4IE	0	R/W	T4IE - Timer 4 interrupt enable 0 Interrupt disabled 1 Interrupt enabled
3	T3IE	0	R/W	T3IE - Timer 3 interrupt enable 0 Interrupt disabled 1 Interrupt enabled
2	T2IE	0	R/W	T2IE – Timer 2 interrupt enable 0 Interrupt disabled 1 Interrupt enabled
1	T1IE	0	R/W	T1IE – Timer 1 interrupt enable 0 Interrupt disabled 1 Interrupt enabled
0	DMAIE	0	R/W	DMAIE – DMA transfer interrupt enable 0 Interrupt disabled 1 Interrupt enabled

IEN2 (0x9A) – Interrupt Enable 2 Register

Bit	Name	Reset	R/W	Description
7:6	–	00	R0	Not used. Read as 0
5	WDTIE	0	R/W	WDTIE – Watchdog timer interrupt enable 0 Interrupt disabled 1 Interrupt enabled
4	P1IE	0	R/W	P1IE – Port 1 interrupt enable 0 Interrupt disabled 1 Interrupt enabled
3	UTX1IE	0	R/W	UTX1IE – USART1 TX interrupt enable 0 Interrupt disabled 1 Interrupt enabled
2	UTX0IE	0	R/W	UTX0IE - USART0 TX interrupt enable 0 Interrupt disabled 1 Interrupt enabled
1	P2IE	0	R/W	P2IE – Port 2 interrupt enable 0 Interrupt disabled 1 Interrupt enabled
0	RFIE	0	R/W	RFIE – RF general interrupt enable 0 Interrupt disabled 1 Interrupt enabled

12.7.2 Interrupt Processing

When an interrupt occurs, the CPU will vector to the interrupt vector address as shown in Table 29. Once an interrupt service has begun, it can be interrupted only by a higher priority interrupt. The interrupt service is terminated by a RETI return from interrupt instruction. When a RETI is performed, the CPU will return to the instruction that would have been next when the interrupt occurred.

When the interrupt condition occurs, the CPU will also indicate this by setting an interrupt flag bit in the interrupt flag registers. This bit is set regardless of whether the interrupt is enabled or disabled. If the interrupt is enabled when an interrupt flag is set, then on the next instruction cycle the interrupt will be acknowledged by hardware forcing an LCALL to the appropriate vector address.

Interrupt response will require a varying amount of time depending on the state of the CPU when the interrupt occurs. If the CPU is performing an interrupt service with equal or greater priority, the new interrupt will be pending until it becomes the interrupt with highest priority. In other cases, the response time depends on current instruction. The fastest possible response to an interrupt is seven instruction cycles. This includes one instruction cycle for detecting the interrupt and six cycles to perform the LCALL.

TCON (0x88) – Interrupt Flag

Bit	Name	Reset	R/W	Description
7	URX1IF	0	R/W H0	URX1IF – USART1 RX interrupt flag. Set to 1 when USART1 RX interrupt occurs and cleared when CPU vectors to the interrupt service routine. 0 Interrupt not pending 1 Interrupt pending
6	–	0	R/W	Not used
5	ADIF	0	R/W H0	ADIF – ADC interrupt flag. Set to 1 when ADC interrupt occurs and cleared when CPU vectors to the interrupt service routine. 0 Interrupt not pending 1 Interrupt pending
4	–	0	R/W	Not used
3	URX0IF	0	R/W H0	URX0IF – USART0 RX interrupt flag. Set to 1 when USART0 interrupt occurs and cleared when CPU vectors to the interrupt service routine. 0 Interrupt not pending 1 Interrupt pending
2	IT1	1	R/W	Reserved. Must always be set to 1.
1	RFTXRXIF	0	R/W H0	RFTXRXIF – RF TX/RX complete interrupt flag. Set to 1 when RFTXRX interrupt occurs and cleared when CPU vectors to the interrupt service routine. 0 Interrupt not pending 1 Interrupt pending
0	IT0	1	R/W	Reserved. Must always be set to 1.

S0CON (0x98) – Interrupt Flag 2

Bit	Name	Reset	R/W	Description
7:6	–	0	R/W	Not used
1	ENCIF_1	0	R/W	ENCIF – AES interrupt. ENCIF has two interrupt flags, ENCIF_1 and ENCIF_0. Setting one of these flags will request interrupt service. Both flags are set when the AES co-processor requests the interrupt. 0 Interrupt not pending 1 Interrupt pending
0	ENCIF_0	0	R/W	ENCIF – AES interrupt. ENCIF has two interrupt flags, ENCIF_1 and ENCIF_0. Setting one of these flags will request interrupt service. Both flags are set when the AES co-processor requests the interrupt. 0 Interrupt not pending 1 Interrupt pending

S1CON (0x9B) – Interrupt Flag 3

Bit	Name	Reset	R/W	Description
7:6	–	0	R/W	Not used
1	RFIF_1	0	R/W	RFIF – RF general interrupt. RFIF has two interrupt flags, RFIF_1 and RFIF_0. Setting one of these flags will request interrupt service. Both flags are set when the radio requests the interrupt. 0 Interrupt not pending 1 Interrupt pending
0	RFIF_0	0	R/W	RFIF – RF general interrupt. RFIF has two interrupt flags, RFIF_1 and RFIF_0. Setting one of these flags will request interrupt service. Both flags are set when the radio requests the interrupt. 0 Interrupt not pending 1 Interrupt pending

IRCON (0xC0) – Interrupt Flag 4

Bit	Name	Reset	R/W	Description
7	STIF	0	R/W	STIF – Sleep timer interrupt flag 0 Interrupt not pending 1 Interrupt pending
6	–	0	R/W	Not used
5	P0IF	0	R/W	P0IF – Port 0 interrupt flag 0 Interrupt not pending 1 Interrupt pending
4	T4IF	0	R/W H0	T4IF – Timer 4 interrupt flag. Set to 1 when Timer 4 interrupt occurs and cleared when CPU vectors to the interrupt service routine. 0 Interrupt not pending 1 Interrupt pending
3	T3IF	0	R/W H0	T3IF – Timer 3 interrupt flag. Set to 1 when Timer 3 interrupt occurs and cleared when CPU vectors to the interrupt service routine. 0 Interrupt not pending 1 Interrupt pending
2	T2IF	0	R/W H0	T2IF – Timer 2 interrupt flag. Set to 1 when Timer 2 interrupt occurs and cleared when CPU vectors to the interrupt service routine. 0 Interrupt not pending 1 Interrupt pending
1	T1IF	0	R/W H0	T1IF – Timer 1 interrupt flag. Set to 1 when Timer 1 interrupt occurs and cleared when CPU vectors to the interrupt service routine. 0 Interrupt not pending 1 Interrupt pending
0	DMAIF	0	R/W	DMAIF – DMA complete interrupt flag. 0 Interrupt not pending 1 Interrupt pending

IRCON2 (0xE8) – Interrupt Flag 5

Bit	Name	Reset	R/W	Description
7:5	–	00	R/W	Not used
4	WDTIF	0	R/W	WDTIF – Watchdog timer interrupt flag. 0 Interrupt not pending 1 Interrupt pending
3	P1IF	0	R/W	P1IF – Port 1 interrupt flag. 0 Interrupt not pending 1 Interrupt pending
2	UTX1IF	0	R/W	UTX1IF – USART1 TX interrupt flag. 0 Interrupt not pending 1 Interrupt pending
1	UTX0IF	0	R/W	UTX0IF – USART0 TX interrupt flag. 0 Interrupt not pending 1 Interrupt pending
0	P2IF	0	R/W	P2IF – Port2 interrupt flag. 0 Interrupt not pending 1 Interrupt pending

12.7.3 Interrupt Priority

The interrupts are grouped into six interrupt priority groups and the priority for each group is set by the registers IP0 and IP1. In order to assign a higher priority to an interrupt, i.e. to its interrupt group, the corresponding bits in IP0 and IP1 must be set as shown in Table 30 on page 54.

The interrupt priority groups with assigned interrupt sources are shown in Table 31. Each group is assigned one of four priority levels. While an interrupt service request is in progress, it cannot be interrupted by a lower or same level interrupt.

In the case when interrupt requests of the same priority level are received simultaneously, the polling sequence shown in Table 32 is used to resolve the priority of each request.

IP1 (0xB9) – Interrupt Priority 1

Bit	Name	Reset	R/W	Description
7:6	–	00	R/W	Not used.
5	IP1_5	0	R/W	Interrupt group 5, priority control bit 1, refer to Table 30
4	IP1_4	0	R/W	Interrupt group 4, priority control bit 1, refer to Table 30
3	IP1_3	0	R/W	Interrupt group 3, priority control bit 1, refer to Table 30
2	IP1_2	0	R/W	Interrupt group 2, priority control bit 1, refer to Table 30
1	IP1_1	0	R/W	Interrupt group 1, priority control bit 1, refer to Table 30
0	IP1_0	0	R/W	Interrupt group 0, priority control bit 1, refer to Table 30

IP0 (0xA9) – Interrupt Priority 0

Bit	Name	Reset	R/W	Description
7:6	–	00	R/W	Not used.
5	IP0_5	0	R/W	Interrupt group 5, priority control bit 0, refer to Table 30
4	IP0_4	0	R/W	Interrupt group 4, priority control bit 0, refer to Table 30
3	IP0_3	0	R/W	Interrupt group 3, priority control bit 0, refer to Table 30
2	IP0_2	0	R/W	Interrupt group 2, priority control bit 0, refer to Table 30
1	IP0_1	0	R/W	Interrupt group 1, priority control bit 0, refer to Table 30
0	IP0_0	0	R/W	Interrupt group 0, priority control bit 0, refer to Table 30

IP1_x	IP0_x	Priority Level
0	0	0 – lowest
0	1	1
1	0	2
1	1	3 – highest

Table 30: Priority Level Setting

Group	Interrupts		
IP0	RFTXRX	RF	DMA
IP1	ADC	P2	T1
IP2	URX0	UTX0	T2
IP3	URX1	UTX1	T3
IP4	ENC	P1INT	T4
IP5	ST	WDT	P0INT

Table 31: Interrupt Priority Groups

Interrupt number	Interrupt name	
0	RFTXRX	Polling sequence ↓
16	RF	
8	DMA	
1	ADC	
9	T1	
2	URX0	
10	T2	
3	URX1	
11	T3	
4	ENC	
12	T4	
5	ST	
13	P0INT	
6	P2INT	
7	UTX0	
14	UTX1	
15	P1INT	
17	WDT	

Table 32: Interrupt Polling Sequence

12.8 Oscillators and clocks

The *CC1110* has one internal system clock. The source for the system clock can be either a 13 MHz high speed RC oscillator or a 26 MHz crystal oscillator. Clock control is performed using the CLKCON SFR register described in section 13.10.

The choice of oscillator allows a trade-off between high-accuracy in the case of the crystal oscillator and low power consumption when the high-frequency RC oscillator is used. Note that operation of the RF transceiver requires that the crystal oscillator is used.

12.9 Debug Interface

The *CC1110* includes a debug interface that provides a two-wire interface to an on-chip debug module. The debug interface allows programming the on-chip flash as well providing access to memory and register contents and debug features such as breakpoints, single-stepping and register modification.

The debug interface uses the I/O pins P2_1 as Debug Data and P2_2 as Debug Clock during Debug mode. These I/O pins can be used as general purpose I/O only while the device is not in Debug mode. Thus, the debug interface does not interfere with any peripheral I/O pins.

12.9.1 Debug Mode

Debug mode is entered by forcing two rising edge transitions on pin P2_2 (Debug Clock) while the RESET_N input is held low.

While in Debug mode, pin P2_1 is the Debug Data bi-directional pin and P2_2 is the Debug Clock input pin.

12.9.2 Debug Communication

The debug interface uses an SPI-like two-wire interface consisting of the Debug Data (P2_1) and Debug Clock (P2_2) pins. Data is driven

on the bi-directional Debug Data pin at the positive edge of Debug Clock and data is sampled on the negative edge of this clock.

Debug commands are sent by an external host and consist of 1 to 4 output bytes (including command byte) from the host and an optional input byte read by the host. Figure 9 shows a timing diagram of data on the debug interface.

The first byte of the debug command is a command byte and is encoded as follows:

- bits 7 to 3 : instruction code
- bit 2 : return input byte to host when high
- bits 1 to 0 : number of output bytes from host following instruction code byte

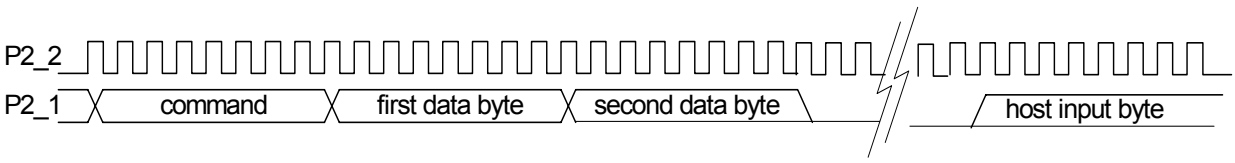


Figure 9: Debug interface timing diagram

12.9.3 Debug Commands

The debug commands are shown in Table 33. Some of the debug commands are described in further detail in the following sections

12.9.4 Debug Lock Bit

For software code security, the Debug Interface may be locked. When the Debug Lock bit , DBGLOCK, is set (see section 13.14.3) all debug commands except CHIP_ERASE, READ_STATUS and GET_CHIP_ID are disabled and will not function.

The CHIP_ERASE command is used to clear the Debug Lock bit.

12.9.5 Debug Configuration

The commands WR_CONFIG and RD_CONFIG are used to access the debug configuration data byte. The format and description of this configuration data is shown in Table 34.

12.9.6 Debug Status

A Debug status byte is read using the READ_STATUS command. The format and description of this debug status is shown in Table 35.

The READ_STATUS command is used e.g. for polling the status of flash chip erase after a CHIP_ERASE command or oscillator stable status required for debug commands HALT, RESUME, DEBUG_INSTR, STEP_REPLACE and STEP_INSTR.

Command	Instruction code	Description
CHIP_ERASE	0001 0x00	Perform flash chip erase (mass erase) and clear lock bits. If any other command, except READ_STATUS, is issued, then the use of CHIP_ERASE is disabled.
WR_CONFIG	0001 1x01	Write configuration data. Refer to Table 34
RD_CONFIG	0010 0100	Read configuration data. Returns value set by WR_CONFIG command.
GET_PC	0010 1000	Return value of 16-bit program counter. Returns 2 bytes regardless of value of bit 2 in instruction code
READ_STATUS	0011 0x00	Read status byte. Refer to Table 35
SET_HW_BRKPNT	0011 1x11	Set hardware breakpoint
HALT	0100 0100	Halt CPU operation
RESUME	0100 1100	Resume CPU operation. The CPU must be in halted state for this command to be run.
DEBUG_INSTR	0101 01xx	Run debug instruction. The supplied instruction will be executed by the CPU without incrementing the program counter. The CPU must be in halted state for this command to be run.
STEP_INSTR	0101 1100	Step CPU instruction. The CPU will execute the next instruction from program memory and increment the program counter after execution. The CPU must be in halted state for this command to be run.
STEP_REPLACE	0110 01xx	Step and replace CPU instruction. The supplied instruction will be executed by the CPU instead of the next instruction in program memory. The program counter will be incremented after execution. The CPU must be in halted state for this command to be run.
GET_CHIP_ID	0110 1000	Return value of 16-bit chip ID and version number. Returns 2 bytes regardless of value of bit 2 of instruction code

Table 33: Debug Commands

Bit	Name	Description
7-4	–	Not used
3	timers_off	Disable timers. Disable timer operation 0 Do not disable timers 1 Disable timers
2	DMA_pause	DMA pause 0 Enable DMA transfers 1 Pause all DMA transfers
1	timer_suspend	Suspend timers. Timer operation is suspended for debug instructions and if a step instruction is a branch. If not suspended these instructions would result in an extra timer count during the clock cycle in which the branch is executed 0 Do not suspend timers 1 Suspend timers
0	sel_flash_info_page	Select flash information page in order to write flash lock bits. 0 Select flash main page 1 Select flash information page

Table 34: Debug Configuration

Bit	Name	Description
7	chip_erase_done	Flash chip erase done 0 Chip erase in progress 1 Chip erase done
6	pcon_idle	PCON idle 0 CPU is running 1 CPU is idle (clock gated)
5	cpu_halted	CPU halted 0 CPU running 1 CPU halted
4	power_mode_0	Power Mode 0 0 Power Mode 1-3 selected 1 Power Mode 0 selected
3	halt_status	Halt status. Returns cause of last CPU halt 0 CPU was halted by HALT debug command 1 CPU was halted by software or hardware breakpoint
2	debug_locked	Debug locked. Returns value of DBGLOCK bit 0 Debug interface is not locked 1 Debug interface is locked
1	oscillator_stable	Oscillators stable. This bit represents the status of the CLKCON.XSOC_STB and CLKCON.HFRC_STB register bits. 0 Oscillators not stable 1 Oscillators stable
0	stack_overflow	Stack overflow. This bit indicates when the CPU writes to DATA memory space at address 0xFF which is possibly a stack overflow 0 No stack overflow 1 Stack overflow

Table 35: Debug Status

12.9.7 Hardware Breakpoints

The debug command SET_HW_BRKPNT is used to set a hardware breakpoint. The ***CC1110*** supports up to four hardware breakpoints. When a hardware breakpoint is enabled, it will compare the CPU address bus with the breakpoint.. When a match occurs, the CPU is halted.

When issuing the SET_HW_BRKPNT, the external host must supply three data bytes that define the hardware breakpoint. The hardware breakpoint itself consists of 18 bits while three bits are used for control purposes. The format of the three data bytes for the SET_HW_BRKPNT command is as follows.

The first data byte consists of the following:

- bits 7-5 : unused
- bits 4-3 : breakpoint number; 0-3
- bit 2 : 1=enable, 0=disable
- bits 1-0 : Memory bank bits. Bits 17-16 of hardware breakpoint.

The second data byte consists of bits 15-8 of the hardware breakpoint.

The third data byte consists of bits 7-0 of the hardware breakpoint.

12.9.8 Flash Programming

Programming of the on-chip flash is performed via the debug interface. The external host must initially send instructions using the DEBUG_INSTR debug command to perform the flash programming with the Flash Controller as described in section 13.14 on page 147.

12.10 RAM

The ***CC1110*** contains static RAM. At power-on the contents of RAM is undefined. The RAM size is 4 KB in total. The RAM retains data in all power modes.

The memory locations 0xFDAA-0xFEFF consisting of 342 bytes in XDATA memory do not retain data when power modes PM2/3 is entered.

12.11 Flash Memory

The on-chip flash memory consists of 32768 bytes. The flash memory is primarily intended to hold program code. The flash memory has the following features:

- Flash page erase time: 20 ms
- Flash chip (mass) erase time: 20 ms
- Flash write time: 20 µs
- Data retention[1]:100 years
- Program/erase endurance: 1,000 cycles

The flash memory consists of the Flash Main Page which is where the CPU reads program code and data. The flash memory also contains a Flash Information Page which contains the Flash Lock Bits. The Flash Information Page and hence the Lock Bits is only accessed by first selecting this page through the Debug Interface. The Flash Controller (see section 13.14) is used to write and erase the contents of the flash memory.

When the CPU reads instructions from flash memory, it fetches the next instruction through a cache. The instruction cache is provided mainly to reduce power consumption by reducing the amount of time the flash memory itself is accessed. The use of the instruction cache may be disabled with the `MEMCTR.CACHDIS` register bit.

12.12 Memory Arbiter

The ***CC1110*** includes a memory arbiter which handles CPU and DMA access to all memory space.

A control register `MEMCTR` is used to control various aspects of the memory sub-system. The `MEMCTR` register is described in the following.

[1] At room temperature

MEMCTR (0xC7) – Memory Arbiter Control

Bit	Name	Reset	R/W	Description
7:6	–	00	R0	Not used
5:4	–	01	R/W	Not used
3:2	–	00	R0	Not used
1	CACHDIS	0	R/W	Flash cache disable. Invalidates contents of instruction cache and forces all instruction read accesses to read straight from flash memory. Disabling will increase power consumption and is provided for debug purposes. 0 Cache enabled 1 Cache disabled
0	–	1	R/W	Reserved. Must always be set to 1.

13 Peripherals

In the following sub-sections, each CC1110 peripheral is described in detail.

The CC1110 has four timers. These timers all run on the tick frequency given by the Power Management Controller register CLKCON.TICKSPD.

13.1 I/O ports

The CC1110 has 21 digital input/output pins that can be configured as general purpose digital I/O or as peripheral I/O signals connected to the ADC, Timers or USART peripherals. The usage of the I/O ports is fully configurable from user software through a set of configuration registers.

The I/O ports have the following key features:

- 21 digital input/output pins
- General purpose I/O or peripheral I/O
- Pull-up or pull-down capability on inputs
- External interrupt capability

The external interrupt capability is available on all 21 I/O pins. Thus, external devices may generate interrupts if required. The external interrupt feature can also be used to wake up from sleep modes.

13.1.1 General Purpose I/O

When used as general purpose I/O, the pins are organized as three 8-bit ports, ports 0-2, denoted P0, P1 and P2. P0 and P1 are complete 8-bit wide ports while P2 has only five usable bits. All ports are both bit- and byte addressable through the SFR registers P0, P1 and P2. Each port pin can individually be set to operate as a general purpose I/O or as a peripheral I/O.

The output drive strength is 4 mA on all outputs, except for the two high-drive outputs, P1_0 and P1_1, which each have 20 mA output drive strength.

To use a port as a general purpose I/O pin the pin must first be configured. The registers PxSEL where x is the port number 0-2 are used to configure each pin in a port either as a general purpose I/O pin or as a peripheral I/O signal. By default, after a reset, all digital input/output pins are configured as general-purpose I/O pins.

By default, all general-purpose I/O pins are configured as inputs. To change the direction of a port pin, at any time, the registers PxDIR are used to set each port pin to be either an input or an output. Thus by setting the appropriate bit within PxDIR to 1, the corresponding pin becomes an output.

When reading the port registers P0, P1 and P2, the logic values on the input pins are returned regardless of the pin configuration. This does not apply during the execution of read-modify-write instructions. The read-modify-write instructions when operating on a port registers are the following: ANL, ORL, XRL, JBC, CPL, INC, DEC, DJNZ and MOV, CLR or SETB, when the destination is an individual bit in a port register P0, P1 or P2. For these read-modify-write instructions, the value of the register, not the value on the pin, is read, modified, and written back to the port register.

When used as an input, the general purpose I/O port pins can be configured to have a pull-up, pull-down or tri-state mode of operation. By default, after a reset, inputs are configured as inputs with pull-up. To deselect the pull-up/pull-down function on an input the appropriate bit within the PxINP must be set to 1. The I/O port pins P1_0 and P1_1 do not have pull-up/pull-down capability.

In power modes PM2 and PM3 the I/O pins retain the I/O mode and output value (if applicable) that was set when PM2/3 was entered.

13.1.2 General Purpose I/O Interrupts

General purpose I/O pins configured as inputs can be used to generate interrupts. The interrupts can be configured to trigger on either a rising or falling edge of the external signal. Each of the P0, P1 and P2 ports have separate interrupt enable bits common for all bits within the port located in the IEN1-2 registers as follows:

- IEN1.P0IE : P0 interrupt enable
- IEN2.P1IE : P1 interrupt enable

- IEN2.P2IE : P2 interrupt enable

In addition to these common interrupt enables, the bits within each port have interrupt enables located in I/O port SFR registers. Each bit within P1 has an individual interrupt enable. In P0 the low-order nibble and the high-order nibble have their individual interrupt enables. For the P2_0 – P2_4 inputs there is a common interrupt enable.

When an interrupt condition occurs on one of the general purpose I/O pins, the corresponding interrupt status flag in the P0-P2 interrupt flag registers, P0IFG , P1IFG or P2IFG will be set to 1. The interrupt status flag is set regardless of whether the pin has its interrupt enable set. When an interrupt is serviced the interrupt status flag is cleared by writing a 0 to that flag.

Note that when clearing the PxIFG interrupt status flags, only one active flag should be cleared at a time. Failure to do this may result in generation of false interrupt requests.

The SFR registers used for I/O interrupts are described in section 12.7 on page 45. The registers are summarized below:

- P1IEN : P1 interrupt enables
- PICTL : P0/P2 interrupt enables and P0-2 edge configuration
- P0IFG : P0 interrupt flags
- P1IFG : P1 interrupt flags
- P2IFG : P2 interrupt flags

13.1.3 General Purpose I/O DMA

When used as general purpose I/O pins, the P0 and P1 ports are each associated with one DMA trigger. These DMA triggers are IOC_0 for P0 and IOC_1 for P1 as shown in Table 37 on page 84.

The IOC_0 or IOC_1 DMA trigger is activated when an input transition occurs on one of the P0 or P1 pins respectively. Note that only input transitions on pins configured as general purpose I/O inputs, will produce the DMA trigger.

13.1.4 Peripheral I/O

This section describes how the digital input/output pins are configured as peripheral I/Os. For each peripheral unit that can interface with an external system through the digital input/output pins, a description of how peripheral I/Os are configured is given in the following sub-sections.

In general, setting the appropriate PxSEL bits to 1 is required to select peripheral I/O function on a digital I/O pin.

Note that peripheral units have two alternative locations for their I/O pins, refer to Table 36. The location to be used is selected by writing to PERCFG.

It is possible to set PERCFG so that several peripherals are assigned to the same port pins. In such cases a set of peripheral priority control bits select the order of precedence between up to two peripherals at a time, when these are assigned to the same port pins.

Table 36: Peripheral I/O Pin Mapping

Periphery / Function	P0								P1								P2				
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	4	3	2	1	0
ADC	A7	A6	A5	A4	A3	A2	A1	A0													
USART0 Alt 1			C	SS	M0	MI															
SPI Alt. 2											MO	MI	C	SS							
USART0 Alt. 1			RT	CT	TX	RX															
UART Alt. 2											TX	RX	RT	CT							
USART1 Alt.1			MI	M0	C	SS															
SPI Alt. 2									MI	M0	C	SS									
USART1 Alt. 1			RX	TX	RT	CT															
UART Alt. 2									RX	TX	RT	CT									
TIMER1 Alt.1				2	1	0															
Alt. 2														0	1	2					
TIMER3 Alt.1												1	0								
Alt. 2									1	0											
TIMER4 Alt.1															1	0					
Alt. 2																		1			0
32.768kHz XOSC																	Q2	Q1			
DEBUG																			DC	DD	

13.1.4.1 USART0

The SFR register bit `PERCFG.U0CFG` selects whether to use alternative 1 or alternative 2 locations.

In Table 36, the USART0 signals are shown as follows:

UART:

- RX : RXDATA
- TX : TXDATA
- RT : RTS
- CT : CTS

SPI:

- MI : MISO
- MO : MOSI
- C : SCK
- SS : SSN

`P2DIR.PRIP0` selects the order of precedence when assigning several peripherals to port 0, i.e. the situation when several peripherals are assigned to the same pin locations. When set to 00, USART0 has precedence. Note that if UART mode is selected and hardware flow control is disabled, USART1 or timer 1 will have precedence to use ports P0_4 and P0_5.

`P2SEL.PRI3P1` and `P2SEL.PRI0P1` select the order of precedence when assigning several peripherals to port 1. USART0 has precedence when both are set to 0. Note that if UART mode is selected and hardware flow control is disabled, timer 1 or timer 3 will have precedence to use ports P1_2 and P1_3.

13.1.4.2 USART1

The SFR register bit `PERCFG.U1CFG` selects whether to use alternative 1 or alternative 2 locations.

In Table 36, the USART1 signals are shown as follows:

- RX : RXDATA

- TX : TXDATA
- RT : RTS
- CT : CTS

SPI:

- MI : MISO
- MO : MOSI
- C : SCK
- SS : SSN

P2DIR.PRIP0 selects the order of precedence when assigning several peripherals to port 0. When set to 01, USART1 has precedence. Note that if UART mode is selected and hardware flow control is disabled, USART0 or timer 1 will have precedence to use ports P0_2 and P0_3.

P2SEL.PRI3P1 and P2SEL.PRI2P1 select the order of precedence when assigning several peripherals to port 1. USART1 has precedence when the former is set to 1 and the latter is set to 0. Note that if UART mode is selected and hardware flow control is disabled, USART0 or timer 3 will have precedence to use ports P2_4 and P2_5.

13.1.4.3 Timer 1

PERCFG.T1CFG selects whether to use alternative 1 or alternative 2 locations.

In Table 36, the Timer 1 signals are shown as the following:

- 0 : Channel 0 capture/compare pin
- 1 : Channel 1 capture/compare pin
- 2 : Channel 2 capture/compare pin

P2DIR.PRIP0 selects the order of precedence when assigning several peripherals to port 0. When set to 10 or 11 the timer 1 channels have precedence.

P2SEL.PRI1P1 and P2SEL.PRI0P1 select the order of precedence when assigning several peripherals to port 1. The timer 1 channels have precedence when the former is set low and the latter is set high.

13.1.4.4 Timer 3

PERCFG.T3CFG selects whether to use alternative 1 or alternative 2 locations.

In Table 36, the Timer 3 signals are shown as the following:

- 0 : Channel 0 capture/compare pin
- 1 : Channel 1 capture/compare pin

P2SEL.PRI2P1 selects the order of precedence when assigning several peripherals to port 1. The timer 3 channels have precedence when the bit is set.

13.1.4.5 Timer 4

PERCFG.T4CFG selects whether to use alternative 1 or alternative 2 locations.

In Table 36, the Timer 4 signals are shown as the following:

- 0 : Channel 0 capture/compare pin
- 1 : Channel 1 capture/compare pin

P2SEL.PRI1P1 selects the order of precedence when assigning several peripherals to port 1. The timer 4 channels have precedence when the bit is set.

13.1.5 ADC

When using the ADC in an application, Port 0 pins must be configured as ADC inputs. Up to eight ADC inputs can be used. The port pins are mapped to the ADC inputs so that P0_7 – P0_0 corresponds to AIN7- AIN0. To configure a Port 0 pin to be used as an ADC input the corresponding bit in the ADCCFG register must be set to 1. The default values in this register select the Port 0 pins as non-ADC input i.e. digital input/outputs. The settings in the ADCCFG register override the settings in P0SEL.

The ADC can be configured to use the general-purpose I/O pin P2_0 as an external trigger to start conversions. P2_0 must be configured as a general-purpose I/O in input mode, when being used for ADC external trigger.

Refer to section 13.7 on page 117 for a detailed description of use of the ADC.

13.1.6 Debug interface

Port pins P2_1 and P2_2 are used for debug data and clock signals, respectively. These are shown as DD (debug data) and DC (debug clock) in Table 36. When the debug interface is in use, P2DIR should select these pins as

inputs. The state of P2SEL is overridden by the debug interface. Also, the direction is overridden when the chip changes the direction to supply the external host with data.

13.1.7 32.768 kHz XOSC input

Ports P2_3 and P2_4 are used to connect an external 32.768 kHz crystal. These port pins will be used by the 32.768 kHz crystal oscillator when CLKCON.OSC32K is low, regardless of register settings. The port pins will be set in analog mode when CLKCON.OSC32K is low.

13.1.8 Unused I/O pins

Unused I/O pins should have a defined level and not be left floating. One way to do this is to leave the pin unconnected and configure the pin as a general purpose I/O input with pull-up resistor. This is also the state of all pins during reset. Alternatively the pin can be configured as a general purpose I/O output. In both cases the pin should not be connected directly to VDD or GND in order to avoid excessive power consumption.

13.1.9 IOC registers

The registers for the IO ports are described in this section. The registers are:

- P0 Port 0
- P1 Port 1
- P2 Port 2
- PERCFG Peripheral control register
- ADCCFG ADC input configuration register
- P0SEL Port 0 function select register
- P1SEL Port 1 function select register
- P2SEL Port 2 function select register
- P0DIR Port 0 direction register
- P1DIR Port 1 direction register
- P2DIR Port 2 direction register
- P0INP Port 0 input mode register
- P1INP Port 1 input mode register
- P2INP Port 2 input mode register
- P0IFG Port 0 interrupt status flag register
- P1IFG Port 1 interrupt status flag register
- P2IFG Port 2 interrupt status flag register
- PICTL Interrupt mask and edge register
- P1IEN Port 1 interrupt mask register

P0 (0x80) – Port 0

Bit	Name	Reset	R/W	Description
7:0	P0[7:0]	0xFF	R/W	Port 0. General purpose I/O port. Bit-addressable.

P1 (0x90) – Port 1

Bit	Name	Reset	R/W	Description
7:0	P1[7:0]	0xFF	R/W	Port 1. General purpose I/O port. Bit-addressable.

P2 (0xA0) – Port 2

Bit	Name	Reset	R/W	Description
7:0	-	000	R0	Not used
4:0	P2[4:0]	0x1F	R/W	Port 2. General purpose I/O port. Bit-addressable.

PERCFG (0xF1) – Peripheral Control

Bit	Name	Reset	R/W	Description
7	–	0	R0	Not used
6	T1CFG	0	R/W	Timer 1 I/O location 0 Alternative 1 location 1 Alternative 2 location
5	T3CFG	0	R/W	Timer 3 I/O location 0 Alternative 1 location 1 Alternative 2 location
4	T4CFG	0	R/W	Timer 4 I/O location 0 Alternative 1 location 1 Alternative 2 location
3:2	–	00	R0	Not used
1	U1CFG	0	R/W	USART1 I/O location 0 Alternative 1 location 1 Alternative 2 location
0	U0CFG	0	R/W	USART0 I/O location 0 Alternative 1 location 1 Alternative 2 location

ADCCFG (0xF2) – ADC Input Configuration

Bit	Name	Reset	R/W	Description
7:0	ADCCFG[7:0]	0x00	R/W	ADC input configuration. ADCCFG[7:0] shall select P0_7 - P0_0 as ADC inputs AIN7 – AIN0 0 ADC input disabled 1 ADC input enabled

P0SEL (0xF3) – Port 0 Function Select

Bit	Name	Reset	R/W	Description
7	SELP0_7	0	R/W	P0_7 function select 0 General purpose I/O 1 Peripheral function
6	SELP0_6	0	R/W	P0_6 function select 0 General purpose I/O 1 Peripheral function
5	SELP0_5	0	R/W	P0_5 function select 0 General purpose I/O 1 Peripheral function
4	SELP0_4	0	R/W	P0_4 function select 0 General purpose I/O 1 Peripheral function
3	SELP0_3	0	R/W	P0_3 function select 0 General purpose I/O 1 Peripheral function
2	SELP0_2	0	R/W	P0_2 function select 0 General purpose I/O 1 Peripheral function
1	SELP0_1	0	R/W	P0_1 function select 0 General purpose I/O 1 Peripheral function
0	SELP0_0	0	R/W	P0_0 function select 0 General purpose I/O 1 Peripheral function

P1SEL (0xF4) – Port 1 Function Select

Bit	Name	Reset	R/W	Description
7	SELP1_7	0	R/W	P1_7 function select 0 General purpose I/O 1 Peripheral function
6	SELP1_6	0	R/W	P1_6 function select 0 General purpose I/O 1 Peripheral function
5	SELP1_5	0	R/W	P1_5 function select 0 General purpose I/O 1 Peripheral function
4	SELP1_4	0	R/W	P1_4 function select 0 General purpose I/O 1 Peripheral function
3	SELP1_3	0	R/W	P1_3 function select 0 General purpose I/O 1 Peripheral function
2	SELP1_2	0	R/W	P1_2 function select 0 General purpose I/O 1 Peripheral function
1	SELP1_1	0	R/W	P1_1 function select 0 General purpose I/O 1 Peripheral function
0	SELP1_0	0	R/W	P1_0 function select 0 General purpose I/O 1 Peripheral function

P2SEL (0xF5) – Port 2 Function Select

Bit	Name	Reset	R/W	Description
7	–	0	R0	Not used
6	PRI3P1	0	R/W	Port 1 peripheral priority control. These bits shall determine the order of priority in the case when PERCFG assigns USART0 and USART1 to the same pins. 0 USART0 has priority 1 USART1 has priority
5	PRI2P1	0	R/W	Port 1 peripheral priority control. These bits shall determine the order of priority in the case when PERCFG assigns USART1 and timer 3 to the same pins. 0 USART1 has priority 1 Timer 3 has priority
4	PRI1P1	0	R/W	Port 1 peripheral priority control. These bits shall determine the order of priority in the case when PERCFG assigns timer 1 and timer 4 to the same pins. 0 Timer 1 has priority 1 Timer 4 has priority
3	PRI0P1	0	R/W	Port 1 peripheral priority control. These bits shall determine the order of priority in the case when PERCFG assigns USART0 and timer 1 to the same pins. 0 USART0 has priority 1 Timer 1 has priority
2	SELP2_4	0	R/W	P2_4 function select 0 General purpose I/O 1 Peripheral function
1	SELP2_3	0	R/W	P2_3 function select 0 General purpose I/O 1 Peripheral function
0	SELP2_0	0	R/W	P2_0 function select 0 General purpose I/O 1 Peripheral function

P0DIR (0xFD) – Port 0 Direction

Bit	Name	Reset	R/W	Description
7	DIRP0_7	0	R/W	P0_7 I/O direction 0 Input 1 Output
6	DIRP0_6	0	R/W	P0_6 I/O direction 0 Input 1 Output
5	DIRP0_5	0	R/W	P0_5 I/O direction 0 Input 1 Output
4	DIRP0_4	0	R/W	P0_4 I/O direction 0 Input 1 Output
3	DIRP0_3	0	R/W	P0_3 I/O direction 0 Input 1 Output
2	DIRP0_2	0	R/W	P0_2 I/O direction 0 Input 1 Output
1	DIRP0_1	0	R/W	P0_1 I/O direction 0 Input 1 Output
0	DIRP0_0	0	R/W	P0_0 I/O direction 0 Input 1 Output

P1DIR (0xFE) – Port 1 Direction

Bit	Name	Reset	R/W	Description
7	DIRP1_7	0	R/W	P1_7 I/O direction 0 Input 1 Output
6	DIRP1_6	0	R/W	P1_6 I/O direction 0 Input 1 Output
5	DIRP1_5	0	R/W	P1_5 I/O direction 0 Input 1 Output
4	DIRP1_4	0	R/W	P1_4 I/O direction 0 Input 1 Output
3	DIRP1_3	0	R/W	P1_3 I/O direction 0 Input 1 Output
2	DIRP1_2	0	R/W	P1_2 I/O direction 0 Input 1 Output
1	DIRP1_1	0	R/W	P1_1 I/O direction 0 Input 1 Output
0	DIRP1_0	0	R/W	P1_0 I/O direction 0 Input 1 Output

P2DIR (0xFF) – Port 2 Direction

Bit	Name	Reset	R/W	Description
7:6	PRIP0[1:0]	0	R/W	Port 0 peripheral priority control. These bits shall determine the order of priority in the case when PERCFG assigns several peripherals to the same pins 00 USART0 – USART1 01 USART1 – USART0 10 Timer 1 channels 0 and 1 – USART1 11 Timer 1 channel 2 – USART0
5	–	0	R0	Not used
4	DIRP2_4	0	R/W	P2_4 I/O direction 0 Input 1 Output
3	DIRP2_3	0	R/W	P2_3 I/O direction 0 Input 1 Output
2	DIRP2_2	0	R/W	P2_2 I/O direction 0 Input 1 Output
1	DIRP2_1	0	R/W	P2_1 I/O direction 0 Input 1 Output
0	DIRP2_0	0	R/W	P2_0 I/O direction 0 Input 1 Output

P0INP (0x8F) – Port 0 Input Mode

Bit	Name	Reset	R/W	Description
7	MDP0_7	0	R/W	P0_7 I/O input mode 0 Pull-up / pull-down 1 Tristate
6	MDP0_6	0	R/W	P0_6 I/O input mode 0 Pull-up / pull-down 1 Tristate
5	MDP0_5	0	R/W	P0_5 I/O input mode 0 Pull-up / pull-down 1 Tristate
4	MDP0_4	0	R/W	P0_4 I/O input mode 0 Pull-up / pull-down 1 Tristate
3	MDP0_3	0	R/W	P0_3 I/O input mode 0 Pull-up / pull-down 1 Tristate
2	MDP0_2	0	R/W	P0_2 I/O input mode 0 Pull-up / pull-down 1 Tristate
1	MDP0_1	0	R/W	P0_1 I/O input mode 0 Pull-up / pull-down 1 Tristate
0	MDP0_0	0	R/W	P0_0 I/O input mode 0 Pull-up / pull-down 1 Tristate

P1INP (0xF6) – Port 1 Input Mode

Bit	Name	Reset	R/W	Description
7	MDP1_7	0	R/W	P1_7 I/O input mode 0 Pull-up / pull-down 1 Tristate
6	MDP1_6	0	R/W	P1_6 I/O input mode 0 Pull-up / pull-down 1 Tristate
5	MDP1_5	0	R/W	P1_5 I/O input mode 0 Pull-up / pull-down 1 Tristate
4	MDP1_4	0	R/W	P1_4 I/O input mode 0 Pull-up / pull-down 1 Tristate
3	MDP1_3	0	R/W	P1_3 I/O input mode 0 Pull-up / pull-down 1 Tristate
2	MDP1_2	0	R/W	P1_2 I/O input mode 0 Pull-up / pull-down 1 Tristate
1:0	–	00	R0	Not used

P2INP (0xF7) – Port 2 Input Mode

Bit	Name	Reset	R/W	Description
7	PDUP2	0	R/W	Port 2 pull-up/down select. Selects function for all Port 2 pins configured as pull-up/pull-down inputs. 0 Pull-up 1 Pull-down
6	PDUP1	0	R/W	Port 1 pull-up/down select. Selects function for all Port 1 pins configured as pull-up/pull-down inputs. 0 Pull-up 1 Pull-down
5	PDUP0	0	R/W	Port 0 pull-up/down select. Selects function for all Port 0 pins configured as pull-up/pull-down inputs. 0 Pull-up 1 Pull-down
4	MDP2_4	0	R/W	P2_4 I/O input mode 0 Pull-up / pull-down 1 Tristate
3	MDP2_3	0	R/W	P2_3 I/O input mode 0 Pull-up / pull-down 1 Tristate
2	MDP2_2	0	R/W	P2_2 I/O input mode 0 Pull-up / pull-down 1 Tristate
1	MDP2_1	0	R/W	P2_1 I/O input mode 0 Pull-up / pull-down 1 Tristate
0	MDP2_0	0	R/W	P2_0 I/O input mode 0 Pull-up / pull-down 1 Tristate

P0IFG (0x89) – Port 0 Interrupt Status Flag

Bit	Name	Reset	R/W	Description
7:0	P0IF[7:0]	0x00	R/W0	Port 0, inputs 7 to 0 interrupt status flags. When an input port pin has an interrupt request pending, the corresponding flag bit will be set.

P1IFG (0x8A) – Port 1 Interrupt Status Flag

Bit	Name	Reset	R/W	Description
7:0	P1IF[7:0]	0x00	R/W0	Port 1, inputs 7 to 0 interrupt status flags. When an input port pin has an interrupt request pending, the corresponding flag bit will be set.

P2IFG (0x8B) – Port 2 Interrupt Status Flag

Bit	Name	Reset	R/W	Description
7:5	–	000	R0	Not used.
4:0	P2IF[4:0]	0x00	R/W0	Port 2, inputs 4 to 0 interrupt status flags. When an input port pin has an interrupt request pending, the corresponding flag bit will be set.

PICTL (0x8C) – Port Interrupt Control

Bit	Name	Reset	R/W	Description
7	–	0	R0	Not used
6	–	0	R/W	Reserved. Write 0
5	P2IEN	0	R/W	Port 2, inputs 4 to 0 interrupt enable. This bit enables interrupt requests for the port 2 inputs 4 to 0. 0 Interrupts are disabled 1 Interrupts are enabled
4	P0IENH	0	R/W	Port 0, inputs 7 to 4 interrupt enable. This bit enables interrupt requests for the port 0 inputs 7 to 4. 0 Interrupts are disabled 1 Interrupts are enabled
3	P0IENL	0	R/W	Port 0, inputs 3 to 0 interrupt enable. This bit enables interrupt requests for the port 0 inputs 3 to 0. 0 Interrupts are disabled 1 Interrupts are enabled
2	P2ICON	0	R/W	Port 2, inputs 4 to 0 interrupt configuration. This bit selects the interrupt request condition for all port 2 inputs 0 Rising edge on input gives interrupt 1 Falling edge on input gives interrupt
1	P1ICON	0	R/W	Port 1, inputs 7 to 0 interrupt configuration. This bit selects the interrupt request condition for all port 1 inputs 0 Rising edge on input gives interrupt 1 Falling edge on input gives interrupt
0	P0ICON	0	R/W	Port 0, inputs 7 to 0 interrupt configuration. This bit selects the interrupt request condition for all port 0 inputs 0 Rising edge on input gives interrupt 1 Falling edge on input gives interrupt

P1IEN (0x8D) – Port 1 Interrupt Mask

Bit	Name	Reset	R/W	Description
7	P1_7IEN	0	R/W	Port P1_7 interrupt enable 0 Interrupts are disabled 1 Interrupts are enabled
6	P1_6IEN	0	R/W	Port P1_6 interrupt enable 0 Interrupts are disabled 1 Interrupts are enabled
5	P1_5IEN	0	R/W	Port P1_5 interrupt enable 0 Interrupts are disabled 1 Interrupts are enabled
4	P1_4IEN	0	R/W	Port P1_4 interrupt enable 0 Interrupts are disabled 1 Interrupts are enabled
3	P1_3IEN	0	R/W	Port P1_3 interrupt enable 0 Interrupts are disabled 1 Interrupts are enabled
2	P1_2IEN	0	R/W	Port P1_2 interrupt enable 0 Interrupts are disabled 1 Interrupts are enabled
1	P1_1IEN	0	R/W	Port P1_1 interrupt enable 0 Interrupts are disabled 1 Interrupts are enabled
0	P1_0IEN	0	R/W	Port P1_0 interrupt enable 0 Interrupts are disabled 1 Interrupts are enabled

13.2 DMA Controller

The CC1110 includes a direct memory access (DMA) controller, which can be used to relieve the 8051 CPU core of handling data movement operations. Thus the CC1110 can achieve high overall performance with good power efficiency. The DMA controller can move data from a peripheral unit such as ADC or RF transceiver to memory with minimum CPU intervention.

The DMA controller module coordinates all DMA transfers, ensuring that DMA requests are prioritized appropriately relative to each other and CPU memory access. The DMA controller contains a number of programmable DMA channels for memory-to-memory data movement.

The DMA controller controls data movement over the entire XDATA memory space. Since the SFR registers are mapped into the DMA memory space these flexible DMA channels can be used to unburden the 8051 in innovative ways, e.g. feed a USART with data from memory, periodically transfer samples between ADC and memory, produce a desired I/O waveform by transferring a pattern in memory to an I/O port output register, etc. Use of the DMA can also reduce system power consumption by keeping the CPU in a low-power mode without having to wake up to move data to or from a peripheral unit.

The main features of the DMA controller are as follows:

- Five independent DMA channels
- Three configurable levels of DMA channel priority
- 30 configurable transfer trigger events
- Independent control of source and destination address
- Single, block and repeated transfer modes
- Supports variable transfer length by including the length field in the transfer data
- Can operate in either word-size or byte-size mode

13.2.1 DMA Operation

There are five DMA channels available in the DMA controller numbered channel 0 to channel 4. Each DMA channel can move data from one place within the DMA memory space to another.

In order to use a DMA channel it must first be configured as described in sections 13.2.2 and 13.2.3.

Once a DMA channel has been configured it must be armed before any transfers are allowed to be initiated. A DMA channel is armed by setting the appropriate bit in the DMA Channel Arm register DMAARM.

When a DMA channel is armed a transfer will begin when the configured DMA trigger event occurs. There are 30 possible DMA trigger events, e.g. UART transfer, Timer overflow etc. The trigger event to be used by a DMA channel is set by the DMA channel configuration. The DMA trigger events are listed in Table 37.

In addition to starting a DMA transfer through the DMA trigger events, the user software may force a DMA transfer to begin by setting the corresponding DMAREQ bit.

Figure 10 shows the DMA state diagram.

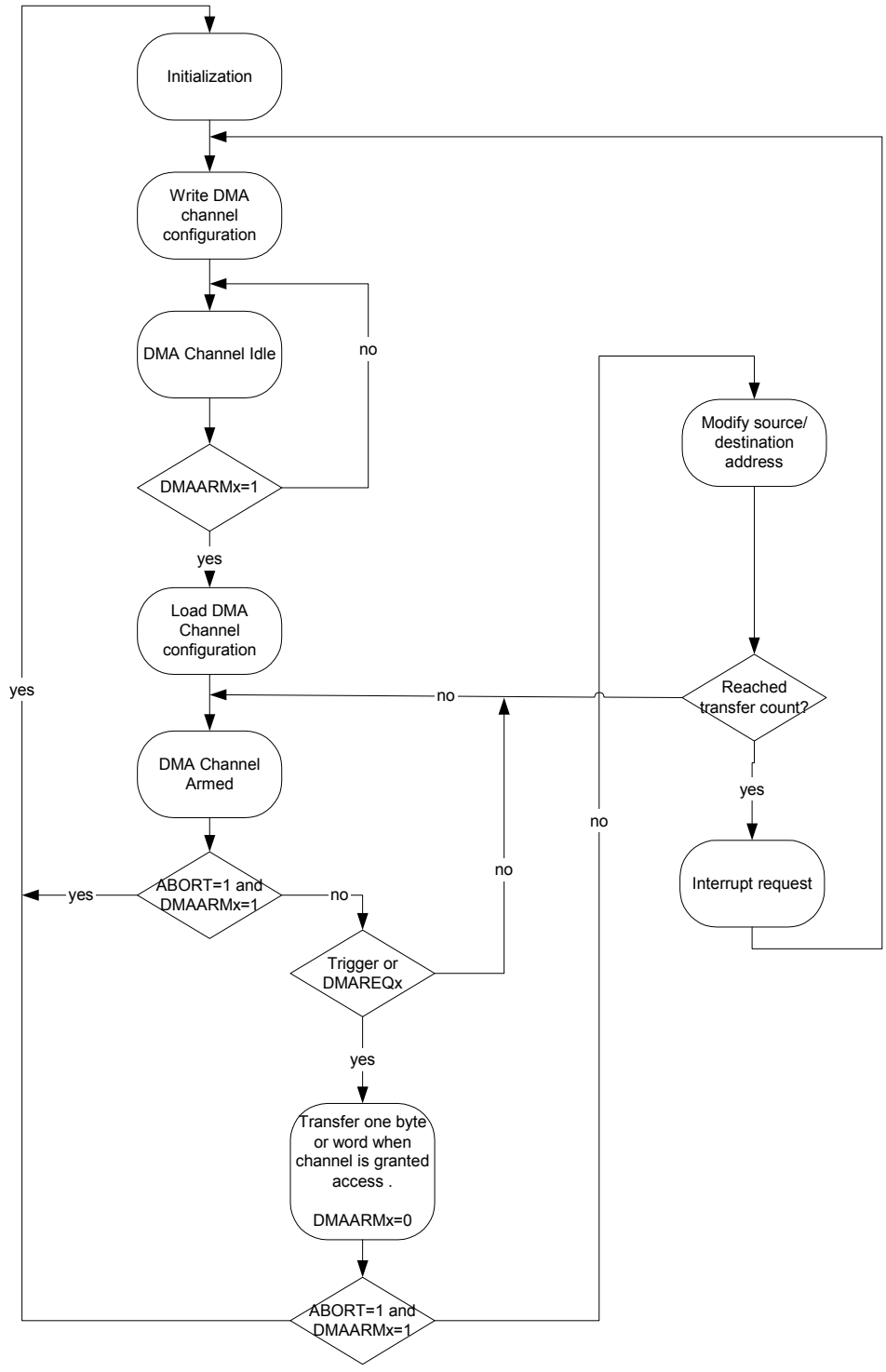


Figure 10: DMA Operation

13.2.2 DMA Configuration Parameters

Setup and control of the DMA operation is performed by the user software. This section describes the parameters that must be configured before a DMA channel can be used. Section 13.2.3 on page 82 describes how the parameters are set up in software and passed to the DMA controller.

The behavior of each of the five DMA channels is configured with the following parameters:

Source address. The first address from which the DMA channel should read data.

Destination address. The first address to which the DMA channel should write the data read from the source address. The user must ensure that the destination is writable.

Transfer count. The number of transfers to perform before rearming or disarming the DMA channel and alerting the CPU with an interrupt request. The length can be defined in the configuration or it can be defined as described next as VLEN setting.

VLEN setting. The DMA channel is capable of variable length transfers using the first byte or word to set the transfer length. When doing this, various options regarding how to count number of bytes to transfer are available.

Priority. The priority of the DMA transfers for the DMA channel in respect to the CPU and other DMA channels and access ports.

Trigger event. All DMA transfers are initiated by so-called DMA trigger events. This trigger either starts a DMA block transfer or a single DMA transfer. In addition to the configured trigger, a DMA channel can always be triggered by setting its designated `DMAREQ.DMAREQx` flag. The DMA trigger sources are described in Table 37 on page 84.

Source and Destination Increment. The source and destination addresses can be controlled to increment, decrement, or not change, in order to give good flexibility for various types of transfers.

Transfer mode. The transfer mode determines whether the transfer should be a single transfer or a block transfer, or repeated versions of these.

Byte or word transfers. Determines whether each DMA transfer should be 8-bit (byte) or 16-bit (word).

Interrupt Mask. An interrupt request is generated upon completion of the DMA transfer. The interrupt mask bit controls if the interrupt generation is enabled or disabled.

M8: Decide whether to use seven or eight bits of length byte for transfer length. Only applicable when doing byte transfers.

A detailed description of all configuration parameters is given in the following sections.

13.2.2.1 Source Address

The address of the location in XDATA memory space where the DMA channel shall start to read data for the transfer.

13.2.2.2 Destination Address

The address of the location in XDATA memory space where the DMA channel shall start to write transfer data. The user must ensure that the destination is writable.

13.2.2.3 Transfer Count

The number of bytes/words needed to be transferred for the DMA transfer to be complete. When the transfer count is reached, the DMA controller rearms or disarms the DMA channel and alerts the CPU with an interrupt request. The transfer count can be defined in the configuration or it can be defined as a variable length described in the next section.

13.2.2.4 VLEN Setting

The DMA channel is capable of using the first byte or word (for word, bits 12:0 are used) in source data as the transfer length. This allows variable length transfers. When using variable length transfer, various options regarding how to count number of bytes to transfer is given. In any case, the LEN setting is used as maximum transfer count. Note that the M8 bit is only used when byte size transfers are chosen.

Options are:

1. *Default :* Transfer number of bytes/words commanded by first byte/word + 1 (transfers length byte/word, and then as many bytes/words as dictated by length byte/word)
2. Transfer number of bytes/words commanded by first byte/word

3. Transfer number of bytes/words commanded by first byte/word + 2 (transfers length byte/word, and then as many bytes/words as dictated by length byte/word + 1)

4. Transfer number of bytes/words commanded by first byte/word + 3 (transfers length byte/word, and then as many bytes/words as dictated by length byte/word + 2)

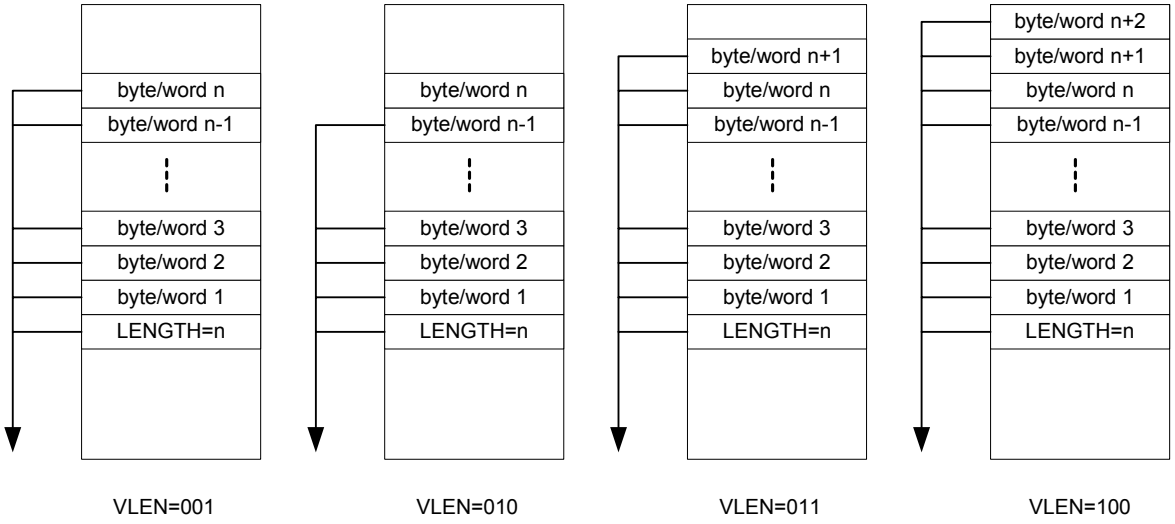


Figure 11: Variable Length (VLEN) Transfer Options

13.2.2.5 Trigger Event

Each DMA channel can be set up to sense on a single trigger. This field determines which trigger the DMA channel shall sense.

13.2.2.6 Source and Destination Increment

When the DMA channel is armed or rearmed the source and destination addresses are transferred to internal address pointers. The possibilities for address increment are :

- *Increment by zero.* The address pointer shall remain fixed after each transfer.
- *Increment by one.* The address pointer shall increment one count after each transfer.
- *Increment by two.* The address pointer shall increment two counts after each transfer.
- *Decrement by one.* The address pointer shall decrement one count after each transfer.

13.2.2.7 DMA Transfer Mode

The transfer mode determines how the DMA channel behaves when it starts transferring data. There are four transfer modes described below:

Single. On a trigger a single DMA transfer occurs and the DMA channel awaits the next trigger. After the number of transfers specified by the transfer count are completed, the CPU is notified and the DMA channel is disarmed.

Block. On a trigger the number of DMA transfers specified by the transfer count is performed as quickly as possible, after which the CPU is notified and the DMA channel is disarmed.

Repeated single. On a trigger a single DMA transfer occurs and the DMA channel awaits the next trigger. After the number of transfers specified by the transfer count are completed, the CPU is notified and the DMA channel is rearmed.

Repeated block. On a trigger the number of DMA transfers specified by the transfer count is performed as quickly as possible, after

which the CPU is notified and the DMA channel is rearmed.

13.2.2.8 DMA Priority

A DMA priority is associated with each DMA access port and is configurable for each DMA channel. The DMA priority is used to determine the winner in the case of multiple simultaneous internal memory requests, and whether the DMA memory access should have priority or not over a simultaneous CPU memory access. In case of an internal tie, a round-robin scheme is used to ensure access for all. There are three levels of DMA priority:

High. Highest internal priority. DMA access will always prevail over CPU access.

Normal. Second highest internal priority. Guarantees that DMA access prevails over CPU on at least every second try.

Low. Lowest internal priority. DMA access will always defer to a CPU access.

13.2.2.9 Byte or Word transfers

Determines whether 8-bit (byte) or 16-bit (word) are done.

13.2.2.10 Interrupt mask

Upon completing a DMA transfer, the channel can generate an interrupt to the processor. This bit will mask the interrupt.

13.2.2.11 Mode 8 setting

This field determines whether to use seven or 8 bits of length byte for transfer length. Only applicable when doing byte transfers.

13.2.3 DMA Configuration Setup

The DMA channel parameters such as address mode, transfer mode and priority described in the previous section have to be configured before a DMA channel can be armed and activated. The parameters are not configured directly through SFR registers, but instead they are written in a special DMA configuration data structure in memory. Each DMA channel in use requires its own DMA configuration data structure. The DMA configuration data structure consists of eight bytes and is described in section 13.2.6 A DMA configuration data structure may reside at any location decided upon by the user software, and the address location is passed to the DMA controller through a set of SFRs DMAxCFGH:DMAxCFGL, Once a channel has been armed, the DMA controller will read the configuration data structure for that channel, given by the address in DMAxCFGH:DMAxCFGL.

It is important to note that the method for specifying the start address for the DMA configuration data structure differs between DMA channel 0 and DMA channels 1-4 as follows:

DMA0CFGH:DMA0CFGL gives the start address for DMA channel 0 configuration data structure.

DMA1CFGH:DMA1CFGL gives the start address for DMA channel 1 configuration data structure followed by channel 2-4 configuration data structures.

Thus the DMA controller expects the DMA configuration data structures for DMA channels 1-4 to lie in a contiguous area in memory, starting at the address held in DMA1CFGH:DMA1CFGL and consisting of 32 bytes.

13.2.4 Stopping DMA Transfers

Ongoing DMA transfer or armed DMA channels will be aborted using the DMAARM register to disarm the DMA channel.

One or more DMA channels are aborted by writing the following to the DMAARM register.

- Writing a 1 to DMAARM.ABORT, and at the same time,
- Select which DMA channels to abort by setting the corresponding, DMAARM.DMAARMx bits.

An example of DMA channel arm and disarm is shown in Figure 12.

```
MOV DMAARM, #0x03   ; arm DMA channel 0 and 1

MOV DMAARM, #0x81   ; disarm DMA channel 0,
                    ; channel 1 is still armed
```

Figure 12: DMA arm/disarm example

13.2.5 DMA Interrupts

Each DMA channel can be configured to generate an interrupt to the CPU upon completing a DMA transfer. This is accomplished with the IRQMASK bit in the channel configuration. The corresponding interrupt flag in the DMAIRQ SFR register will be set when the interrupt is generated.

Regardless of the IRQMASK bit in the channel configuration, the interrupt flag will be set upon DMA channel complete. Thus software should always check (and clear) this register when rearming a channel with a changed IRQMASK setting. Failure to do so could generate an interrupt based on the stored interrupt flag.

13.2.6 DMA Configuration Data Structure

For each DMA channel, the DMA configuration data structure consists of eight bytes. The configuration data structure is described in Table 38.

DMA Trigger number	DMA Trigger name	Functional unit	Description
0	NONE	DMA	No trigger, setting DMAREQ.DMAREQx bit starts transfer
1	PREV	DMA	DMA channel is triggered by completion of previous channel
2	T1_CH0	Timer 1	Timer 1, compare, channel 0
3	T1_CH1	Timer 1	Timer 1, compare, channel 1
4	T1_CH2	Timer 1	Timer 1, compare, channel 2
5	-	-	Not in use.
6	T2_OVFL	Timer 2	Timer 2, overflow
7	T3_CH0	Timer 3	Timer 3, compare, channel 0
8	T3_CH1	Timer 3	Timer 3, compare, channel 1
9	T4_CH0	Timer 4	Timer 4, compare, channel 0
10	T4_CH1	Timer 4	Timer 4, compare, channel 1
11	ST	Sleep Timer	Sleep Timer compare
12	IOC_0	IO Controller	IO pin input transition
13	IOC_1	IO Controller	IO pin input transition
14	URX0	USART0	USART0 RX complete
15	UTX0	USART0	USART0 TX complete
16	URX1	USART1	USART1 RX complete
17	UTX1	USART1	USART1 TX complete
18	FLASH	Flash controller	Flash data write complete
19	RADIO	Radio	RF packet byte received/transmit
20	ADC_CHALL	ADC	ADC end of a conversion in a sequence, sample ready
21	ADC_CH11	ADC	ADC end of conversion channel 0 in sequence, sample ready
22	ADC_CH21	ADC	ADC end of conversion channel 1 in sequence, sample ready
23	ADC_CH32	ADC	ADC end of conversion channel 2 in sequence, sample ready
24	ADC_CH42	ADC	ADC end of conversion channel 3 in sequence, sample ready
25	ADC_CH53	ADC	ADC end of conversion channel 4 in sequence, sample ready
26	ADC_CH63	ADC	ADC end of conversion channel 5 in sequence, sample ready
~~27~~	ADC_CH74	ADC	ADC end of conversion channel 6 in sequence, sample ready
28	ADC_CH84	ADC	ADC end of conversion channel 7 in sequence, sample ready
29	ENC_DW	AES	AES encryption processor requests download input data
30	ENC_UP	AES	AES encryption processor requests upload output data
31	-	-	Not in use.

Table 37: DMA Trigger Sources

Table 38: DMA Configuration Data Structure

Byte Offset	Bit	Field Name	Description
0	7:0	SRCADDR[15:8]	The DMA channel source address, high
1	7:0	SRCADDR[7:0]	The DMA channel source address, low
2	7:0	DESTADDR[15:8]	The DMA channel destination address, high. Note that flash memory is not directly writeable.
3	7:0	DESTADDR[7:0]	The DMA channel destination address, high. Note that flash memory is not directly writeable.
4	7:5	VLEN[2:0]	Variable length transfer mode. In word mode, bits 12:0 of the first word is considered as the transfer length. 000/111 Use LEN for transfer count 001 Transfer the number of bytes/words specified by first byte/word + 1 (up to a maximum specified by LEN). Thus transfer count excludes length byte/word 010 Transfer the number of bytes/words specified by first byte/word (up to a maximum specified by LEN). Thus transfer count includes length byte/word. 011 Transfer the number of bytes/words specified by first byte/word + 2 (up to a maximum specified by LEN). 100 Transfer the number of bytes/words specified by first byte/word + 3 (up to a maximum specified by LEN). 101 reserved 110 reserved
4	4:0	LEN[12:8]	The DMA channel transfer count. Used as maximum allowable length when VLEN = 000/111. The DMA channel counts in words when in WORDSIZE mode, and otherwise in bytes.
5	7:0	LEN[7:0]	The DMA channel transfer count. Used as maximum allowable length when VLEN = 000/111. The DMA channel counts in words when in WORDSIZE mode, and otherwise in bytes.
6	7	WORDSIZE	Selects whether each DMA transfer shall be 8-bit (0) or 16-bit (1).
6	6:5	TMODE[1:0]	The DMA channel transfer mode: 00 : Single 01 : Block 10 : Repeated single 11 : Repeated block
6	4:0	TRIG[4:0]	Select DMA trigger to use 00000 : No trigger (writing to DMAREQ is only trigger) 00001 : The previous DMA channel finished 00010 – 11111 : Selects one of the triggers shown in Table 37. The trigger is selected in the order shown in the table.
7	7:6	SRCINC[1:0]	Source address increment mode (after each transfer): 00 : 0 bytes/words 01 : 1 bytes/words 10 : 2 bytes/words 11 : -1 bytes/words
7	5:4	DESTINC[1:0]	Destination address increment mode (after each transfer): 00 : 0 bytes/words 01 : 1 bytes/words 10 : 2 bytes/words 11 : -1 bytes/words

Byte Offset	Bit	Field Name	Description
7	3	IRQMASK	Interrupt Mask for this channel. 0 : Disable interrupt generation 1 : Enable interrupt generation upon DMA channel done
7	2	M8	Mode of 8th bit for VLEN transfer length; only applicable when WORDSIZE=0. 0 : Use all 8 bits for transfer count 1 : Use 7 LSB for transfer count
7	1:0	PRIORITY[1:0]	The DMA channel priority: 00 : Low, CPU has priority. 01 : Guaranteed, DMA at least every second try. 10 : High, DMA has priority 11 : Highest, DMA has priority. Reserved for DMA port access.

13.2.7 DMA registers

This section describes the SFR registers associated with the DMA Controller

DMAARM (0xD6) – DMA Channel Arm

Bit	Name	Reset	R/W	Description
7	ABORT	0	R0/W	DMA abort. This bit is used to stop ongoing DMA transfers. Writing a 1 to this bit will abort all channels which are selected by setting the corresponding DMAARM bit to 1 0 : Normal operation 1 : Abort channels all selected channels
6:5	-	00	R/W	Not used
4	DMAARM4	0	R/W	DMA arm channel 4 This bit must be set in order for any DMA transfers to occur on the channel. For non-repetitive transfer modes, the bit is automatically cleared upon completion.
3	DMAARM3	0	R/W	DMA arm channel 3 This bit must be set in order for any DMA transfers to occur on the channel. For non-repetitive transfer modes, the bit is automatically cleared upon completion.
2	DMAARM2	0	R/W	DMA arm channel 2 This bit must be set in order for any DMA transfers to occur on the channel. For non-repetitive transfer modes, the bit is automatically cleared upon completion.
1	DMAARM1	0	R/W	DMA arm channel 1 This bit must be set in order for any DMA transfers to occur on the channel. For non-repetitive transfer modes, the bit is automatically cleared upon completion.
0	DMAARM0	0	R/W	DMA arm channel 0 This bit must be set in order for any DMA transfers to occur on the channel. For non-repetitive transfer modes, the bit is automatically cleared upon completion.

DMAREQ (0xD7) – DMA Channel Start Request and Status

Bit	Name	Reset	R/W	Description
7:5	-	000	R0	Not used
4	DMAREQ4	0	R/W1 H0	DMA transfer request, channel 4 When set to 1 activate the DMA channel (has the same effect as a single trigger event.). Only by setting the armed bit to 0 in the DMAARM register, can the channel be stopped if already started. This bit is cleared when the DMA channel is granted access.
3	DMAREQ3	0	R/W1 H0	DMA transfer request, channel 3 When set to 1 activate the DMA channel (has the same effect as a single trigger event.). Only by setting the armed bit to 0 in the DMAARM register, can the channel be stopped if already started. This bit is cleared when the DMA channel is granted access.
2	DMAREQ2	0	R/W1 H0	DMA transfer request, channel 2 When set to 1 activate the DMA channel (has the same effect as a single trigger event.). Only by setting the armed bit to 0 in the DMAARM register, can the channel be stopped if already started. This bit is cleared when the DMA channel is granted access.
1	DMAREQ1	0	R/W1 H0	DMA transfer request, channel 1 When set to 1 activate the DMA channel (has the same effect as a single trigger event.). Only by setting the armed bit to 0 in the DMAARM register, can the channel be stopped if already started. This bit is cleared when the DMA channel is granted access.
0	DMAREQ0	0	R/W1 H0	DMA transfer request, channel 0 When set to 1 activate the DMA channel (has the same effect as a single trigger event.). Only by setting the armed bit to 0 in the DMAARM register, can the channel be stopped if already started. This bit is cleared when the DMA channel is granted access.

DMA0CFGH (0xD5) – DMA Channel 0 Configuration Address High Byte

Bit	Name	Reset	R/W	Description
7:0	DMA0CFG[15:8]	0x00	R/W	The DMA channel 0 configuration address, high order

DMA0CFGL (0xD4) – DMA Channel 0 Configuration Address Low Byte

Bit	Name	Reset	R/W	Description
7:0	DMA0CFG[7:0]	0x00	R/W	The DMA channel 0 configuration address, low order

DMA1CFGH (0xD3) – DMA Channel 1-4 Configuration Address High Byte

Bit	Name	Reset	R/W	Description
7:0	DMA1CFG[15:8]	0x00	R/W	The DMA channel 1-4 configuration address, high order

DMA1CFGL (0xD2) – DMA Channel 1-4 Configuration Address Low Byte

Bit	Name	Reset	R/W	Description
7:0	DMA1CFG[7:0]	0x00	R/W	The DMA channel 1-4 configuration address, low order

DMAIRQ (0xD1) – DMA Interrupt Flag

Bit	Name	Reset	R/W	Description
7:5	-	000	R/W0	Not used
4	DMAIF4	0	R/W0	DMA channel 4 interrupt flag. 0 : DMA channel transfer not complete 1 : DMA channel transfer complete/interrupt pending
3	DMAIF3	0	R/W0	DMA channel 3 interrupt flag. 0 : DMA channel transfer not complete 1 : DMA channel transfer complete/interrupt pending
2	DMAIF2	0	R/W0	DMA channel 2 interrupt flag. 0 : DMA channel transfer not complete 1 : DMA channel transfer complete/interrupt pending
1	DMAIF1	0	R/W0	DMA channel 1 interrupt flag. 0 : DMA channel transfer not complete 1 : DMA channel transfer complete/interrupt pending
0	DMAIF0	0	R/W0	DMA channel 0 interrupt flag. 0 : DMA channel transfer not complete 1 : DMA channel transfer complete/interrupt pending

13.3 16-bit Timer, Timer 1

Timer 1 is an independent 16-bit timer which supports typical timer/counter functions such as input capture, output compare and PWM functions. The timer has three independent capture/compare channels. The timer uses one I/O pin per channel. The timer is used for a wide range of control and measurement applications and the availability of up/down count mode with three channels will for example allow implementation of motor control applications.

The features of Timer 1 are as follows:

- Three capture/compare channels
- Rising, falling or any edge input capture
- Set, clear or toggle output compare
- Free-running, modulo or up/down counter operation
- Clock prescaler for divide by 1, 8, 32 or 128
- Interrupt request generated on each capture/compare and terminal count
- Capture triggered by radio
- DMA trigger function

13.3.1 16-bit Timer Counter

The timer consists of a 16-bit counter that increments or decrements at each active clock edge. The period of the active clock edges is defined by the register bits `CLKCON.TICKSPD` which sets the global division of the system clock giving a variable clock tick frequency from 0.203 MHz to 26 MHz. This is further divided in Timer 1 by the prescaler value set by `T1CTL.DIV`. This prescaler value can be from 1 to 128. Thus the lowest clock frequency used by Timer 1 is 1586.9 Hz and the highest is 26 MHz when the 26 MHz crystal oscillator is used as system clock source. When the 13 MHz RC oscillator is used as system clock source, then the highest clock frequency used by Timer 1 is 13 MHz.

The counter operates as either a free-running counter, a modulo counter or as an up/down counter for use in centre-aligned PWM.

It is possible to read the 16-bit counter value through the two 8-bit SFRs; `T1CNTH` and `T1CNTL`, containing the high-order byte and low-order byte respectively. When the `T1CNTL` is read, the high-order byte of the counter at that instant is buffered in `T1CNTH` so that the high-order byte can be read from `T1CNTH`. Thus `T1CNTL` shall always be read first before reading `T1CNTH`.

All write accesses to the `T1CNTL` register will reset the 16-bit counter.

The counter produces an interrupt request when the terminal count value (overflow) is reached. It is possible to clear and halt the counter with `T1CTL` control register settings. The counter is started when a value other than 00 is written to `T1CTL.MODE`. If 00 is written to `T1CTL.MODE` the counter halts at its present value.

13.3.2 Timer 1 Operation

In general, the control register `T1CTL` is used to control the timer operation. The various modes of operation are described below.

13.3.3 Free-running Mode

In the free-running mode of operation the counter starts from 0x0000 and increments at each active clock edge. When the counter reaches 0xFFFF the counter is loaded with 0x0000 and continues incrementing its value as shown in Figure 13. When the terminal count value 0xFFFF is reached, the flag `T1CTL.OVFIF` is set. An interrupt request is generated if the corresponding interrupt mask bit `TIMIF.OVFIM` is set. The free-running mode can be used to generate independent time intervals and output signal frequencies.

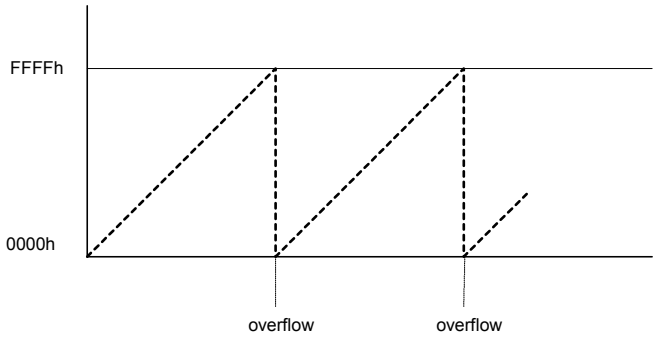


Figure 13: Free-running mode

13.3.4 Modulo Mode

When the timer operates in modulo mode the 16-bit counter starts at 0x0000 and increments at each active clock edge. When the counter reaches the terminal count value held in registers T1CC0H:T1CC0L, the counter is reset to 0x0000 and continues to increment. The flag T1CTL.OVFIF is set when the terminal count value (overflow) is reached. An interrupt request is generated if the corresponding interrupt mask bit TIMIF.OVFIM is set. The modulo mode can be used for applications where a period other then 0xFFFF is required. The counter operation is shown in Figure 14.

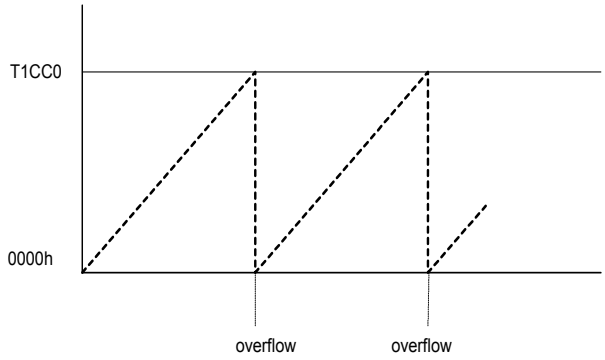


Figure 14: Modulo mode

13.3.5 Up/down Mode

In the up/down timer mode, the counter repeatedly starts from 0x0000 and counts up until the value held in T1CC0H:T1CC0L is reached and then the counter counts down until 0x0000 is reached as shown in Figure 15. This timer mode is used when symmetrical output pulses are required with a period other than 0xFFFF, and therefore allows implementation of centre-aligned PWM output applications. The flag T1CTL.OVFIF is not set when the terminal count value is reached in the up/down mode.

Clearing the counter by writing to T1CNTL will also reset the count direction to the count up from 0x0000 mode.

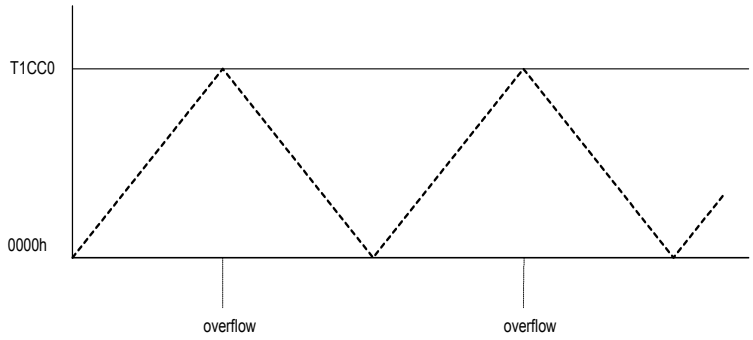


Figure 15 : Up/down mode

13.3.6 Channel Mode Control

The channel mode is set with each channel's control and status register T1CCTLn. The settings include input capture and output compare modes.

13.3.7 Input Capture Mode

When a channel is configured as an input capture channel, the I/O pin associated with that channel, is configured as an input. After the timer has been started, a rising edge, falling edge or any edge on the input pin will trigger a capture of the 16-bit counter contents into the associated capture register. Thus the timer is able to capture the time when an external event takes place.

Note: before an I/O pin can be used by the timer, the required I/O pin must be configured as a Timer 1 peripheral pin as described in section 13.1.4 on page 62 .

The channel input pin is synchronized to the internal system clock. Thus pulses on the input pin must have a minimum duration greater than the system clock period.

The content of the 16-bit capture register is read out from registers T1CCnH:T1CCnL.

When the capture takes place the interrupt flag for the channel is set. This bit is T1CTL.CH0IF for channel 0, T1CTL.CH1IF for channel 1, and T1CTL.CH2IF for channel 2. An interrupt request is generated if the corresponding interrupt mask bit on T1CCTL0.IM, T1CCTL1.IM, or T1CCTL2.IM, respectively, is set.

13.3.7.1 RF Event Capture

Each timer channel may be configured so that an RF interrupt RFIF event will trigger a capture instead of the normal input pin capture. This function is selected with the register bit T1CCTLx.CPSEL which selects to use either the input pin or the RFIF interrupt as capture event. When RFIF is selected as capture input, the interrupt source(s) enabled by RFIM (see section 15.3.1 on page 156) will trigger a capture. In this way the timer can be used to capture a value when e.g. a start of frame delimiter (SFD) is detected.

13.3.8 Output Compare Mode

In output compare mode the I/O pin associated with a channel is set as an output. After the timer has been started, the contents of the counter is compared with the contents of the channel compare register T1CCnH:T1CCnL. If the compare register equals the counter contents, the output pin is set, reset or toggled according to the compare output mode setting of T1CCTLn.CMP. Note that all edges on output pins are glitch-free when operating in a given output compare mode. Writing to the compare register T1CCnL is buffered so that a value written to T1CCnL does not take effect until the corresponding high order register, T1CCnH is written. For output compare modes 1-3, a new value written to the compare register T1CCnH:T1CCnL takes effect after the registers have been written. For other output compare modes the new value written to the compare register take effect when the timer reaches 0x0000.

Note that channel 0 has fewer output compare modes than channel 1 and 2 because T1CC0H:T1CC0L has a special function in modes 6 and 7, meaning these modes would not be useful for channel 0.

When a compare occurs, the interrupt flag for the channel is set. This bit is T1CTL.CH0IF

for channel 0, T1CTL.CH1IF for channel 1, and T1CTL.CH2IF for channel 2. An interrupt request is generated if the corresponding interrupt mask bit on T1CCTL0.IM, T1CCTL1.IM, or T1CCTL2.IM, respectively, is set.

Examples of output compare modes in various timer modes are given in the following figures.

Edge-aligned PWM output signals can be generated using the timer modulo mode and channels 1 and 2 in output compare mode 6 or 7 as shown in Figure 17. The period of the PWM signal is determined by the setting T1CC0 and the duty cycle for the channel output is determined by T1CCn. The timer free-running mode may also be used. In this case CLKCON.TICKSPD and the prescaler divider value T1CTL.DIV set the period of the PWM signal. The polarity of the PWM signal is determined by whether output compare mode 6 or 7 is used. PWM output signals can also be generated using output compare modes 4 and 5 as shown in the same figure, or by using modulo mode as shown in Figure 17. Using output compare mode 4 and 5 is preferred for simple PWM.

Centre-aligned PWM outputs can be generated when the timer up/down mode is selected. The channel output compare mode 4 or 5 is selected depending on required polarity of the PWM signal. The period of the PWM signal is determined by T1CC0 and the duty cycle for the channel output is determined by T1CCn.

The centre-aligned PWM mode is required by certain types of motor drive applications and typically less noise is produced than the edge-aligned PWM mode because the I/O pin transitions are not lined up on the same clock edge.

In some types of applications, a defined delay or dead time is required between outputs. Typically this is required for outputs driving an H-bridge configuration to avoid uncontrolled cross-conduction in one side of the H-bridge. The delay or dead-time can be obtained in the PWM outputs by using T1CCn as shown in the following:

Assuming that channel 1 and channel 2 are used to drive the outputs using timer up/down mode and the channels use output compare modes 4 and 5 respectively. If T1CC1 is greater than T1CC2, then the timer period (in Timer 1 clock periods) is:

T_P = T1CC0 x 2

and the dead time, i.e. the time from when the channel 1 output goes low until the channel 2 output goes high, (in Timer 1 clock periods) is given by:

T_D = T1CC1 – T1CC2

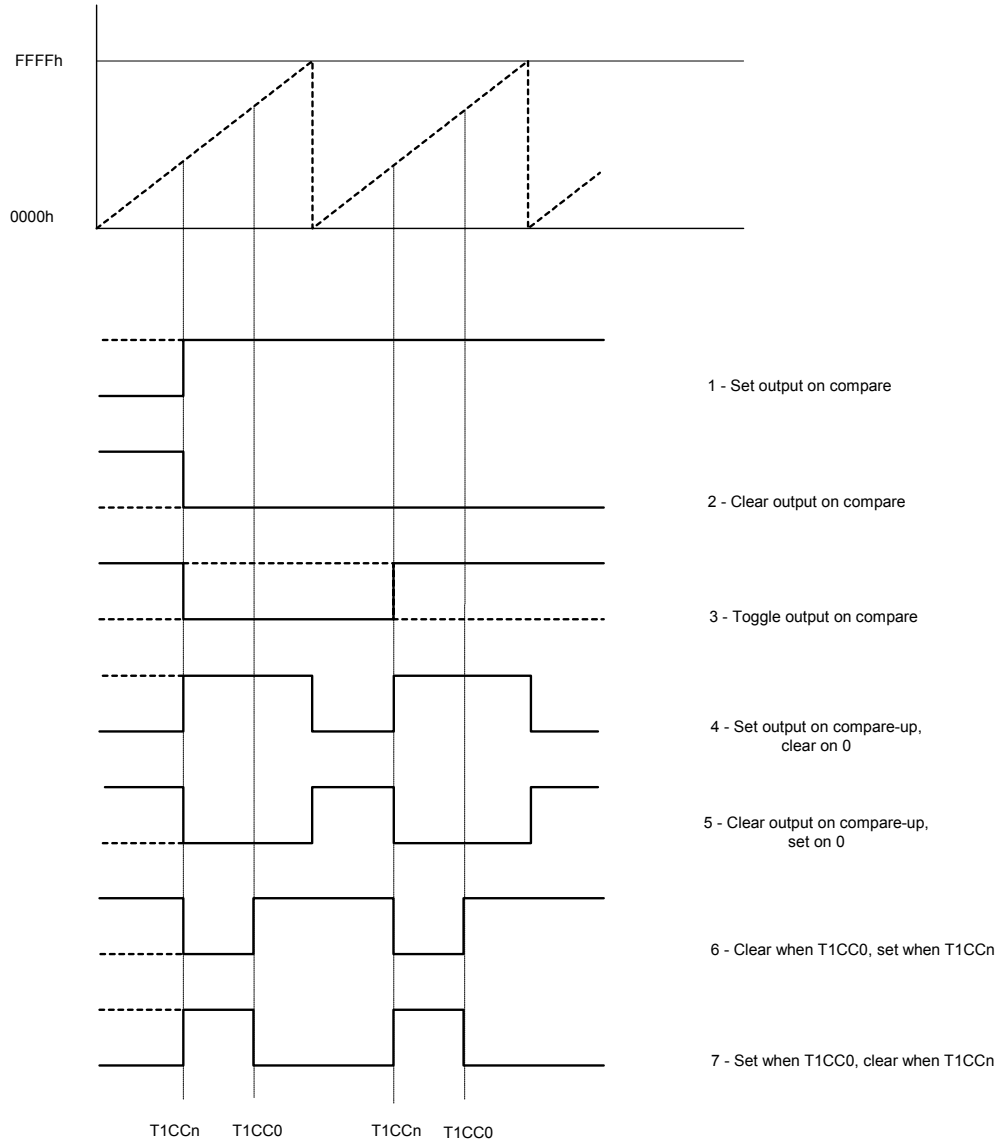


Figure 16: Output compare modes, timer free-running mode

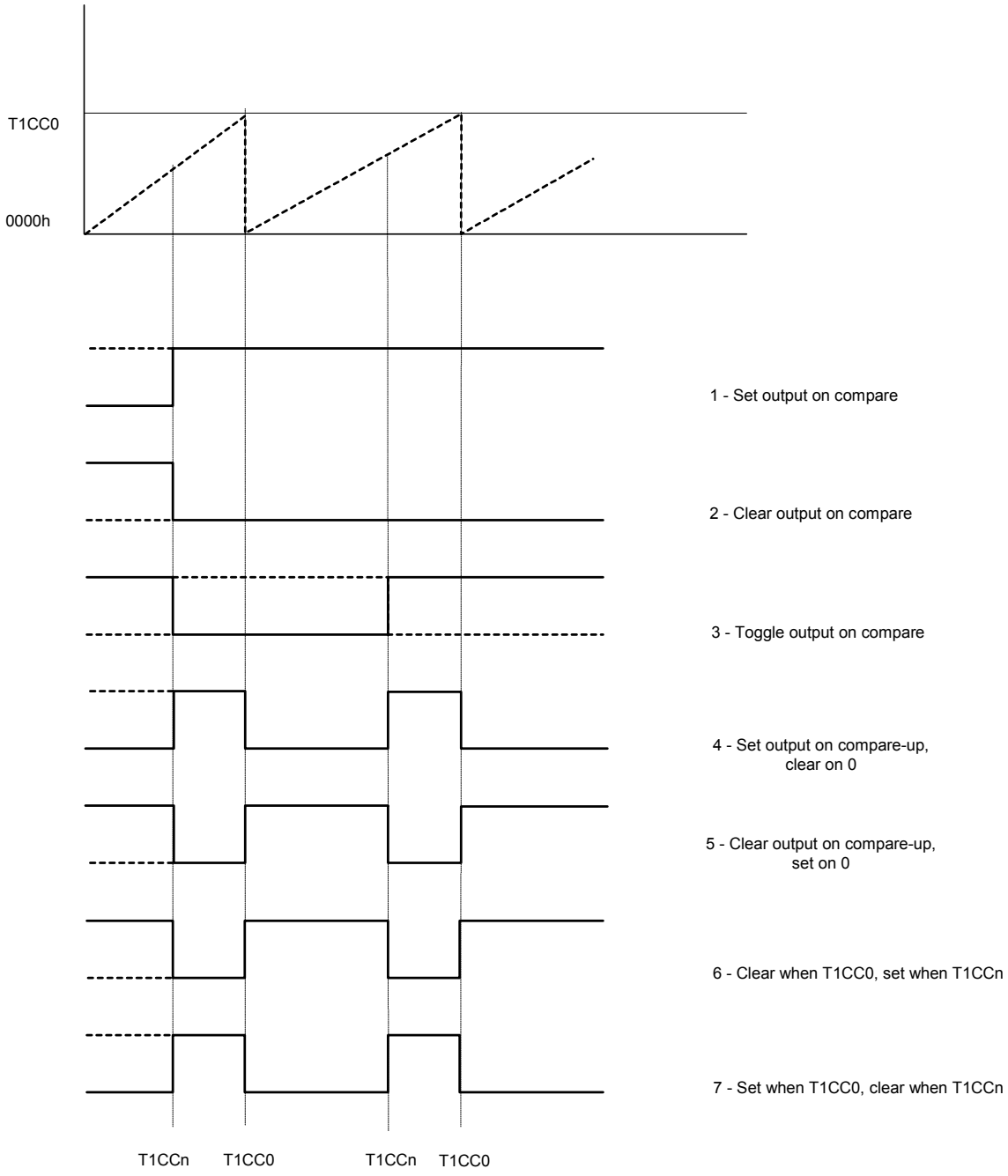


Figure 17: Output compare modes, timer modulo mode

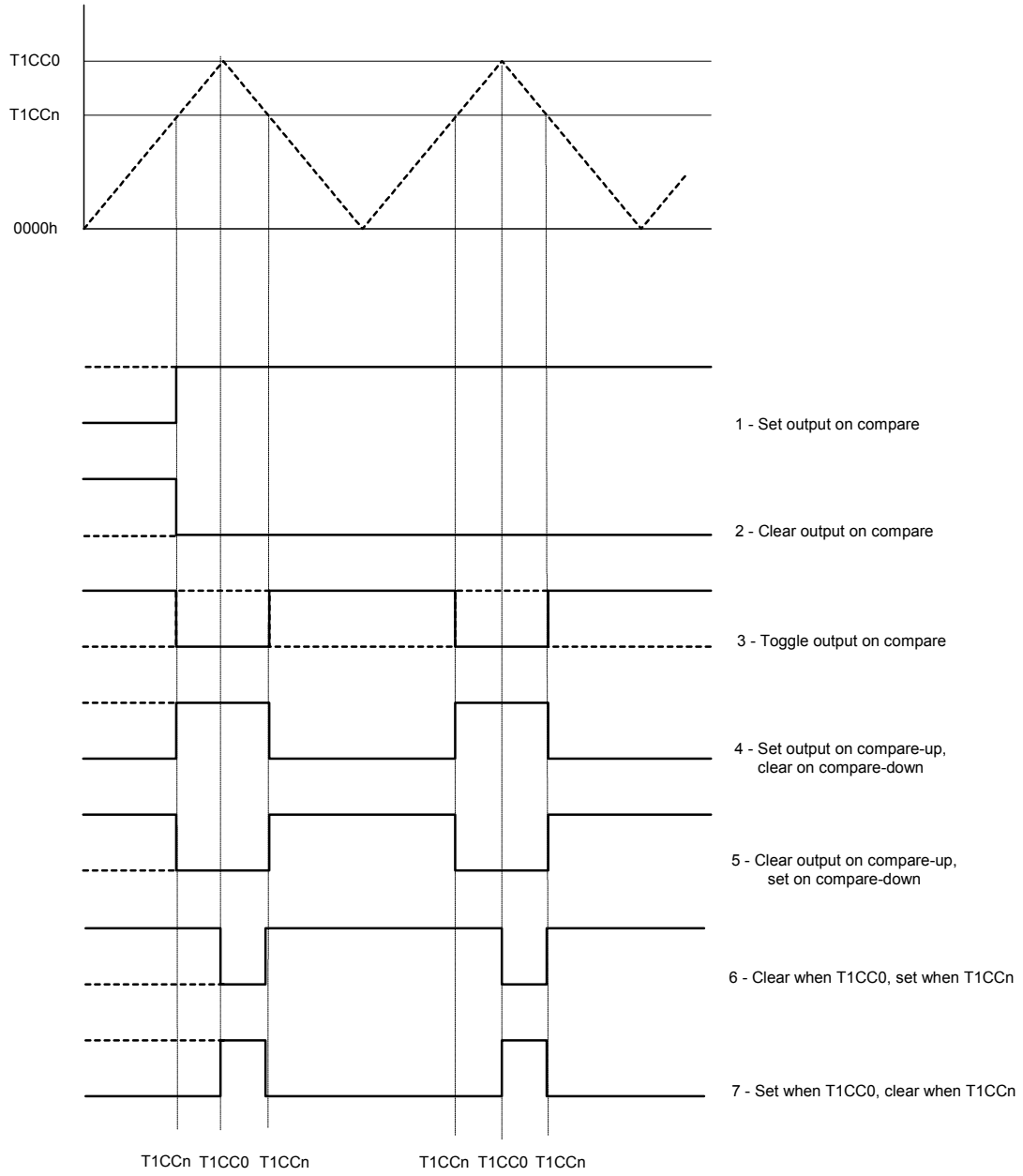


Figure 18: Output modes, timer up/down mode

13.3.9 Timer 1 Interrupts

There is one interrupt vector assigned to the timer. An interrupt request is generated when one of the following timer events occur:

- Counter reaches terminal count value.
- Input capture event.
- Output compare event

The register bits `T1CTL.OVFIF`, `T1CTL.CH0IF`, `T1CTL.CH1IF`, and `T1CTL.CH2IF` contains the interrupt flags for the terminal count value event, and the three channel compare/capture events, respectively. An interrupt request is only generated when the corresponding interrupt mask bit is set. The interrupt mask bits are `T1CCTL0.IM`, `T1CCTL1.IM`, `T1CCTL2.IM` and

TIMIF.OVFIM. If there are other pending interrupts, the corresponding interrupt flag must be cleared by software before a new interrupt request is generated. Also, enabling an interrupt mask bit will generate a new interrupt request if the corresponding interrupt flag is set.

13.3.10 Timer 1 DMA Triggers

There are three DMA triggers associated with Timer 1. These are DMA triggers T1_CH0, T1_CH1 and T1_CH2 which are generated on timer compare events as follows:

- T1_CH0 – channel 0 compare
- T1_CH1 – channel 1 compare
- T1_CH2 – channel 2 compare

See Table 37 for a list of all DMA triggers.

13.3.11 Timer 1 Registers

This section describes the Timer 1 registers that consist of the following registers:

- T1CNTH – Timer 1 Count High
- T1CNTL – Timer 1 Count Low
- T1CTL – Timer 1 Control and Status
- T1CCTLx – Timer 1 Channel x Capture/Compare Control
- T1CCxH – Timer 1 Channel x Capture/Compare Value High
- T1CCxL – Timer 1 Channel x Capture/Compare Value Low

T1CNTH (0xE3) – Timer 1 Counter High

Bit	Name	Reset	R/W	Description
7:0	CNT[15:8]	0x00	R	Timer count high order byte. Contains the high byte of the 16-bit timer counter buffered at the time T1CNTL is read.

T1CNTL (0xE2) – Timer 1 Counter Low

Bit	Name	Reset	R/W	Description
7:0	CNT[7:0]	0x00	R/W	Timer count low order byte. Contains the low byte of the 16-bit timer counter. Writing anything to this register results in the counter being cleared to 0x0000.

T1CTL (0xE4) – Timer 1 Control and Status

Bit	Name	Reset	R/W	Description
7	CH2IF	0	R/W0	Timer 1 channel 2 interrupt flag. Set when the channel 2 interrupt condition occurs. Writing a 1 has no effect.
6	CH1IF	0	R/W0	Timer 1 channel 1 interrupt flag. Set when the channel 1 interrupt condition occurs. Writing a 1 has no effect.
5	CH0IF	0	R/W0	Timer 1 channel 0 interrupt flag. Set when the channel 0 interrupt condition occurs. Writing a 1 has no effect.
4	OVFIF	0	R/W0	Timer 1 counter overflow interrupt flag. Set when the counter reaches the terminal count value in free-running or modulo mode. Writing a 1 has no effect.
3:2	DIV[1:0]	00	R/W	Prescaler divider value. Generates the active clock edge used to update the counter as follows: 00 Tick frequency/1 01 Tick frequency/8 10 Tick frequency/32 11 Tick frequency/128
1:0	MODE[1:0]	00	R/W	Timer 1 mode select. The timer operating mode is selected as follows: 00 Operation is suspended 01 Free-running, repeatedly count from 0x0000 to 0xFFFF 10 Modulo, repeatedly count from 0x0000 to T1CC0 11 Up/down, repeatedly count from 0x0000 to T1CC0 and from T1CC0 down to 0x0000

T1CCTL0 (0xE5) – Timer 1 Channel 0 Capture/Compare Control

Bit	Name	Reset	R/W	Description
7	CPSEL	0	R/W	Capture select. Timer 1 channel 0 captures on RFIF interrupt from RF transceiver or capture input pin. 0 Use normal capture input 1 Use RFIF interrupt from RF transceiver for capture
6	IM	1	R/W	Channel 0 interrupt mask. Enables interrupt request when set.
5:3	CMP[2:0]	000	R/W	Channel 0 compare mode select. Selects action on output when timer value equals compare value in T1CC0 000 Set output on compare 001 Clear output on compare 010 Toggle output on compare 011 Set output on compare-up, clear on 0 (clear on compare-down in up/down mode) 100 Clear output on compare-up, set on 0 (set on compare-down in up/down mode) 101 Not used 110 Not used 111 Not used
2	MODE	0	R/W	Mode. Select Timer 1 channel 0 capture or compare mode 0 Capture mode 1 Compare mode
1:0	CAP[1:0]	00	R/W	Channel 0 capture mode select 00 No capture 01 Capture on rising edge 10 Capture on falling edge 11 Capture on all edges

T1CC0H (0xDB) – Timer 1 Channel 0 Capture/Compare Value High

Bit	Name	Reset	R/W	Description
7:0	T1CC0[15:8]	0x00	R/W	Timer 1 channel 0 capture/compare value, high order byte

T1CC0L (0xDA) – Timer 1 Channel 0 Capture/Compare Value Low

Bit	Name	Reset	R/W	Description
7:0	T1CC0[7:0]	0x00	R/W	Timer 1 channel 0 capture/compare value, low order byte

T1CCTL1 (0xE6) – Timer 1 Channel 1 Capture/Compare Control

Bit	Name	Reset	R/W	Description
7	CPSEL	0	R/W	Capture select. Timer 1 channel 1 captures on RFIF interrupt from RF transceiver or capture input pin 0 Use normal capture input 1 Use RFIF interrupt from RF transceiver for capture
6	IM	1	R/W	Channel 1 interrupt mask. Enables interrupt request when set.
5:3	CMP[2:0]	000	R/W	Channel 1 compare mode select. Selects action on output when timer value equals compare value in T1CC1 000 Set output on compare 001 Clear output on compare 010 Toggle output on compare 011 Set output on compare-up, clear on 0 (clear on compare-down in up/down mode) 100 Clear output on compare-up, set on 0 (set on compare-down in up/down mode) 101 Clear when equal T1CC0, set when equal T1CC1 110 Set when equal T1CC0, set when equal T1CC1 111 Not used
2	MODE	0	R/W	Mode. Select Timer 1 channel 1 capture or compare mode 0 Capture mode 1 Compare mode
1:0	CAP[1:0]	00	R/W	Channel 1 capture mode select 00 No capture 01 Capture on rising edge 10 Capture on falling edge 11 Capture on all edges

T1CC1H (0xDD) – Timer 1 Channel 1 Capture/Compare Value High

Bit	Name	Reset	R/W	Description
7:0	T1CC1[15:8]	0x00	R/W	Timer 1 channel 1 capture/compare value, high order byte

T1CC1L (0xDC) – Timer 1 Channel 1 Capture/Compare Value Low

Bit	Name	Reset	R/W	Description
7:0	T1CC1[7:0]	0x00	R/W	Timer 1 channel 1 capture/compare value, low order byte

T1CCTL2 (0xE7) – Timer 1 Channel 2 Capture/Compare Control

Bit	Name	Reset	R/W	Description
7	CPSEL	0	R/W	Capture select. Timer 1 channel 2 captures on RFIF from RF transceiver or capture input pin 0 Use normal capture input 1 Use RFIF from RF transceiver for capture
6	IM	1	R/W	Channel 2 interrupt mask. Enables interrupt request when set.
5:3	CMP[2:0]	000	R/W	Channel 2 compare mode select. Selects action on output when timer value equals compare value in T1CC2 000 Set output on compare 001 Clear output on compare 010 Toggle output on compare 011 Set output on compare-up, clear on 0 (clear on compare-down in up/down mode) 100 Clear output on compare-up, set on 0 (set on compare-down in up/down mode) 101 Clear when equal T1CC0, set when equal T1CC2 110 Set when equal T1CC0, set when equal T1CC2 111 Not used
2	MODE	0	R/W	Mode. Select Timer 1 channel 2 capture or compare mode 0 Capture mode 1 Compare mode
1:0	CAP[1:0]	00	R/W	Channel 2 capture mode select 00 No capture 01 Capture on rising edge 10 Capture on falling edge 11 Capture on all edges

T1CC2H (0xDF) – Timer 1 Channel 2 Capture/Compare Value High

Bit	Name	Reset	R/W	Description
7:0	T1CC2[15:8]	0x00	R/W	Timer 1 channel 2 capture/compare value, high order byte

T1CC2L (0xDE) – Timer 1 Channel 2 Capture/Compare Value Low

Bit	Name	Reset	R/W	Description
7:0	T1CC2[7:0]	0x00	R/W	Timer 1 channel 2 capture/compare value, low order byte

The TIMIF.OVFIM register bit resides in the TIMIF register, which is described together with timer 3 and timer 4

13.4 MAC Timer (Timer 2)

The MAC timer is designed for slot timing operations for MAC layers of RF protocols. The timer includes a highly tunable prescaler allowing the user to select a timer tick interval that equals, or is an integer fraction of a transmission slot.

- 8-bit timer
- 18-bit tunable prescaler

13.4.1 Timer Operation

This section describes the operation of the timer.

The timer count can be read from the T2CT SFR register. The timer decrements by 1 at each timer tick. When the timer count reaches 0x00 the timer expires and does not wrap around. When the timer expires, the register bit T2CTL.TEX is set to 1. An interrupt request is generated when the timer expires, if the interrupt mask T2CTL.INT is 1.

When a new value is written to the timer count register, T2CT, then this value is stored in the counter immediately. If a tick and a write to T2CT occurs at the same time, the written value will be decremented before it is stored.

The timer tick period T, is given as:

T = T2PR * Val(T2CTL.TIP) clock cycles.

where the function Val(x) is set by the tick period, T2CTL.TIP and defined as

Val(00)=64

Val(01)=128

Val(10)=256

Val(11)=1024

The tick generator can be set to run freely or to run only when the timer holds a non-zero value. Whenever the tick generator is started it starts from its zero state. At this point there will be T2PR*Val(T2CTL.TIP)-1 clock cycles until the next tick.

13.4.2 Timer 2 Registers

The SFR registers associated with Timer 2 are listed in this section. These registers are the following:

- T2CTL – Timer 2 Control
- T2PRE – Timer 2 Prescaler
- T2CT – Timer 2 Count

T2CTL (0x9E) – Timer 2 Control

Bit	Name	Reset	R/W	Description
7	–	0	R/W0	Not used.
6	TEX	0	R/W0	Timer expired. This bit is set when the timer expires. Writing a 1 to this bit has no effect
5	–	0	R/W	Reserved. Should always be written as 0
4	INT	0	R/W	Interrupt enable. Select interrupt generated on timer expiration 0 Interrupt disabled 1 Interrupt enabled
3	–	0	R/W	Reserved. Should always be written as 0
2	TIG	0	R/W	Tick generator mode. 0 Tick generator is running when T2CT not equal to 00h. The tick generator will always start running form its null state. 1 Tick generator is in free-running mode. If it is not already running it will start from its null state when a '1' is written to TIG.
1:0	TIP[1:0]	00	R/W	Tick period. Selects tick period based on prescaler multiplier value. 00 Tick period is T2PR * 64 clock cycles 01 Tick period is T2PR * 128 clock cycles 10 Tick period is T2PR * 256 clock cycles 11 Tick period is T2PR * 1024 clock cycles

T2CT (0x9C) – Timer 2 Count

Bit	Name	Reset	R/W	Description
7:0	CNT[7:0]	0x00	R/W	Timer count. Contents of 8-bit counter

T2PR (0x9D) – Timer 2 Prescaler

Bit	Name	Reset	R/W	Description
7:0	PR[7:0]	0x00	R/W	Timer prescaler multiplier. 0x00 is interpreted as 256

13.5 Sleep Timer

The Sleep Timer is used to control when the CC1110 exits from the low-power modes PM1 or PM2. Thus the Sleep Timer can be used to implement a wake up functionality which enables CC1110 to periodically wake up from low-power mode and listen for incoming RF packets. Additionally the Sleep Timer can be used as a real time clock when the 32.768 kHz crystal oscillator is used.

13.5.1 Sleep Timer Operation

This section describes the operation of the timer.

The Sleep Timer consists of a 31-bit counter. The appropriate bits of this counter are selected according to a resolution setting determined by the WORCTL.WOR_RES register bits. The Sleep Timer is clocked by the 32.768 kHz clock. The source for this clock is either the 32.768 kHz crystal oscillator or the 34 kHz low power RC oscillator. The timer runs in all power modes except PM3 where all oscillators are powered off. The timer can be reset by writing 1 to the WORCTL.WOR_RESET register bit.

The Sleep Timer has two programmable timing events called Event 0 and Event 1. While in power mode PM1 or PM2, reaching Event 0 will turn on the digital voltage regulator and start the crystal oscillator. Event 1 always follows Event 0 after a programmed timeout.

The time between two consecutive Event 0's is programmed with timeout value set by a mantissa value given by WOREVT1.EVENT0 and WOREVT0.EVENT0, and an exponent value set by WORCTRL.WOR_RES. The equation is:

$$t_{Event0} = \frac{750}{f_{XOSC}} \cdot EVENT0 \cdot 2^{5 \cdot WOR\_RES}$$

The Event 1 timeout is programmed with WORCTRL.EVENT1.

Figure 19 shows the timing relationship between Event 0 timeout and Event 1 timeout.

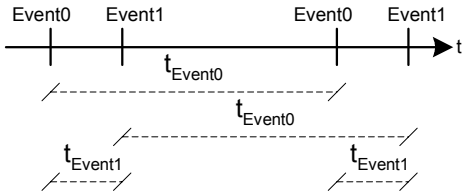


Figure 19: Event 0 and Event 1 Relationship

13.5.2 Low power RC oscillator and timing

This section applies to using the low power RC oscillator as clock source for the Sleep Timer.

The frequency of the low-power RC oscillator, which can be used as clock source for the Sleep Timer, varies with temperature and supply voltage. In order to keep the frequency as accurate as possible, the RC oscillator will be calibrated whenever possible, which is when the 26 MHz crystal oscillator is running and the chip is in the PM0 power mode. When the chip goes to PM1 or PM2, the RC oscillator will use the last valid calibration result. The frequency of the low power RC oscillator is therefore locked to the 26 MHz crystal oscillator frequency divided by 750.

13.5.3 Sleep Timer Interrupts

The Sleep Timer generates the Sleep Timer interrupt, ST, when one of the timing events Event 0 or Event 1, occur. Each of these interrupt sources can be masked using the WORIRQ interrupt mask bits. The interrupt flag bits in WORIRQ will be set when the corresponding event occurs.

13.5.4 Sleep Timer Registers

The SFR registers associated with the Sleep Timer are described in the following

WORTIME0 (0xA5) – Sleep Timer Low Byte

Bit	Name	Reset	R/W	Description
7:0	WORTIME0[7:0]	0x00	R	Timer 16-bit value, low byte. The 16 bits are selected from the 31-bit Sleep Timer according to the setting of WORCTL.WOR_RES[1:0]

WORTIME1 (0xA6) – Sleep Timer High Byte

Bit	Name	Reset	R/W	Description
7:0	WORTIME1[7:0]	0x00	R	Timer 16-bit value, high byte. The 16 bits are selected from the 31-bit Sleep Timer according to the setting of WORCTL.WOR_RES[1:0]

WOREVT1 (0xA4) – Sleep Timer Event0 Timeout High

Bit	Name	Reset	R/W	Description
7:0	EVENT0[15:8]	0x87	R/W	High byte of Event 0 timeout register $t_{Event0} = \frac{750}{f_{XOSC}} \cdot EVENT0 \cdot 2^{5 \cdot WOR\_RES}$

WOREVT0 (0xA3) – Sleep Timer Event0 Timeout Low

Bit	Name	Reset	R/W	Description
7:0	EVENT0[7:0]	107 (0x6B)	R/W	Low byte of Event 0 timeout register. The default Event 0 value gives 1.0s timeout, assuming a 26.0 MHz crystal.

WORCTL (0xA2) – Sleep Timer Control

Bit	Name	Reset	R/W	Description
7	-	0	R0	Not used
6:4	EVENT1[2:0]	111	R/W	Event 1 timeout. RC oscillator clock frequency equals F_{XOSC}/750, which is 34 – 36 kHz, depending on crystal frequency. The table below lists the number of clock cycles after Event 0 before Event 1 times out. 000 4 (0.107 ms – 0.125 ms) 001 6 (0.161 ms – 0.188 ms) 010 8 (0.214 ms – 0.25 ms) 011 12 (0.321 ms – 0.375 ms) 100 16 (0.429 ms – 0.5 ms) 101 24 (0.643 ms – 0.75 ms) 110 32 (0.857 ms – 1 ms) 111 48 (1.286 ms – 1.5 ms)
3	-	-	R0	Not used
2	WOR_RESET	0	R0/W1	Reset timer. If a 1 is written to this bit location, the timer is reset. Writing 0 will have no effect. Always read as 0
1:0	WOR_RES[1:0]	00	R/W	Timer resolution Controls the resolution and maximum timeout of the WOR timer. Adjusting the resolution does not affect the clock cycle counter: 00 31.25 us resolution, 2 s max timeout (15:0) 01 1 ms resolution, 65 s max timeout (20:5) 10 32 ms resolution, 35 min max timeout (25:10) 11 1 s resolution, 18 h max timeout (30:15)

WORIRQ (0xA1) – Sleep Timer Interrupt Control

Bit	Name	Reset	R/W	Description
7:6	–	00	R0	Not used
5	EVENT1_MASK	0	R/W	Event 1 interrupt mask 0 : interrupt is disabled 1 : interrupt is enabled
4	EVENT0_MASK	0	R/W	Event 0 interrupt mask 0 : interrupt is disabled 1 : interrupt is enabled
3:2	–	00	R0	Not used
1	EVENT1_FLAG	0	R/W0	Event 1 interrupt flag 0 : no interrupt is pending 1 : interrupt is pending
0	EVENT0_FLAG	0	R/W0	Event 0 interrupt flag 0 : no interrupt is pending 1 : interrupt is pending

13.6 8-bit Timer 3 and Timer 4

Timer 3 and 4 are 8-bit timers which support typical input capture and output compare operations using two capture/compare channels each. The timer allows general-purpose timer and waveform generation functions.

Features of Timer 3/4 are as follows:

- Dual channel operation
- Rising, falling or any edge input compare
- Set, clear or toggle output compare
- Clock prescaler for divide by 1, 2, 4, 8, 16, 32, 64, 128
- Interrupt request generated on each capture/compare and terminal count event
- DMA trigger function

13.6.1 8-bit Timer Counter

All timer functions are based on the main 8-bit counter found in Timer 3/4. The counter increments or decrements at each active clock edge. The period of the active clock edges is defined by the register bits CLKCON.TICKSPD which is further divided by the prescaler value set by TxCTL.DIV (where x refers to the timer number, 3 or 4). The counter operates as either a free-running counter, a down counter, a modulo counter or as an up/down counter.

It is possible to read the 8-bit counter value through the SFR TxCNT where x refers to the timer number, 3 or 4.

The possibility to clear and halt the counter is given with TxCTL control register settings. The counter is started when a 1 is written to TxCTL.START. If a 0 is written to TxCTL.START the counter halts at its present value.

13.6.2 Timer 3/4 Mode Control

In general the control register TxCTL is used to control the timer operation. The timer modes are described in the following sections.

13.6.2.1 Free-running Mode

In the free-running mode of operation the counter starts from 0x00 and increments at each active clock edge. When the counter reaches 0xFF the counter is loaded with 0x00 and continues incrementing its value. When the terminal count value 0xFF is reached (i.e. an overflow occurs), the interrupt flag TIMIF.TxOVFIF is set. If the corresponding interrupt mask bit TxCTL.OVFIM is set, an interrupt request is generated. The free-running mode can be used to generate independent time intervals and output signal frequencies.

13.6.2.2 Down mode

In the down mode, after the timer has been started, the counter is loaded with the contents in TxCC. The counter then counts down to 0x00 and remains at 0x00. The flag TIMIF.TxOVFIF is set when 0x00 is reached. If the corresponding interrupt mask bit TxCTL.OVFIM is set, an interrupt request is generated. The timer down mode can generally be used in applications where an event timeout interval is required.

13.6.2.3 Modulo Mode

When the timer operates in modulo mode the 8-bit counter starts at 0x00 and increments at each active clock edge. When the counter reaches the terminal count value held in register TxCC the counter is reset to 0x00 and continues to increment. The flag TIMIF.TxOVFIF is set when on this event. If the corresponding interrupt mask bit TxCTL.OVFIM is set, an interrupt request is generated. The modulo mode can be used for applications where a period other than 0xFF is required.

13.6.2.4 Up/down Mode

In the up/down timer mode, the counter repeatedly starts from 0x00 and counts up until the value held in TxCC is reached and then the counter counts down until 0x00 is reached. This timer mode is used when symmetrical output pulses are required with a period other than 0xFF, and therefore allows implementation of centre-aligned PWM output applications.

Clearing the counter by writing to TxCTL.CLR will also reset the count direction to the count up from 0x00 mode.

13.6.3 Channel Mode Control

The channel modes for each channel are set by the control and status registers TxCCTLn, where n is the channel number, 0 or 1. The settings include input capture and output compare modes.

13.6.4 Input Capture Mode

When the channel is configured as an input capture channel, the I/O pin associated with that channel is configured as an input. After the timer has been started, either a rising edge, falling edge or any edge on the input pin triggers a capture of the 8-bit counter contents into the associated capture register. Thus the timer is able to capture the time when an external event takes place.

The channel input pins are synchronized to the internal system clock. Thus pulses on the input pins must have a minimum duration greater than the system clock period.

Note: before an input/output pin can be used by the timer, the required I/O pin must be configured as a Timer 3/4 peripheral pin as described in sections 13.1.4.4 and 13.1.4.5.

The contents of the 8-bit capture registers, is read out from registers TxCCn.

When a capture takes place the interrupt flag corresponding to the actual channel is set. This interrupt flag is TIMIF.TxCHnIF. An interrupt request is generated if the corresponding interrupt mask bit TxCCTLn.IM is set.

13.6.5 Output Compare Mode

In output compare mode the I/O pin associated with a channel should be configured as an output. After the timer has been started, the contents of the counter is compared with the contents of the channel compare register TxCCn. If the compare register equals the counter contents, the output pin is set, reset or toggled according to the compare output mode setting of TxCCTL.CMP1:0. Note that all edges on output pins are glitch-free when operating in a given compare output mode.

For simple PWM use, output compare modes 4 and 5 are preferred.

Writing to the compare register TxCC0 does not take effect on the output compare value until the counter value is 0x00. Writing to the compare register TxCC1 takes effect immediately.

When a compare occurs the interrupt flag corresponding to the actual channel is set. This interrupt flag is TIMIF.TxCHnIF. An interrupt request is generated if the corresponding interrupt mask bit TxCCTLn.IM is set.

13.6.6 Timer 3 and 4 interrupts

There is one interrupt vector assigned to each of the timers. These are T3IF (interrupt 11) and T4IF (interrupt 12). An interrupt request is generated when one of the following timer events occur:

- Counter reaches terminal count value.
- Input capture event.
- Output compare event

The SFR register TIMIF contains all interrupt flags for Timer 3 and Timer 4. The register bits TIMIF.TxOVFIF and TIMIF.TxCHnIF contain the interrupt flags for the two terminal count value events and the four channel compare/capture events, respectively. An interrupt request is only generated when the corresponding interrupt mask bit is set. If there are other pending interrupts, the corresponding interrupt flag must be cleared by the CPU before a new interrupt request can be generated. Also, enabling an interrupt mask bit will generate a new interrupt request if the corresponding interrupt flag is set.

13.6.7 Timer 3 and Timer 4 DMA triggers

There are two DMA triggers associated with Timer 3 and two DMA triggers associated with Timer 4. These are the following:

- T3_CH0 : Timer 3 channel 0 compare
- T3_CH1 : Timer 3 channel 1 compare
- T4_CH0 : Timer 4 channel 0 compare
- T4_CH0 : Timer 4 channel 1 compare

Refer to section 13.2 on page 78, for a description on use of DMA channels.

13.6.8 Timer 3 and 4 registers

The Timer 3 and 4 registers are described on the following pages.

T3CNT (0xCA) – Timer 3 Counter

Bit	Name	Reset	R/W	Description
7:0	CNT[7:0]	0x00	R	Timer count byte. Contains the current value of the 8-bit counter.

T3CTL (0xCB) – Timer 3 Control

Bit	Name	Reset	R/W	Description
7:5	DIV[2:0]	00	R/W	Prescaler divider value. Generates the active clock edge used to clock the timer from CLKCON.TICKSPD as follows: 000 Tick frequency /1 001 Tick frequency /2 010 Tick frequency /4 011 Tick frequency /8 100 Tick frequency /16 101 Tick frequency /32 110 Tick frequency /64 111 Tick frequency /128
4	START	0	R/W	Start timer. Normal operation when set, suspended when cleared
3	OVFIM	1	R/W0	Overflow interrupt mask 0 : interrupt is disabled 1 : interrupt is enabled
2	CLR	0	R0/W1	Clear counter. Writing high resets counter to 0x00
1:0	MODE[1:0]	00	R/W	Timer 3 mode. Select the mode as follows: 00 Free running, repeatedly count from 0x00 to 0xFF 01 Down, count from T3CC0 to 0x00 10 Modulo, repeatedly count from 0x00 to T3CC0 11 Up/down, repeatedly count from 0x00 to T3CC0 and down to 0x00

T3CCTL0 (0xCC) – Timer 3 Channel 0 Capture/Compare Control

Bit	Name	Reset	R/W	Description
7	–	0	R0	Unused
6	IM	1	R/W	Channel 0 interrupt mask 0 : interrupt is disabled 1 : interrupt is enabled
5:3	CMP[2:0]	000	R/W	Channel 0 compare output mode select. Specified action on output when timer value equals compare value in T3CC0 000 Set output on compare 001 Clear output on compare 010 Toggle output on compare 011 Set output on compare-up, clear on 0 (clear on compare-down in up/down mode) 100 Clear output on compare-up, set on 0 (set on compare-down in up/down mode) 101 Set output on compare, clear on 0xFF 110 Clear output on compare, set on 0x00 111 Not used
2	MODE	0	R/W	Mode. Select Timer 3 channel 0 capture or compare mode 0 Capture mode 1 Compare mode
1:0	CAP[1:0]	00	R/W	Channel 0 capture mode select 00 No capture 01 Capture on rising edge 10 Capture on falling edge 11 Capture on all edges

T3CC0 (0xCD) – Timer 3 Channel 0 Capture/Compare Value

Bit	Name	Reset	R/W	Description
7:0	VAL[7:0]	0x00	R/W	Timer capture/compare value channel 0

T3CCTL1 (0xCE) – Timer 3 Channel 1 Capture/Compare Control

Bit	Name	Reset	R/W	Description
7	–	0	R0	Unused
6	IM	1	R/W	Channel 1 interrupt mask 0 : interrupt is disabled 1 : interrupt is enabled
5:3	CMP[2:0]	000	R/W	Channel 1 compare output mode select. Specified action on output when timer value equals compare value in T3CC1 000 Set output on compare 001 Clear output on compare 010 Toggle output on compare 011 Set output on compare-up, clear on 0 (clear on compare-down in up/down mode) 100 Clear output on compare-up, set on 0 (set on compare-down in up/down mode) 101 Set output on compare, clear on T3CC0 110 Clear output on compare, set on T3CC0 111 Not used
2	MODE	0	R/W	Mode. Select Timer 3 channel 1 capture or compare mode 0 Capture mode 1 Compare mode
1:0	CAP[1:0]	00	R/W	Channel 1 capture mode select 00 No capture 01 Capture on rising edge 10 Capture on falling edge 11 Capture on all edges

T3CC1 (0xCF) – Timer 3 Channel 1 Capture/Compare Value

Bit	Name	Reset	R/W	Description
7:0	VAL[7:0]	0x00	R/W	Timer capture/compare value channel 1

T4CNT (0xEA) – Timer 4 Counter

Bit	Name	Reset	R/W	Description
7:0	CNT[7:0]	0x00	R	Timer count byte. Contains the current value of the 8-bit counter.

T4CTL (0xEB) – Timer 4 Control

Bit	Name	Reset	R/W	Description
7:5	DIV[2:0]	00	R/W	Prescaler divider value. Generates the active clock edge used to clock the timer from CLKCON.TICKSPD as follows: 000 Tick frequency /1 001 Tick frequency /2 010 Tick frequency /4 011 Tick frequency /8 100 Tick frequency /16 101 Tick frequency /32 110 Tick frequency /64 111 Tick frequency /128
4	START	0	R/W	Start timer. Normal operation when set, suspended when cleared
3	OVFIM	1	R/W0	Overflow interrupt mask
2	CLR	0	R0/W1	Clear counter. Writing high resets counter to 0x00
1:0	MODE[1:0]	00	R/W	Timer 4 mode. Select the mode as follows: 00 Free running, repeatedly count from 0x00 to 0x00 01 Down, count from T4CC0 to 0x00 10 Modulo, repeatedly count from 0x00 to T4CC0 11 Up/down, repeatedly count from 0x00 to T4CC0 and down to 0x00

T4CCTL0 (0xEC) – Timer 4 Channel 0 Capture/Compare Control

Bit	Name	Reset	R/W	Description
7	–	0	R0	Unused
6	IM	1	R/W	Channel 0 interrupt mask
5:3	CMP[2:0]	000	R/W	Channel 0 compare output mode select. Specified action on output when timer value equals compare value in T4CC0 000 Set output on compare 001 Clear output on compare 010 Toggle output on compare 011 Set output on compare-up, clear on 0 (clear on compare-down in up/down mode) 100 Clear output on compare-up, set on 0 (set on compare-down in up/down mode) 101 Set output on compare, clear on 0x00 110 Clear output on compare, set on 0x00 111 Not used
2	MODE	0	R/W	Mode. Select Timer 4 channel 0 capture or compare mode 0 Capture mode 1 Compare mode
1:0	CAP[1:0]	00	R/W	Channel 0 capture mode select 00 No capture 01 Capture on rising edge 10 Capture on falling edge 11 Capture on all edges

T4CC0 (0xED) – Timer 4 Channel 0 Capture/Compare Value

Bit	Name	Reset	R/W	Description
7:0	VAL[7:0]	0x00	R/W	Timer capture/compare value channel 0

T4CCTL1 (0xEE) – Timer 4 Channel 1 Capture/Compare Control

Bit	Name	Reset	R/W	Description
7	–	0	R0	Unused
6	IM	1	R/W	Channel 1 interrupt mask
5:3	CMP[2:0]	000	R/W	Channel 1 compare output mode select. Specified action on output when timer value equals compare value in T4CC1 000 Set output on compare 001 Clear output on compare 010 Toggle output on compare 011 Set output on compare-up, clear on 0 (clear on compare-down in up/down mode) 100 Clear output on compare-up, set on 0 (set on compare-down in up/down mode) 101 Set output on compare, clear on T4CC0 110 Clear output on compare, set on T4CC0 111 Not used
2	MODE	0	R/W	Mode. Select Timer 4 channel 1 capture or compare mode 0 Capture mode 1 Compare mode
1:0	CAP[1:0]	00	R/W	Channel 1 capture mode select 00 No capture 01 Capture on rising edge 10 Capture on falling edge 11 Capture on all edges

T4CC1 (0xEF) – Timer 4 Channel 1 Capture/Compare Value

Bit	Name	Reset	R/W	Description
7:0	VAL[7:0]	0x00	R/W	Timer capture/compare value channel 1

TIMIF (0xD8) – Timers 1/3/4 Interrupt Mask/Flag

Bit	Name	Reset	R/W	Description
7	–	0	R0	Unused
6	OVFIM	1	R/W	Timer 1 overflow interrupt mask
5	T4CH1IF	0	R/W0	Timer 4 channel 1 interrupt flag 0 : no interrupt is pending 1 : interrupt is pending Writing a 1 has no effect.
4	T4CH0IF	0	R/W0	Timer 4 channel 0 interrupt flag 0 : no interrupt is pending 1 : interrupt is pending Writing a 1 has no effect.
3	T4OVFIF	0	R/W0	Timer 4 overflow interrupt flag 0 : no interrupt is pending 1 : interrupt is pending Writing a 1 has no effect.
2	T3CH1IF	0	R/W0	Timer 3 channel 1 interrupt flag 0 : no interrupt is pending 1 : interrupt is pending Writing a 1 has no effect.
1	T3CH0IF	0	R/W0	Timer 3 channel 0 interrupt flag 0 : no interrupt is pending 1 : interrupt is pending Writing a 1 has no effect.
0	T3OVFIF	0	R/W0	Timer 3 overflow interrupt flag 0 : no interrupt is pending 1 : interrupt is pending Writing a 1 has no effect.

13.7 ADC

13.7.1 ADC Introduction

The ADC supports up to 14-bit analog-to-digital conversion. The ADC includes an analog multiplexer with up to eight individually configurable channels, reference voltage generator and conversion results written to memory through DMA. Several modes of operation are available.

The main features of the ADC are as follows:

- Selectable decimation rates which also sets the resolution (8 to 13 bits).
- Eight individual input channels, single-ended or differential
- Reference voltage selectable as internal, external single ended, external differential or AVDD.
- Interrupt request generation
- DMA triggers at end of conversions
- Temperature sensor input
- Battery measurement capability

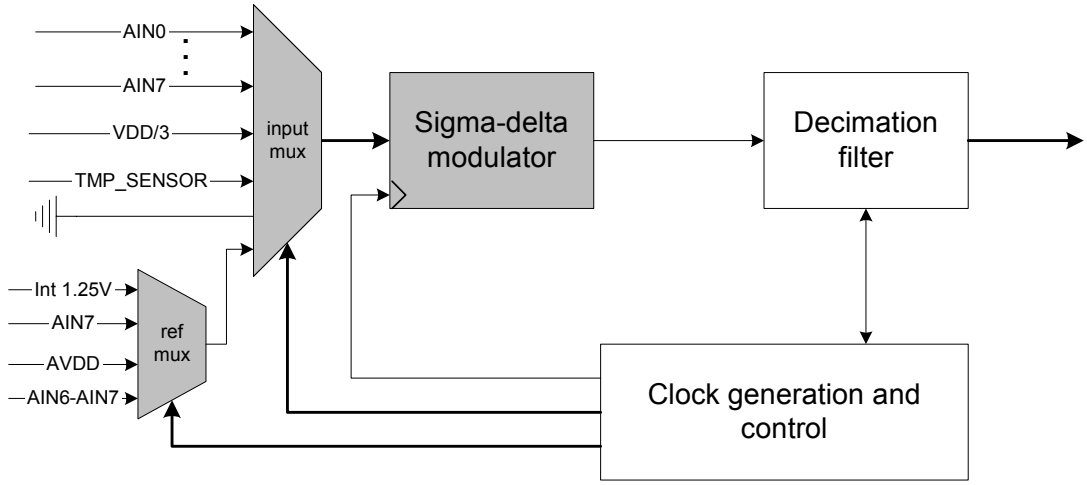


Figure 20: ADC block diagram.

13.7.2 ADC Operation

This section describes the general setup and operation of the ADC and describes the usage of the ADC control and status registers accessed by the CPU.

13.7.2.1 ADC Core

The ADC includes an ADC capable of converting an analog input into a digital representation with up to 14 bits resolution. The ADC uses a selectable positive reference voltage.

13.7.2.2 ADC Inputs

The signals on the `P0` port pins can be used as ADC inputs. In the following these port pint will be referred to as the AIN0-AIN7 pins. The input pins AIN0-AIN7 are connected to the ADC. The ADC can be set up to automatically perform a sequence of conversions and optionally perform an extra conversion from any channel when the sequence is completed.

It is possible to configure the inputs as single-ended or differential inputs. In the case where differential inputs are selected,

the differential inputs consist of the input pairs AIN0-1, AIN2-3, AIN4-5 and AIN6-7.

In addition to the input pins AIN0-AIN7, the output of an on-chip temperature sensor can be selected as an input to the ADC for temperature measurements.

It is also possible to select a voltage corresponding to AVDD/3 as an ADC input. This input allows the implementation of e.g. a battery monitor in applications where this feature is required.

13.7.2.3 ADC conversion sequences

The ADC will perform a sequence of conversions, and move the results to memory (through DMA) without any interaction from the CPU.

The ADCCON2.SCH register bits are used to define an ADC conversion sequence, from the ADC inputs. A conversion sequence will contain a conversion from each channel from 0 up to and including the channel number programmed in ADCCON2.SCH when ADCCON2.SCH is set to a value less than 8.

The single-ended inputs AIN0 to AIN7 are represented by channel numbers 0 to 7 in ADCCON2.SCH. Channel numbers 8 to 11 represent the differential inputs consisting of AIN0-AIN1, AIN2-AIN3, AIN4-AIN5 and AIN6-AIN7. Channel numbers 12 to 15 represent GND, internal voltage reference, temperature sensor and AVDD/3, respectively.

When ADCCON2.SCH is set to a value between 8 and 12, the sequence will start at channel 8. For even higher settings, only single conversions are performed. In addition to this sequence of conversions, the ADC can be programmed to perform a single conversion from any channel as soon as the sequence has completed. This is called an extra conversion and is controlled with the ADCCON3 register.

The conversion sequence can also be influenced with the ADCCFG register (see section 13.1.5 on page 64). The eight analog inputs to the ADC come from I/O pins, which are not necessarily programmed to be analog inputs. If a channel should normally be part of a sequence, but the corresponding analog input is disabled in the ADCCFG, then that channel will be skipped. For channels 8 to 12, both input pins must be enabled.

13.7.2.4 ADC Operating Modes

This section describes the operating modes and initialization of conversions.

The ADC has three control registers: ADCCON1, ADCCON2 and ADCCON3. These registers are used to configure the ADC and to report status.

The ADCCON1.EOC bit is a status bit that is set high when a conversion ends and cleared when ADCH is read.

The ADCCON1.ST bit is used to start a sequence of conversions. A sequence will start when this bit is set high, ADCCON1.STSEL=”11” and no conversion is currently running. When the sequence is completed, this bit is automatically cleared.

The ADCCON1.STSEL bits select which event that will start a new sequence of conversions. The options which can be selected are rising edge on external pin P2_0, end of previous sequence, a Timer 1 channel 0 compare event or ADCCON1.ST=’1’.

The ADCCON2 register controls how the sequence of conversions is performed.

ADCCON2.SREF is used to select the reference voltage. The reference voltage should only be changed when no conversion is running.

The ADCCON2.SDIV bits select the decimation rate (and thereby also the resolution and time required to complete a conversion and sample rate). The decimation rate should only be changed when no conversion is running.

The last channel of a sequence is selected with the ADCCON2.SCH bits.

The ADCCON3 register controls the channel number, reference voltage and decimation rate for the extra conversion. The extra conversion will take place immediately after the ADCCON3 register is updated. The coding of the register bits is exactly as for ADCCON2.

13.7.2.5 ADC Conversion Results

The digital conversion result is represented in two's complement form. For 13-bit resolution, the digital conversion result is 8191 when the analogue input is equal to the VREF, and the conversion result is -8192 when the analogue input is equal to –VREF, where VREF is the selected positive voltage reference.

The digital conversion result is available when ADCCON1.EOC is set to 1, in ADCH and ADCL.

When the ADCCON2.SCH bits are read, they will indicate which channel the conversion result in ADCL and ADCH apply to.

13.7.2.6 ADC Reference Voltage

The positive reference voltage for analogue-to-digital conversions is selectable as either an internally generated 1.25V voltage, the AVDD pin, the external voltage applied to the AIN7 input pin or the differential voltage applied to the AIN6-AIN7 inputs.

It is possible to select the reference voltage as the input to the ADC in order to perform a conversion of the reference voltage e.g. for calibration purposes. Similarly, it is possible to select the ground terminal GND as an input.

13.7.2.7 ADC Conversion Timing

The 26 MHz crystal oscillator should be selected when the ADC is used. The ADC runs on a clock which is the 26 MHz system clock source divided by 6 to give a 4.33 MHz ADC clock. The delta sigma modulator and decimation filter both use the ADC clock for their calculations.

13.7.2.10 ADC Registers

This section describes the ADC registers.

The time required to perform a conversion depends on the selected decimation rate. When the decimation rate is set to for instance 128, the decimation filter uses exactly 128 of the ADC clock periods to calculate the result. When a conversion is started, the input multiplexer is allowed 16 ADC clock cycles to settle in case the channel has been changed since the previous conversion. The 16 clock cycles settling time applies to all decimation rates. Thus in general, the conversion time is given by:

Tconv = (decimation rate + 16) x 0.23 µs

13.7.2.8 ADC Interrupts

The ADC will generate an interrupt when an extra conversion has completed. An interrupt is not generated when a conversion from the sequence is completed.

13.7.2.9 ADC DMA Triggers

The ADC will generate a DMA trigger every time a conversion from the sequence has completed. When an extra conversion completes, no DMA trigger is generated.

There is one DMA trigger for each of the eight channels defined by the first eight possible settings for ADCCON2.SCH . The DMA trigger is active when a new sample is ready from the conversion for the channel. The DMA triggers are named ADC_CHx in Table 37 on page 84.

In addition there is one DMA trigger, ADC_CHALL, which is active when new data is ready from any of the channels in the ADC conversion sequence.

ADCL (0xBA) – ADC Data Low

Bit	Name	Reset	R/W	Description
7:2	ADC[5:0]	0x00	R	Least significant part of ADC conversion result.
1:0	-	00	R0	Not used. Always read as 0

ADCH (0xBB) – ADC Data High

Bit	Name	Reset	R/W	Description
7:0	ADC[13:6]	0x00	R	Most significant part of ADC conversion result.

ADCCON1 (0xB4) – ADC Control 1

Bit	Name	Reset	R/W	Description
7	EOC	0	R H0	End of conversion Cleared when both ADCH and ADCL has been read. If a new conversion is completed before the previous data has been read, the EOC bit will remain high. 0 conversion not complete 1 conversion completed
6	ST	0	R/W1	Start conversion. Read as 1 until conversion has completed 0 no conversion in progress 1 start a conversion sequence if ADCCON1.STSEL = “11” and no sequence is running.
5:4	STSEL[1:0]	11	R/W	Start select. Selects which event that will start a new conversion sequence. 00 External trigger on P2_0 pin. 01 Full speed. Do not wait for triggers. 10 Timer1 channel1 output = 1 11 ADCCON1.ST = 1
3:2	RCTRL[1:0]	00	R/W	Controls the 16 bit random generator. When written “01” or “10”, the setting will automatically return to “00” when operation has completed. 00 Normal operation. (13x unrolling) 01 Clock the LFSR once (no unrolling). 10 Seeding from modulator. NOTE: The ADC must be running in order for the seeding to start. 11 Stopped. Random generator is turned off.
1:0	-	11	R/W	Reserved. Set to 11.

ADCCON2 (0xB5) – ADC Control 2

Bit	Name	Reset	R/W	Description
7:6	SREF[1:0]	00	R/W	Selects reference voltage used for the sequence of conversions 00 Internal 1.25V reference 01 External reference on AIN7 pin 10 AVDD pin 11 External reference on AIN6-AIN7 differential input
5:4	SDIV[1:0]	01	R/W	Sets the decimation rate for channels included in the sequence of conversions. The decimation rate also determines the resolution and time required to complete a conversion. 00 64 dec rate (8 bits resolution) 01 128 dec rate (10 bits resolution) 10 256 dec rate (12 bits resolution) 11 512 dec rate (13 bits resolution)
3:0	SCH[3:0]	00	R/W	Sequence Channel Select. Selects the end of the sequence. A sequence can either be from AIN0 to AIN7 (SCH<=7) or from the differential input AIN0-AIN1 to AIN6-AIN7 (8<=SCH<=11). For other settings, only single conversions are performed. When read, these bits will indicate the channel number of current conversion result. 0000 AIN0 0001 AIN1 0010 AIN2 0011 AIN3 0100 AIN4 0101 AIN5 0110 AIN6 0111 AIN7 1000 AIN0-AIN1 1001 AIN2-AIN3 1010 AIN4-AIN5 1011 AIN6-AIN7 1100 GND 1101 Positive voltage reference 1110 Temperature sensor 1111 AVDD/3

ADCCON3 (0xB6) – ADC Control 3

Bit	Name	Reset	R/W	Description
7:6	EREF[1:0]	00	R/W	Selects reference voltage used for the extra conversion 00 Internal 1.25V reference 01 External reference on AIN7 pin 10 AVDD pin 11 External reference on AIN6-AIN7 differential input
5:4	EDIV[1:0]	00	R/W	Sets the decimation rate used for the extra conversion. The decimation rate also determines the resolution and time required to complete the conversion. 00 64 dec rate (8 bits resolution) 01 128 dec rate (10 bits resolution) 10 256 dec rate (12 bits resolution) 11 512 dec rate (13 bits resolution)
3:0	ECH[3:0]	0000	R/W	Extra channel select. Selects the channel number of the extra conversion that is carried out after a conversion sequence has ended. As long as these bits remain at “0000”, no extra conversion is performed. If the ADC is not running, writing to these bits will trigger an immediate single conversion from the selected extra channel. The bits are automatically cleared when the extra conversion has finished. 0000 AIN0 0001 AIN1 0010 AIN2 0011 AIN3 0100 AIN4 0101 AIN5 0110 AIN6 0111 AIN7 1000 AIN0-AIN1 1001 AIN2-AIN3 1010 AIN4-AIN5 1011 AIN6-AIN7 1100 GND 1101 Positive voltage reference 1110 Temperature sensor 1111 VDD/3

13.8 Random Number Generator

13.8.1 Introduction

The random number generator has the following features.

- Generate pseudo-random bytes which can be read by the CPU or used directly by the Command Strobe Processor.
- Calculate CRC16 of bytes that are written to RNDH.
- Seeded by a random bit-stream from the delta-sigma modulator in the ADC.
- Seeded by value written to RNDL.

The random number generator is a 16-bit Linear Feedback Shift Register (LFSR) with polynomial $X^{16} + X^{15} + X^2 + 1$ (i.e. CRC16). It uses different levels of unrolling depending on the operation it performs. The basic version (no unrolling) is shown below.

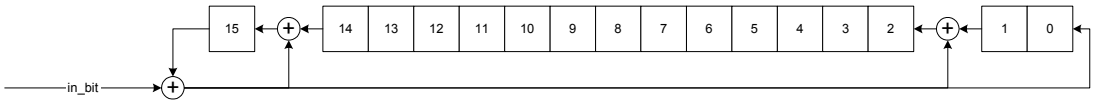


Figure 21: Basic structure of the Random Number Generator

The random number generator is turned off when ADCCON1.RCTRL=”11”.

13.8.2 Random Number Generator Operation

The operation of the random number generator is controlled through a combination of the ADCCON1.RCTRL bits and input signals from other modules. The current value of the 16-bit shift register in the LFSR can be read from the RNDH and RNDL registers.

13.8.2.1 Semi random sequence generation

The default operation (ADCCON1.RCTRL=”00”) is to clock the LFSR once (13x unrolling) each time the Command Strobe Processor reads the random value. This leads to the availability of a fresh pseudo-random byte from the LSB end of the LFSR.

Another way to update the LFSR is to set ADCCON1.RCTRL=”01”. This will clock the LFSR once (no unrolling) and the ADCCON1.RCTRL bits will automatically be cleared when the operation has completed.

13.8.2.2 Seeding

When a true random value is required, the LFSR can be seeded with random bits from the delta-sigma modulator. For seeding option, the ADC must be performing a conversion, so the actual seeding will not start until the ADC is running. This seeding process is started by setting ADCCON1.RCTRL=”10”. When the seeding has completed, these bits are automatically cleared.

The LFSR can also be seeded from software by simply writing to the RNDL register twice. Each time the RNDL register is written, the 8 LSB of the LFSR is copied to the 8 MSB and the 8 LSBs are replaced with the new data byte that was written to RNDL.

13.8.2.3 CRC16

The LFSR can also be used to calculate the CRC value of a sequence of bytes. Writing to the RNDH register will trigger a CRC calculation. The new byte is processed from the MSB end and an 8x unrolling is used, so that a new byte can be written to RNDH every clock cycle.

Note that the LFSR must be properly seeded by writing to RNDL, before the

CRC calculations start. Commonly used seed values are 0x0000 or 0xFFFF.

13.8.3 Registers

The random number generator registers are described in this section.

RNDL (0xBC) – Random Number Generator Data Low Byte

Bit	Name	Reset	R/W	Description
[7:0]	RNDL[7:0]	0xFF	R/W	Random value/seed or CRC result, low byte When used for random number generation writing this register twice will seed the random number generator. Writing to this register copies the 8 LSBs of the LFSR to the 8 MSBs and replaces the 8 LSBs with the data value written. The value returned when reading from this register is the 8 LSBs of the LSFR. When used for random number generation, reading this register returns the 8 LSBs of the random number. When used for CRC calculations, reading this register returns the 8 LSBs of the CRC result.

RNDH (0xBD) – Random Number Generator Data High Byte

Bit	Name	Reset	R/W	Description
[7:0]	RNDH[7:0]	0xFF	R/W	Random value or CRC result/input data, high byte When written, a CRC16 calculation will be triggered, and the data value written is processed starting with the MSB bit. The value returned when reading from this register is the 8 MSBs of the LSFR. When used for random number generation, reading this register returns the 8 MSBs of the random number. When used for CRC calculations, reading this register returns the 8 MSBs of the CRC result.

13.9 AES Coprocessor

With the CC1110, data encryption can be performed using a dedicated coprocessor which supports Advanced Encryption Standard, AES. The coprocessor allows encryption/decryption to be performed with minimal CPU usage.

The coprocessor has the following features:

- ECB, CBC, CFB, OFB, CTR and CBC-MAC modes.
- Hardware support for CCM mode
- 128-bits key and IV/Nonce
- DMA transfer trigger capability

13.9.1 AES Operation

To encrypt a message, the following procedure must be followed:

- Load key
- Load initialization vector (IV)
- Download and upload data for encryption/decryption.

The AES coprocessor works on blocks of 128 bits. A block of data is loaded into the coprocessor, encryption is performed and the result must be read out before the next block can be processed. Before each block load, a dedicated start command must be sent to the coprocessor.

13.9.2 Key and IV

Before a key or IV/nonce load starts, an appropriate load key or IV must be issued to the coprocessor. When loading the IV it is important to also set the correct mode.

A key load or IV load operation aborts any processing that could be running.

The key, once loaded, stays valid until a key reload takes place.

The IV must be downloaded before the beginning of each message (not block).

Both key and IV are cleared by a reset.

13.9.3 Padding of input data

AES works on blocks of 128 bits. If the last block contains less than 128 bits, it must be padded with zeros when written to the coprocessor.

13.9.4 Interface to CPU

The CPU communicates with the coprocessor using three SFR registers:

- ENCCS, Encryption control and status register
- ENCDI, Encryption input register
- ENCDO, Encryption output register

Read/write to the status register is done by the CPU, while read/write the input/output register is intended for use together with direct memory access (DMA).

Two DMA channels must be used, one for input data and one for output data. The DMA channels must be initialized before a start command is written to the ENCCS. Writing a start command generates a DMA trigger and the transfer is started. After each block is processed, an interrupt is generated. The interrupt is used to issue a new start command to the ENCCS.

13.9.5 Modes of operation

ECB and CBC modes are performed as described in section 13.9.1

When using CFB, OFB and CTR mode, the 128 bits blocks are divided into four 32 bits blocks. 32 bits are loaded into the AES coprocessor and the resulting 32 bits are read out. This continues until all 128 bits are encrypted. The only time one has to consider this is if data is loaded/read directly using the CPU. When using DMA, this is handled automatically by the DMA triggers generated by the AES coprocessor.

Both encryption and decryption are performed similarly.

The CBC-MAC mode is a variant of the CBC mode. When performing CBC-MAC, data is downloaded to the coprocessor as one block at a time, except for the last block. Before the last block is loaded, the mode must be changed to CBC. The last block is then downloaded and the block uploaded will be the MAC value.

CCM is a combination of CBC-MAC and CTR. Parts of the CCM must therefore be done in software. The following section gives a short explanation of the necessary steps to be done.

13.9.5.1 CBC-MAC

When performing CBC-MAC encryption, data is only downloaded to the coprocessor in CBC-MAC mode except for the last block, one block at a time. Before the last block is loaded, the mode is changed to CBC. The last block is downloaded and the block uploaded is the message MAC.

CBC-MAC decryption is similar to encryption. The message MAC uploaded must be compared with the MAC to be verified.

13.9.5.2 CCM mode

To encrypt a message under CCM mode, the following sequence can be conducted (key is already loaded):

Message Authentication Phase

This phase takes place during steps 1-6 shown in the following.

(1) The software loads the IV with zeros.

(2) The software creates the block B0. The layout of block B0 is shown in Figure 22.

.

	Name B0					Designation First block for authentication in CCM mode									
Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Name	Flag	NONCE									L_M					

Figure 22: Message Authentication Phase Block 0

There is no restriction on the NONCE value. L_M is the message length in bytes.

The content of the Authentication Flag byte is described in Figure 23.

L is set to 6 in this example. So, L-1 is set to 5. M and A_Data can be set to any value.

	Name FLAG/B0		Designation Authentication Flag Field for CCM mode					
Bit	7	6	5	4	3	2	1	0
Name	Reserved	A_Data	(M-2)/2			L-1		
Value	0	x	x	x	x	1	0	1

Figure 23: Authentication Flag Byte

(3) If some Additional Authentication Data (called a later) is needed (that is A_Data =1), the software creates the A_Data length field, called L(a) by :

- **(3a)** If l(a)=0, (that is A_Data =0), then L(a) is the empty string. We note l(a) the length of a in octets.
- **(3b)** If $0 < l(a) < 2^{16} - 2^8$, then L(a) is the 2-octets encoding of l(a).

The Additional Authentication Data is appended to the A_Data length field L(a). The Additional Authentication Blocks is padded with zeros until the last Additional Authentication Block is full. There is no restriction on the length of a.

AUTH-DATA = L(a) + Authentication Data + (zero padding)

(**4)** The last block of the message is padded with 0s until full (that is if its length is not a multiple of 128).

(5) The software concatenates the block B0, the Additional Authentication Blocks if any, and the message;

Input message = B0 + AUTH-DATA + Message + (zero padding of message)

(6) Once the input message authentication by CBC-MAC is finished, the software leaves the uploaded buffer contents unchanged (M=16), or keeps only the buffer's higher M bytes unchanged, while setting the lower bits to 0 (M != 16).

The result is called T.

Message Encryption

(7) The software creates the key stream block A0. Note that L=6, with the current example of the CTR generation. The content is shown in Figure 24:

	Name A0					Designation First CTR value for CCM mode										
Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Name	Flag	NONCE											CTR			

Figure 24: Message Encryption Phase Block 0

Note that any value but zero works for the CTR value.

The content of the Encryption Flag byte is described in Figure 25

	Name FLAG/A0		Designation Encryption Flag Field for CCM mode					
Bit	7	6	5	4	3	2	1	0
Name	Reserved		-			L-1		
Value	0	0	0	0	0	1	0	1

Figure 25: Encryption Flag Byte

(**8)** The software loads A0 by selecting a Load IV/Nonce command. To do so, it sets Mode to CFB or OFB at the same time it selects the Load IV/Nonce command.

(9) The software calls a CFB or an OFB encryption on the authenticated data T. The uploaded buffer contents stay unchanged (M=16), or only its first M bytes stay unchanged, the others being set to 0 (M-16). The result is U, which will be used later.

(**10)** The software calls a CTR mode encryption right now on the still padded message blocks. It does not have to reload the IV/CTR.

(11) The encrypted authentication data U is appended to the encrypted message. This gives the final result, c.

Result c = encrypted message(m) + U

Message Decryption

CCM Mode decryption

In the coprocessor, the automatic generation of CTR works on 32 bits, therefore the maximum length of a message is 128 x 2^{32} bits, that is 2^{36} bytes, which can be written in a six-bit word. So, the value L is set to 6. To decrypt a CCM mode processed message, the following sequence can be conducted (key is already loaded):

Message Parsing Phase

(1) The software parses the message by separating the M rightmost octets, namely U, and the other octets, namely string C.

(2) C is padded with zeros until it can fill an integer number of 128-bit blocks;

(3) U is padded with zeros until it can fill a 128-bit block.

(4) The software creates the key stream block A0. It is done the same way as for CCM encryption.

(5) The software loads A0 by selected a Load IV/Nonce command. To do so, it sets Mode to CFB or OFB at the same time it selects the IV load.

(6) The software calls a CFB or an OFB encryption on the encrypted authenticated data U. The uploaded buffer contents stay unchanged (M=16), or only its first M bytes stay unchanged, the others being set to 0 (M!=16). The result is T.

(**7)** The software calls a CTR mode decryption right now on the encrypted message blocks C. It does not have to reload the IV/CTR.

Reference Authentication tag generation

This phase is identical to the Authentication Phase of CCM encryption. The only difference is that the result is named MACTag (instead of T).

Message Authentication checking Phase

The software compares T with MACTag.

13.9.6 Sharing the AES coprocessor between layers

The AES coprocessor is a common resource shared by all layers. The AES coprocessor can only be used by one instance one at a time. It is therefore necessary to implement some kind of software semaphore to allocate and de-allocate the resource.

13.9.7 AES Interrupts

The AES interrupt, ENC, is produced when encryption or decryption of a block is completed. The interrupt enable bit is `IEN0.ENCIE` and the interrupt flag is `S0CON.ENCIF`.

13.9.8 AES DMA Triggers

There are two DMA triggers associated with the AES coprocessor. These are ENC_DW which is active when input data needs to be downloaded to the `ENCDI` register, and ENC_UP which is active when output data needs to be uploaded from the `ENCDO` register.

The `ENCDI` and `ENCDO` registers should be set as destination and source locations for DMA channels used to transfer data to or from the AES coprocessor.

13.9.9 AES Registers

The AES coprocessor registers have the layout shown in this section.

ENCCS (0xB3) – Encryption Control and Status

Bit	Name	Reset	R/W	Description
7	–	0	R0	Not used, always read as 0
6:4	MODE[2:0]	000	R/W	Encryption/decryption mode 000 CBC 001 CFB 010 OFB 011 CTR 100 ECB 101 CBC MAC 110 Not used 111 Not used
3	RDY	1	R	Encryption/decryption ready status 0 Encryption/decryption in progress 1 Encryption/decryption is completed
2:1	CMD[1:0]	0	R/W	Command to be performed when a 1 is written to ST. 00 encrypt block 01 decrypt block 10 load key 11 load IV/nonce
0	ST	0	R/W1 H0	Start processing command set by CMD. Must be issued for each command or 128 bits block of data. Cleared by hardware

ENCDI (0xB1) – Encryption Input Data

Bit	Name	Reset	R/W	Description
7:0	DIN[7:0]	0x00	R/W	Encryption input data

ENCDO (0xB2) – Encryption Output Data

Bit	Name	Reset	R/W	Description
7:0	DOUT[7:0]	0x00	R/W	Encryption output data

13.10 Power Management

This section describes the Power Management Controller. The Power Management Controller controls the use of power modes and clock control to achieve low-power operation.

13.10.1 Power Management Introduction

The ***CC1110*** uses different operating modes, or power modes, to allow low-power operation. Ultra-low-power operation is obtained by turning off power supply to modules to avoid static (leakage) power consumption and also by using clock gating to reduce dynamic power consumption.

The various operating modes are enumerated and shall be designated as power modes (PMx). The power modes are:

- PM0
  Clock oscillators on, voltage regulator on
- PM1
  32.768/34 kHz oscillators on, voltage regulator on
- PM2
  32.768/34 kHz oscillators on, voltage regulator off
- PM3
  All clock oscillators off, voltage regulator off

13.10.1.1 PM0

PM0 is the full functional mode of operation where the CPU and peripherals are active. The voltage regulator is turned on.

PM0 is used for normal operation.

13.10.1.2 PM1

In PM1, the high-speed oscillators are powered down thereby halting the CPU and peripherals. The voltage regulator, the power-on reset, external interrupts, the 32.768/34 kHz oscillators and sleep timer peripherals are active. I/O pins retain the I/O mode and output value set before entering PM1. When PM1 is entered, a power down sequence is run. When the device is taken out of PM1 to PM0, the high-speed oscillators are started. The device will run on the high speed RC oscillator until the high speed crystal oscillator has settled and has been selected.

PM1 is used when the expected time until a wakeup event is relatively short since PM1 uses a fast power down/up sequence.

13.10.1.3 PM2

PM2 has the second lowest power consumption. In PM2 the power-on reset, external interrupts, 32.768/34 kHz oscillators and sleep timer peripherals are active. I/O pins retain the I/O mode and output value set before entering PM2. All other internal circuits are powered down. The voltage regulator is also turned off. When PM2 is entered, a power down sequence is run.

PM2 is used when it is relatively long until the expected time of a wakeup event, since the power up/down sequence is relatively long. PM2 is typically entered when using the sleep timer as the wakeup event.

13.10.1.4 PM3

PM3 is used to achieve the operating mode with the lowest power consumption. In PM3 all internal circuits that are powered from the voltage regulator are turned off. The internal voltage regulator and all oscillators are also turned off.

Power-on reset and external interrupts are the only functions that are operating in this mode. I/O pins retain the I/O mode and output value set before entering PM3. Only a reset or external interrupt condition will wake the device up and place it into PM0. The contents of RAM and registers are preserved in this mode. PM3 uses the same power down/up sequence as PM2.

PM3 is used to achieve ultra low power consumption when waiting for an external event.

13.10.2 Power Management Control

The required power mode is selected by the MODE bits in the SLEEP control register. Setting the SFR register PCON.IDLE bit after setting the MODE bits, enters the selected power mode.

An interrupt from port pins, or sleep timer or a power-on reset will wake the device and bring it into PM0 by resetting the MODE bits.

13.10.3 System Clock

The system clock is derived from the selected system clock source, which is the high-speed (26 MHz) crystal oscillator or the high-speed (13 MHz) RC oscillator. The CLKCON.OSC bit selects the source of the system clock. Note that to use the RF transceiver the high speed crystal oscillator must be selected and stable.

Note that each time the CLKCON.OSC bit is altered, then the CLKCON.CLKSPD bit must also be altered in order for the system to know the frequency of the system clock.

When the SLEEP.XOSC_STB is 1, the 26 MHz crystal oscillator is stable and can be used as the source for the system clock.

The oscillator not selected as the system clock source, will be set in power-down mode by setting SLEEP.OSC_PD to 1. Thus the high-speed RC oscillator may be turned off when the 26 MHz crystal oscillator has been selected as system clock source and vice versa. When SLEEP.OSC_PD is 0, both oscillators are powered up and running.

13.10.4 High-speed oscillators

Two high speed oscillators are present in the device. The high-speed crystal oscillator startup time may be too long for some applications, therefore the device will run on the high-speed RC oscillator until crystal oscillator is stable. The high-speed RC oscillator consumes less power than the crystal oscillator, but since it is not as accurate as the crystal oscillator it can not be used for RF transceiver operation.

13.10.5 32.768/34 kHz oscillators

Two low power oscillators are present in the device. By default the 34 kHz low power RC oscillator is enabled (see Table 10 on page 16). The low power RC oscillator consumes less power, but is less accurate than the 32.768 kHz crystal oscillator. When the high speed crystal oscillator is running the low power RC oscillator is continuously calibrated.

13.10.6 Timer Tick generation

The power management controller generates a tick or enable signal for the peripheral timers, thus acting as a prescaler for the timers. This is a global clock division for Timer 1, Timer 3 and Timer 4. The tick speed is programmed from 0.203 to 26 MHz in the CLKCON.TICKSPD register.

13.10.7 Data Retention

In power modes PM2 and PM3 parts of SRAM will retain its contents. The content of internal registers is also retained in PM2/3.

The XDATA memory locations 0xF000-0xFFFF (4096 bytes) retains data in PM2/3. Please note one exception as given below.

The XDATA memory locations 0xFDAA-0xFEFF (342 bytes) will lose all data when PM2/3 is entered. These locations will contain undefined data when PM0 is re-entered.

The registers which retain their contents are the CPU registers, peripheral registers and RF registers therefore switching to the low-power modes PM2/3 appears transparent to software.

13.10.8 Power Management Registers

This section describes the Power Management registers.

PCON (0x87) – Power Mode Control

Bit	Name	Reset	R/W	Description
7:2	–	0x00	R/W	Not used.
1	–	0	R0	Not used, always read as 0.
0	IDLE	0	R0/W H0	Power mode control. Writing a 1 to this bit forces entry to the power mode set by SLEEP.MODE. This bit is always read as 0

SLEEP (0xBE) – Sleep Mode Control

Bit	Name	Reset	R/W	Description
7	-	0	R0	Unused
6	XOSC_STB	0	R	26 MHz Crystal oscillator stable status: 0 – oscillator is not powered up or not yet stable 1 – oscillator is powered up and stable
5	HFRC_STB	0	R	High speed RC oscillator stable status: 0 – oscillator is not powered up or not yet stable 1 – oscillator is powered up and stable
4:3	RST[1:0]	XX	R	Status bit indicating the cause of the last reset. If there are multiple resets, the register will only contain the last event. 00 – Power-on reset 01 – External reset 10 – Watchdog timer reset
2	OSC_PD	0	R/W H0	XOSC and HS RCOSC power down setting. The bit is cleared if the CLKCON.OSC bit is toggled. Also, if there is a calibration in progress and the CPU attempts to set the bit, the bit will be updated at the end of calibration: 0 – Both oscillators powered up 1 – Oscillator not selected by CLKCON.OSC bit powered down
1:0	MODE[1:0]	00	R/W	Sleep mode setting: 00 – Power mode 0 01 – Power mode 1 10 – Power mode 2 11 – Power mode 3

CLKCON (0xC6) – Clock Control

Bit	Name	Reset	R/W	Description
7	OSC32K	1	R/W	32 kHz clock oscillator select: 0 – 32.768 kHz crystal oscillator 1 – 34 kHz low power RC oscillator
6	OSC	1	R/W	System clock oscillator select: 0 – 26 MHz crystal oscillator 1 – 13 MHz HF RC oscillator This setting will only take effect when the selected oscillator is powered up and stable. If the selected oscillator is not powered up, then writing this bit will power it up.
5:3	TICKSPD[2:0]	001	R/W	Timer ticks output setting, can not be higher than system clock setting given by OSC bit setting 000 – System clock 001 – System clock / 2 010 – System clock / 4 011 – System clock / 8 100 – System clock / 16 101 – System clock / 32 110 – System clock / 64 111 – System clock / 128
2:1	–	00	R/W	Reserved. Must always be set to 000.
0	CLKSPD	1	R/W	Clock Speed. Sets system clock frequency together with the OSC bit setting 0 – 32 MHz 1 – 16 MHz This bit must always follow the value of the OSC bit, which selects which oscillator to use.

13.11 Power On Reset

The CC1110 includes a Power On Reset (POR) in order to protect the memory contents during supply voltage variations and provide correct initialization during power-on.

When power is initially applied to the CC1110 the Power On Reset (POR) will hold the device in reset state until the supply voltage reaches above the Power On Reset voltage as defined in Table 4 on page 12.

Figure 26 shows the POR operation with the 1.8V (typical) regulated supply voltage together with the active low reset signal shown in the bottom of the figure.

The cause of the last reset can read from the register bits SLEEP.RST.

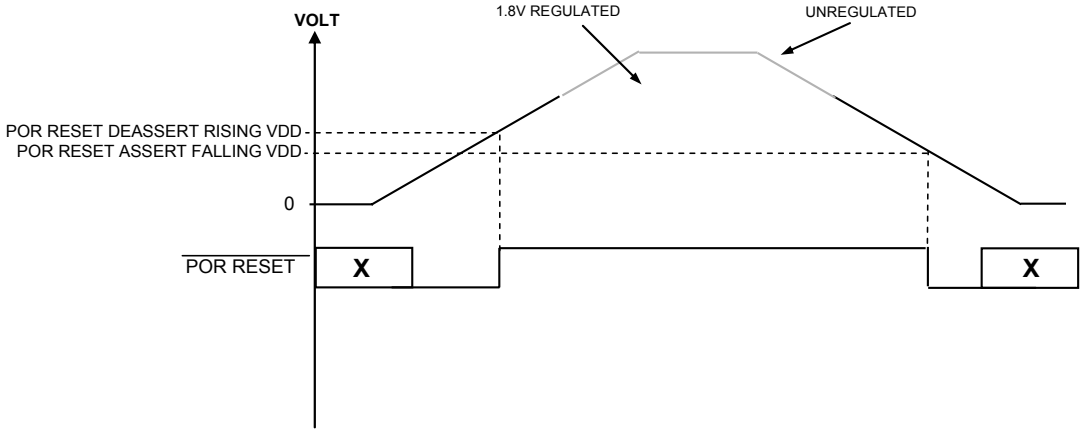


Figure 26 : Power On Reset Operation

13.12 Watchdog Timer

The watchdog timer (WDT) is intended as a recovery method in situations where the CPU may be subjected to a software upset. The WDT shall reset the system when software fails to clear the WDT within a selected time interval. The watchdog can be used in applications that are subject to electrical noise, power glitches, electrostatic discharge etc., or where high reliability is required. If the watchdog function is not needed in an application, it is possible to configure the watchdog timer to be used as an interval timer that can be used to generate interrupts at selected time intervals.

The features of the watchdog timer are as follows:

- Four selectable timer intervals
- Watchdog mode
- Timer mode
- Interrupt request generation in timer mode
- Clock independent from system clock

The WDT is configured as either a watchdog timer or as a timer for general-purpose use. The operation of the WDT module is controlled by the WDCTL register. The watchdog timer consists of an 15-bit counter clocked by the 32.768 or 34 kHz oscillator clock. Note that the contents of the 15-bit counter is not user-accessible.

13.12.1 Watchdog mode

The watchdog timer is disabled after a system reset. To set the WDT in watchdog mode the WDCTL.MODE bit is set to 0. The watchdog timer counter starts incrementing when the enable bit WDCTL.EN is set to 1. When the timer is enabled in watchdog mode it is not possible to disable the timer. Therfore, writing a 0 to WDCTL.EN has no effect if a 1 was already written to this bit when WDCTL.MODE was 0.

The WDT can operate with a watchdog timer clock frequency of 32.768 kHz. This clock frequency gives time-out periods equal to 1.9 ms, 15.625 ms, 0.25 s and 1 s corresponding to the count value settings 64, 512, 8192 and 32768 respectively.

If the counter reaches the selected timer interval value, the watchdog timer generates a reset signal for the system. If a watchdog clear sequence is performed before the counter reaches the selected timer interval value, the counter is reset to 0x0000 and continues incrementing its value. The watchdog clear sequence consists of writing 0xA to WDCTL.CLR[3:0] followed by writing 0x5 to the same register bits within one half of a watchdog clock period. If this complete sequence is not performed, the watchdog timer generates a reset signal for the system. Note that as long as a correct watchdog clear sequence begins within the selected timer interval, the counter is reset when the complete sequence has been received.

When the watchdog timer has been enabled in watchdog mode, it is not possible to change the mode by writing to the WDCTL.MODE bit. The timer interval value can be changed by writing to the WDCTL.INT[1:0] bits.

Note that it is recommended that user software clears the watchdog timer at the same time as the timer interval value is changed, in order to avoid an unwanted watchdog reset.

In watchdog mode, the WDT does not produce an interrupt request.

13.12.2 Timer mode

To set the WDT in normal timer mode, the WDCTL.MODE bit is set to 1. When register bit WDCTL.EN is set to 1, the timer is started and the counter starts incrementing. When the counter reaches the selected interval value, the timer will produce an interrupt request.

In timer mode, it is possible to clear the timer contents by writing a 1 to WDCTL.CLR[0]. When the timer is cleared the contents of the counter is set to 0x0000. Writing a 0 to the enable bit

WDCTL.EN stops the timer and writing 1 restarts the timer from 0x0000.

The timer interval is set by the WDCTL.INT[1:0] bits. In timer mode, a reset will not be produced when the timer interval has been reached.

13.12.3 Watchdog Timer Example

Figure 27 shows an example of periodical clearing of an active watchdog timer.

```
; clear watchdog timer
   MOV    WDCTL,#ABh
   MOV    WDCTL,#5Bh
```

Figure 27: WDT Example

13.12.4 Watchdog Timer Register

This section describes the register for the Watchdog Timer.

WDCTL (0xC9) – Watchdog Timer Control

Bit	Name	Reset	R/W	Description
7:4	CLR[3:0]	0000	R/W	Clear timer. When 0xA followed by 0x5 is written to these bits, the timer is loaded with 0x0. Note the timer will only be cleared when 0x5 is written within 0.5 watchdog clock period after 0xA was written. Writing to these bits when EN is 0 has no effect. These bits are always be read as 0000.
3	EN	0	R/W	Enable timer. When a 1 is written to this bit the timer is enabled and starts incrementing. Writing a 0 to this bit in timer mode stops the timer. Writing a 0 to this bit in watchdog mode has no effect. 0 Timer disabled (stop timer) 1 Timer enabled
2	MODE	0	R/W	Mode select. This bit selects the watchdog timer mode. 0 Watchdog mode 1 Timer mode
1:0	INT[1:0]	00	R/W	Timer interval select. These bits select the timer interval defined as a given number of 32.768 or 34 kHz oscillator periods. 00 clock period x 32768 (typical 1 s) 01 clock period x 8192 (typical 0.25 s) 10 clock period x 512 (typical 15.625 ms) 11 clock period x 64 (typical 1.9 ms)

13.13 USART

USART0 and USART1 are serial communications interfaces that can be operated separately in either asynchronous UART mode or in synchronous SPI mode. The two USARTs have identical function, and are assigned to separate I/O pins. Refer to section 13.1 for I/O configuration.

13.13.1 UART mode

For asynchronous serial interfaces, the UART mode is provided. In the UART mode the interface uses a two-wire or four-wire interface consisting of the pins RXD, TXD and optionally RTS and CTS. The UART mode of operation includes the following features:

- 8 or 9 data bits
- Odd, even or no parity
- Configurable start and stop bit level
- Configurable LSB or MSB first transfer
- Independent receive and transmit interrupts
- Independent receive and transmit DMA triggers
- Parity and framing error status

The UART mode provides full duplex asynchronous transfers, and the synchronization of bits in the receiver does not interfere with the transmit function. A UART byte transfer consists of a start bit, eight data bits, an optional ninth data or parity bit, and one or two stop bits. Note that the data transferred is referred to as a byte, although the data can actually consist of eight or nine bits.

The UART operation is controlled by the USART Control and Status registers, `UxCSR` and the UART Control register `UxUCR` where x is the USART number, 0 or 1.

The UART mode is selected when `UxCSR.MODE` is set to 1.

13.13.1.1 UART Transmit

A UART transmission is initiated when the USART Receive/Transmit Data Buffer, `UxDBUF` register is written. The byte is transmitted on TXDx output pin. The `UxDBUF` register is double-buffered.

The `UxCSR.ACTIVE` bit goes high when the byte transmission starts and low when it ends. When byte transmission ends the `UxCSR.TX_BYTE` bit is set. An interrupt request is generated when the `UxDBUF` register is ready to accept new transmit data. This happens immediately after the transmission has been started, hence a new data byte value can be loaded into the data buffer while the byte is being transmitted.

13.13.1.2 UART Receive

Data reception on the UART is initiated when a 1 is written to the `UxCSR.RE` bit. The UART will then search for a valid start bit on the RXDx input pin and set the `UxCSR.ACTIVE` bit high. When a valid start bit has been detected the received byte is shifted into the receive register. The `UxCSR.RX_BYTE` bit is set when the operation has completed, and a receive interrupt is generated. At the same time `UxCSR.ACTIVE` will go low.

The received data byte is available through the `UxDBUF` register. When `UxDBUF` is read, `UxCSR.RX_BYTE` is cleared by hardware.

13.13.1.3 UART Hardware Flow Control

Hardware flow control is enabled when the `UxUCR.FLOW` bit is set to 1. The *RTS* output will then be driven low when the receive register is empty and reception is enabled. Transmission of a byte will not occur before the CTS input goes low.

13.13.1.4 UART Character Format

If the `BIT9` and `PARITY` bits in register `UxUCR` are set high, parity generation and detection is enabled. The parity is computed and transmitted as the ninth bit, and during reception, the parity is computed and compared to the received ninth bit. If there is a parity error, the `UxCSR.ERR` bit is set high. This bit is cleared when `UxCSR` is read.

The number of stop bits to be transmitted is set to one or two bits determined by the register bit UxUCR.SPB. The receiver will always check for one stop bit. If the first stop bit received during reception is not at the expected stop bit level, a framing error is signaled by setting register bit UxCSR.FE high. UxCSR.FE is cleared when UxCSR is read. The receiver will check both stop bits when UxUCR.SPB is set.

13.13.2 SPI Mode

This section describes the SPI mode of operation for synchronous communication. In SPI mode, the USART communicates with an external system through a 3-wire or 4-wire interface. The interface consists of the pins MOSI, MISO, SCK and SS_N. Refer to section 13.1 for description of how the USART pins are assigned to the I/O pins.

The SPI mode includes the following features:

- 3-wire and 4-wire SPI interface
- Master and slave modes
- Configurable SCK polarity and phase
- Configurable LSB or MSB first transfer

The SPI mode is selected when UxCSR.MODE is set to 0.

In SPI mode, the USART can be configured to operate either as an SPI master or as an SPI slave by writing the UxCSR.SLAVE bit.

13.13.2.1 SPI Master Operation

An SPI byte transfer in master mode is initiated when the UxDBUF register is written. The USART generates the SCK serial clock using the baud rate generator (see section 13.13.3) and shifts the provided byte from the transmit register onto the MOSI output. At the same time the receive register shifts in the received byte from the MISO input pin.

The UxCSR.ACTIVE bit goes high when the transfer starts and low when the transfer ends. When the transfer ends, the UxCSR.RX_BYTE and UxCSR.TX_BYTE bits are set. A receive interrupt is generated when new received data is ready in the UxDBUF USART Receive/Transmit Data register.

The polarity and clock phase of the serial clock SCK is selected by UxGCR.CPOL and UxGCR.CPHA as shown in Figure 28. The order of the byte transfer is selected by the UxGCR.ORDER bit.

At the end of the transfer, the received data byte is available for reading from the UxDBUF.

A transmit interrupt is generated when the unit is ready to accept another data byte for transmission. Since UxDBUF is double-buffered, this happens just after the transmission has been initiated.

13.13.2.2 SPI Slave Operation

An SPI byte transfer in slave mode is controlled by the external system. The data on the MISO input is shifted into the receive register controlled by the serial clock SCK which is an input in slave mode. At the same time the byte in the transmit register is shifted out onto the MOSI output.

The UxCSR.ACTIVE bit goes high when the transfer starts and low when the transfer ends. Then the UxCSR.RX_BYTE and UxCSR.TX_BYTE bits are set and a receive interrupt is generated.

The expected polarity and clock phase of SCK is selected by UxGCR.CPOL and UxGCR.CPHA as shown in Figure 28. The expected order of the byte transfer is selected by the UxGCR.ORDER bit.

At the end of the transfer, the received data byte is available for reading from UxDBUF

The transmit interrupt is generated at the start of the operation.

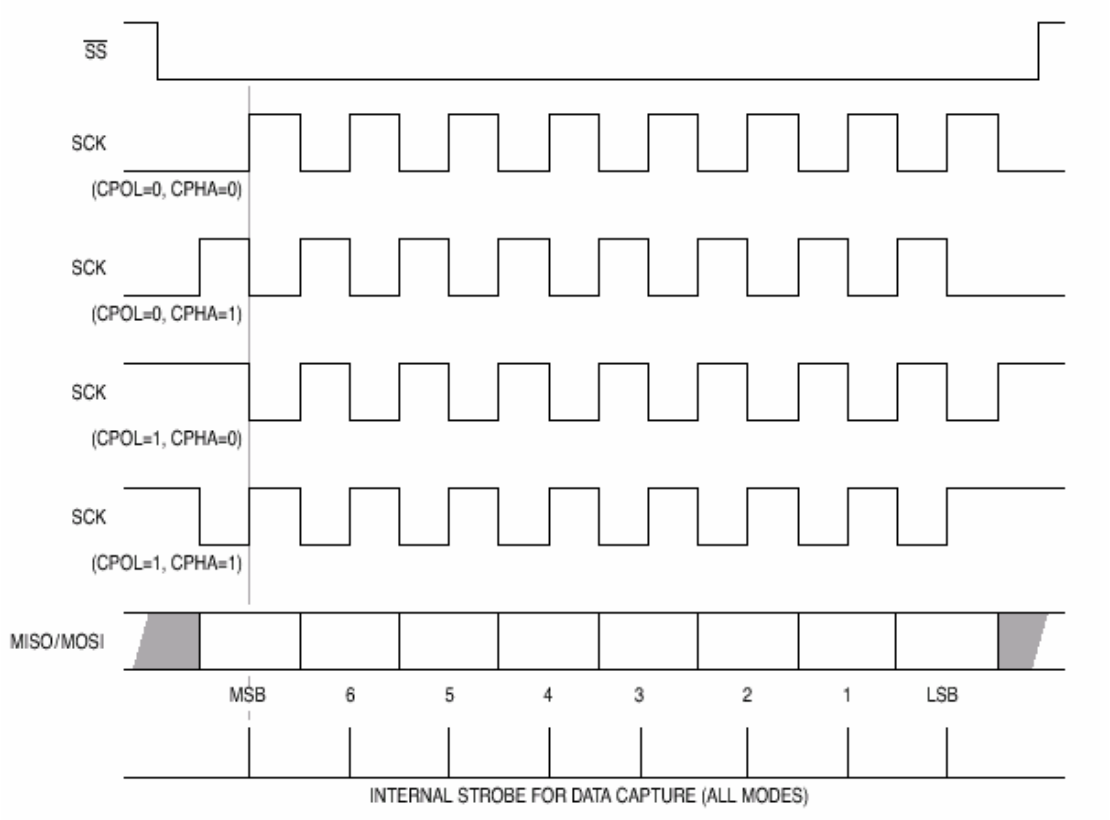


Figure 28: SPI Dataflow

13.13.3 Baud Rate Generation

An internal baud rate generator sets the UART baud rate when operating in UART mode and the SPI master clock frequency when operating in SPI mode.

The `UxBAUD.BAUD_M[7:0]` and `UxGCR.BAUD_E[4:0]` registers define the baud rate used for UART transfers and the rate of the serial clock for SPI transfers. The baud rate is given by the following equation:

$$Baudrate = \frac{(256 + BAUD\_M) * 2^{BAUD\_E}}{2^{28}} * F$$

where F is the system clock frequency set by the selected system clock source.

The register values required for standard baud rates are shown in Table 39 for a typical system clock set to 26 MHz. The table also gives the difference in actual baud rate to standard baud rate value as a percentage error.

The maximum baud rate for UART mode is F/16 when BAUD_E is 16 and BAUD_M is 0, and where F is the system clock frequency.

The maximum SPI clock frequency for SPI mode is F/2 when BAUD_E is 19 and BAUD_M is 0, and where F is the system clock frequency. Setting higher clock frequencies than this will give erroneous results.

Baud rate (bps)	UxBAUD.BAUD_M	UxGCR.BAUD_E	Error (%)
2400	131	6	0.04
4800	131	7	0.04
9600	131	8	0.04
14400	34	9	0.13
19200	131	9	0.04
28800	34	10	0.13
38400	131	10	0.04
57600	34	11	0.13
76800	131	11	0.04
115200	34	12	0.13
230400	164	13	0.02

Table 39: Commonly used baud rate settings for 26 MHz system clock

13.13.4 USART flushing

The current operation can be aborted by setting the UxUCR.FLUSH register bit. This event will immediately stop the current operation and clear all data buffers.

13.13.5 USART Interrupts

Each USART has two interrupts. These are the RX complete interrupt (URXx) and the TX complete interrupt (UTXx).

The USART interrupt enable bits are found in the IEN0 and IEN2 registers. The interrupt flags are located in the TCON and IRCON2 registers. Refer to section 12.7 on page 45 for details of these registers. The interrupt enables and flags are summarized below.

Interrupt enables:

- USART0 RX : IEN0.URX0IE
- USART1 RX : IEN0.URX1IE
- USART0 TX : IEN2.UTX0IE
- USART1 TX : IEN2.UTX1IE

Interrupt flags:

- USART0 RX : TCON.URX0
- USART1 RX : TCON.URX1
- USART0 TX : IRCON2.UTX0
- USART1 TX : IRCON2.UTX1

13.13.6 USART DMA Triggers

There are two DMA triggers associated with each USART. The DMA triggers are activated by RX complete and TX complete events i.e. the same events as the USART interrupt requests. A DMA channel can be configured using a USART Receive/Transmit buffer, UxDBUF, as source or destination address.

Refer to Table 37 on page 84 for an overview of the DMA triggers.

13.13.7 USART Registers

The registers for the USART are described in this section. For each USART there are five registers consisting of the following (x refers to USART number i.e. 0 or 1):

- UxCSR USART x Control and Status
- UxUCR USART x UART Control
- UxGCR USART x Generic Control
- UxDBUF USART x Receive/Transmit data buffer
- UxBAUD USART x Baud Rate Control

U0CSR (0x86) – USART 0 Control and Status

Bit	Name	Reset	R/W	Description
7	MODE	0	R/W	USART mode select 0 SPI mode 1 UART mode
6	RE	0	R/W	UART receiver enable 0 Receiver disabled 1 Receiver enabled
5	SLAVE	0	R/W	SPI master or slave mode select 0 SPI master 1 SPI slave
4	FE	0	R/W0	UART framing error status 0 No framing error detected 1 Byte received with incorrect stop bit level
3	ERR	0	R/W0	UART parity error status 0 No parity error detected 1 Byte received with parity error
2	RX_BYTE	0	R/W0	Receive byte status 0 No byte received 1 Received byte ready
1	TX_BYTE	0	R/W0	Transmit byte status 0 Byte not transmitted 1 Last byte written to Data Buffer register transmitted
0	ACTIVE	0	R	USART transmit/receive active status 0 USART idle 1 USART busy in transmit or receive mode

U0UCR (0xC4) – USART 0 UART Control

Bit	Name	Reset	R/W	Description
7	FLUSH	0	R0/W1	Flush unit. When set, this event will immediately stop the current operation and return the unit to idle state.
6	FLOW	0	R/W	UART hardware flow enable. Selects use of hardware flow control with RTS and CTS pins 0 Flow control disabled 1 Flow control enabled
5	D9	0	R/W	UART data bit 9 contents. This value is used when 9 bit transfer is enabled. When parity is disabled the value written to D9 is transmitted as the bit 9 when 9 bit data is enabled. If parity is enabled then this bit sets the parity level as follows. 0 Odd parity 1 Even parity
4	BIT9	0	R/W	UART 9-bit data enable. When this bit is 1, data is 9 bits and the contents of data bit 9 is given by D9 and PARITY. 0 8 bits transfer 1 9 bits transfer
3	PARITY	0	R/W	UART parity enable. 0 Parity disabled 1 Parity enabled
2	SPB	0	R/W	UART number of stop bits. Selects the number of stop bits to transmit 0 1 stop bit 1 2 stop bits
1	STOP	1	R/W	UART stop bit level 0 Low stop bit 1 High stop bit
0	START	0	R/W	UART start bit level. The polarity of the idle line is assumed to be the opposite of the selected start bit level. 0 Low start bit 1 High start bit

U0GCR (0xC5) – USART 0 Generic Control

Bit	Name	Reset	R/W	Description
7	CPOL	0	R/W	SPI clock polarity 0 Negative clock polarity 1 Positive clock polarity
6	CPHA	0	R/W	SPI clock phase 0 Data is output on *MOSI* when *SCK* goes from **CPOL** inverted to **CPOL**, and data input is sampled on *MISO* when *SCK* goes from **CPOL** to **CPOL** inverted. 1 Data is output on *MOSI* when *SCK* goes from **CPOL** to **CPOL** inverted, and data input is sampled on *MISO* when *SCK* goes from **CPOL** inverted to **CPOL.**
5	ORDER	0	R/W	Bit order for transfers 0 LSB first 1 MSB first
4:0	BAUD_E[4:0]	0x00	R/W	Baud rate exponent value. **BAUD_E** along with **BAUD_M** decides the UART baud rate and the SPI master SCK clock frequency

U0DBUF (0xC1) – USART 0 Receive/Transmit Data Buffer

Bit	Name	Reset	R/W	Description
7:0	DATA[7:0]	0x00	R/W	USART receive and transmit data. When writing this register the data written is written to the internal transmit data register. When reading this register, the data from the internal read data register is read.

U0BAUD (0xC2) – USART 0 Baud Rate Control

Bit	Name	Reset	R/W	Description
7:0	BAUD_M[7:0]	0x00	R/W	Baud rate mantissa value. **BAUD_E** along with **BAUD_M** decides the UART baud rate and the SPI master SCK clock frequency

U1CSR (0xF8) – USART 1 Control and Status

Bit	Name	Reset	R/W	Description
7	MODE	0	R/W	USART mode select 0 SPI mode 1 UART mode
6	RE	0	R/W	UART receiver enable 0 Receiver disabled 1 Receiver enabled
5	SLAVE	0	R/W	SPI master or slave mode select 0 SPI master 1 SPI slave
4	FE	0	R/W0	UART framing error status 0 No framing error detected 1 Byte received with incorrect stop bit level
3	ERR	0	R/W0	UART parity error status 0 No parity error detected 1 Byte received with parity error
2	RX_BYTE	0	R/W0	Receive byte status 0 No byte received 1 Received byte ready
1	TX_BYTE	0	R/W0	Transmit byte status 0 Byte not transmitted 1 Last byte written to Data Buffer register transmitted
0	ACTIVE	0	R	USART transmit/receive active status 0 USART idle 1 USART busy in transmit or receive mode

U1UCR (0xFB) – USART 1 UART Control

Bit	Name	Reset	R/W	Description
7	FLUSH	0	R0/W1	Flush unit. When set, this event will immediately stop the current operation and return the unit to idle state.
6	FLOW	0	R/W	UART hardware flow enable. Selects use of hardware flow control with RTS and CTS pins 0 Flow control disabled 1 Flow control enabled
5	D9	0	R/W	UART data bit 9 contents. This value is used 9 bit transfer is enabled. When parity is disabled the value written to D9 is transmitted as the bit 9 when 9 bit data is enabled. If parity is enabled then this bit sets the parity level as follows. 0 Odd parity 1 Even parity
4	BIT9	0	R/W	UART 9-bit data enable. When this bit is 1, data is 9 bits and the contents of data bit 9 is given by D9 and PARITY. 0 8 bits transfer 1 9 bits transfer
3	PARITY	0	R/W	UART parity enable. 0 Parity disabled 1 Parity enabled
2	SPB	0	R/W	UART number of stop bits. Selects the number of stop bits to transmit 0 1 stop bit 1 2 stop bits
1	STOP	1	R/W	UART stop bit level 0 Low stop bit 1 High stop bit
0	START	0	R/W	UART start bit level. The polarity of the idle line is assumed to be the opposite of the selected start bit level. 0 Low start bit 1 High start bit

U1GCR (0xFC) – USART 1 Generic Control

Bit	Name	Reset	R/W	Description
7	CPOL	0	R/W	SPI clock polarity 0 Negative clock polarity 1 Positive clock polarity
6	CPHA	0	R/W	SPI clock phase 0 Data is output on *MOSI* when *SCK* goes from **CPOL** inverted to **CPOL**, and data input is sampled on *MISO* when *SCK* goes from **CPOL** to **CPOL** inverted. 1 Data is output on *MOSI* when *SCK* goes from **CPOL** to **CPOL** inverted, and data input is sampled on *MISO* when *SCK* goes from **CPOL** inverted to **CPOL.**
5	ORDER	0	R/W	Bit order for transfers 0 LSB first 1 MSB first
4:0	BAUD_E[4:0]	0x00	R/W	Baud rate exponent value. **BAUD_E** along with **BAUD_M** decides the UART baud rate and the SPI master SCK clock frequency

U1DBUF (0xF9) – USART 1 Receive/Transmit Data Buffer

Bit	Name	Reset	R/W	Description
7:0	DATA[7:0]	0x00	R/W	USART receive and transmit data. When writing this register the data written is written to the internal transmit data register. When reading this register, the data from the internal read data register is read.

U1BAUD (0xFA) – USART 1 Baud Rate Control

Bit	Name	Reset	R/W	Description
7:0	BAUD_M[7:0]	0x00	R/W	Baud rate mantissa value. **BAUD_E** along with **BAUD_M** decides the UART baud rate and the SPI master SCK clock frequency

13.14 Flash Controller

The CC1110 contains 32 KB flash memory for storage of program code. The flash memory is programmable from the user software.

The Flash Controller handles writing and erasing the embedded 32 KB flash memory. The embedded flash memory consists of 32 pages of 1024 bytes each. The flash memory is byte-addressable from the CPU and 16-bit word-programmable.

The flash controller has the following features:

- 16-bit word programmable
- Page erase
- Lock bits for write-protection and code security
- Flash erase timing 20 ms
- Flash write timing 20 µs

13.14.1 Flash Write

Data is written to the flash memory by using a program command initiated by writing the Flash Control register, FCTL. Flash write operations can program any number of locations in the flash memory at a time – it is however important to make sure the pages to be written are erased first.

A write operation is performed using one out of two methods;

- Through DMA transfer
- Through CPU SFR access.

The DMA transfer method is the preferred way to write to the flash memory.

A write operation is initiated by writing a 1 to FCTL.WRITE. The address to start writing at, is given by FADDRH:FADDRL. During each single write operation FCTL.SWBSY is set high. During a write operation the data written to the FWDATA register is forwarded to the flash memory. The flash memory is 16-bit word-programmable, meaning data is written as 16-bit words. Therefore the actual writing to flash memory takes place each time two bytes have been written to FWDATA.

The CPU will not be able to access the flash, e.g. to read program code, while a flash write operation is in progress. Therefore the program code executing the flash write must be executed from RAM, meaning that the program code must reside in the area 0xF000 to 0xFF00 in CODE memory space.

When a flash write operation is executed from RAM, the CPU continues to execute code from the next instruction after the write to FWDATA, which initiated the flash write operation.

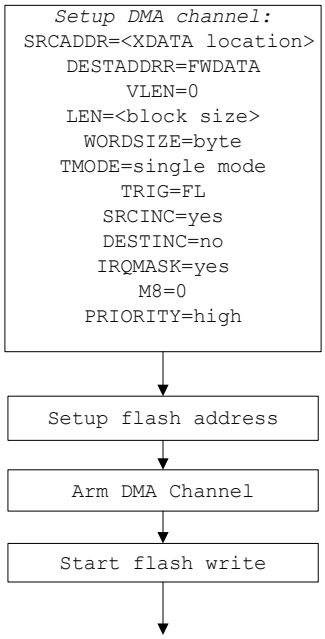
The FCTL.SWBSY bit must be 0 before accessing the flash after a flash write, otherwise an access violation occurs. This also means that FCTL.SWBSY must be 0 before program execution can continue at a location in flash memory i.e. from the area 0x0000 to 0x7FFF in CODE memory space.

13.14.1.1 DMA Flash Write

When using DMA write operations, the data to be written into flash is stored in DATA/XDATA memory. A DMA channel is configured to read the data to be written from memory and write this data to the Flash Write Data register, FWDATA with the DMA trigger event FL enabled. Thus the Flash Controller will trigger a DMA transfer when the Flash Write Data register, FWDATA, is ready to receive new data. The DMA channel should be configured to perform a block to fixed, single mode, byte size transfers.

When the DMA channel is armed, starting a flash write will trigger the first DMA transfer.

Figure 29 shows an example how a DMA channel is configured and how a DMA transfer is initiated to write a block of data from a location in XDATA to flash memory.




```
; Write a consecutive block of data from XDATA to consecutive locations in
; flash memory using DMA
; Assumes 26 MHz system clock is used
;
       MOV    DPTR,#DMACFG         ;load data pointer with address for DMA
                                   ;channel configuration and
                                   ;start writing DMA configuration
       MOV    A,#SRC_HI            ;source data high address
       MOVX   @DPTR,A              ;
       INC    DPTR                 ;
       MOV    A,#SRC_LO            ;source data low address
       MOVX   @DPTR,A              ;
       INC    DPTR                 ;
       MOV    A,#0DFh              ;destination high address = HIGH(X_FWDATA)
       MOVX   @DPTR,A              ;
       INC    DPTR                 ;
       MOV    A,#0AFh              ;destination low address = LOW(X_FWDATA)
       MOVX   @DPTR,A              ;
       INC    DPTR                 ;
       MOV    A,#BLK_LEN           ;block length
       MOVX   @DPTR,A              ;
       INC    DPTR                 ;
       MOV    A,#012h              ;8 bits, single mode, use FL trigger
       MOVX   @DPTR,A              ;
       INC    DPTR                 ;
       MOV    A,#042h              ;increment source by 1, don't increment
       MOVX   @DPTR,A              ;destination, mask interrupt, high DMA
                                   ;priority
       MOV    DMA0CFGL,#DMACFG_LO  ;setup start address for current DMA
       MOV    DMA0CFGH,#DMACFG_HI  ;configuration
       MOV    DMAARM,#01h          ;arm DMA channel 0
       MOV    FADDRH,#00h          ;setup flash address high
       MOV    FADDRL,#01h          ;setup flash address low
       MOV    FWT,#2Ah             ;setup flash timing
       MOV    FCTL,#02h            ;start flash page write => trigger DMA
       .
       .
```

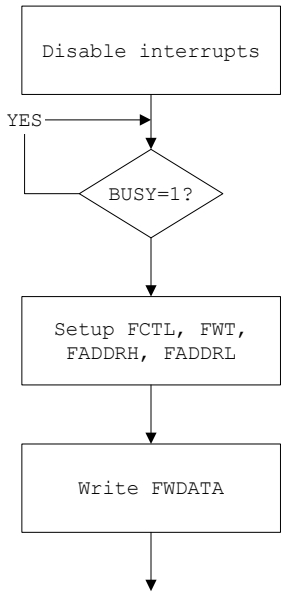
Figure 29: Flash write using DMA

13.14.1.2 CPU Flash Write

The CPU can also write directly to the flash. The CPU writes data to the Flash Write Data register, FWDATA. The flash memory is written each time two bytes have been written to FWDATA. The CPU can poll the FCTL.SWBSY status to determine when the flash is ready for two more bytes to be written to FWDATA..

Performing flash write from XDATA

The steps required to start a flash write operation from XDATA are shown in Figure 30 on page 149.




```
; Write 32-bit word from XDATA
; Assumes 26 MHz system clock is used
;
       CLR    EA                  ;mask interrupts
C1:    MOV    A,FCTL              ;wait until flash controller is ready
       JB     ACC.7,C1
       MOV    FADDRH,#00h         ;setup flash address high
       MOV    FADDRL,#01h         ;setup flash address low
       MOV    FWT,#2Ah            ;setup flash timing
       MOV    FCTL,#02h           ;set flash page write
       MOV    FWDATA,#12h         ;first byte
       MOV    FWDATA,#34h         ;second byte, initiates write
```

Figure 30 : Flash write performed from XDATA

13.14.2 Flash Page Erase

A page erase is initiated by setting FCTL.ERASE to 1. The page addressed by FADDRH[6:1] is erased when a page erase is initiated. Note that if a page erase is initiated simultaneously with a page write, i.e. FCTL.WRITE is set to 1, the page erase will be performed before the page write operation. The FCTL.BUSY bit can be polled to see when the page erase has completed.

Note: if flash erase operations are performed from within flash memory and the watchdog timer is enabled, a watchdog timer interval must be selected that is longer than 20 ms, the duration of the flash erase operation, so that the CPU will manage to clear the watchdog timer.

Performing flash erase from flash memory.

The steps required to perform a flash page erase from within flash memory are outlined in Figure 31.

Note that, while executing program code from within flash memory, when a flash erase or write operation is initiated, program execution will resume from the next instruction when the flash controller has completed the operation.

```
; Erase page in flash memory
; Assumes 26 MHz system clock is used
;
       CLR    EA                  ;mask interrupts
C1:    MOV    A,FCTL              ;wait until flash controller is ready
       JB     ACC.7,C1
       MOV    FADDRH,#00h         ;setup flash address high
       MOV    FADDRL,#01h         ;setup flash address low
       MOV    FWT,#2Ah            ;setup flash timing
       MOV    FCTL,#01h           ;erase page
       RET                        ;continues here when flash is ready
```

Figure 31: Flash page erase performed from flash memory

13.14.3 Flash Lock Protection

For software protection purposes a set of lock protection bits can be written once after each chip erase has been performed. The lock protect bits can only be written through the Debug Interface. There are three kinds of lock protect bits as described in this section. The flash lock bits reside at location 0x000 in the Flash Information page as described in section 12.11.

The LSIZE[2:0] lock protect bits are used to define a section of the flash memory which is write protected. The size of the write protected area can be set by the LSIZE[2:0] lock protect bits in sizes of eight steps from 0 to 32 KB.

The second type of lock protect bits is BBLOCK, which is used to lock the boot sector page (page 0 ranging from address 0 to 0x07FF). When BBLOCK is set to 0, the boot sector page is locked.

The third type of lock protect bit is DBGLOCK, which is used to disable hardware debug support through the Debug Interface. When DBGLOCK is set to 0, all debug commands are disabled.

The lock protect bits are written as a normal flash write to FWDATA, but the Debug Interface needs to select the Flash Information Page first instead of the Flash Main Page which is the default setting. The Information Page is selected through the Debug Configuration which is written through the Debug Interface only. Refer to section 12.9 on page 55 for details on how to select the Flash Information Page using the Debug Interface.

Table 40 defines the byte containing the flash lock protection bits. Note that this is not an SFR register, but instead the byte stored at location 0x000 in Flash Information Page.

Table 40: Flash Lock Protection Bits Definition

Bit	Name	Description
7:5	-	Reserved, write as 0
4	BBLOCK	Boot Block Lock 0 Page 0 is write protected 1 Page 0 is writeable, unless LSIZE is 000
3:1	LSIZE[2:0]	Lock Size. Sets the size of the upper Flash area which is write protected. Byte sizes are listed below 000 32K bytes (All pages) 001 16K bytes 010 8K bytes 011 4K bytes 100 2K bytes 101 1K bytes 110 512 bytes 111 0 bytes (no pages)
0	DBGLOCK	Debug lock bit 0 Disable debug commands 1 Enable debug commands

13.14.4 Flash Write Timing

The Flash Controller contains a timing generator which controls the timing sequence of flash write and erase operations. The timing generator uses the information set in the Flash Write Timing register, FWT.FWT[5:0], to set the internal timing. FWT.FWT[5:0] must be set to a value according to the currently selected system clock frequency.

The value set in the FWT.FWT[5:0] shall be set according to the system clock frequency by the following equation.

$$FWT = \frac{21000 * F}{16 * 10^9}$$

Where *F* is the system clock frequency. The initial value held in FWT.FWT[5:0] after a reset is 0x25 which corresponds to 26 MHz CPU clock frequency.

The FWT values for the possible system clock frequencies are given in Table 41.

System clock frequency (MHz)	FWT
16	0x15
26	0x2A

Table 41: Flash timing (FWT) values

13.14.5 Flash Controller Registers

The Flash Controller registers are described in this section.

FCTL (0xAE) – Flash Control

Bit	Name	Reset	R/W	Description
7	BUSY	0	R	Indicates that write or erase is in operation
6	SWBSY	0	R	Indicates that single write is busy; avoid writing to FWDATA register while this is true
5	–	0	R/W	Not used.
4	CONTRD	R/W	0	Continuous read enable mode 0 Avoid wasting power; turn on read enables to flash only when needed 1 Enable continuous read enables to flash when read is to be done. Reduces internal switching of read enables, but greatly increases power consumption.
3:2		0	R/W	Not used.
1	WRITE	0	R0/W	Page Write. Start writing page given by FADDRH:FADDRL. If ERASE is set to 1, a page erase is performed before the write.
0	ERASE	0	R0/W	Page Erase. Erase page that is given by FADDRH:FADDRL

FWDATA (0xAF) – Flash Write Data

Bit	Name	Reset	R/W	Description
7:0	FWDATA[7:0]	0x00	R/W	Flash write data. Data written to FWDATA is written to flash when FCTL.WRITE is set to 1.

FADDRH (0xAD) – Flash Address High Byte

Bit	Name	Reset	R/W	Description
7:6	-	00	R/W	Not used
5:0	FADDRH[6:0]	0x00	R/W	High byte of flash address Bits 5:1 will select page to access, while bit 0 is MSB of row access.

FADDRL (0xAC) – Flash Address Low Byte

Bit	Name	Reset	R/W	Description
7:0	FADDRL[7:0]	0x00	R/W	Low byte of flash address Bit 0 of FADDRH and bits 7:6 will select which row to write to, while bits 5:0 will select which location to write to.

FWT (0xAB) – Flash Write Timing

Bit	Name	Reset	R/W	Description
7:6	–	00	R/W	Not used
5:0	FWT[5:0]	0x25	R/W	Flash Write Timing. Controls flash timing generator.

14 Crystal Oscillator

A crystal in the frequency range 26 MHz-27 MHz must be connected between the XOSC_Q1 and XOSC_Q2 pins. The oscillator is designed for parallel mode operation of the crystal. In addition, loading capacitors (C201 and C211) for the crystal are required. The loading capacitor values depend on the total load capacitance, C_L, specified for the crystal. The total load capacitance seen between the crystal terminals should equal C_L for the crystal to oscillate at the specified frequency.

$$C_L = \frac{1}{\frac{1}{C_{211}} + \frac{1}{C_{201}}} + C_{parasitic}$$

The parasitic capacitance is constituted by pin input capacitance and PCB stray capacitance. Total parasitic capacitance is typically 2.5 pF.

The crystal oscillator circuit is shown in Figure 6. Typical component values for different values of C_L are given in Table 8.

The crystal oscillator is amplitude regulated. This means that a high current is used to start up the oscillations. When the amplitude builds up, the current is reduced to what is necessary to maintain approximately 0.4 Vpp signal swing. This ensures a fast start-up, and keeps the drive level to a minimum. The ESR of the crystal should be within the specification in order to ensure a reliable start-up (see section Table 8 on page 16).

15 Radio

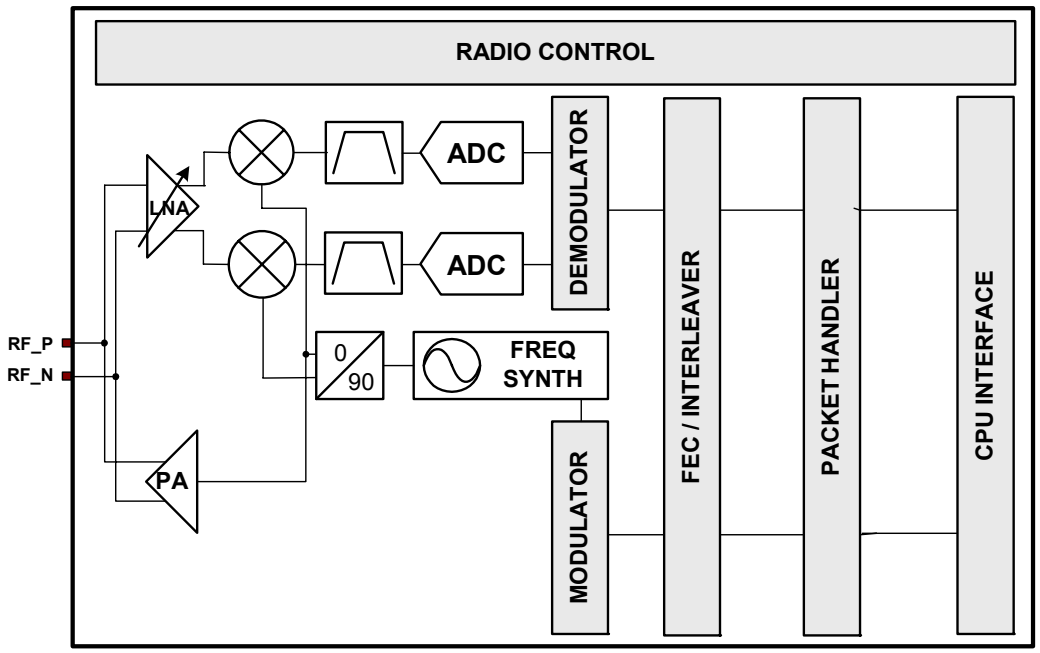


Figure 32: CC1110 Radio Module

A simplified block diagram of the radio module in the CC1110 is shown in Figure 32.

CC1110 features a low-IF receiver. The received RF signal is amplified by the low-noise amplifier (LNA) and down-converted in quadrature (I and Q) to the intermediate frequency (IF). At IF, the I/Q signals are digitized by the ADCs. Automatic gain control (AGC), fine channel filtering, demodulation bit/packet synchronization is performed digitally.

The transmitter part of CC1110 is based on direct synthesis of the RF frequency. The frequency synthesizer includes a completely on-chip LC VCO and a 90 degrees phase shifter for generating the I and Q LO signals to the down-conversion mixers in receive mode.

The 26 MHz crystal oscillator generates the reference frequency for the synthesizer, as well as clocks for the ADC and the digital part.

An SFR register interface is used for data buffer access from the CPU. Configuration and status registers are accessed through registers mapped to XDATA memory.

The digital baseband includes support for channel configuration, packet handling and data buffering. An on-chip voltage regulator delivers a regulated 1.8 V supply voltage.

15.1 Command strobes

The CPU uses a set of *command strobes* to control operation of the radio in CC1110.

Command strobes may be viewed as single byte instructions which each control some function of the radio. These command strobes must be used to enable the frequency synthesizer, enable receive

mode, enable transmit mode and other functions.

The command strobes are issued by writing to the RFST SFR register. The list of all strobe commands which are defined is given in Table 42.

Figure 33 shows a simplified state diagram that explains the main states, together with typical usage and current consumption in the radio part. The diagram shows the state transitions that are controlled by the command strobes. For detailed information on controlling the radio state machine, and a complete state diagram, see section 15.12 on page 167.

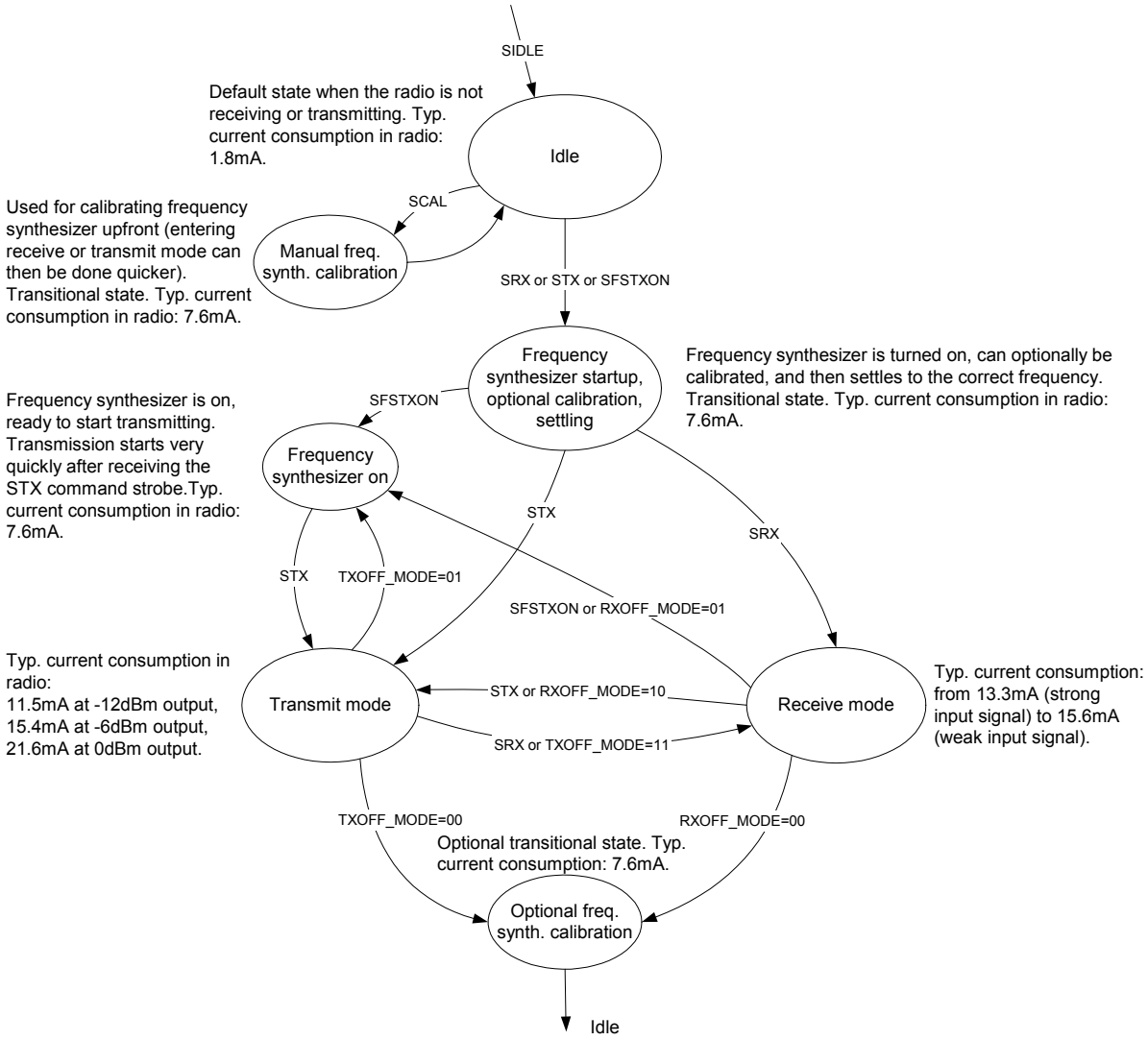


Figure 33: Simplified state diagram, with typical usage and current consumption in radio

RFST Value	Command Strobe Name	Description
0x00	SFSTXON	Enable and calibrate frequency synthesizer (if MCSM0.FS_AUTOCAL=1). While in RX / TX issuing this command strobe will force the radio to go to a wait state where only the synthesizer is running (for quick RX / TX turnaround).
0x01	SCAL	Calibrate frequency synthesizer and turn it off (enables quick start).
0x02	SRX	Enable RX. Perform calibration first if coming from IDLE and MCSM0.FS_AUTOCAL=1.
0x03	STX	In IDLE state: Enable TX. Perform calibration first if MCSM0.FS_AUTOCAL=1. If in RX state and CCA is enabled: Only go to TX if channel is clear.
0x04	SIDLE	Exit RX / TX, turn off frequency synthesizer.
0x05	SAFC	Perform AFC adjustment of the frequency synthesizer
all others	SNOP	No operation.

Table 42: Strobe commands

15.2 Radio Registers

The operation of the radio is configured through a set of RF registers. These RF registers are mapped to XDATA memory space as shown in Figure 7 on page 33.

In addition to configuration registers, the RF registers also provide status information from the radio.

The RF registers control/status bits are referred to where appropriate in the following sections while section 15.17 on page 172 gives a full description of all RF registers.

15.3 Interrupts

The radio is associated with two interrupt vectors on the CPU. These are the RFTXRX interrupt (interrupt 0) and the RFIF interrupt (interrupt 12) with the following functions

- RFTXRX: RX data ready or TX data complete
- RF: all other RFIF interrupt flags

The RF interrupt vector combines the interrupts in RFIF shown on page 157. Note that these RF interrupts are rising-edge triggered. Thus an interrupt is generated when e.g. the SFD status flag goes from 0 to 1.

The RF interrupt can also be used to trigger a timer capture in Timer 1.

The RF interrupt flags are described in the next section.

15.3.1 Interrupt registers

Two of the main interrupt control SFR registers are used to enable the RF and RFTXRX interrupts. These are the following:

- RFTXRX : IEN0. RFTXRXIE
- RF : IEN2.RFIE

Two main interrupt flag SFR registers hold the RF and RFERR interrupt flags. These are the following:

- RFTXRX : TCON. RFTXRX
- RF : S1CON.RFIF

Refer to section 12.7 for details about the interrupts.

The RF interrupt is the combined interrupt from six different sources in the radio. Two SFR registers are used for setting the six individual RFIF radio interrupt flags and interrupt enables. These are the RFIF and RFIM registers.

The interrupt flags in SFR register RFIF show the status for each interrupt source for the RF interrupt vector.

The interrupt enable bits in RFIM are used to disable individual interrupt sources for the RF interrupt vector. Note that masking an interrupt source in RFIM does not affect the update of the status in the RFIF register.

Due to the use of the individual interrupt masks in RFIM, and the main interrupt mask for the RF interrupt given by IEN2.RFIE there is two-layered masking of this interrupt. Special attention needs to be taken when processing this type of interrupt as described below.

To clear the RF interrupt, S1CON.RFIF and the interrupt flag in RFIF need to be cleared. The order and method of doing this is shown in Figure 34. Note that S1CON is cleared after RFIF, otherwise S1CON.RFIF could be set once again due to the same interrupt.

```
MOV    RFIF,#00h          ;clear all interrupt flags
MOV    S1CON,#00h         ;clear main interrupt flags
MOV    RFIM,RFIM          ;set interrupt mask
```

Figure 34: Clearing RF Interrupt

RFIF (0xE9) – RF Interrupt Flags

Bit	Name	Reset	R/W	Description
7:6	–	00	R0	Not used
5	IRQ_TIMEOUT	0	R/W0	RX timeout, no packet has been received in the programmed period. 0 No interrupt pending 1 Interrupt pending
4	IRQ_DONE	0	R/W0	Packet received/transmitted. Also used to detect underflow/overflow conditions. 0 No interrupt pending 1 Interrupt pending
3	IRQ_CS	0	R/W0	Carrier sense. 0 No interrupt pending 1 Interrupt pending
2	IRQ_PQT	0	R/W0	Preamble quality reached. 0 No interrupt pending 1 Interrupt pending
1	IRQ_CCA	0	R/W0	Clear Channel Assessment 0 No interrupt pending 1 Interrupt pending
0	IRQ_SFD	0	R/W0	Start of Frame Delimiter, sync word detected 0 No interrupt pending 1 Interrupt pending

RFIM (0x91) – RF Interrupt Mask

Bit	Name	Reset	R/W	Description
7:6	–	00	R0	Not used
5	IM_TIMEOUT	0	R/W	RX timeout, no packet has been received in the programmed period. 0 Interrupt disabled 1 Interrupt enabled
4	IM_DONE	0	R/W	Packet received/transmitted. Also used to detect underflow/overflow conditions. 0 Interrupt disabled 1 Interrupt enabled
3	IM_CS	0	R/W	Carrier sense. 0 Interrupt disabled 1 Interrupt enabled
2	IM_PQT	0	R/W	Preamble quality reached. 0 Interrupt disabled 1 Interrupt enabled
1	IM_CCA	0	R/W	Clear Channel Assessment 0 Interrupt disabled 1 Interrupt enabled
0	IM_SFD	0	R/W	Start of Frame Delimiter, sync word detected 0 Interrupt disabled 1 Interrupt enabled

15.4 TX/RX Data Transfer

Transmit data is written to the radio when writing to the RF Data register, RFD. Received data is returned when the RFD register is read.

It is required that the user software uses FIFO structures in memory to implement RX and TX FIFOs. In most cases it is recommended that the transfer of data between FIFOs in memory and the RF Data register, RFD, involves the use of DMA channels with RFD as source/destination and DMA trigger RADIO. For description on the usage of DMA, refer to section 13.2 on page 78.

A simple example of writing TX data to the radio is shown in Figure 35. This example does not use DMA, but illustrates some of the basic principles.

```
; Start radio TX
; simple example, assumes required frequency, modulation format and
; data rate has been set in RF registers
;
       MOV    DPTR,PA_TABLE0      ;setting PA output power
       MOV    A,#0FFh             ;
       MOVX   @DPTR,A             ;
       MOV    CLKCON,#00H         ;select 26 MHz XOSC
       MOV    RFST,#03H           ;start TX with STX command strobe
C1:    JNB    IE0,C1              ;wait for interrupt flag telling radio is
       CLR    IE0                 ;ready to accept data, then write
       MOV    RFD,#02H            ;first data to radio, packet length=2
C2:    JNB    IE0,C2              ;wait for radio
       CLR    IE0                 ;
       MOV    RFD,#12H            ;send first byte in payload
C3:    JNB    IE0,C3              ;wait for radio
       CLR    IE0                 ;
       MOV    RFD,#34H            ;send second byte in payload
                                  ;done
```

Figure 35: Simple RF transmit example

15.5 Data Rate Programming

The data rate used when transmitting, or the data rate expected in receive is programmed by the MDMCFG3.DRATE_M and the MDMCFG4.DRATE_E configuration registers. The data rate is given by the formula below. As the formula shows, the programmed data rate depends on the crystal frequency.

$$R_{DATA} = \frac{(256 + DRATE\_M) \cdot 2^{DRATE\_E}}{2^{28}} \cdot f_{XOSC}$$

The following approach can be used to find suitable values for a given data rate:

$$DRATE\_E = \left\lfloor \log_2 \left(\frac{R_{DATA} \cdot 2^{20}}{f_{XOSC}} \right) \right\rfloor$$

$$DRATE\_M = \frac{R_{DATA} \cdot 2^{28}}{f_{XOSC} \cdot 2^{DRATE\_E}} - 256$$

If DRATE_M is rounded to the nearest integer and becomes 256, increment DRATE_E and use DRATE_M=0.

The data rate can be set from 1.2 kbps to 500 kbps with the minimum step size as shown in Table 43.

Data rate start	Typical data rate	Data rate stop	Data rate step size
0.8 kbps	1.2/2.4 kbps	3.17 kbps	0.0062 kbps
3.17 kbps	4.8 kbps	6.35 kbps	0.0124 kbps
6.35 kbps	9.6 kbps	12.7 kbps	0.0248 kbps
12.7 kbps	19.6 kbps	25.4 kbps	0.0496 kbps
25.4 kbps	38.4 kbps	50.8 kbps	0.0992 kbps
50.8 kbps	76.8 kbps	101.6 kbps	0.1984 kbps
101.6 kbps	153.6 kbps	203.1 kbps	0.3967 kbps
203.1 kbps	250 kbps	406.3 kbps	0.7935 kbps
406.3 kbps	500 kbps	500 kbps	1.5869 kbps

Table 43: Data rate step size

15.6 Receiver Channel Filter Bandwidth

In order to meet different channel width requirements, the receiver channel filter is programmable. The MDMCFG4.CHANBW_E and MDMCFG4.CHANBW_M configuration registers control the receiver channel filter bandwidth, which scales with the crystal oscillator frequency. The following formula gives the relation between the register settings and the channel filter bandwidth:

$$BW_{channel} = \frac{f_{XOSC}}{8 \cdot (4 + CHANBW\_M) \cdot 2^{CHANBW\_E}}$$

The CC1110 supports channel filter bandwidths shown in Table 44.

MDMCFG4. CHANBW_M	MDMCFG4.CHANBW_E			
	00	01	10	11
00	812	406	203	102
01	650	325	162	81
10	541	270	135	68
11	464	232	116	58

Table 44: Channel filter bandwidths [kHz] (26 MHz crystal)

Above 300kHz bandwidth, however, the sensitivity and blocking performance may be somewhat degraded.

For best performance, the channel filter bandwidth should be selected so that the signal bandwidth occupies at most 80% of the channel filter bandwidth. The channel centre tolerance due to crystal accuracy should also be subtracted from the signal bandwidth. The following example illustrates this:

With the channel filter bandwidth set to 500kHz, the signal should stay within 80% of 500kHz, which is 400kHz. Assuming 915MHz frequency and ±20ppm frequency uncertainty for both the transmitting device and the receiving device, the total frequency uncertainty is ±40ppm of 915MHz, which is ±37kHz. If the whole transmitted signal bandwidth is to be received within 400kHz, the transmitted signal bandwidth should be maximum 400kHz–2·37kHz, which is 326kHz

.

15.7 Demodulator, Symbol Synchronizer and Data Decision

CC1110 contains an advanced and highly configurable demodulator. Channel filtering and frequency offset compensation is performed digitally. To generate the RSSI level (see section 15.10.2 for more information) the signal level in the channel is estimated. Data filtering is also included for enhanced performance.

15.7.1 Frequency Offset Compensation

When using FSK or MSK modulation, the demodulator will compensate for the offset between the transmitter and receiver frequency, within certain limits, by estimating the centre of the received data. This value is available in the FREQEST status register.

By issuing the SAFC command strobe, the measured offset,

FREQEST.FREQOFF_EST, can automatically be used to adjust the frequency offset programming in the frequency synthesizer. This will add the current RX frequency offset estimate to the value in FSCTRL0.FREQOFF, which adjust the synthesizer frequency. Thus, the frequency offset will be compensated in both RX and TX when the SAFC command strobe is used.

To avoid compensating for frequency offsets measured without a valid signal in the RF channel, FREQEST.FREQOFF_EST is copied to an internal register when issuing the SAFC strobe in RX, and when a synch word is detected. If SAFC was issued in RX, this internal value is added to FSCTRL0.FREQOFF after exiting RX. Issuing SAFC when not in RX will immediately add the internal register value to FSCTRL0.FREQOFF. Thus, the SAFC strobe should be issued when currently receiving a packet, or outside the RX state.

Note that frequency offset compensation is not supported for ASK or OOK modulation.

15.7.2 Bit Synchronization

The bit synchronization algorithm extracts the clock from the incoming symbols. The algorithm requires that the expected data rate is programmed as described in Section 15.5 on page 159. Re-synchronization is performed continuously to adjust for error in the incoming symbol rate.

15.7.3 Byte synchronization

Byte synchronization is achieved by a continuous sync word search. The sync word is a 16 or 32 bit configurable field that is automatically inserted at the start of the packet by the modulator in transmit mode. The demodulator uses this field to find the byte boundaries in the stream of bits. The sync word will also function as a system identifier, since only packets with the correct predefined sync word will be received. The sync word detector correlates against the user-configured 16-bit sync word. The correlation threshold can be set to 15/16 bits match or 16/16 bits match. The sync word can be further qualified using the preamble quality indicator mechanism described below and/or a carrier sense condition. The sync word is programmed with SYNC1 and SYNC0.

In order to make false detections of sync words less likely, a mechanism called preamble quality indication (PQI) can be used to qualify the sync word. A threshold value for the preamble quality must be exceeded in order for a detected sync word to be accepted. See section 15.10.1 on page 165 for more details.

15.8 Packet Handling Hardware Support

The *CC1110* has built-in hardware support for packet oriented radio protocols.

In transmit mode, the packet handler will add the following elements to the packet stored to be transmitted:

- A programmable number of preamble bytes. Four preamble bytes is recommended.
- A two byte Synchronization Word. Can be duplicated to give a 4-byte sync word. (Recommended).
- Optionally whiten the data with a PN9 sequence.
- Optionally Interleave and Forward Error Code the data.
- Optionally compute and add a CRC checksum over the data field.

In receive mode, the packet handling support will de-construct the data packet:

- Preamble detection.
- Sync word detection.
- Optional one byte address check.
- Optionally compute and check CRC.
- Optionally append two status bytes (see Table 45 and Table 46) with RSSI value, Link Quality Indication and CRC status.

Bit	Field name	Description
7:0	RSSI	RSSI value

Table 45: Received packet status byte 1 (first byte appended after the data)

Bit	Field name	Description
7	CRC_OK	1: CRC for received data OK (or CRC disabled) 0: CRC error in received data
6:0	LQI	The Link Quality Indicator estimates how easily a received signal can be demodulated

Table 46: Received packet status byte 2 (second byte appended after the data)

Note that register fields that control the packet handling features should only be altered when CC1110 is in the IDLE state.

15.8.1 Data whitening

From a radio perspective, the ideal over the air data are random and DC free. This results in the smoothest power distribution over the occupied bandwidth. This also gives the control loops in the receiver uniform operation conditions (no data dependencies).

Real world data often contain long sequences of zeros and ones. Performance can then be improved by whitening the data before transmitting, and de-whitening in the receiver. With CC1110, this can be done automatically by setting PKTCTRL0.WHITE_DATA=1. All data, except the preamble and the sync word, are then XOR-ed with a 9-bit pseudo-random (PN9) sequence before being transmitted. At the receiver end, the data are XOR-ed with the same pseudo-random sequence. This way, the whitening is reversed, and the original data appear in the receiver.

Setting PKTCTRL0.WHITE_DATA=1 is recommended for all uses, except when over-the-air compatibility with other systems is needed.

15.8.2 Packet format

The format of the data packet can be configured and consists of the following items:

- Preamble
- Synchronization word
- Length byte or constant programmable packet length
- Optional Address byte
- Payload
- Optional 2 byte CRC

The preamble pattern is an alternating sequence of ones and zeros (01010101…). The minimum length of the preamble is programmable. When enabling TX, the modulator will start transmitting the preamble. When the programmed number of preamble bytes has been transmitted, the modulator will send the sync word and then data from the RFD register.

The number of preamble bytes is programmed with the MDMCFG1.NUM_PREAMBLE value.

The synchronization word is a two-byte value set in the SYNC1 and SYNC0 registers. The sync word provides byte synchronization of the incoming packet. A one-byte sync word can be emulated by setting the SYNC1 value to the preamble pattern. It is also possible to emulate a 32 bit sync word by using

MDMCFG2.SYNC_MODE=3 or 7. The sync word will then be repeated twice.

CC1110 supports both fixed packet length protocols and variable length protocols. The maximum packet length is 255 bytes. For longer packets, infinite packet length mode must be used.

Fixed packet length mode is selected by setting PKTCTRL0.LENGTH_CONFIG=0. The desired packet length is set by the PKTLEN register. The packet length is defined as the payload data, excluding the length byte and the optional automatic CRC. In variable length mode, the PKTLEN register is used to set the maximum packet length allowed in RX. Any packet received with a length byte with a value greater than PKTLEN will be discarded.

Note that the minimum packet length supported (excluding the optional length byte and CRC) is one byte of payload data.

With PKTCTRL0.LENGTH_CONFIG=2, the packet length is set to infinite and transmission and reception will continue until turned off manually. The infinite mode can be turned off while a packet is being transmitted or received. As described in the next section, this can be used to support packet formats with different length configuration than natively supported by CC1110.

15.8.2.1 Arbitrary length field configuration

The fixed length field can be reprogrammed during receive and transmit. This opens the possibility to have a different length field configuration than supported for variable length packets. At the start of reception, the packet length is set to a large value. The CPU reads out enough bytes to interpret the length field in the packet. Then the PKTLEN value is set according to this value. The end of packet will occur when the byte counter in the packet handler is equal to the PKTLEN register. Thus, the CPU must be able to program the correct length, before the internal counter reaches the packet length.

By utilizing the infinite packet length option, arbitrary packet length is available. At the start of the packet, the infinite mode must be active. When less than 256 bytes remains of the packet, the CPU sets the PKTLEN register to mod(length, 256), disables infinite packet length and activates fixed length packets. When the internal byte counter reaches the PKTLEN value, the transmission or reception ends. Automatic CRC appending/checking can be used (by setting PKTCTRL0.CRC_EN to 1).

When for example a 454-byte packet is to be transmitted, the CPU does the following:

- Set PKTCTRL0.LENGTH_CONFIG=2 (10).
- Pre-program the PKTLEN register to mod(454,256)=198.
- Transmit at least 198 bytes, for example by filling the 64-byte TX FIFO four times (256 bytes transmitted).
- Set PKTCTRL0.LENGTH_CONFIG=0 (00).

The transmission ends when the packet counter reaches 198. A total of 256+198=454 bytes are transmitted.

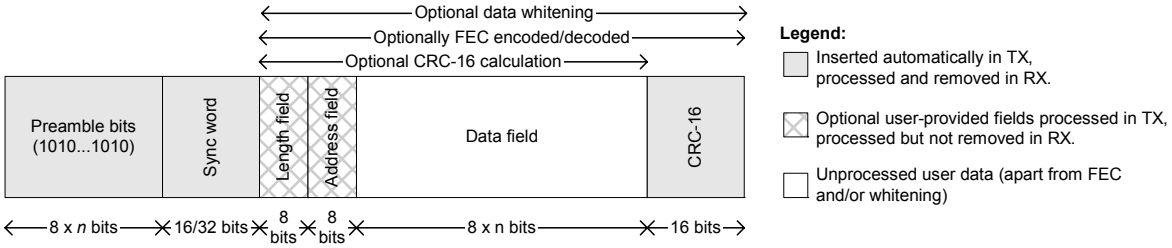


Figure 36: Packet Format

15.8.3 Packet filtering in Receive Mode

CC1110 supports two different packet-filtering criteria: address filtering and maximum length filtering.

15.8.3.1 Address Filtering

Setting PKTCTRL1.ADR_CHK to any other value than zero enables the packet address filter. The packet handler engine will compare the destination address byte in the packet with the programmed node address in the ADDR register and the 0x00 broadcast address when PKTCTRL1.ADR_CHK=10 or both 0x00 and 0xFF broadcast addresses when PKTCTRL1.ADR_CHK=11. If the received address matches a valid address, the packet is received and written into the RFD register. If the address match fails, the packet is discarded and receive mode restarted (regardless of the MCSM1.RXOFF_MODE setting).

15.8.3.2 Maximum Length Filtering

In the variable packet length mode the PKTLEN.PACKET_LENGTH register value is used to set the maximum allowed packet length. If the received length byte has a larger value than this, the packet is discarded and receive mode restarted (regardless of the MCSM1.RXOFF_MODE setting). If the received length byte has a larger value than this, the packet is discarded and receive mode restarted (regardless of the MCSM1.RXOFF_MODE setting).

15.8.4 Packet Handling in Transmit Mode

The payload that is to be transmitted must be written into RFD. The first byte written must be the length byte when variable packet length is enabled. The length byte has a value equal to the payload of the packet (including the optional address byte). If fixed packet length is enabled, then the first byte written to RFD is interpreted as the destination address, if this feature is enabled in the device that receives the packet.

The modulator will first send the programmed number of preamble bytes. If data is written to RFD, the modulator will send the two-byte (optionally 4-byte) sync word and then the payload written to RFD. If CRC is enabled, the checksum is calculated over all the data pulled from RFD and the result is sent as two extra bytes at the end of the payload data.

If whitening is enabled, the length byte, payload data and the two CRC bytes will be whitened. This is done before the optional FEC/Interleaver stage. Whitening is enabled by setting PKTCTRL0.WHITE_DATA=1.

If FEC/Interleaving is enabled, the length byte, payload data and the two CRC bytes will be scrambled by the interleaver, and FEC encoded before being modulated.

15.8.5 Packet Handling in Receive Mode

In receive mode, the demodulator and packet handler will search for a valid preamble and the sync word. When found, the demodulator has obtained both bit and byte synchronism and will receive the first payload byte.

If FEC/Interleaving is enabled, the FEC decoder will start to decode the first payload byte. The interleaver will de-scramble the bits before any other processing is done to the data.

If whitening is enabled, the data will be de-whitened at this stage.

When variable packet length is enabled, the first byte is the length byte. The packet handler stores this value as the packet length and receives the number of bytes indicated by the length byte. If fixed packet length is used, the packet handler will accept the programmed number of bytes.

Next, the packet handler optionally checks the address and only continues the reception if the address matches. If automatic CRC check is enabled, the packet handler computes CRC and matches it with the appended CRC checksum.

At the end of the payload, the packet handler will optionally write two extra packet status bytes that contain CRC status, link quality indication and RSSI value.

15.9 Modulation Formats

CC1110 supports amplitude, frequency and phase shift modulation formats. The desired modulation format is set in the MDMCFG2.MOD_FORMAT register.

Optionally, the data stream can be Manchester coded by the modulator and decoded by the

demodulator. This option is enabled by setting MDMCFG2.MANCHESTER_EN=1. Manchester encoding is not supported at the same time as using the FEC/Interleaver option. Manchester coding can be used with the 2-ary modulation formats (2-FSK, GFSK, OOK and MSK).

15.9.1 Frequency Shift Keying

2-FSK can optionally be shaped by a Gaussian filter with BT=1, producing a GFSK modulated signal.

The frequency deviation is programmed with the DEVIATION_M and DEVIATION_E values in the DEVIATN register. The value has an exponent/mantissa form, and the resultant deviation is given by:

$$f_{dev} = \frac{f_{xosc}}{2^{17}} \cdot (8 + DEVIATION\_M) \cdot 2^{DEVIATION\_E}$$

The symbol encoding is shown in Table 47.

Format	Symbol	Coding
2FSK/GFSK	'0'	– Deviation
	'1'	+ Deviation

Table 47: Symbol encoding for FSK modulation

15.9.2 Minimum Shift Keying

When using MSK[2], the complete transmission (preamble, sync word and payload) will be MSK modulated.

Phase shifts are performed with a constant transition time. This means that the rate of change for the 180-degree transition is twice that of the 90-degree transition.

The fraction of a symbol period used to change the phase can be modified with the DEVIATN.DEVIATION_M setting. This is equivalent to changing the shaping of the symbol. Setting DEVIATN.DEVIATION_M=7 will generate a standard shaped MSK signal.

The MSK modulation format implemented in CC1110 inverts the sync word and data compared to e.g. signal generators.

[2] Identical to offset QPSK with half-sine shaping (data coding may differ)

15.9.3 Amplitude Modulation

CC1110 supports two different forms of amplitude modulation: On-Off Keying (OOK) and Amplitude Shift Keying (ASK). OOK modulation simply turns on or off the PA to modulate 1 and 0 respectively. When using ASK the modulation depth (the difference between 1 and 0) can be programmed, and the power ramping will be shaped. This will produce a more bandwidth constrained output spectrum. Output power is limited to 0 dBm when ASK is used.

15.10 Received Signal Qualifiers and Link Quality Information

CC1110 has several qualifier values that are used to increase the requirements that must be fulfilled before a search for a valid sync word is started.

15.10.1 Preamble Quality Threshold (PQT)

The Preamble Quality Threshold (PQT) sync-word qualifier adds the requirement that the received sync word must be preceded with a preamble with a quality above the programmed threshold.

Another use of the preamble quality threshold is as a qualifier for the optional RX termination timer. See section 15.12.3 on page 169 for details.

The preamble quality estimator increases an internal counter by one each time a bit is received that is different from the previous bit, and decreases the counter by 4 each time a bit is received that is the same as the last bit. The counter saturates at 0 and 31. The threshold is configured with the register field PKTCTRL1.PQT. A threshold of 4·PQT for this counter is used to gate sync word detection. By setting the value to zero, the preamble quality qualifier of the synch word is disabled.

A "Preamble Quality reached" flag can also be observed in the status register bit PKTSTATUS.PQT_REACHED. This flag asserts when the received signal exceeds the PQT.

15.10.2 RSSI

The RSSI value is an estimate of the signal level in the current channel. This value is based on the current gain setting in the RX chain and the measured signal level in the channel.

In RX mode, the RSSI value can be read continuously from the RSSI status register, until the demodulator detects a sync word (when sync word detection is enabled). At that point, the RSSI readout value is frozen until the next time the chip enters the RX state. The RSSI value is in dB with 0.5dB resolution.

If PKTCTRL1.APPEND_STATUS is enabled, a snapshot of the RSSI during the first 8 bytes of the packet is automatically added to the end of each received packet.

The RSSI value read from the RSSI status register is a 2's complement number. The following procedure can be used to convert the RSSI reading to an absolute power level (RSSI_dBm).

15.10.3 Carrier Sense (CS)

The Carrier Sense flag is used as a sync word qualifier and for CCA. The CS flag can be set based on two conditions, which can be individually adjusted:

- CS is asserted when the RSSI is above a programmable absolute threshold, and de-asserted when RSSI is below the same threshold (with hysteresis).
- CS is asserted when the RSSI has increased with a programmable number of dB from one RSSI sample to the next, and de-asserted when RSSI has decreased with the same number of dB. This setting is not dependent on the absolute signal level and is thus useful to detect signals in environments with time varying noise floor.

Carrier Sense (CS) can be used as a sync word qualifier that requires the signal level to be higher than the threshold for a sync word search to be performed. The signal can also be observed in the status register bit PKTSTATUS.CS.

Other uses of Carrier Sense include the TX-If-CCA function (see section 15.10.4 on page 166) and the optional fast RX termination (see section See section 15.12.3 on page 169 for details.).

15.10.4 Clear Channel Assessment (CCA)

The Clear Channel Assessment is used to indicate if the current channel is free or busy. The current CCA state is viewable in the PKTSTATUS register

MCSM1.CCA_MODE selects the mode to use when determining CCA.

When the STX or SFSTXON command strobe is given while **CC1110** is in the RX state, the TX state is only entered if the clear channel requirements are fulfilled. The chip will otherwise remain in RX. This feature is called *TX if CCA*.

Four CCA requirements can be programmed:

- Always (CCA disabled, always goes to TX)
- If RSSI is below threshold
- Unless currently receiving a packet
- Both the above (RSSI below threshold and not currently receiving a packet)

15.10.5 Link Quality Indicator (LQI)

The Link Quality Indicator is a metric of the current quality of the received signal. If PKTCTRL1.APPEND_STATUS is enabled, the value is automatically appended to the end of each received packet. The value can also be read from the LQI status register. The LQI is calculated over the 64 symbols following the sync word (first 8 packet bytes for 2-ary modulation). LQI is best used as a relative measurement of the link quality, since the value is dependent on the modulation format.

15.11 Forward Error Correction with Interleaving

CC1110 has built-in support for Forward Error Correction (FEC). To enable this option, set MDMCFG1.FEC_EN to 1. FEC is employed on the data field and CRC word in order to reduce the gross bit error rate when operating near the sensitivity limit. Redundancy is added to the transmitted data in such a way that the receiver can restore the original data in the presence of some bit errors.

The use of FEC allows correct reception at a lower SNR, thus extending communication range. Alternatively, for a given SNR, using FEC decreases the bit error rate (BER). As the packet error rate (PER) is related to BER by:

$$PER = 1 - (1 - BER)^{packet\_length},$$

a lower BER can be used to allow significantly longer packets, or a higher percentage of packets of a given length, to be transmitted successfully. Finally, in realistic ISM radio

environments, transient and time-varying phenomena will produce occasional errors even in otherwise good reception conditions. FEC will mask such errors and, combined with interleaving of the coded data, even correct relatively long periods of faulty reception (burst errors).

The FEC scheme adopted for ***CC1110*** is convolutional coding, in which *n* bits are generated based on *k* input bits and the *m* most recent input bits, forming a code stream able to withstand a certain number of bit errors between each coding state (the *m*-bit window).

The convolutional coder is a rate 1/2 code with a constraint length of m=4. The coder codes one input bit and produces two output bits; hence, the effective data rate is halved.

15.11.1 Interleaving

Data received through real radio channels will often experience burst errors due to interference and time-varying signal strengths. In order to increase the robustness to errors spanning multiple bits, interleaving is used when FEC is enabled. After de-interleaving, a continuous span of errors in the received stream will become single errors spread apart.

CC1110 employs matrix interleaving, which is illustrated in Figure 37. The on-chip interleaving and de-interleaving buffers are 4 x 4 matrices. In the transmitter, the data bits are written into the rows of the matrix, whereas the bit sequence to be transmitted is read from the columns of the matrix and fed to the rate ½ convolutional coder. Conversely, in the receiver, the received symbols are written into the columns of the matrix, whereas the data passed onto the convolutional decoder is read from the rows of the matrix.

When FEC and interleaving is used, the amount of data transmitted over the air must be a multiple of the size of the interleaver buffer (two bytes). In addition, at least one extra byte is required for trellis termination. The packet control hardware therefore automatically inserts one or two extra bytes at the end of the packet, so that the total length of the data to be interleaved is an even number. Note that these extra bytes are invisible to the user, as they are removed before the received packet enters the `RFD` data register.

Due to the implementation of the FEC and interleaver, the data to be interleaved must be at least two bytes. One byte long fixed length packets without CRC is therefore not supported when FEC/interleaving is enabled.

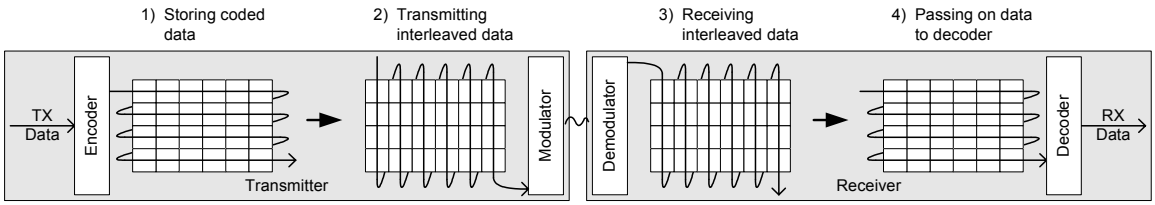


Figure 37: General principle of matrix interleaving

15.12 Radio Control

CC1110 has a built-in radio state machine that is used to switch between different operational states (modes) of the radio. The change of state is done by using command strobes given in Table 42.

A simplified state diagram, together with typical usage and current consumption, is shown in Figure 33 on page 155. The complete radio control state diagram is shown in Figure 38. The state names and numbering shown in the figure, refer to the state value returned by the `MARCSTATE` status register. This functionality is primarily for test and debug purposes.

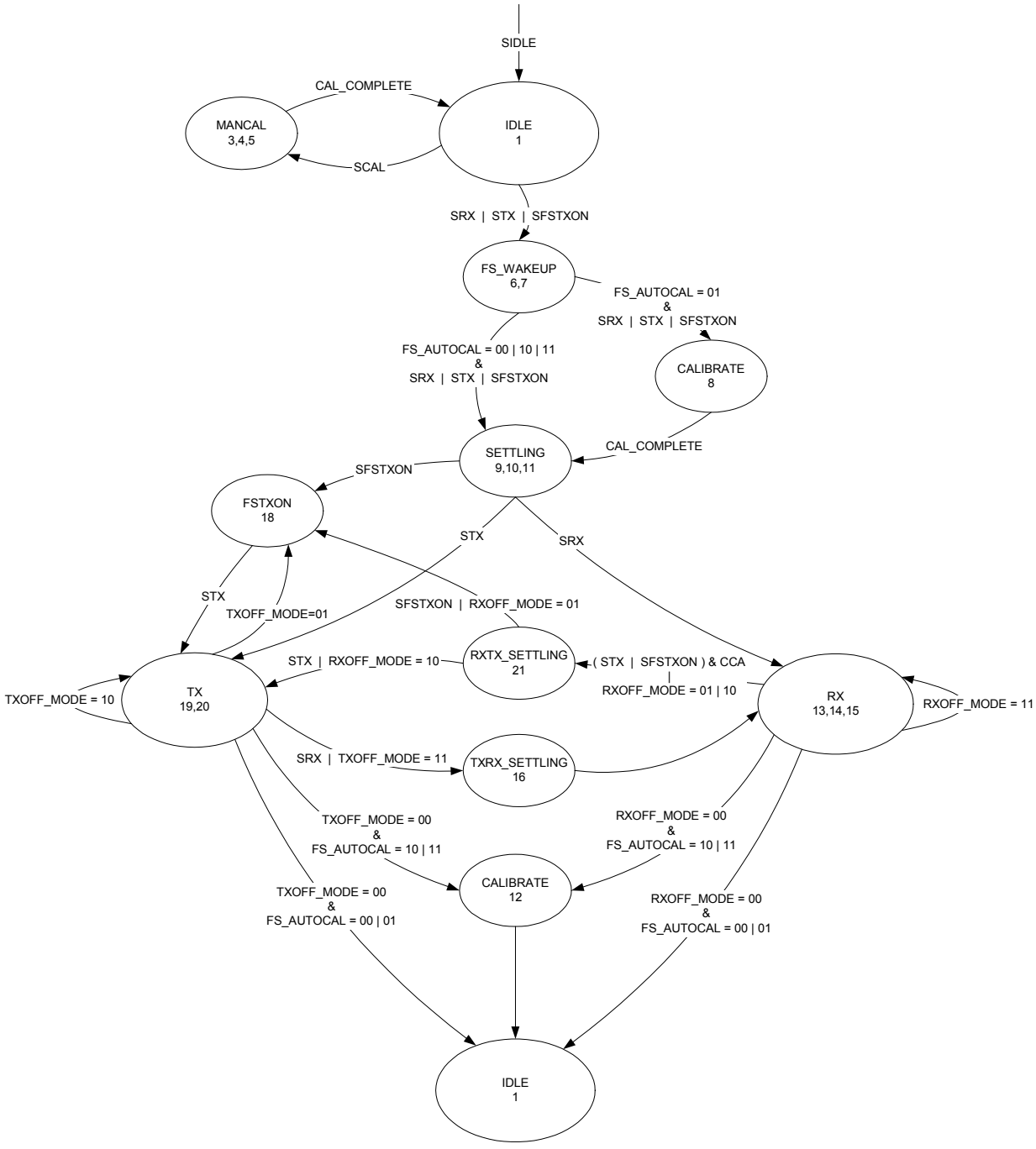


Figure 38: Complete Radio Control State Diagram

15.12.1 Active Modes

The ***CC1110*** radio has two active modes: receive and transmit. These modes are activated directly by the CPU by writing the SRX and STX command strobes to the RFST register.

The frequency synthesizer must be calibrated regularly. ***CC1110*** has one manual calibration option (using the SCAL strobe), and three automatic calibration options. The automatic calibration options are controlled by the MCSM0.FS_AUTOCAL setting as follows:

- Calibrate when going from IDLE to either RX or TX (or FSTXON)
- Calibrate when going from either RX or TX to IDLE
- Calibrate every fourth time when going from either RX or TX to IDLE

The calibration takes a constant number of XOSC cycles (see Table 48 for timing details).

When RX is activated, the chip will remain in receive mode until the RX termination timer expires (see section 15.12.3) or a packet has been successfully received. Note: the probability that a false sync word is detected can be reduced by using PQT, CS, maximum sync word length and sync word qualifier mode as describe in section 15.10. After a packet is successfully received the radio controller will then go to the state indicated by the MCSM1.RXOFF_MODE setting. The possible states are:

- IDLE
- FSTXON: Frequency synthesizer on and ready at the TX frequency. Activate TX with STX.
- TX: Start sending preambles
- RX: Start search for a new packet

Similarly, when TX is active the chip will remain in the TX state until the current packet has been successfully transmitted. Then the state will change as indicated by the MCSM1.TXOFF_MODE setting. The possible destinations are the same as for RX.

The CPU can change the state from RX to TX and vice versa by using the command strobes. If the radio controller is currently in transmit and the SRX strobe is written, the current transmission will be ended and the transition to RX will be done.

If the radio controller is in RX when the STX or SFSTXON command strobes are issued, the "TX if clear channel" function will be used. If the channel is not clear, the chip will remain in RX. The MCSM1.CCA_MODE setting controls the conditions for clear channel assessment. See section 15.10.4 on page 166 for details.

The SIDLE command strobe can always be issued to force the radio controller to go to the IDLE state.

15.12.2 Timing

The radio controller controls most timing in CC1110, such as synthesizer calibration, PLL lock and RT/TX turnaround times. Timing from IDLE to RX and IDLE to TX is constant, dependent on the auto calibration setting. RX/TX and TX/RX turnaround times are constant. The calibration time is constant 18739 clock periods. Table 48 shows timing in crystal clock cycles for key state transitions.

Power on time and XOSC start-up times are variable, but within the limits stated in Table 8.

Description	XOSC periods	26MHz crystal
Idle to RX, no calibration	2298	88.4µs
Idle to RX, with calibration	~21037	809µs
Idle to TX/FSTXON, no calibration	2298	88.4µs
Idle to TX/FSTXON, with calibration	~21037	809µs
TX to RX switch	560	21.5µs
RX to TX switch	250	9.6µs
RX or TX to IDLE, no calibration	2	0.1µs
RX or TX to IDLE, with calibration	~18739	721µs
Manual calibration	~18739	721µs

Table 48: State transition timing

15.12.3 RX Termination Timer

CC1110 has optional functions for automatic termination of RX after a programmable time. The main use for this functionality is wake-on-radio (WOR), but it may be useful for other applications. The termination timer starts when enabling the demodulator. The timeout is programmable with the MCSM2.RX_TIME setting. When the timer expires, the radio controller will check the condition for staying in RX; if the condition is not met, RX will terminate. After the timeout, the condition will be checked continuously.

The programmable conditions are:

- MCSM2.RX_TIME_QUAL=0: Continue receive if sync word has been found
- MCSM2.RX_TIME_QUAL=1: Continue receive if sync word has been found or preamble quality is above threshold (PQT)

If the system can expect the transmission to have started when enabling the receiver, the

MCSM2.RX_TIME_RSSI function can be used. The radio controller will then terminate RX if the first valid carrier sense sample indicates no carrier (RSSI below threshold). See Section 15.10.3 on page 166 for details on Carrier Sense.

For OOK modulation, lack of carrier sense is only considered valid after eight symbol periods. Thus, the MCSM2.RX_TIME_RSSI function can be used in OOK mode when the distance between "1" symbols is 8 or less.

If RX terminates due to no carrier sense when the MCSM2.RX_TIME_RSSI function is used, or if no sync word was found when using the MCSM2.RX_TIME timeout function, the chip will always go back to IDLE. Otherwise, the MCSM1.RXOFF_MODE setting determines the state to go to when RX ends.

15.13 Frequency Programming

The frequency programming in CC1110 is designed to minimize the programming needed in a channel-oriented system.

To set up a system with channel numbers, the desired channel spacing is programmed with the MDMCFG0.CHANSPC_M and MDMCFG1.CHANSPC_E registers. The channel spacing registers are mantissa and exponent respectively.

The base or start frequency is set by the 24 bit frequency word located in the FREQ2, FREQ1 and FREQ0 registers. This word will typically be set to the centre of the lowest channel frequency that is to be used.

The desired channel number is programmed with the 8-bit channel number register, CHANNR.CHAN, which is multiplied by the channel offset. The resultant carrier frequency is given by the expression shown below.

$$f_{carrier} = \frac{f_{XOSC}}{2^{16}} \cdot \left(FREQ + CHAN \cdot \left((256 + CHANSPC\_M) \cdot 2^{CHANSPC\_E-2} \right) \right)$$

With a 26 MHz crystal the maximum channel spacing is 405 kHz. To get e.g. 1 MHz channel spacing one solution is to use 333 kHz channel spacing and select each third channel in CHANNR.CHAN.

The preferred IF frequency is programmed with the FSCTRL1.FREQ_IF register. The IF frequency is given by:

$$f_{IF} = \frac{f_{XOSC}}{2^{10}} \cdot FREQ\_IF$$

Note that the SmartRF® Studio software automatically calculates the optimum FSCTRL1.FREQ_IF register setting based on channel spacing and channel filter bandwidth.

If any frequency programming register is altered when the frequency synthesizer is running, the synthesizer may give an undesired response. Hence, the frequency programming should only be updated when the radio is in the IDLE state.

15.14 VCO

The VCO is completely integrated on-chip.

15.14.1 VCO and PLL Self-Calibration

The VCO characteristics will vary with temperature and supply voltage changes, as well as the desired operating frequency. In order to ensure reliable operation, CC1110 includes frequency synthesizer self-calibration circuitry. This calibration should be done regularly, and must be performed after turning on power and before using a new frequency (or channel). The number of XOSC cycles for completing the PLL calibration is given in Table 48 on page 169.

The calibration can be initiated automatically or manually. The synthesizer can be automatically calibrated each time the synthesizer is turned on, or each time the synthesizer is turned off. This is configured with the MCSM0.FS_AUTOCAL register setting. In manual mode, the calibration is initiated when the SCAL command strobe is activated in the IDLE mode.

Note that the calibration values are maintained in power-down modes PM2/3, so the calibration is still valid after waking up from these power-down modes (unless supply

voltage or temperature has changed significantly).

15.15 Output Power Programming

The RF output power level from the device has two levels of programmability, as illustrated in Figure 39. Firstly, the PA_TABLE7-PA_TABLE0 registers can hold up to eight user selected output power settings. Secondly, the 3-bit FREND0.PA_POWER value selects which PA_TABLE7-0 register to use. This two-level functionality provides flexible PA power ramp up and ramp down at the start and end of transmission, as well as ASK modulation shaping.n each case, all the PA power settings from PA_TABLE0 from i up to the FREND0.PA_POWER value are used.

The power ramping at the start and at the end of a packet can be turned off by setting FREND0.PA_POWER to zero and then programming the desired output power in PA_TABLE0. Table 49 contains recommended PA_TABLE settings for various output levels and frequency bands, together with current consumption in the RF transceiver.

With ASK modulation, the eight power settings are used for shaping. The modulator contains a counter that counts up when transmitting a one and down when transmitting a zero. The counter counts at a rate equal to 8 times the symbol rate. The counter saturates at FREND0.PA_POWER and 0 respectively. This counter value is used as an index for a lookup in the power table. Thus, in order to utilize the whole table, FREND0.PA_POWER should be 7 when ASK is active. The shaping of the ASK signal is dependent on the configuration in the PA_TABLE7-0 registers.

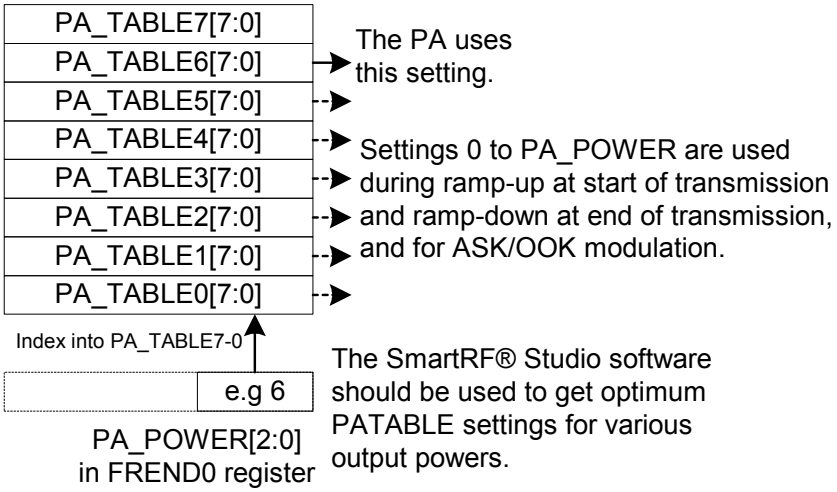


Figure 39: PA_POWER and PA_TABLE7-0

	315MHz		433MHz		868MHz		915MHz	
Output power [dBm]	Setting	Current consumption, typ. [mA]	Setting	Current consumption, typ. [mA]	Setting	Current consumption, typ. [mA]	Setting	Current consumption, typ. [mA]
-30	0x04	10.9	0x68	11.7	0x03	12.0	0x11	11.9
-20	0x17	11.5	0x6C	12.2	0x0D	12.6	0x0B	12.4
-15	0x1D	12.2	0x1C	12.8	0x1C	13.2	0x1B	13.1
-10	0x26	13.4	0x06	14.3	0x34	14.6	0x6D	13.7
-5	0x69	13.0	0x3A	13.8	0x67	14.4	0x67	14.2
0	0x51	15.1	0x51	16.1	0x60	16.8	0x50	16.5
5	0x86	18.3	0x85	19.3	0x85	19.9	0x85	19.3
7	0xCC	22.2	0xC8	24.0	0xCC	25.6	0xC9	25.6
10	0xC3	26.9	0xC0	28.8	0xC3	30.3	0xC1	30.2

Table 49: Optimum PA_TABLE settings for various output power levels (subject to changes)

	315MHz		433MHz		868MHz		915MHz	
Default power setting	Output power [dBm]	Current consumption, typ. [mA]	Output power [dBm]	Current consumption, typ. [mA]	Output power [dBm]	Current consumption, typ. [mA]	Output power [dBm]	Current consumption, typ. [mA]
0xC6	8.9	25.1	7.8	25.0	8.9	28.3	8.1	26.8

Table 50: Output power and current consumption for default PATABLE setting

15.16 Antenna Interface

The balanced RF input and output of ***CC1110*** share two common pins and are designed for a simple, low-cost matching and balun network on the printed circuit board. The receive and transmit switching at the ***CC1110*** front-end is controlled by a dedicated on-chip function, eliminating the need for an external RX/TX-switch.

A few passive external components combined with the internal RX/TX switch/termination circuitry ensures match in both RX and TX mode.

Although ***CC1110*** has a balanced RF input/output, the chip can be connected to a single-ended antenna with few external low cost capacitors and inductors.

The passive matching/filtering network connected to ***CC1110*** should have the following differential impedance as seen from the RF-port (RF_P and RF_N) towards the antenna:

$Z_{out\ 315MHz} = 122 + j31\ \Omega$

$Z_{out\ 433MHz} = 116 + j41\ \Omega$

$Z_{out\ 868MHz} = 87 + j43\ \Omega$

15.17 Radio Registers

This section describes all RF registers used for control and status for the radio. The RF registers reside in XDATA memory space in the region 0xDF00-0xDF3D. Table 51 gives an overview of register addresses while the remaining tables in this section describe each register.

XDATA Address	Register	Description
0xDF00	SYNC1	Sync word, high byte
0xDF01	SYNC0	Sync word, low byte
0xDF02	PKTLEN	Packet length
0xDF03	PKTCTRL1	Packet automation control
0xDF04	PKTCTRL0	Packet automation control
0xDF05	ADDR	Device address
0xDF06	CHANNR	Channel number
0xDF07	FSCTRL1	Frequency synthesizer control
0xDF08	FSCTRL0	Frequency synthesizer control
0xDF09	FREQ2	Frequency control word, high byte
0xDF0A	FREQ1	Frequency control word, middle byte
0xDF0B	FREQ0	Frequency control word, low byte
0xDF0C	MDMCFG4	Modem configuration
0xDF0D	MDMCFG3	Modem configuration
0xDF0E	MDMCFG2	Modem configuration
0xDF0F	MDMCFG1	Modem configuration
0xDF10	MDMCFG0	Modem configuration
0xDF11	DEVIATN	Modem deviation setting
0xDF12	MCSM2	Main Radio Control State Machine configuration
0xDF13	MCSM1	Main Radio Control State Machine configuration
0xDF14	MCSM0	Main Radio Control State Machine configuration
0xDF15	FOCCFG	Frequency Offset Compensation configuration
0xDF16	BSCFG	Bit Synchronization configuration
0xDF17	AGCTRL2	AGC control
0xDF18	AGCTRL1	AGC control
0xDF19	AGCTRL0	AGC control
0xDF1A	FREND1	Front end RX configuration
0xDF1B	FREND0	Front end TX configuration
0xDF1C	FSCAL3	Frequency synthesizer calibration
0xDF1D	FSCAL2	Frequency synthesizer calibration
0xDF1E	FSCAL1	Frequency synthesizer calibration
0xDF1F	FSCAL0	Frequency synthesizer calibration
0xDF20	-	Reserved
0xDF21	-	Reserved
0xDF22	-	Reserved
0xDF23	-	Reserved
0xDF24	-	Reserved
0xDF25	-	Reserved
0xDF27	PA_TABLE7	PA output power setting
0xDF28	PA_TABLE6	PA output power setting
0xDF29	PA_TABLE5	PA output power setting
0xDF2A	PA_TABLE4	PA output power setting
0xDF2B	PA_TABLE3	PA output power setting
0xDF2C	PA_TABLE2	PA output power setting
0xDF2D	PA_TABLE1	PA output power setting
0xDF2E	PA_TABLE0	PA output power setting
0xDF2F	IOCFG2	GDO2 output pin configuration

XDATA Address	Register	Description
0xDF30	IOCFG1	GDO1 output pin configuration
0xDF31	IOCFG0	GDO0 output pin configuration
0xDF36	PARTNUM	Chip Identifier
0xDF37	VERSION	configuration
0xDF38	FREQEST	Frequency Offset Estimate
0xDF39	LQI	Link Quality Indicator
0xDF3A	RSSI	Received Signal Strength Indication
0xDF3B	MARCSTATE	Main Radio Control State
0xDF3C	PKSTATUS	Packet status
0xDF3D	VCO_VC_DAC	PLL calibration current

Table 51: Overview of RF registers

0xDF2F: IOCFG2 – GDO2 output pin configuration

Bit	Field Name	Reset	R/W	Description
7	-		R0	Reserved
6	GDO2_INV	0	R/W	Invert output, i.e. select active low / high
5:0	GDO2_CFG[5:0]	0x00	R/W	Debug output on P1_7 pin. See Table 52 for description of internal signals which can be output on this pin for debug purpose

0xDF30: IOCFG1 – GDO1 output pin configuration

Bit	Field Name	Reset	R/W	Description
7	GDO_DS	0	R/W	Set high (1) or low (0) output drive strength on the GDO pins.
6	GDO1_INV	0	R/W	Invert output, i.e. select active low / high
5:0	GDO1_CFG[5:0]	0x00	R/W	Debug output on P1_6 pin. See Table 52 for description of internal signals which can be output on this pin for debug purpose

0xDF31: IOCFG0 – GDO0 output pin configuration

Bit	Field Name	Reset	R/W	Description
7	TEMP_SENSOR_ENABLE	0	R/W	Enable analog temperature sensor. Write 0 in all other register bits when using temperature sensor.
6	GDO0_INV	0	R/W	Invert output, i.e. select active low / high
5:0	GDO0_CFG[5:0]	0x00	R/W	Debug output on P1_5 pin. See Table 52 for description of internal signals which can be output on this pin for debug purpose

0xDF00: SYNC1 – Sync word, high byte

Bit	Field Name	Reset	R/W	Description
7:0	SYNC[15:8]	0xD3	R/W	8 MSB of 16-bit sync word

0xDF01: SYNC0 – Sync word, low byte

Bit	Field Name	Reset	R/W	Description
7:0	SYNC[7:0]	0x91	R/W	8 LSB of 16-bit sync word

0xDF02: PKTLEN – Packet length

Bit	Field Name	Reset	R/W	Description
7:0	PACKET_LENGTH	0xFF	R/W	Indicates the packet length when fixed length packets are enabled. If variable length packets are used, this value indicates the maximum length packets allowed.

0xDF03: PKTCTRL1 – Packet automation control

Bit	Field Name	Reset	R/W	Description
7:5	PQT[2:0]	000	R/W	Preamble quality estimator threshold. The preamble quality estimator increases an internal counter by one each time a bit is received that is different from the previous bit, and decreases the counter by 4 each time a bit is received that is the same as the last bit. The counter saturates at 0 and 31. A threshold of 4·PQT for this counter is used to gate sync word detection. When PQT=0 a sync word is always accepted.
4:3	-	00	R0	Reserved
2	APPEND_STATUS	1	R/W	When enabled, two status bytes will be appended to the payload of the packet. The status bytes contain RSSI and LQI values, as well as the CRC OK flag.
1:0	ADR_CHK[1:0]	00	R/W	Controls address check configuration of received packages. Setting — Address check configuration 0 (00) — No address check 1 (01) — Address check, no broadcast 2 (10) — Address check, 0 (0x00) broadcast 3 (11) — Address check, 0 (0x00) and 255 (0xFF) broadcast

0xDF04: PKTCTRL0 – Packet automation control

Bit	Field Name	Reset	R/W	Description
7	-	0	R0	Reserved
6	WHITE_DATA	1	R/W	Turn data whitening on / off 0: Whitening off 1: Whitening on
5:4	PKT_FORMAT[1:0]	00	R/W	Format of RX and TX data Setting: Packet format 0 (00): Normal mode, use FIFOs for RX and TX 1 (01): Serial Synchronous mode, used for backwards compatibility 2 (10): Random TX mode; sends random data using PN9 generator. Used for test. Works as normal mode, setting 0 (00), in RX. 3 (11): Asynchronous transparent mode. Data in on GDO0 and Data out on either of the GDO pins
3	CC2400_EN	0	R/W	Enable CC2400 support. Use same CRC implementation as CC2400.
2	CRC_EN	1	R/W	1: CRC calculation in TX and CRC check in RX enabled 0: CRC disabled for TX and RX
1:0	LENGTH_CONFIG[1:0]	01	R/W	Configure the packet length Setting: Packet length configuration 0 (00): Fixed length packets, length configured in PKTLEN register 1 (01): Variable length packets, packet length configured by the first byte after sync word 2 (10): Enable infinite length packets 3 (11): Reserved

0xDF05: ADDR – Device address

Bit	Field Name	Reset	R/W	Description
7:0	DEVICE_ADDR[7:0]	0x00	R/W	Address used for packet filtration. Optional broadcast addresses are 0 (0x00) and 255 (0xFF).

0xDF06: CHANNR – Channel number

Bit	Field Name	Reset	R/W	Description
7:0	CHAN[7:0]	0x00	R/W	The 8-bit unsigned channel number, which is multiplied by the channel spacing setting and added to the base frequency.

0xDF07: FSCTRL1 – Frequency synthesizer control

Bit	Field Name	Reset	R/W	Description
7:5	-	000	R0	Not used
4:0	FREQ_IF[4:0]	01111	R/W	The desired IF frequency to employ in RX. Subtracted from FS base frequency in RX and controls the digital complex mixer in the demodulator. $f_{IF} = \frac{f_{XOSC}}{2^{10}} \cdot FREQ\_IF$ The default value gives an IF frequency of 254 kHz, assuming a 26.0 MHz crystal.

0xDF08: FSCTRL0 – Frequency synthesizer control

Bit	Field Name	Reset	R/W	Description
7:0	FREQOFF[7:0]	0x00	R/W	Frequency offset added to the base frequency before being used by the FS. (2-complement). Resolution is $F_{XTAL}/2^{14}$ (1.5 kHz-1.7 kHz); range is ±186 kHz to ±217 kHz, dependent of XTAL frequency. The SAFC strobe command and the automatic AFC mechanism add the current FREQEST value to FREQOFF.

0xDF09: FREQ2 – Frequency control word, high byte

Bit	Field Name	Reset	R/W	Description
7:6	FREQ[23:22]	00	R	FREQ[23:22] is always (the FREQ2 register is less than 36 with 26MHz or higher crystal)

Bit	Field Name	Reset	R/W	Description
5:0	FREQ[21:16]	0x1E	R/W	FREQ[23:0] is the base frequency for the frequency synthesizer in increments of $F_{XOSC}/2^{16}$. $f_{carrier} = \frac{f_{XOSC}}{2^{16}} \cdot FREQ[23:0]$ The default frequency word gives a base frequency of 800 MHz, assuming a 26.0 MHz crystal. With the default channel spacing settings, the following FREQ2 values and channel numbers can be used: FREQ2 / Base frequency / Frequency range (CHAN numbers) 10 (0x0A) / 280MHz / 300.2MHz-331MHz (101-255) 11 (0x0B) / 306MHz / 306MHz-347.8MHz (0-209) 14 (0x0E) / 384MHz / 400.2MHz-435MHz (81-255) 15 (0x0F) / 410MHz / 410MHz-461MHz (0-255) 16 (0x10) / 436MHz / 436MHz-463.8MHz (0-139) 17 (0x11) / 462MHz / 462MHz-463.8MHz (0-9) 30 (0x1E) / 800MHz / 800.2MHz-851MHz (1-255) 31 (0x1F) / 826MHz / 826MHz-877MHz (0-255) 32 (0x20) / 852MHz / 852MHz-903MHz (0-255) 33 (0x21) / 878MHz / 878MHz-927.8MHz (0-249) 34 (0x22) / 904MHz / 904MHz-927.8MHz (0-119)

0xDF0A: FREQ1 – Frequency control word, middle byte

Bit	Field Name	Reset	R/W	Description
7:0	FREQ[15:8]	0xC4	R/W	Ref. FREQ2 register

0xDF0B: FREQ0 – Frequency control word, low byte

Bit	Field Name	Reset	R/W	Description
7:0	FREQ[7:0]	0xEC	R/W	Ref. FREQ2 register

0xDF0C: MDMCFG4 – Modem configuration

Bit	Field Name	Reset	R/W	Description
7:6	CHANBW_E[1:0]	10	R/W	
5:4	CHANBW_M[1:0]	00	R/W	Sets the decimation ratio for the delta-sigma ADC input stream and thus the channel bandwidth. $BW_{channel} = \frac{f_{XOSC}}{8 \cdot (4 + CHANBW\_M) \cdot 2^{CHANBW\_E}}$ Note that the combination CHANBW_E=0 and CHANBW_M=0 is not supported. The default values give 203kHz channel filter bandwidth, assuming a 26.0MHz crystal.
3:0	DRATE_E[3:0]	1100	R/W	The exponent of the user specified symbol rate

0xDF0D: MDMCFG3 – Modem configuration

Bit	Field Name	Reset	R/W	Description
7:0	DRATE_M[7:0]	0x22	R/W	The mantissa of the user specified symbol rate. The symbol rate is configured using an unsigned, floating-point number with 9-bit mantissa and 4-bit exponent. The 9th bit is a hidden '1'. The resulting data rate is: $R_{DATA} = \frac{(256 + DRATE\_M) \cdot 2^{DRATE\_E}}{2^{28}} \cdot f_{XOSC}$ The default values give a data rate of 115.051kbps (closest setting to 115.2kbps), assuming a 26.0MHz crystal.

0xDF0E: MDMCFG2 – Modem configuration

Bit	Field Name	Reset	R/W	Description
7	DEM_DCFILT_OFF	0	R/W	Disable digital DC blocking filter before demodulator. 0 = Enable (better sensitivity for data rates ≤ 250 kbps) 1 = Disable (reduced power consumption) The recommended IF frequency changes when the DC blocking is disabled.
6:4	MOD_FORMAT[2:0]	000	R/W	The modulation format of the radio signal Setting — Modulation format 0 (000) — 2-FSK 1 (001) — GFSK 2 (010) — - 3 (011) — ASK/OOK 4 (100) — - 5 (101) — - 6 (110) — - 7 (111) — MSK
3	MANCHESTER_EN	0	R/W	Enables Manchester encoding/decoding. 0 = Disable 1 = Enable
2:0	SYNC_MODE[2:0]	010	R/W	Combined sync-word qualifier mode. The values 0 (000) and 4 (100) disables sync word transmission in TX and sync word detection in RX. The values 1 (001), 2 (001), 5 (101) and 6 (110) enables 16-bit sync word transmission in TX and 16-bits sync word detection in RX. Only 15 of 16 bits need to match in RX when using setting 1 (001) or 5 (101). The values 3 (011) and 7 (111) enables repeated sync word transmission in RX and 32-bits sync word detection in RX (only 30 of 32 bits need to match). Setting — Sync-word qualifier mode 0 (000) — No preamble/sync 1 (001) — 15/16 sync word bits detected 2 (010) — 16/16 sync word bits detected 3 (011) — 30/32 sync word bits detected 4 (100) — No preamble/sync, carrier-sense above threshold 5 (101) — 15/16 + carrier-sense above threshold 6 (110) — 16/16 + carrier-sense above threshold 7 (111) — 30/32 + carrier-sense above threshold

0xDF0F: MDMCFG1 – Modem configuration

Bit	Field Name	Reset	R/W	Description
7	FEC_EN	0	R/W	Enable Forward Error Correction (FEC) with interleaving for packet payload 0 = Disable 1 = Enable
6:4	NUM_PREAMBLE[2:0]	010	R/W	Sets the minimum number of preamble bytes to be transmitted Setting — Number of preamble bytes 0 (000) — 2 1 (001) — 3 2 (010) — 4 3 (011) — 6 4 (100) — 8 5 (101) — 12 6 (110) — 16 7 (111) — 24
3:2	-	0	R0	Reserved
1:0	CHANSPC_E[1:0]	10	R/W	2 bit exponent of channel spacing

0xDF10: MDMCFG0 – Modem configuration

Bit	Field Name	Reset	R/W	Description
7:0	CHANSPC_M[7:0]	0xF8	R/W	8-bit mantissa of channel spacing (initial 1 assumed). The channel spacing is multiplied by the channel number CHAN and added to the base frequency. It is unsigned and has the format: $\Delta f_{CHANNEL} = \frac{f_{XOSC}}{2^{18}} \cdot (256 + CHANSPC\_M) \cdot 2^{CHANSPC\_E} \cdot CHAN$ The default values give 199.951 kHz channel spacing (the closest setting to 200 kHz), assuming 26.0 MHz crystal frequency.

0xDF11: DEVIATN – Modem deviation setting

Bit	Field Name	Reset	R/W	Description
7	-	0	R0	Reserved
6:4	DEVIATION_E[2:0]	100	R/W	Deviation exponent
3	-	0	R0	Reserved
2:0	DEVIATION_M[2:0]	111	R/W	When MSK modulation is enabled: Sets fraction of symbol period used for phase change. When 2-FSK/GFSK modulation is enabled: Deviation mantissa, interpreted as a 4-bit value with MSB implicit 1. The resulting FSK deviation is given by: $f_{dev} = \frac{f_{xosc}}{2^{17}} \cdot (8 + DEVIATION\_M) \cdot 2^{DEVIATION\_E}$ The default values give ±47.607 kHz deviation, assuming 26.0 MHz crystal frequency.

0xDF12: MCSM2 – Main Radio Control State Machine configuration

Bit	Field Name	Reset	R/W	Description
7:5	-	000	R0	Reserved
4	RX_TIME_RSSI	0	R/W	Direct RX termination based on RSSI measurement (carrier sense). For ASK/OOK modulation, RX times out if there is no carrier sense in the first 8 symbol periods.
3	RX_TIME_QUAL	0	R/W	When the RX_TIME timer expires, the chip checks if sync word is found when RX_TIME_QUAL=0, or either sync word is found or PQT is set when RX_TIME_QUAL=1.
2:0	RX_TIME[2:0]	111	R/W	Timeout for sync word search in RX. The timeout is relative to the programmed EVENT0 timeout. The RX timeout is scaled by 1 bit less than the EVENT0 timeout with respect to the `WORCTRL.WOR_RES` setting, as very long timeouts probably also will use very low RX duty cycles. Setting / RX timeout / Duty cycle, WOR 0 (000) / $T_{EVENT0} / 2^{(3+WOR\_RES)}$ / 12.5% / 2^{WOR_RES} 1 (001) / $T_{EVENT0} / 2^{(4+WOR\_RES)}$ / 6.25% / 2^{WOR_RES} 2 (010) / $T_{EVENT0} / 2^{(5+WOR\_RES)}$ / 3.125% / 2^{WOR_RES} 3 (011) / $T_{EVENT0} / 2^{(6+WOR\_RES)}$ / 1.563% / 2^{WOR_RES} 4 (100) / $T_{EVENT0} / 2^{(7+WOR\_RES)}$ / 0.781% / 2^{WOR_RES} 5 (101) / $T_{EVENT0} / 2^{(8+WOR\_RES)}$ / 0.391% / 2^{WOR_RES} 6 (110) / $T_{EVENT0} / 2^{(9+WOR\_RES)}$ / 0.195% / 2^{WOR_RES} 7 (111) / Until end of packet / N/A (no timeout) Note that the RC oscillator must be enabled in order to use setting 0-6, because the timeout counts RC oscillator periods. The timeout counter resolution is limited: With `RX_TIME=0`, the timeout count is given by the 13MSBs of EVENT0, decreasing to the 7MSBs of EVENT0 with `RX_TIME=6`.

0xDF13: MCSM1 – Main Radio Control State Machine configuration

Bit	Field Name	Reset	R/W	Description
7:6	-	00	R0	Reserved
5:4	CCA_MODE[1:0]	11	R/W	Selects CCA_MODE; Reflected in CCA signal Setting — Clear channel indication 0 (00) — Always 1 (01) — If RSSI below threshold 2 (10) — Unless currently receiving a packet 3 (11) — If RSSI below threshold unless currently receiving a packet
3:2	RXOFF_MODE[1:0]	00	R/W	Select what should happen when a packet has been received Setting — Next state after finishing packet reception 0 (00) — IDLE 1 (01) — FSTXON 2 (10) — TX 3 (11) — Stay in RX
1:0	TXOFF_MODE[1:0]	00	R/W	Select what should happen when a packet has been sent (TX) Setting — Next state after finishing packet transmission 0 (00) — IDLE 1 (01) — FSTXON 2 (10) — Stay in TX (start sending preamble) 3 (11) — RX

0xDF14: MCSM0 – Main Radio Control State Machine configuration

Bit	Field Name	Reset	R/W	Description
7:6	-	00	R0	Reserved
5:4	FS_AUTOCAL[1:0]	00	R/W	Automatically calibrate when going to RX or TX, or back to IDLE Setting — When to perform automatic calibration 0 (00) — Never (manually calibrate using SCAL strobe) 1 (01) — When going from IDLE to RX or TX (or FSTXON) 2 (10) — When going from RX or TX back to IDLE 3 (11) — Every 4th time when going from RX or TX to IDLE In some automatic wake-on-radio (WOR) applications, using setting 3 (11) can significantly reduce current consumption.
3:0	-	0100	R	Reserved

0xDF15: FOCCFG – Frequency Offset Compensation configuration

Bit	Field Name	Reset	R/W	Description
7:6	-	00	R0	Reserved
5:0	FOCCFG[5:0]	0x76	R/W	Frequency offset compensation configuration. The value to use in this register is given by the SmartRF® Studio software.

0xDF16: BSCFG – Bit Synchronization configuration

Bit	Field Name	Reset	R/W	Description
7:0	BSCFG[7:0]	0x6C	R/W	Bit Synchronization configuration. The value to use in this register is given by the SmartRF® Studio software.

0xDF17: AGCCTRL2 – AGC control

Bit	Field Name	Reset	R/W	Description
7:6	MAX_DVGA_GAIN[1:0]	0 (00)	R/W	Reduces the maximum allowable DVGA gain. Setting — Allowable DVGA settings 0 (00) — All gain settings can be used 1 (01) — The highest gain setting can not be used 2 (10) — The 2 highest gain settings can not be used 3 (11) — The 3 highest gain settings can not be used
5:3	MAX_LNA_GAIN[2:0]	0 (000)	R/W	Sets the maximum allowable LNA + LNA 2 gain relative to the maximum possible gain. Setting — Maximum allowable LNA + LNA 2 gain 0 (000) — Maximum possible LNA + LNA 2 gain 1 (001) — Approx. 2.6 dB below maximum possible gain 2 (010) — Approx. 6.1 dB below maximum possible gain 3 (011) — Approx. 7.4 dB below maximum possible gain 4 (100) — Approx. 9.2 dB below maximum possible gain 5 (101) — Approx. 11.5 dB below maximum possible gain 6 (110) — Approx. 14.6 dB below maximum possible gain 7 (111) — Approx. 17.1 dB below maximum possible gain
2:0	MAGN_TARGET[2:0]	3 (011)	R/W	These bits set the target value for the averaged amplitude from the digital channel filter (1 LSB = 0 dB). Setting — Target amplitude from channel filter 0 (000) — 24 dB 1 (001) — 27 dB 2 (010) — 30 dB 3 (011) — 33 dB 4 (100) — 36 dB 5 (101) — 38 dB 6 (110) — 40 dB 7 (111) — 42 dB

0xDF18: AGCCTRL1 – AGC control

Bit	Field Name	Reset	R/W	Description
7	Reserved		R0	
6	AGC_LNA_PRIORITY	1	R/W	Selects between two different strategies for LNA and LNA 2 gain adjustment. When 1, the LNA gain is decreased first. When 0, the LNA 2 gain is decreased to minimum before decreasing LNA gain.
5:4	CARRIER_SENSE_REL_THR[1:0]	0 (00)	R/W	Sets the relative change threshold for asserting carrier sense Setting – Carrier sense relative threshold 0 (00) – Relative carrier sense threshold disabled 1 (01) – 6 dB increase in RSSI value 2 (10) – 10 dB increase in RSSI value 3 (11) – 14 dB increase in RSSI value
3:0	CARRIER_SENSE_ABS_THR[3:0]	0 (0000)	R/W	Sets the absolute RSSI threshold for asserting carrier sense. The 2-complement signed threshold is programmed in steps of 1 dB and is relative to the MAGN_TARGET setting. Setting – Carrier sense absolute threshold (Equal to channel filter amplitude when AGC has not decreased gain) -8 (1000) – Absolute carrier sense threshold disabled -7 (1001) – 7 dB below MAGN_TARGET setting … – … -1 (1111) – 1 dB below MAGN_TARGET setting 0 (0000) – At MAGN_TARGET setting 1 (0001) – 1 dB above MAGN_TARGET setting … – … 7 (0111) – 7 dB above MAGN_TARGET setting

0xDF19: AGCCTRL0 – AGC control

Bit	Field Name	Reset	R/W	Description
7:0	AGCCTRL0[7:0]	145 (0x91)	R/W	AGC control register. The value to use in this register is given by the SmartRF® Studio software.

0xDF1A: FREND1 – Front end RX configuration

Bit	Field Name	Reset	R/W	Description
7:0	FREND1[7:0]	0x56	R/W	Front end RX configuration. The value to use in this register is given by the SmartRF® Studio software.

0xDF1B: FREND0 – Front end TX configuration

Bit	Field Name	Reset	R/W	Description
7:6	-	00	R0	Reserved
5:4	LODIV_BUF_CURRENT_TX[1:0]	01	R/W	Adjusts current TX LO buffer (input to PA). The value to use in this field is given by the SmartRF® Studio software.
3	-	0	R0	Reserved
2:0	PA_POWER[2:0]	000	R/W	Selects PA power setting. This value is an index to the PATABLE, which can be programmed with up to 8 different PA settings. In ASK mode, this selects the PATABLE index to use when transmitting a '1'. PATABLE index zero is used in ASK when transmitting a '0'. The PATABLE settings from index '0' to the PA_POWER value are used for ASK TX shaping, and for power ramp-up/ramp-down at the start/end of transmission in all TX modulation formats.

0xDF1C: FSCAL3 – Frequency synthesizer calibration

Bit	Field Name	Reset	R/W	Description
7:6	FSCAL3[7:6]	2 (10)	R/W	Frequency synthesizer calibration configuration. The value to write in this register before calibration is given by the SmartRF® Studio software.
5:4	CHP_CURR_CAL_EN[1:0]	2 (10)	R/W	Disable charge pump calibration stage when 0
3:0	FSCAL3[3:0]	9 (1001)	R/W	Frequency synthesizer calibration result register. Fast frequency hopping without calibration for each hop can be done by calibrating upfront for each frequency and saving the resulting FSCAL3, FSCAL2 and FSCAL1 register values. Between each frequency hop, calibration can be replaced by writing the FSCAL3, FSCAL2 and FSCAL1 register values corresponding to the next RF frequency.

0xDF1D: FSCAL2 – Frequency synthesizer calibration

Bit	Field Name	Reset	R/W	Description
7:6	-	00	R0	Reserved
5:0	FSCAL2[5:0]	0x0A	R/W	Frequency synthesizer calibration result register. Fast frequency hopping without calibration for each hop can be done by calibrating upfront for each frequency and saving the resulting FSCAL3, FSCAL2 and FSCAL1 register values. Between each frequency hop, calibration can be replaced by writing the FSCAL3, FSCAL2 and FSCAL1 register values corresponding to the next RF frequency.

0xDF1E: FSCAL1 – Frequency synthesizer calibration

Bit	Field Name	Reset	R/W	Description
7:6	-	00	R0	Reserved
5:0	FSCAL1[5:0]	0x02	R/W	Frequency synthesizer calibration result register. Fast frequency hopping without calibration for each hop can be done by calibrating upfront for each frequency and saving the resulting FSCAL3, FSCAL2 and FSCAL1 register values. Between each frequency hop, calibration can be replaced by writing the FSCAL3, FSCAL2 and FSCAL1 register values corresponding to the next RF frequency.

0xDF1F: FSCAL0 – Frequency synthesizer calibration

Bit	Field Name	Reset	R/W	Description
7	-	0	R0	Reserved
4:0	FSCAL0[6:0]	0x0D	R/W	Frequency synthesizer calibration control. The value to use in this register is given by the SmartRF® Studio software.

0xDF27: PA_TABLE7 – PA power setting 7

Bit	Field Name	Reset	R/W	Description
7:0	PA_TABLE7[7:0]	0x00	R/W	Power amplifier output power setting 7 Currently used PA output power is selected by FREND0.PA_POWER[2:0]

0xDF28: PA_TABLE6 – PA power setting 6

Bit	Field Name	Reset	R/W	Description
7:0	PA_TABLE6[7:0]	0x00	R/W	Power amplifier output power setting 6 Currently used PA output power is selected by FREND0.PA_POWER[2:0]

0xDF29: PA_TABLE5 – PA power setting 5

Bit	Field Name	Reset	R/W	Description
7:0	PA_TABLE5[7:0]	0x00	R/W	Power amplifier output power setting 5 Currently used PA output power is selected by FREND0.PA_POWER[2:0]

0xDF2A: PA_TABLE4 – PA power setting 4

Bit	Field Name	Reset	R/W	Description
7:0	PA_TABLE4[7:0]	0x00	R/W	Power amplifier output power setting 4 Currently used PA output power is selected by FREND0.PA_POWER[2:0]

0xDF2B: PA_TABLE3 – PA power setting 3

Bit	Field Name	Reset	R/W	Description
7:0	PA_TABLE3[7:0]	0x00	R/W	Power amplifier output power setting 3 Currently used PA output power is selected by FREND0.PA_POWER[2:0]

0xDF2C: PA_TABLE2 – PA power setting 2

Bit	Field Name	Reset	R/W	Description
7:0	PA_TABLE2[7:0]	0x00	R/W	Power amplifier output power setting 2 Currently used PA output power is selected by FREND0.PA_POWER[2:0]

0xDF2D: PA_TABLE1– PA power setting 1

Bit	Field Name	Reset	R/W	Description
7:0	PA_TABLE1[7:0]	0x00	R/W	Power amplifier output power setting 1 Currently used PA output power is selected by FREND0.PA_POWER[2:0]

0xDF2E: PA_TABLE0 – PA power setting 0

Bit	Field Name	Reset	R/W	Description
7:0	PA_TABLE0[7:0]	0x00	R/W	Power amplifier output power setting 0 Currently used PA output power is selected by FREND0.PA_POWER[2:0]

0xDF36: PARTNUM – Chip Identifier

Bit	Field Name	Reset	R/W	Description
7:0	PARTNUM[7:0]	0x01	R	Chip part number

0xDF37: VERSION – Chip Version

Bit	Field Name	Reset	R/W	Description
7:0	VERSION[7:0]	0x00	R	Chip version number.

0xDF38: FREQEST – Frequency Offset Estimate from demodulator

Bit	Field Name	Reset	R/W	Description

Bit	Field Name	Reset	R/W	Description
7:0	FREQOFF_EST		R	The estimated frequency offset (two's complement) of the carrier. Resolution is $F_{XTAL}/2^{14}$ (1.5kHz-1.7kHz); range is ±186kHz to ±217kHz, dependent of XTAL frequency. Frequency offset compensation is only supported for 2-FSK, GFSK and MSK modulation. This register will read 0 when using ASK or OOK modulation.

0xDF39: LQI – Demodulator estimate for Link Quality

Bit	Field Name	Reset	R/W	Description
7	CRC OK		R	The last CRC comparison matched. Cleared when entering/restarting RX mode.
6:0	LQI_EST[6:0]		R	The Link Quality Indicator estimates how easily a received signal can be demodulated. Calculated over the 64 symbols following the sync word (first 8 packet bytes for 2-ary modulation, first 16 packet bytes for 4-ary modulation).

0xDF3A: RSSI – Received signal strength indication

Bit	Field Name	Reset	R/W	Description
7:0	RSSI		R	Received signal strength indicator

0xDF3B: MARCSTATE – Main Radio Control State Machine state

Bit	Field Name	Reset	R/W	Description
7:5	Reserved		R0	
4:0	MARC_STATE[4:0]		R	Main Radio Control FSM State

Value	State name	State (Figure 38, page 168)
0 (0x00)	SLEEP	SLEEP
1 (0x01)	IDLE	IDLE
2 (0x02)	XOFF	XOFF
3 (0x03)	VCOON_MC	MANCAL
4 (0x04)	REGON_MC	MANCAL
5 (0x05)	MANCAL	MANCAL
6 (0x06)	VCOON	FS_WAKEUP
7 (0x07)	REGON	FS_WAKEUP
8 (0x08)	STARTCAL	CALIBRATE
9 (0x09)	BWBOOST	SETTLING
10 (0x0A)	FS_LOCK	SETTLING
11 (0x0B)	IFADCON	SETTLING
12 (0x0C)	ENDCAL	CALIBRATE
13 (0x0D)	RX	RX
14 (0x0E)	RX_END	RX
15 (0x0F)	RX_RST	RX
16 (0x10)	TXRX_SWITCH	TXRX_SETTLING
17 (0x11)	RX_OVERFLOW	RX_OVERFLOW
18 (0x12)	FSTXON	FSTXON
19 (0x13)	TX	TX
20 (0x14)	TX_END	TX
21 (0x15)	RXTX_SWITCH	RXTX_SETTLING
22 (0x16)	TX_UNDERFLOW	TX_UNDERFLOW

0xDF3C: PKTSTATUS – Current GDOx status and packet status

Bit	Field Name	Reset	R/W	Description
7	CRC_OK	0	R	The last CRC comparison matched. Cleared when entering/restarting RX mode.
6	CS	0	R	Carrier sense
5	PQT_REACHED	0	R	Preamble Quality reached
4	CCA	0	R	Clear channel assessment
3	SFD	0	R	Start of Frame Delimiter found
2:0	-	000	R0	Not used.

0xDF3D: VCO_VC_DAC – Current setting from PLL calibration module

Bit	Field Name	Reset	R/W	Description
7:0	VCO_VC_DAC[7:0]		R	Status register for test only.

16 Voltage Regulators

The CC1110 includes a low drop-out voltage regulator. This is used to provide a 1.8 V power supply to the CC1110 digital power supply. The voltage regulator should not be used to provide power to external circuits because of limited power sourcing capability and also due to noise considerations.

The voltage regulator input pin AVDD_DREG is to be connected to the unregulated 2.0 V to 3.6 V power supply. The output of the digital regulator is connected internally in the CC1110 to the digital power supply.

The voltage regulator requires an external decoupling capacitor connected to the DCOUPL pin as described in section 11 on page 29.

16.1 Voltage Regulator Power-on

The voltage regulator is disabled when the CC1110 is placed in power modes PM2 or PM3 (see section 13.10). When the voltage regulator is disabled, register and RAM contents will be retained while the unregulated 2.0 V - 3.6 V power supply is present.

17 Radio Test Output signals

For debug and test purposes, a number of internal status signals in the radio may be output on the port pins P1_7 – P1_5. This debug option is controlled through the RF registers IOCFG2-IOCFG0 . Table 52 shows the value written to IOCFGx.GDOx_CFG[5:0] with the corresponding internal signals that will be output in each case.

GDO0_CFG[5:0] GDO1_CFG[5:0] GDO2_CFG[5:0]	Description
0-7	Not in use
8	Preamble Quality Reached. Asserts when the PQI is above the programmed PQT value.
9	Clear channel assessment. High when RSSI level is below threshold (dependent on the current CCA_MODE setting)
10	Lock detector output
11	Serial Clock. Synchronous to the data in synchronous serial mode. Data is set up on the falling edge and is read on the rising edge of SERIAL_CLK.
12	Serial Synchronous Data Output. Used for synchronous serial mode. The MCU must read DO on the rising edge of SERIAL_CLK. Data is set up on the falling edge by CC1100.
13	Serial transparent Data Output. Used for asynchronous serial mode.
14	Carrier sense. High if RSSI level is above threshold.
15	CRC OK. The last CRC comparison matched. Cleared when entering/restarting RX mode.
16	ADC I/Q – serialized
17	Decimation filter output I/Q + channel filter I/Q + CORDIC + GAIN
18	Demodulator backend key signals (PSK)
19	Demodulator backend key signals (FSK)
20	Data filter output
~~21~~	Not in use
22	RX_HARD_DATA[1]
23	RX_HARD_DATA[0]
24	FPLL
25	CLK_PRE
26	VCO_CURR_COMP
27	PA_PD
28	LNA_PD
29	RX_SYMBOL_TICK
30-46	Not in use
47	Hard-wired to 0 (hard-wired to 1 achieved with inverted polarity selected on output test signal)
48-63	Not in use

Table 52: Debug output signals

18 Evaluation Software

Chipcon provides users of CC1110 with a software program, SmartRF® Studio, which may be used for radio performance and functionality evaluation. SmartRF® Studio runs on Microsoft Windows 95/98 and Microsoft Windows NT/XP/2000. SmartRF® Studio can be downloaded from Chipcon's web page: http://www.chipcon.com

19 Register overview

20 Package Description (QLP 36)

All dimensions are in millimeters, angles in degrees. NOTE: The ***CC1110*** is available in RoHS lead-free package only. Compliant with JEDEC: MO-220.

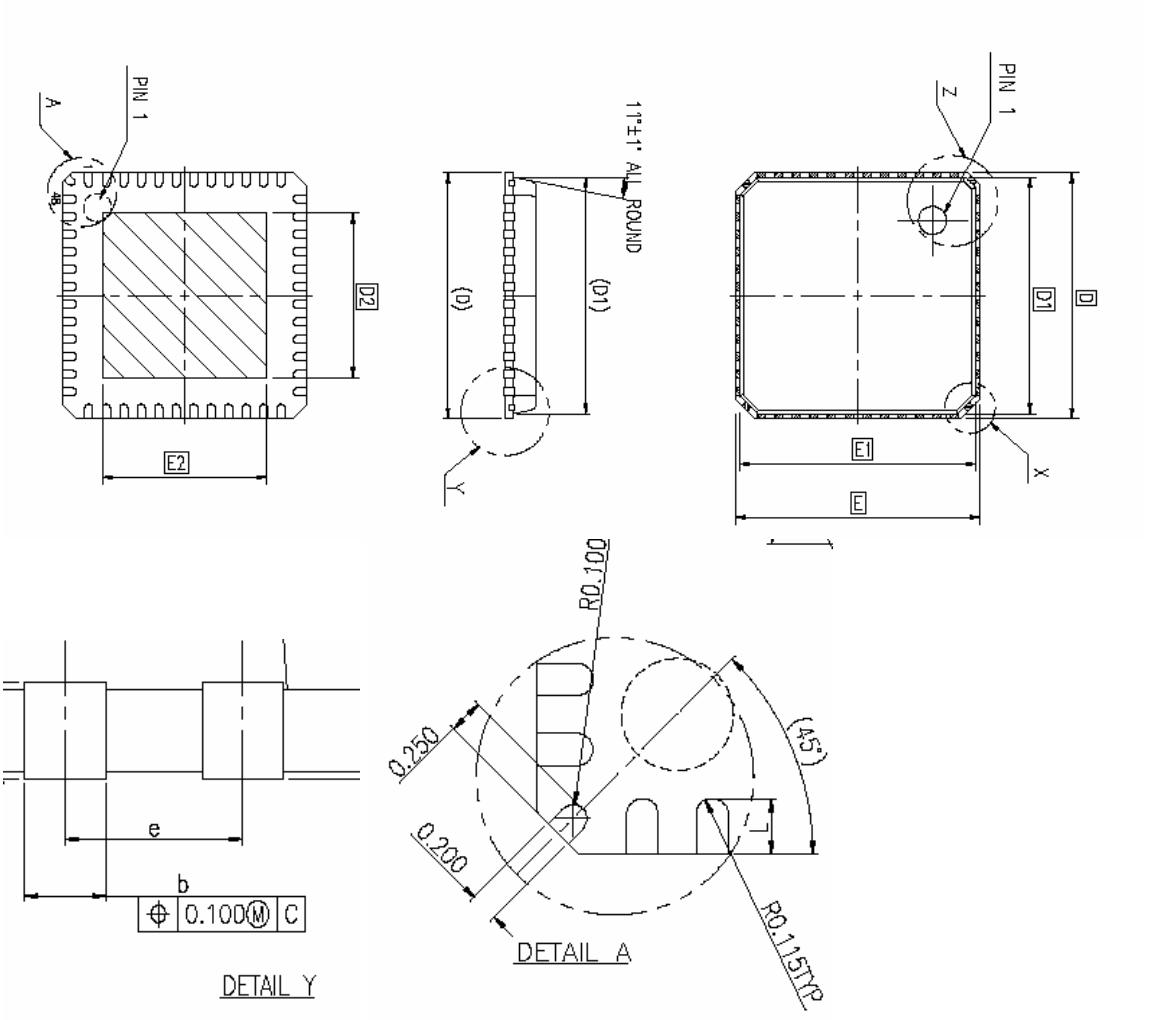


Figure 40: Package dimensions drawing

Quad Leadless Package (QLP)										
		D	D1	E	E1	e	b	L	D2	E2
QLP 36	Min	5.90	5.65	5.90	5.65		0.18	0.45	1.75	1.75
		6.00	5.75	6.00	5.75	0.50	0.23	0.55		
	Max	6.10	5.85	6.10	5.85		0.30	0.65	4.40	4.40
The overall package height is 0.85 +/- 0.05 All dimensions in mm										

Table 53: Package dimensions

20.1 Package thermal properties

Thermal resistance	
Air velocity [m/s]	0
Rth,j-a [K/W]	25.6

Table 54: Thermal properties of QLP 36 package

20.2 Soldering information

The recommendations for lead-free reflow in IPC/JEDEC J-STD-020C should be followed.

20.3 Tray specification

Tray Specification				
Package	Tray Length	Tray Width	Tray Height	Units per Tray
QLP 36	322.6 mm	135.9 mm	7.62 mm	490

Table 55: Tray specification

20.4 Carrier tape and reel specification

Carrier tape and reel is in accordance with EIA Specification 481.

Tape and Reel Specification						
Package	Carrier Tape Width	Component Pitch	Hole Pitch	Reel Diameter	Reel Hub Diameter	Units per Reel
QLP 36	16 mm	12 mm	4 mm	13 inches	7 inches	2500

Table 56: Carrier tape and reel specification

21 Ordering Information

Ordering part number		Description	Minimum Order Quantity (MOQ)
1267	CC1110-RTY1	CC1110, QLP36 package, RoHS compliant Pb-free assembly, trays with 490 pcs per tray. System-on-Chip RF Transceiver	490
1268	CC1110-RTR1	CC1110, QLP36 package, RoHS compliant Pb-free assembly, tape and reel with 2500 pcs per reel. System-on-Chip RF Transceiver	2500
1269	CC1110DK	CC1110 Development Kit	1

Table 57: Ordering Information

22 General Information

22.1 Document History

Revision	Date	Description/Changes
1.0	2006-01-04	First release
1.01	2006-05-11	Preliminary status updated

Table 58: Document history

22.2 Product Status Definitions

Data Sheet Identification	Product Status	Definition
Advance Information	Planned or Under Development	This data sheet contains the design specifications for product development. Specifications may change in any manner without notice.
Preliminary	Engineering Samples and Pre-Production Prototypes	This data sheet contains preliminary data, and supplementary data will be published at a later date. Chipcon reserves the right to make changes at any time without notice in order to improve design and supply the best possible product. The product is not yet fully qualified at this point.
No Identification Noted	Full Production	This data sheet contains the final specifications. Chipcon reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.
Obsolete	Not In Production	This data sheet contains specifications on a product that has been discontinued by Chipcon. The data sheet is printed for reference information only.

Table 59: Product Status Definitions

22.3 Disclaimer

Chipcon AS believes the information contained herein is correct and accurate at the time of this printing. However, Chipcon AS reserves the right to make changes to this product without notice. Chipcon AS does not assume any responsibility for the use of the described product; neither does it convey any license under its patent rights, or the rights of others. The latest updates are available at the Chipcon website or by contacting Chipcon directly.

As far as possible, major changes of product specifications and functionality, will be stated in product specific Errata Notes published at the Chipcon website. Customers are encouraged to sign up for the Chipcon Newsletter for the most recent updates on products and support tools.

When a product is discontinued this will be done according to Chipcon's procedure for obsolete products as described in Chipcon's Quality Manual. This includes informing about last-time-buy options. The Quality Manual can be downloaded from Chipcon's website.

Compliance with regulations is dependent on complete system performance. It is the customer's responsibility to ensure that the system complies with regulations.

This Chipcon product contains Flash memory code protection. However, Chipcon does not guarantee the security of this protection. Chipcon customers using or selling these products with program code do so at their own risk and agree to fully indemnify Chipcon AS for any damages resulting from the use or sale of such products.

Chipcon believes that the Flash memory protection used in this product is one of the most secure in the market today when used in the intended manner and under normal conditions. However, there might be methods to breach the code protection feature. Neither Chipcon nor any other semiconductor manufacturer can guarantee the security of their code protection. Code protection does not mean that we are guaranteeing the product as "unbreakable".

This Chipcon product contains hardware AES encryption. Chipcon does not guarantee the security of the key protection or the security of the encryption scheme. Chipcon customers using or selling products with AES do so at their own risk and agree to fully indemnify Chipcon AS for any damages resulting from the use or sale of such products.

It is the Chipcon customer's responsibility to ensure that sale or export/import of products including this Chipcon product with AES encryption is sold with the required export/import licenses, if necessary, and does not violate any applicable export/import and/or other trade restrictions.

22.4 Trademarks

SmartRF® is a registered trademark of Chipcon AS. SmartRF® is Chipcon's RF technology platform with RF library cells, modules and design expertise. Based on SmartRF® technology Chipcon develops standard component RF circuits as well as full custom ASICs based on customer requirements and this technology.

All other trademarks, registered trademarks and product names are the sole property of their respective owners.

22.5 Life Support Policy

This Chipcon product is not designed for use in life support appliances, devices, or other systems where malfunction can reasonably be expected to result in significant personal injury to the user, or as a critical component in any life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness. Chipcon AS customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Chipcon AS for any damages resulting from any improper use or sale.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265