CC2430ZDK ZigBee Development Kit Pro

User Manual Rev. 1.2

Table of contents

1 Introduction

The CC2430 is the first single-chip, IEEE 802.15.4 compliant and ZigBee™ SOC (System On Chip) RF transceiver with integrated microcontroller. It provides a highly integrated, flexible low-cost solution for applications using the world wide unlicensed 2.4 GHz frequency band.

The CC2430ZDK is the most powerful tool in the market today for developing complete ZigBee™ applications in the minimum timeframe. The kit includes all required hardware and software to evaluate, demonstrate, prototype and develop various ZigBee™ applications.

The CC2430ZDK provides you with Figure 8 Wireless' ZigBee™ Software Development Suite including the industry-leading Z-Stack™, a seat in the 2-day Z-Training and six months email support.

The CC2430ZDK is a very flexible kit that can be used to develop everything from simple light switches to advanced nodes with many peripherals. A set of ZigBee™ sample applications is included with the kit to allow users to test performance right out of the box.

The hardware is representative of an actual application, and is well suited as a prototyping, evaluation and demonstration platform targeting various ZigBee applications. With this kit a ZigBee Logical device type mapped to the IEEE 802.15.4 Full Function and Reduced Function Devices (FFD and RFD) using the CC2430 can be demonstrated. An FFD can take the role of a ZigBee Coordinator, Router, or End Device depending on the ZigBee logical device type configuration. An RFD can act as ZigBee End Device and cannot serve as a ZigBee Coordinator or ZigBee Router.

The CC2430ZDK has the capability to demonstrate ZigBee low power capabilities. The CC2430ZDK is bundled with the Figure 8 Wireless ZigBee stack for evaluation, demonstration, prototyping and developing purposes of the Z-Stack features. The hardware contains an integrated PCB antenna, the IEEE 802.15.4 compliant RF transceiver CC2430 with necessary support components, joystick, buttons and LED’s that can be used to demonstrate the current ZigBee Home Lightning Profile.

This User Manual describes how to use the CC2430 ZigBee Development Kit Pro. The Figure 8 Wireless Z-Stack, tools and documentation will be provided as a distribution package available for download from the Chipcon ZigBee Developer Site after the purchase of this kit. When purchased, customers will be provided with a user name and password to access this site to download the Z-Stack End Customer distribution package.

2 About this manual

This manual contains both tutorial and reference information, and covers both the hardware and software components of the development kit. The following components of the kit have its own user manual that can be downloaded from the Chipcon web site:

- SmartRF® Studio
- Chipcon general packet sniffer
- IAR EW8051 C-compiler and C-Spy debugger (From www.IAR.se)

3 Definitions

CC2430ZDK	CC2430 ZigBee Development Kit Pro. This development kit
SmartRF®04DK	A collective term used for all development kits for the SmartRF®04 platform, i.e. CC2430ZDK
SmartRF®04EB	Motherboard for Evaluation Modules. Used for PC interface, emulator and programming tool as well as user interface.
CC2430DB	CC2430 ZigBee Development Kit Pro. The board described in this user manual
CC2430EM	CC2430 Evaluation Module, a small plug-in module for SmartRF04EB, should be used as reference design for antenna and RF layout
USB MCU	The Silicon Labs C8051F320 MCU used to provide a USB interface on the CC2430ZDK
Factory firmware	The firmware that is supplied programmed into the USB MCU from the factory. This firmware supports SmartRF® Studio operation as well as a stand-alone PER tester
PER	Packet Error Rate. Counts the number of lost and/or faulty packets and displays the ratio: (lost/faulty packets)/number of packets sent
SoC	System on a Chip. A collective term used to refer to Chipcon ICs with on-chip MCU and RF transceiver
ICE	In Circuit Emulator

4 CC2430ZDK contents

4.1 Hardware

2 x SmartRF04EB Evaluation Boards
2 x CC2430EM Evaluation Modules
2 x 2.4GHz Antennas
5 x CC2430DB Demonstration Boards
7 x USB cables
10 x AA batteries
1 x 10-wire flat cable for using SmartRF04EB as emulator for external target
5 x samples of CC2430-F128

4.2 Software

4.2.1 Chipcon Software

Z-Stack™ Industry leading ZigBee™ compliant protocol stack
Z-Stack Configurator™
Z-Stack Profile Builder™
Z-Trace™ debug tool
SmartRF® Studio configuration and test tool
Chipcon Packet Sniffer

4.2.2 3rd party Software

IAR EW8051 C-compiler with C-SPY debugger CD ROM
(60 days evaluation license)

4.3 Training

2-day Z-Training

4.4 Support

6 months email support

Important:

Contact local telecommunication authorities before transmitting an RF signal to ensure that there are no local restrictions on the use of the 2.400 –2483.5 MHz ISM band. The CC2420 operates in the 2.4 GHz frequency band. Although this frequency band is usually described as “world-wide”, some countries do not allow unlicensed operation in this band.

5 Getting started

This section describes how to perform the most common tasks associated with the ZigBee Development Kit Pro

The hardware for the ZigBee Development Kit Pro is based on the CC2430DK. To get started with the ZigBee Development Kit Pro please follow the Quick Start guide on the Chipcon web page for step-by-step information on how to set up the kit.

Make sure to install SmartRF® Studio **before** connecting the ZigBee Development Kit Pro to a PC. SmartRF® Studio can be downloaded from the Chipcon web pages.

5.1 Powering SmartRf04

If several power sources are connected, the ZigBee Development Kit Pro will be powered from the supply that supplies the highest voltage.

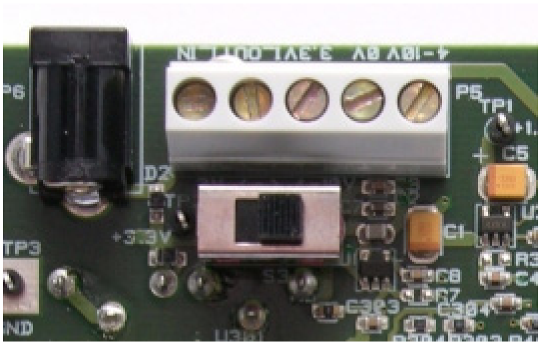

Figure 1: Power switch setting

The Power Switch S3 must be set according to what power source is used. If 3.3V is applied using the power terminal block, the switch should be set to the left position as shown in Figure 1 above. In all other cases, the switch should be set to the right position for power to be applied to the ZigBee Development Kit Pro. This switch can be used to turn off the SmartRF04EB by switching it to the opposite position of that used to turn it on.

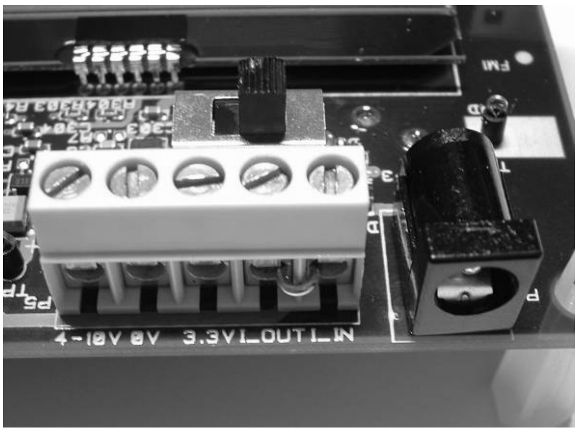

Figure 2: Power connector

The 4-10V input is used for powering the ZigBee Development Kit Pro using the on-board voltage regulator. 0V is the ground connection. The 3.3V terminal is used for powering the ZigBee Development Kit Pro, bypassing the on-board voltage regulator. A voltage between 2.7-3.6V can be used when USB is not active, when using USB the voltage should be limited to 3.0-3.6V.

When shipped from the factory, a jumper is placed between I_OUT and I_IN. To measure the current draw of only the EM, remove the jumper and measure the current using an amperemeter.

The SmartRF04EB can be powered in several different ways:

- **DC jack** connector with standard DC jack power connectors with a 2.5mm centre pin. The centre pin is used for the positive voltage. A 4-10V DC power supply should be used. The onboard voltage regulator supplies 3.3V to the board.
- **Laboratory power supply**. Ground should be connected to the 0V terminal on the power connector (see Figure 2). A 4-10V supply can be connected to the “4-10V” terminal, or a 3.3V supply can be connected to the “3.3V” terminal (the on-board voltage regulators will be bypassed in this case). If a 3.3V supply is used, the supply selection switch should be set to the “3V” position, otherwise the “4-10V” position should be used. If the “4-10V” position is selected, a voltage regulator supplies the circuitry on the ZigBee Development Kit Pro; otherwise the 3.3V supply is applied directly to the SmartRF04EB.
- **USB power**. If the SmartRF04EB is connected to a USB socket on a PC, it will draw power from the USB bus. The onboard voltage regulator supplies 3.3V to the board.
- **Battery power**. The evaluation board includes a 9V-type battery connector on the bottom side of the PCB. A 9V battery or a battery pack that uses a 9V-type connector can be connected to this battery connector. The onboard voltage regulator supplies 3.3V to the board.

Please note that while the SmartRF04 devices have a wide supply range, the components on the ZigBee Development Kit Pro limit the total voltage supply range to 2.7V – 3.6V (3.0V – 3.6V while the USB is active). The ZigBee Development Kit Pro has been designed for a temperature range of –40°C to +85°C (excluding the LCD display). The EM is designed for a temperature range of -40°C to +85°C.

5.2 Powering CC2430DB

The CC2430ZDK can be powered in three different ways:

- DC jack connector with standard DC jack power connectors with a 2.5mm centre pin. The centre pin is used for the positive voltage. A 4-10V DC power supply should be used. The onboard voltage regulator supplies 3.3V to the board.
- USB power. If the CC2430DB is connected to a USB socket on a PC, it will draw power from the USB bus. The onboard voltage regulator supplies 3.3V to the board.
- Battery power. The ZigBee Development Kit Pro includes a battery clip for two AA type batteries

If several power sources are connected, the CC2430 will be powered from the supply that supplies the highest voltage. The USB MCU is only powered when USB or DC jack power is used.

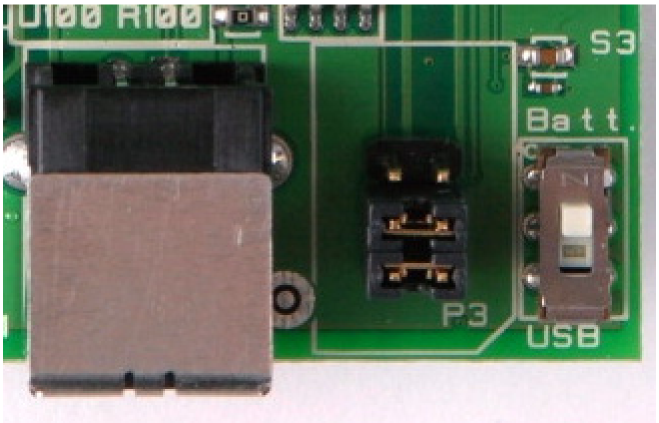

Figure 3: Power switch setting

The Power Switch S3 must be set according to what power source is used. If DC jack or USB power is applied, the switch should be set to the USB position. If the board is powered from batteries, the switch should be set to the BATT position as shown in Figure 3 above. When either DC power or batteries are used to power the board, the switch can be used as an on/off switch.

When powering the CC2430DB from batteries, the CC2430 will work down to 1.8V. Please note that when using USB, the voltage range is limited to 3.0V – 3.6V. The CC2430DB has been designed for a temperature range of –40 C to +85 C.

6 Hardware description SmartRF04EB

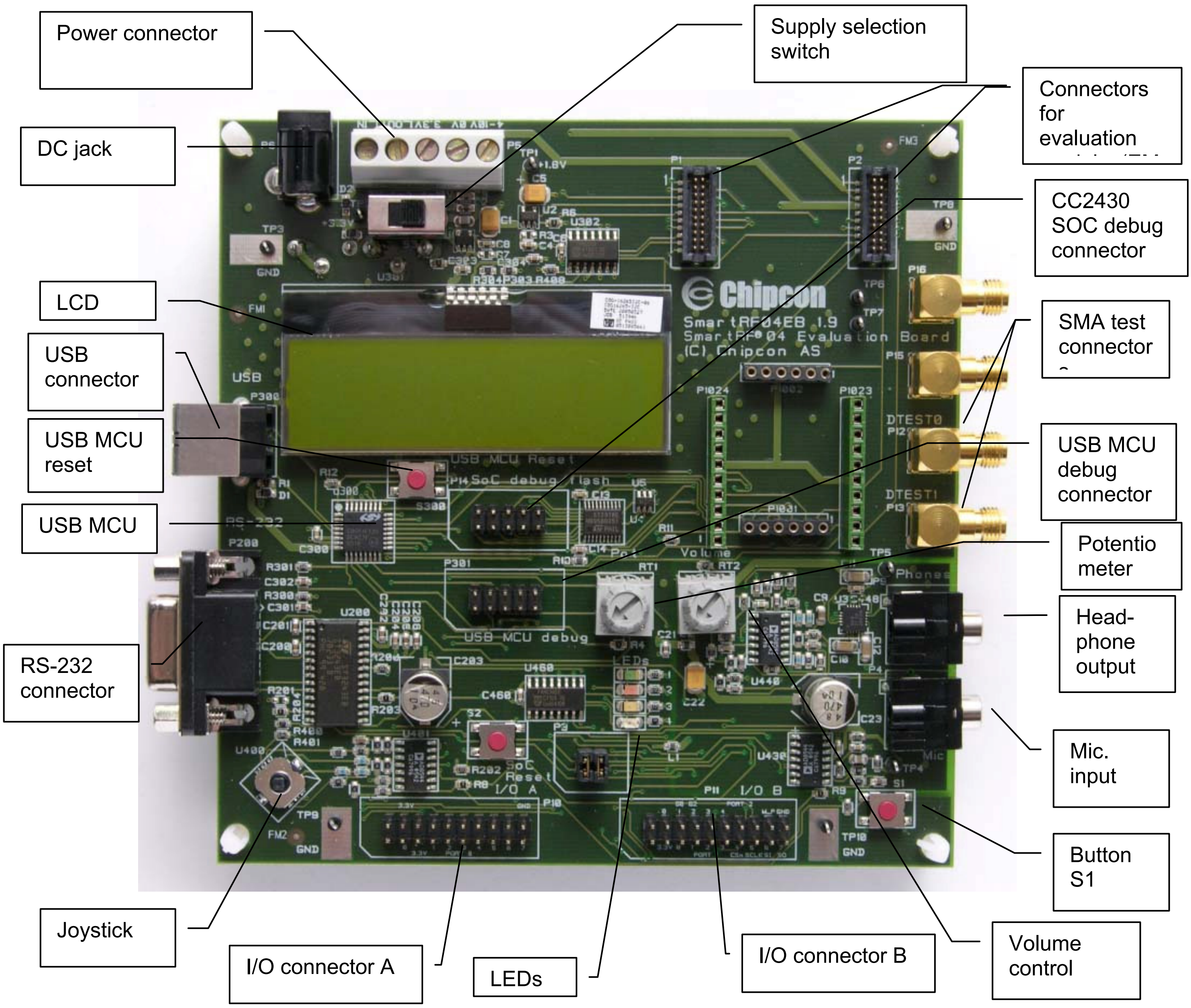

Figure 4: CC2430ZDK overview

The figure above shows the major parts of the SmartRF04. The SmartRF04serves as main platform in the development kit and is designed so that it will support future Chipcon ICs as well.

6.1 USB Interface

The USB interface is used to interface to a PC to run SmartRF® Studio and for programming and debugging using the PC debugging tools and programmers. If SmartRF® Studio connects to the SmartRF04EB and detects an old version of the USB MCU the USB MCU will be upgraded via this interface. The SmartRF04EB can be bus-powered from the USB interface.

6.2 RS-232 interface

The RS-232 can be used by custom applications for communication with other devices. The RS-232 interface utilises a voltage translation device so that the RS-232 port is compatible with bipolar RS-232 levels.

Note that this RS-232 level converter contains a charge-pump power supply that generates electric noise.

6.3 User interface

The CC2430ZDK includes a joystick and a push button as user input devices, and four LEDs and a 2x16 character LCD display as user output devices. The display and user interface is controlled by the application example program loaded in the CC2430.

The factory firmware uses the joystick, push button and LCD display to implement a menu system used to control the packet error rate handling.

6.4 Audio interface

The SmartRF04EB includes a microphone input and headphone output. This interface is not used by Zigbee and IEEE 802.15.4 specific Chipcon devices.

The audio output section consists of a volume control, followed by a 4^{th} order Chebyshev filter. This filter serves to attenuate frequencies above 6 kHz, and so converts the PWM signal to an analogue audio signal. A headphone amplifier IC (TPA4411 from Texas Instruments) is used to drive the headphones.

Note that the headphone amplifier IC uses switch-mode power supply techniques to generate a negative noise, and this may cause electrical noise. The headphone amplifier can be disabled by driving a pin on the USB MCU low (the same pin is used to disable the RS-232 voltage converter).

The audio input section consists of a microphone amplifier that also includes a low-pass anti-aliasing filter.

A standard PC-type headset with separate microphone and headphone mini-jacks can be connected directly to the audio interface.

6.5 SPI interface swap jumpers

Due to a bug in early revisions of CC2430, the MISO and MOSI pins changed directions when switching between SPI slave and master modes. The default settings of the jumpers are SPI master mode, see Figure 5: SPI jumper. To use CC2430 in SPI slave mode, the jumpers must be mounted 90°. SPI slave mode is used by the packet sniffer.

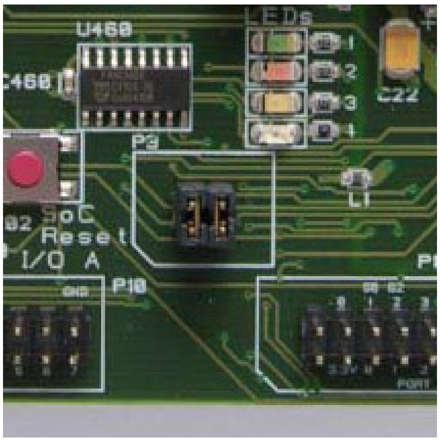

Figure 5: SPI jumper

6.6 I/O connectors

The I/O connectors bring out all the signals from the EM connectors. These connectors are compatible with Agilent logic analyser probes. The connectors allow easy access to I/O signals and to connect prototyping boards.

Pin	Function
1	N/C
2	N/C
3	P0_0/MIC_IN
4	VDD
5	VDD
6	N/C
7	P0_1/BUTTON_PUSH
8	N/C
9	P0_2/UART_RD
10	N/C
11	P0_3/UART_TD
12	N/C
13	P0_4/RTS
14	N/C
15	P0_5/JOY_PUSH
16	N/C
17	P0_6/JOY
18	N/C
19	P0_7/POT
20	GND

Table 1: I/O connector A (P10) pin-out

Pin	Function
1	N/C
2	N/C
3	VDD
4	P2_0*/LED4
5	P1_0/LED1
6	P2_1/DD
7	P1_1/PWM_OUTPUT
8	P2_0/DC
9	P1_2*/LED2
10	P2_3*/SDA
11	P1_3/LED3
12	P2_4*/SCL
13	P1_4/CSn
14	N/C
15	P1_5/SCLK
16	RESET_N
17	P1_6/MOSI
18	Debug Data Direction(DD_DIR)
19	P1_7/MISO
20	GND

Table 2: I/O connector B (P11) pin-out, * see chapter 6.9.

6.7 EM connectors

The EM connectors are used for plugging the EM to the CC2430ZDK. The connectors P1 and P2 are used as the main interface.

The EM includes RF reference design with the Chipcon IC and all required RF circuitry. It is recommended to copy the RF reference design when designing applications with Chipcon devices in order to achieve best RF performance.

Note that while it is possible to plug in EM not belonging to the SmartRF®04 product range into the CC2430ZDK the factory firmware and SmartRF® Studio do not support the use of older devices. This means that CC2430ZDK can be used with custom firmware for prototype using the CC2430ZDK together with older Chipcon RF devices.

6.8 Signal flow

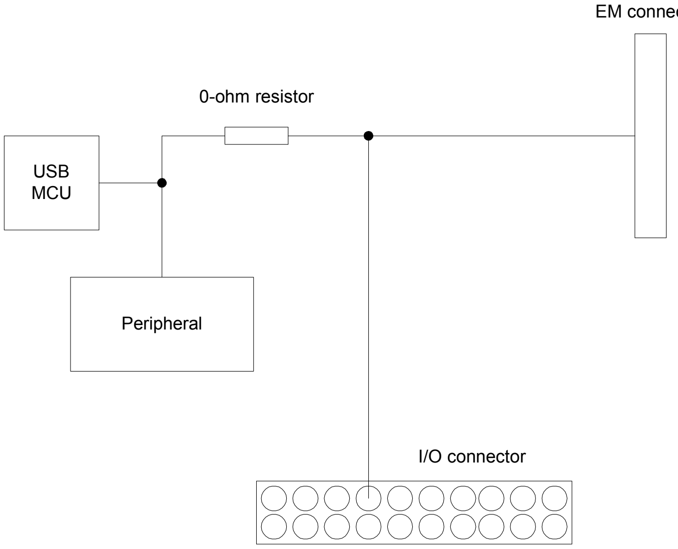

Figure 6: Evaluation Board Signal flow

The signal lines from the EM connectors run via 0-ohm resistors to the USB MCU and the various peripherals on the SmartRF04. This allows connecting an EM module to other applications. The USB MCU can be disconnected from the signal pins by removing the 0-ohm resistors. The I/O connectors are located on the “outside” of the 0-ohm resistors, so they are still connected to the EM connectors even if the 0-ohm resistors are removed. Please refer to Table 11 on page 29 for a list of what resistors correspond to which signals.

6.9 CC2430EM

On CC2430EM port P2_3 and 2_4 is connected to a 32kHz X-oscillator. The LCD display is therefore internally, on CC2430EM, swapped from P1_2 to P2_3 and P2_0 to P2_4.

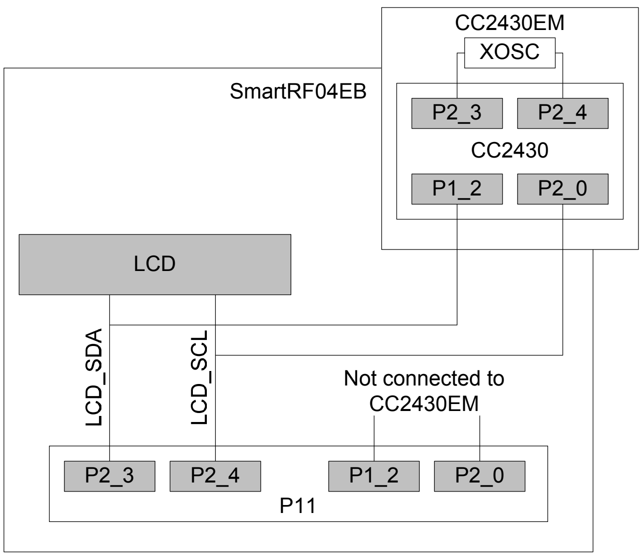

Figure 7: CC2430EM LCD connection

7 Hardware description CC2430DB

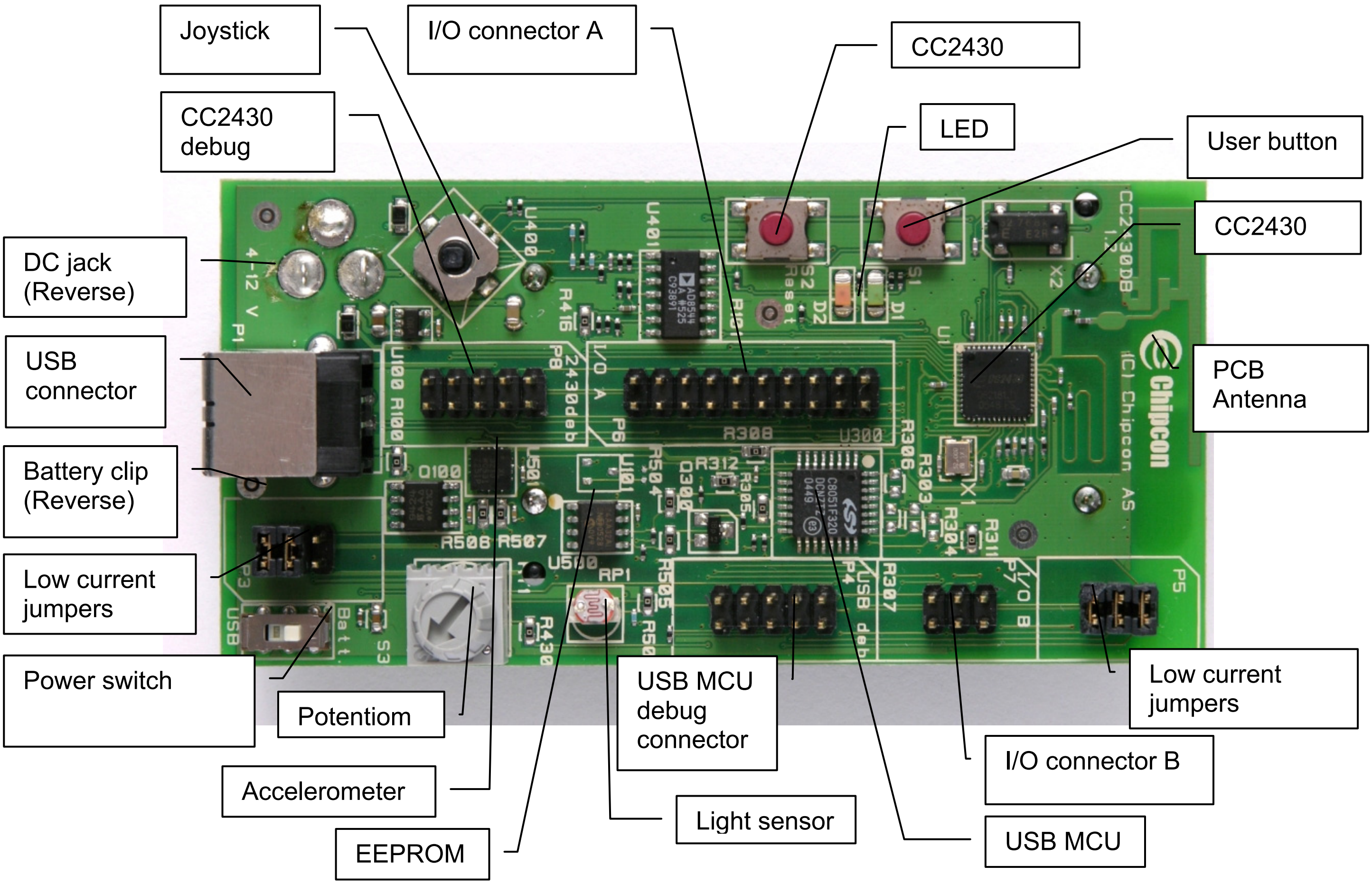

Figure 8: CC2430DB overview

7.1 USB Interface

The USB interface is used to interface to a PC to run the In-Circuit Emulator in the IAR Embedded Workbench and using SmartRF® Studio. It is also possible to reprogram the CC2430DB via this interface with the Chipcon programming software, and use the Chipcon packet sniffer. When connected to the USB port the CC2430DB is powered from the USB and no other voltage supplies are required.

7.2 User interface

The CC2430DB includes a joystick and a push button as user input devices, and two LEDs as user output devices.

7.3 EEPROM

A 32Kbit EEPROM is included for non volatile storage of data that is frequently updated. The EEPROM have 1 Million guaranteed write cycles.

7.4 Accelerometer

The accelerometer can be used measure movements in 2-axis, it can also be used for tilt measurement by measuring the earths gravitation. Accelerometer self test can be initiated by shorting pin 5 and 6 on connector P3. The accelerometer has a 20 ms start-up time after power on. See the Analog Devices ADXL321 datasheet for details about the accelerometer.

7.5 Potentiometer

The potentiometer resistance depends on the rotation. Typical value is 10K.

7.6 Light sensor

A light dependent resistor (LDR) measures light level and gives an analogue signal that is measured by the CC2430 A/D converter. The light sensor resistance ranges from 5KΩ (light) to 20MΩ (dark).

7.7 I/O connectors

The I/O connectors bring out all the signals from CC2430. These connectors allow easy access to all CC2430 I/O pins. For prototyping external circuitry can be connected using these connectors.

Pin	Function
1	VDD
2	VDD_SW_CONTROLLED[1]
3	P0_0/LDR
4	RESET_N
5	P0_1/BUTTON PUSH
6	P1_7/SO/MISO/UART_RD
7	P0_2/EE_SDA
8	P1_6/SI/MOSI/UART_TD
9	P0_3/EE_SCL
10	P1_5/SCLK/RTS
11	P2_1/DD
12	P1_4/CSN/SS/CTS
13	P2_2/DC
14	P2_0/JOY PUSH
15	P0_6/JOY
16	P1_2/VDD_SW_CTRL
17	P0_7/POT
18	P1_1/LED2
19	P1_0/LED1
20	GND

Table 3: I/O connector A (P6) pin-out

Pin	Function
1	P1_3/GPIO
2	DC_JACK_PWR
3	P0_4/ACC_X
4	VDD
5	P0_5/ACC_Y
6	GND

Table 4: I/O connector B (P7) pin-out

7.8 Jumper settings

[1] VDD_SW_CONTROLLED is controlled by the signal named VDD_SW_CTRL

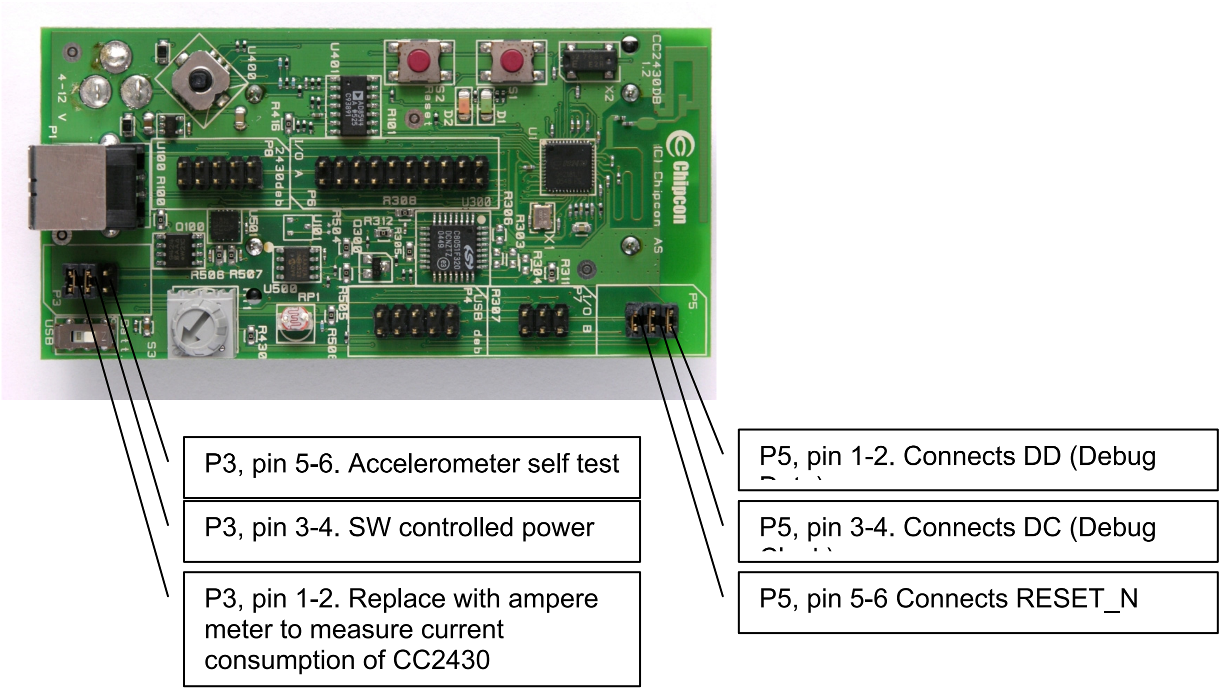

Figure 9: CC2430DB Jumper settings

Header	Pin	Description	Default setting
P3	1-2	Replace with ampere meter to measure current consumption of CC2430	Mounted
P3	3-4	SW controlled power. Can be removed when performing current measurement to reduce leakage current	Mounted
P3	5-6	Accelerometer self test. Mount to perform accelerometer self test function	Not mounted
P4	7-9	Manual reset of USB MCU	Not mounted
P4	9-10	Mount during power on to force USB MCU into boot loader	Not mounted
P5	1-2	Connects DD (Debug Data) between USB MCU and CC2430. Should be removed when connecting ICE to “SOC debug connector”	Mounted
P5	3-4	Connects DC (Debug Clock) between USB MCU and CC2430. Should be removed when connecting ICE to “SOC debug connector”	Mounted
P5	5-6	Connects RESET_N signal between USB MCU and CC2430. Can be removed when performing current measurement to reduce leakage current	Mounted

Table 5: Jumper settings summary

7.8.1 P3 Jumpers

The jumper between pin 1-2 on P3 can be replaced with an ampere meter to measure current consumption of CC2430. The jumper must be mounted for normal operation.

The jumper between pin 2-3 can be removed to when performing current measurement to reduce leakage current of the peripheral devices connected to the CC2430. The jumper must be mounted for normal operation

The jumper between pin 5-6 can be mounted to perform accelerometer self test function. By mounting this jumper, a voltage is applied to the accelerometer outputs. See the Analog

Devices ADXL321 datasheet for details about the accelerometer self test. The jumper must be removed for normal operation.

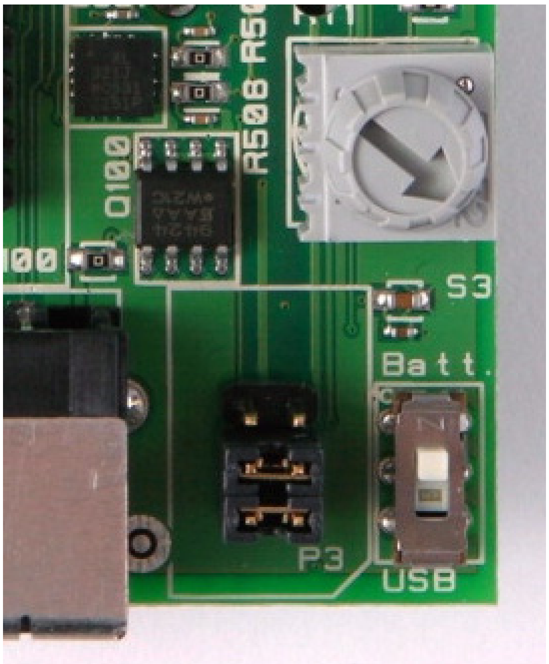

Figure 10: Default P3 jumper settings

7.8.2 P5 Jumpers

All jumpers on header P5 must be mounted for normal operation.

The jumper between pin 1-2 on P5 connects DD (Debug Data) between USB MCU and CC2430..

The jumper between pin 3-4 on P5 connects DC (Debug Clock) between USB MCU and CC2430DB..

When the CC2430DB is used with emulator connected to the USB port both the DD and DC jumpers must be mounted. They should only be removed if an external emulator is used. See chapter 9.1 for instructions how to connect external emulator to the CC2430DB.

The jumper between pin 5-6 on P5 connects RESET_N signal between USB MCU and CC2430. The jumper can be removed when performing current measurement to reduce leakage current for the circuits connected to the RESET pin. The jumper must be mounted for normal operation.

Figure 11: P5 default jumper settings

7.9 Signal flow

The signal lines from the I/O connectors run via 0Ω resistors to the CC2430. and the various peripherals on the ZigBee Development Kit Pro. This way peripheral can be disconnected from the CC2430 signal pins by removing the 0Ω resistors. Please refer to Table 8 for a list of what resistors correspond to which signals.

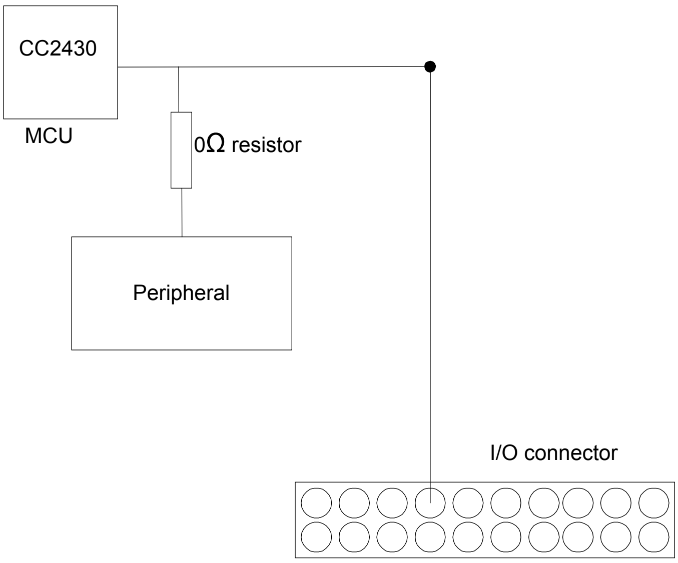

Figure 12: CC2430DB Signal flow

8 Using the CC2430ZDK for prototyping

SmartRF04EB includes a debug and programming interface. The debug interface is controlled by 2 communication pins. On the SmartRF04EB the interface is controlled by the USB MCU. This allows programming and emulator interface using the USB port. For developing applications, a set of example libraries are include with the SmartRF04EB.

Table 11 on page 29 shows what pins on the MCU are used for various peripheral functions. P10 and P11 pin-row connectors can be used to connect the SmartRF04EB to another PCB or prototyping board.

Note: The SmartRF04EB may not work with In-Circuit Emulator (ICE), SmartRF® Studio or the examples when the 0-ohm resistors are removed.

9 Using the CC2430DB for prototyping

The CC2430DB can be used for prototyping by programming the CC2430 with custom applications. All I/O ports on the CC2430 are available on pin row header at the edge of the board, and the USB interface can be used as In Circuit Emulator (ICE) interface allowing real time in circuit emulation of the CC2430.

Code examples and libraries are provided by Chipcon to speed up development with the CC2430DB and CC2430. Table 8 on page 22 shows what pins on the MCU are used for various functions.

P6 and P7 pin-row connectors can be used to connect the CC2430DB to other PCB or prototyping boards. See Table 3 on page 15 and Table 4 on page 15 for the pin-out of these connectors.

0Ω resistors are included to isolate the CC2430 from the external components on the CC2430DB. By removing these resistors, signals can be accessed on the pin headers.

9.1 Debugging using the USB interface

The most common way to use the CC2430DB for development is to use the USB interface to control the CC2430 on-chip In-Circuit Emulator. The USB interface supplies power to the board, so there is no need for additional DC power or batteries. The USB interface can also be used to program the CC2430 in circuit using the Chipcon programming software.

To use the USB interface, simply connect the USB cable to the board and start the IAR Embedded Workbench software supplied with the kit. Follow the user guide for the IAR tools for instructions how to compile and download the code.

9.2 Debugging with the CC2430 debug connector

It is possible to use the CC2430DB with the Chipcon packet sniffer or other application that requires USB interface for communication with PC and debugging with an emulator simultaneously. In these cases the CC2430 SOC debug connector can be used for connecting the ICE. The SmartRF04EB can be used as emulator interface with a cable from P14 “SOC debug flash” on SmartRF04EB to P8 “SOC debug connector” on CC2430DB. See Figure 13 below. The jumpers between pin 1-2, 3-4 and 4-5 on header P5 must be removed.

CC2430DB must be powered from battery, USB or DC jack when debugged from SmartRF04EB.

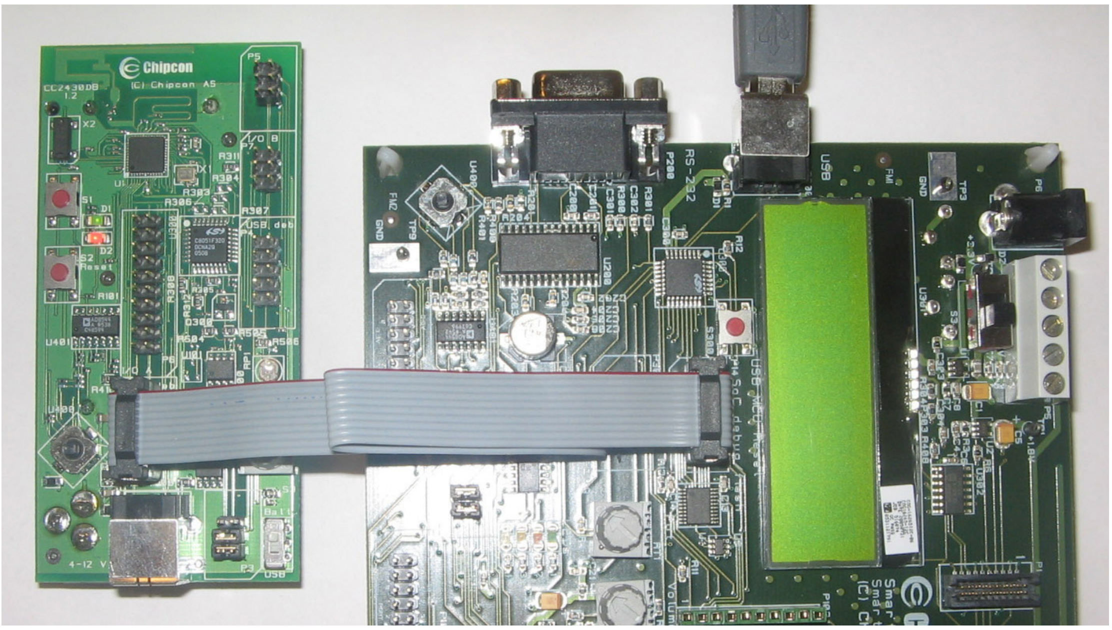

Figure 13: CC2430 Debug Connector

9.3 Low power operation

CC2430DB is designed for low power operation when running from batteries. Only the CC2430 is powered in this mode, the USB MCU is not powered. The voltage to the peripheral functions connected to the CC2430 is controlled by an I/O pin P1.2 on the CC2430. Table 6 lists the current consumption for each of the peripherals.

To obtain minimum current consumption with the CC2430DB, the P1.2 output must be configured as output.

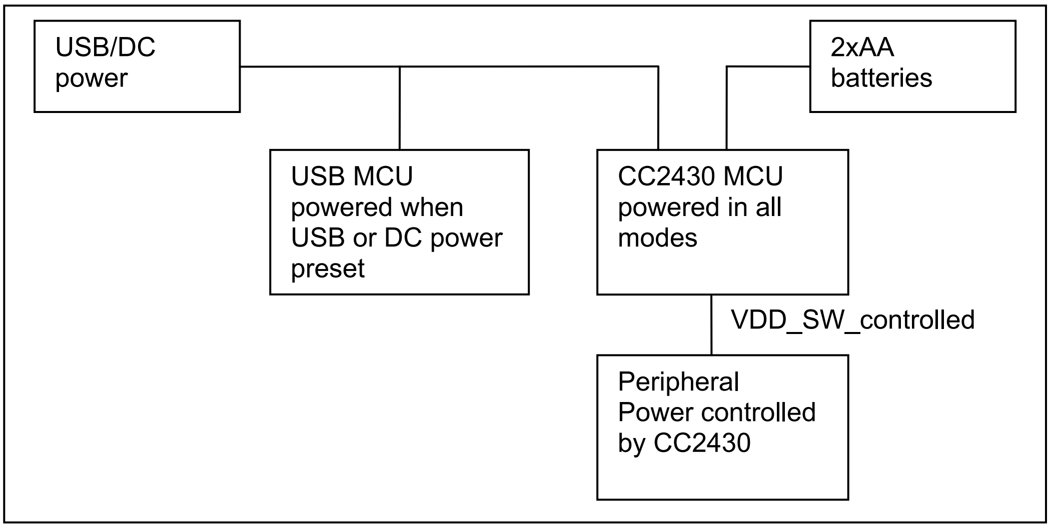

Figure 14: Voltage supply distribution

Table 6 below shows the current consumption for all modules and operation modes with 3.3V supply voltage

	Active mode(mA)		Power down mode(µA)	
	Typical	Max	32Khz	Sleep
CC2430	7,2	27	1µA	0.3µA
EEPROM	1	3	0	
Accelerometer	0,49		0	
Potentiometer	0,33		0	
Light sensor	0,001	0.015	0	
LEDs		24	0	
Total	**9mA**	**54mA**	**1µA**	

Table 6: Current consumption summary

To accurately measure the current consumption of the C the jumpers listed in Table 7 should be removed.

Header	Pin	Description	Low power setting
P3	1-2	Replace with ampere meter to measure current consumption of CC2430	Mounted, or connect ampere meter
P3	3-4	SW controlled power. Can be removed when performing current measurement to reduce leakage current	Not mounted
P5	5-6	Connects RESET_N signal between USB MCU and CC2430. Can be removed when performing current measurement to reduce leakage current	Not mounted

Table 7: Jumper setting for low power measurements

In order to achieve lowest possible current consumption, the I/O ports of the CC2430 should be configured as listed in Power down state in Table 8 below.

2430 pin	Name	Schematic Name	Description	Active state[2]	Power down state	0Ω ohm resistor
11	P0_0	P0_0/LDR	Light dependent resistor analog input	IN ANALOG	OUT LOW	R506
12	P0_1	P0_1/BUTTON PUSH	Push button interrupt active low	IN PULL-UP	IN	
13	P0_2	P0_2/EE_SDA	EEPROM SDA	OUT HIGH	OUT LOW	R505
14	P0_3	P0_3/EE_SCL	EEPROM SCL	OUT HIGH	OUT LOW	R504
15	P0_4	P0_4/ACC_X	Accelerometer x-axis	IN ANALOG	OUT LOW	R507
16	P0_5	P0_5/ACC_Y	Accelerometer y-axis	IN ANALOG	OUT LOW	R508
17	P0_6	P0_6/JOY	Joystick analog signal	IN ANALOG	OUT LOW	R416
18	P0_7	P0_7/POT	Potentiometer analog input	IN ANALOG	OUT LOW	R430

[2]Active state means that Software controlled VDD is on, this should only be enabled when required by an I/O module

9	P1_0	P1_0/LED1	Green LED	OUT HIGH	OUT HIGH	
8	P1_1	P1_1/LED2	Red LED	OUT HIGH	OUT HIGH	
6	P1_2	P1_2/VDD_SW_CTRL	Voltage control for I/O modules	OUT HIGH	OUT LOW	
5	P1_3	P1_3/GPIO	Free I/O for controlling external signal	OUT LOW	OUT LOW	
4	P1_4	P1_4/CSn/SS/CTS		OUT LOW	OUT LOW	R305
3	P1.5	P1_5/SCLK/RTS		OUT LOW	OUT LOW	R308
2	P1_6	P1_6/SI/MOSI/UART_TD		OUT LOW	OUT LOW	R306
1	P1_7	P1_7/SO/MISO/UART_RD		OUT LOW	OUT LOW	R307
48	P2_0	P2_0/Joy_push	Joystick push interrupt active high	IN	IN	
46	P2_1	P2_1/DD	Debug Data	IN		Jumper P5
45	P2_2	P2_2/DC	Debug Clock	IN		Jumper P5
10	RESE T	RESET_N	Reset			Jumper P5

Table 8: Pinout CC2430

The joystick output is coded as an analog voltage. This has been done in order to save the number of pins required on the MCU to interface with the joystick. The libraries contain functions to decode the ADC values and indicate in what direction the joystick is moved. The push function of the joystick is connected to a digital input pin.

9.4 Using the CC2430serial port to interface to RS232

The CC2430serial port is accessible on the header connectors as 3.3V signals. An RS232 driver circuit is required to connect to a RS232 port on a PC. The RS232 port on a SmartRF04EB can be connected to the CC2430DB to allow applications to interface to a PC serial port.

Table 9 shows the connection between the CC2430DB and the SmartRF04EB. Both the P8 (10-pin) connector and the P8 (20-pin) connector on the CC2430DB can be used, but only one of them should be connected.

To use the serial port on CC2430DBUSART1 must be used on alternative location 2 in the CC2430. Please see the CC2430datasheet for instructions how to select serial port and location. Using the SmartRF04EB will only work if the CC2430DB is powered from a USB or DC jack. The serial port signals will not be accessible if the CC2430DB is powered from batteries.

Signal Name	Smartrf04EB (P10)	CC2430ZDK (P8) ”2430 Debug”	CC2430ZDK (P6) ”I/O A”
RD	9	10	6

TD	11	8	8
RTS	13	6	10
CTS	15	5	12
GND	20	1	20

Table 9: CC2430ZDK to SmartRF04EB serial port connections

10 SmartRF04EB programming

The CC2430 can be programmed from the USB interface using the Chipcon programming software. For programming CC2430, the System-On-Chip tab should be selected, and a hex file can be programmed. Figure 15 shows the flash programming interface.

When designing applications with CC2430 it is recommended to include a pin header or test points to allow in-circuit emulation or programming using SmartRF04EB or other programming tools. The pinout used on the SmartRF04EB and the CC2430DB is explained in Table 10 and Figure 16.

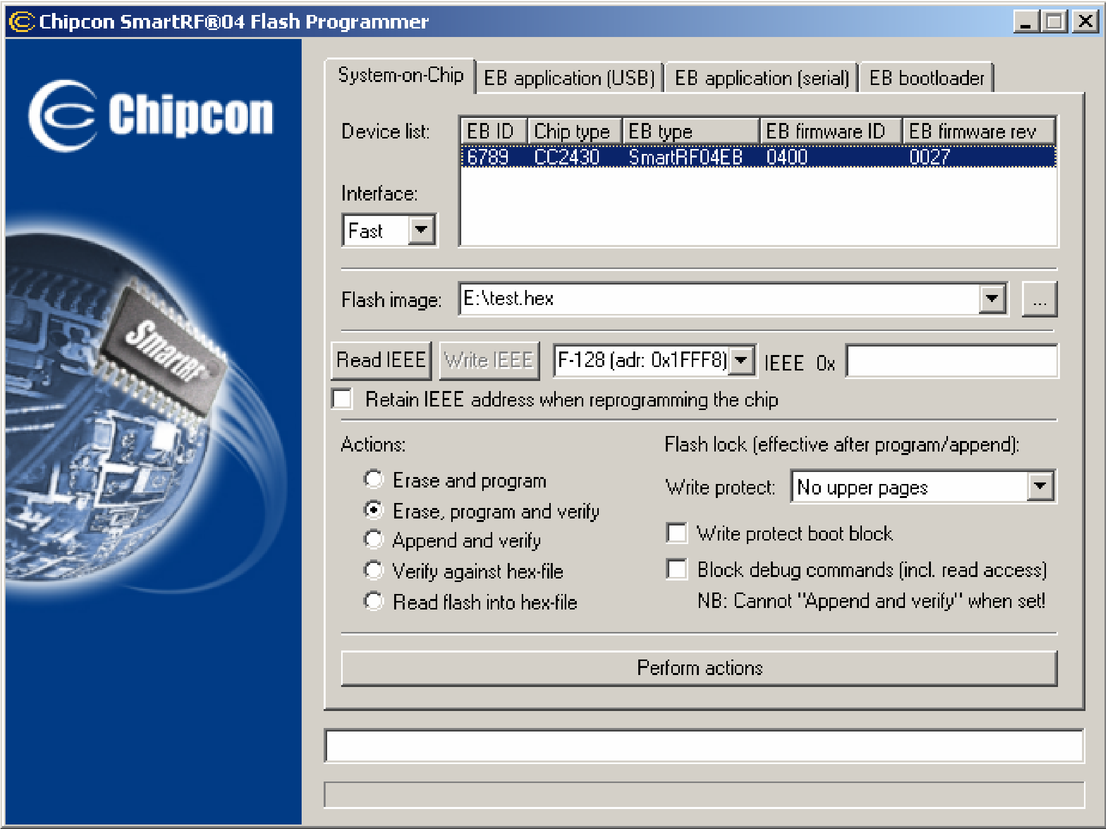

Figure 15: Chipcon Flash programmer software

11 Using SmartRF04EB as an In-Circuit Emulator (ICE)

The SmartRF04EB can be used as ICE both for devices connected to the EM socket and for external system with custom applications.

To use the SmartRF04EB as ICE, the IAR Embedded Workbench software must be installed. The Embedded Workbench is a C-Compiler, Simulator, and ICE debugger. See the IAR Embedded Workbench documentation supplied with the kit for instructions how to set up the ICE debugger for use as an emulator.

When the SmartRF04EB is connected to a PC with the USB port, the debugger will connect to the SmartRF04EB. Several SmartRF04EB can be connected to USB ports simultaneously. A selection window will display the connected SmartRF04EB, and the user can select which device to load.

When designing applications with CC2430 it is recommended to include a pin header or test points to allow in-circuit emulation or programming using SmartRF04EB or other 3rd party programming tools. Please see CC2430section of the Chipcon web site for an updated list of 3rd party programming tools

The pin-out used on the SmartRF04EB and the CC2430DB is explained in Table 10 and Figure 16. The connector includes 4 SPI control signals. These are currently not used, but they are included to give users flexibility.

Pin	**Function**
1	**Gnd**
2	**VDD**
3	**Debug Clock(DC)**
4	**Debug Data(DD)**
5	CSn(optional)
6	SCLK(optional)
7	**Reset_N**
8	MOSI(optional)
9	3.3V VDD, alt. NC
10	MISO

Table 10: P14 SOC debug connector pin-out

VDD note: The SmartRF04EB includes a voltage converter to support programming and debugging of external systems with different voltage than the SmartRF04EB. The debug connector includes two VDD connections on pin 2 and pin 9. The function is different for these connections.

Pin 2 VDD supplies voltage to the voltage converter. If the target application is self powered pin 2 should be connected to VDD to assure that the correct supply voltage is used for the voltage converter. This pin must always be connected to VDD.

Pin 9 VDD supplies 3.3V from the SmartRF04EB. If the target application is powered from the SmartRF04EB supply during programming and debugging this pin can be connected to VDD. If the target voltage differs from 3.3V, this pin should not be connected

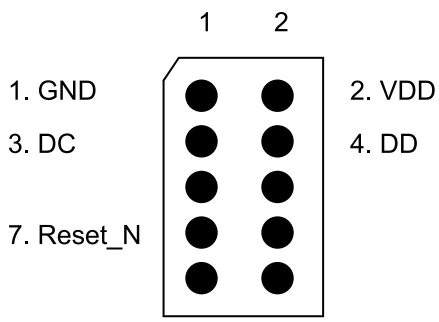

Figure 16: Recommended debug connector layout (Top view)

Figure 16 shows the required signal for a minimum connector layout.

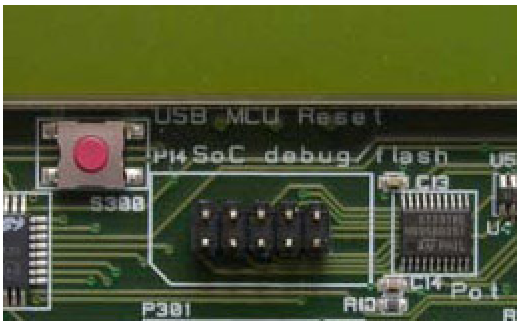

Figure 17: SmartRF04EB debug connector for debugging of external target systems

12 Firmware CC2430DB

When shipped, the CC2430DB is programmed with the USB bootloader and firmware that allows the ZigBee Development Kit Pro to communicate with SmartRF® Studio and run application examples including stand-alone packet error rate (PER) test.

Chipcon provides libraries that make it easy to control the various peripherals on the CC2430DB. Please see the Examples and Libraries document for more information about these libraries and application examples.

Please see Silicon Labs' datasheet for technical information about the USB MCU.

12.1 The USB MCU bootloader

The bootloader allows programming of new code into the USB MCU without the Silicon Labs' serial adapter. The bootloader communicates with SmartRF® Studio or a custom program via USB. The CC2430DB is programmed with the bootloader when it is shipped from the factory.

If it is required to update the firmware of the USB MCU, it can be forced to enter the bootloader code by mounting jumper between pin 9-10 on header connector P4 and apply power to the board. New firmware can then be programmed using the SmartRF04 programming software.

13 Programming the USB MCU

To download programs to the USB MCU without using the bootloader, a Silicon Labs' EC2 serial adapter is required. This adapter should be connected to P301 (marked "USB MCU debug" on the CC2430ZDK).

Figure 18: EC2 serial adapter

13.1 USB MCU signal names

The USB MCU is a C8051F320 from Silicon Labs. Please see the Silicon Labs web site for detailed information about this MCU.

The following table shows what pins on the USB MCU are used for what functions.

Pin no.	Pin name (USB MCU)	Signal name (on CC2430ZDK schematic)	0-ohm resistor	Function
1	P0.1	P1.7/SO/GDO1/MISO	R101	SPI MISO signal, transceiver/transmitter SO/GDO2
2	P0.0	P1.5/SCLK	R115	SPI Serial clock
10	P3.0/C2D			USB MCU Debug pin
11	P2.7	P1.3/LED3	R113	LED3 (yellow), active low
12	P2.6	P0.4/RTS	R100	
13	P2.5	RS232_POWER		Turns RS-232 level converter on/off
14	P2.4	P2.0/LED_4	R120	LED4 (Blue), active low
15	P2.3	RESET_N		LCD Power on reset signal, SoC RESET
16	P2.2	SOC_PRESENT		Tells USB MCU whether a SoC is present. (0=transmitter/transceiver, 1=SoC)
17	P2.1	P0.6/JOY	R106	Joystick input (analogue coded voltage)
18	P2.0	P1.1/LED2	R111	LED2 (Red), active low
19	P1.7	P1.0/LED1	R110	LED1, (Green), active low
20	P1.6	P0.7/POT	R107	Potentiometer input
21	P1.5	P0.5/JOY_PUSH	R112	Joystick pushed
22	P1.4	P1.2/PWM_OUTPUT	R105	PWM audio output
23	P1.3	P0.1/BUTTON_PUSH	R101	Button pushed
24	P1.2	P0.0/MIC_INPUT	R104	Audio input
25	P1.1	P2.4/SCL	R124	I2S clock (for LCD)
26	P1.0	P2.3/SDA	R123	I2S data (for LCD)
27	P0.7	P2.2/GDO2/DC	R122	Transceiver/transmitter GDO3, SoC debug signal
28	P0.6/CNVSTR	P2.1/GDO1/DD	R121	Transceiver/transmitter GDO1, SoC debug signal
29	P0.5	P0.2/UART_RD	R102	UART RD
30	P0.4	P0.3/UART_TD	R103	UART TD
31	P0.3/XTAL2	P1.4/CS/SS	R114	SPI slave select signal
32	P0.2/XTAL1	P1.6/MOSI	R116	SPI MOSI signal, Transceiver/Transmitter SI

Table 11: USB MCU pin-out

As mentioned in the table, the joystick output is coded as an analogue voltage. This has been done in order to save the number of pins required on the MCU to interface with the joystick. The libraries contain functions to decode the ADC values and indicate in what direction the joystick is moved. The push function of the joystick is treated as a separate digital signal.

14 Schematics

See SmartRF04DK User Manual and CC2430DB Reference Design for schematics.

15 Document History

Revision	Date	Description/Changes
1.2	2006-02-16	Changed layout, Changed P0.4/CTS to P0.4/RTS in table “USB MCU pin-out”
1.1.1	2006-01-17	Changed picture CC2430 Debug Connector
1.1	2006-01-09	Added picture CC2430 Debug Connector
1.0	2005-12-16	Initial release