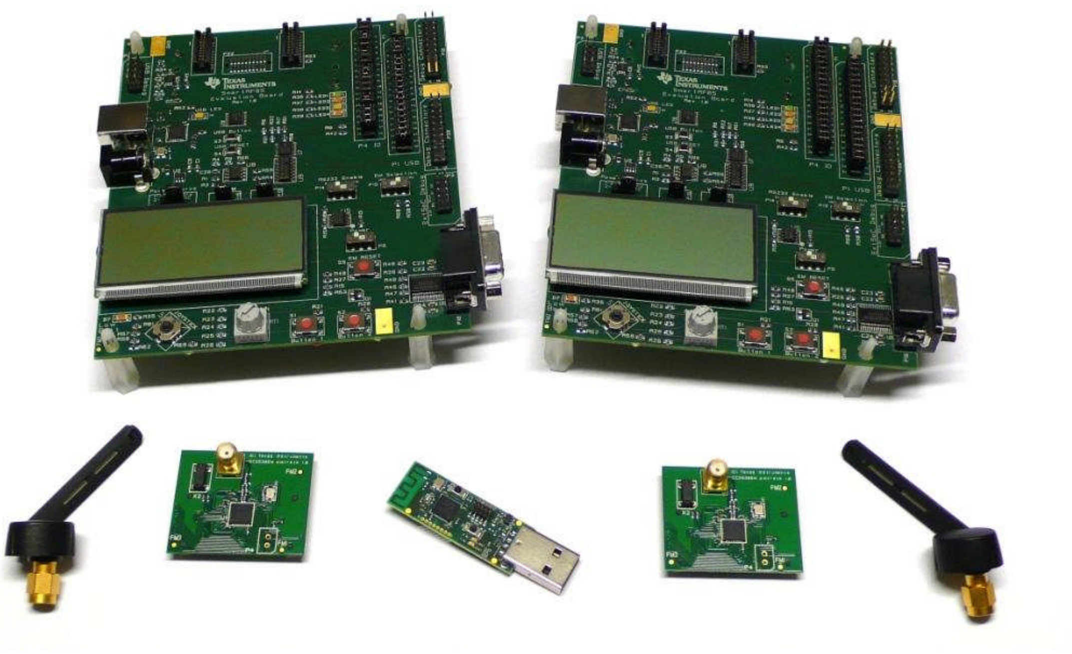
CC2530 Development Kit User's Guide



swru208a

Table of contents

CC2530 DEVELOPMENT KIT USER'S GUIDE **1**
1 INTRODUCTION **3**
2 ABOUT THIS MANUAL **3**
3 ACRONYMS **4**
4 DEVELOPMENT KIT CONTENTS **5**
5 GETTING STARTED **7**
5.1 SETTING UP THE HARDWARE 7
5.2 RUNNING THE PREPROGRAMMED PER TEST ON THE CC2530EM 8
5.3 EVALUATE THE CC2530 RADIO USING SMARTRF STUDIO 9
5.4 SETTING UP THE SOFTWARE DEVELOPMENT ENVIRONMENT 11
6 RF TESTING **12**
6.1 TX PARAMETER TESTING BASICS 12
6.2 RX PARAMETER TESTING BASICS 13
7 CC2530EM **14**
8 CC2531 USB DONGLE **15**
9 SMARTRF05 EVALUATION BOARD **17**
10 FREQUENTLY ASKED QUESTIONS **18**
11 REFERENCES **20**
12 DOCUMENT HISTORY **20**
APPENDIX A SETTING UP THE SOFTWARE ENVIRONMENT **21**
A.1 CREATE THE PROJECT 21
A.2 PROJECT OPTIONS 22
A.3 SELECT DEVICE 22
A.4 SELECT CODE AND MEMORY MODEL 23
A.5 CONFIGURE THE LINKER 25
A.6 CONFIGURE THE DEBUGGER 26
A.7 WRITE SOFTWARE 27
A.8 COMPILE AND DEBUG 28
A.9 DONE! 28
APPENDIX B SOFTWARE SOLUTIONS FOR CC2530 FROM TI **29**
B.1 SIMPLICITI™ NETWORK PROTOCOL 29
B.2 TIMAC SOFTWARE 29
B.3 REMOTI™ NETWORK PROTOCOL 30
B.4 Z-STACK™ SOFTWARE 30
APPENDIX C SCHEMATICS **32**

1 Introduction

Thank you for purchasing a CC2530 Development Kit.

The CC2530 is Texas Instrument's second generation ZigBee/IEEE 802.15.4 compliant System-on-Chip with an optimized 8051 MCU core and radio for the 2.4 GHz unlicensed ISM/SRD band. This device enables industrial grade applications by offering state-of-the-art noise immunity, excellent link budget, operation up to 125 degrees and low voltage operation.

In addition, the CC2530 provides extensive hardware support for packet handling, data buffering, burst transmissions, data encryption, data authentication, clear channel assessment, link quality indication and packet timing information.

The CC2530 product folder on the web [1] has more information, with datasheets, user guides and application notes.

The CC2530 Development Kit includes all the necessary hardware to properly evaluate, demonstrate, prototype and develop software targeting not only IEEE802.15.4 or ZigBee compliant applications, but also proprietary applications for which a DSSS radio is required or wanted.

2 About this manual

This manual describes all the hardware included in the CC2530 Development Kit (CC2530DK) and points the user to other useful information sources.

Chapter 4 briefly describes the contents of the development kit and chapter 5 gives a quick introduction to how to get started with the kit. In particular, it describes how to install SmartRF Studio to get the required drivers for the evaluation board, how the hardware can be used, and lists the software that is available for the development kit. Chapter 6 explains some simple methods for performing practical RF testing with the development kit. Chapter 7, 8, and 9 describe the hardware in the kit and where to find more information about how to use it. A troubleshooting guide can be found in chapter 10.

Appendix A gives a detailed description of how to set up the software development environment for the CC2530. Appendix B lists available software solutions for CC2530.

The CC2530DK Quick Start Guide [4] has a short tutorial on how to get started with the kit. The CC2530 Software User's Guide [5] provides details about the software examples and information about other software options for the CC2530.

The PC tools SmartRF Studio and SmartRF Flash Programmer have their own user manuals.

Please visit the CC2530 development kit web page [3] and CC2530 product page [1] for additional information. Further information can be found on the TI LPRF Online Community [7].

See chapter 11 for a list of relevant documents and links.

3 Acronyms

ACM	Abstract Control Model
ADC	Analog to Digital Converter
CDC	Communications Device Class
DK	Development Kit
EB	Evaluation Board
EM	Evaluation Module
HID	Human Interface Device
IC	Integrated Circuit
ISM	Industrial, Scientific and Medical
KB	Kilo Byte (1024 byte)
LCD	Liquid Crystal Display
LED	Light Emitting Diode
LPRF	Low Power RF
MCU	Micro Controller
NC	Not connected
PER	Packet Error Rate
RF	Radio Frequency
RX	Receive
SoC	System on Chip
SPI	Serial Peripheral Interface
SRD	Short Range Device
TI	Texas Instruments
TX	Transmit
UART	Universal Asynchronous Receive Transmit
USB	Universal Serial Bus

4 Development Kit contents

The CC2530 Development Kit (CC2530DK) includes hardware and software that allows quick testing of the CC2530 RF performance and offers a complete platform for development of advanced prototype RF systems.

- Evaluate the CC2530 right out of the box. The kit can be used for range testing using the pre-programmed PER tester running on the CC2530.
- Use SmartRF Studio to perform RF measurements. The radio can be easily configured to measure sensitivity, output power and other RF parameters.
- Prototype development. All I/O pins from the CC2530 are available on pin connectors on the SmartRF05EB, allowing easy interconnection to peripherals on the EB board or other external sensors and devices.

The CC2530DK contains the following components

- 2 x SmartRF05EB (the two large boards)
- 2 x CC2530 Evaluation Modules (the two small boards)
- 2 x Antennas
- 1 x CC2531 USB Dongle
- Cables
- Batteries
- Documents

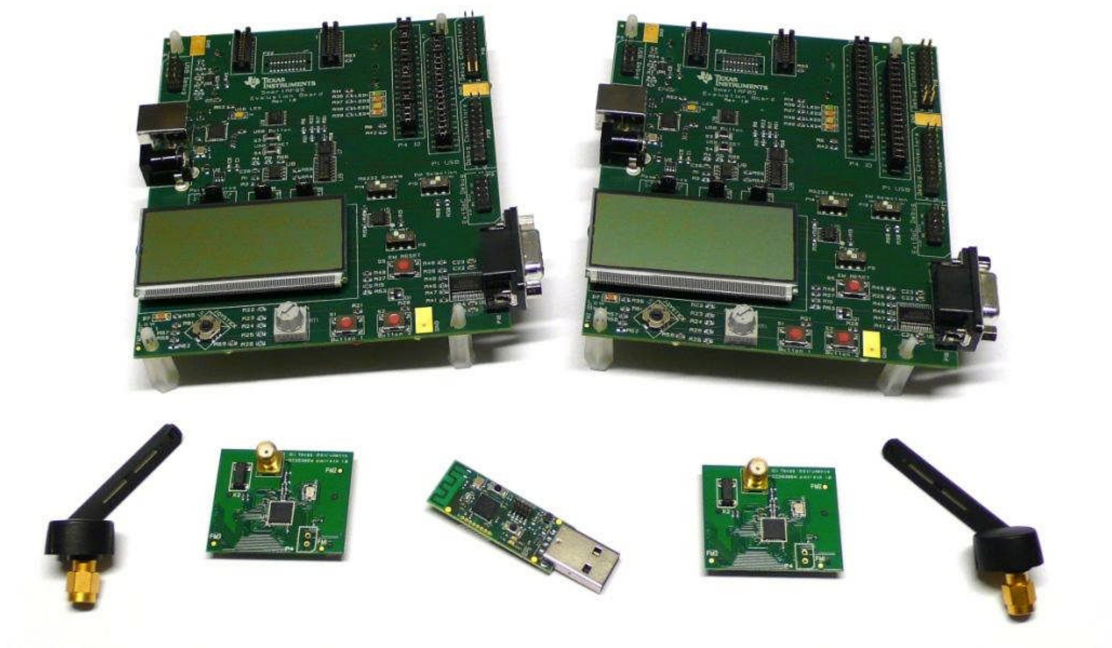


Figure 1 - CC2530 Development Kit Contents

SmartRF05EB

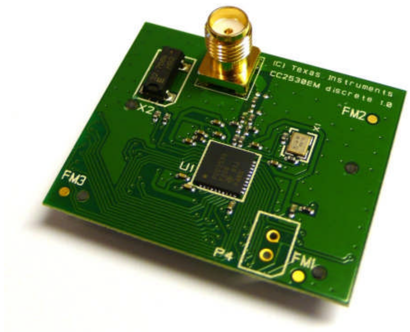
The SmartRF05EB (evaluation board) is the main board in the kit with a wide range of user interfaces:

- 3x16 character serial LCD
- Full speed USB 2.0 interface
- UART
- LEDs
- Serial Flash
- Potentiometer
- Joystick
- Buttons

The EB is the platform for the evaluation modules (EM) and can be connected to the PC via USB to control the EM.

CC2530EM

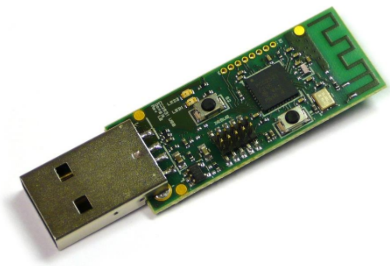
The CC2530EM (evaluation module) contains the RF IC and necessary external components and matching filters for getting the most out of the radio. The module can be plugged into the SmartRF05EB. **Use the EM as reference design for RF layout.** The schematics are included at the end of this document and the layout files can be found on the web [1].



CC2531 USB Dongle

The CC2531 USB Dongle is a fully operational USB device that can be plugged into a PC. The dongle has 2 LEDs, two small push-buttons and connector holes that allow connection of external sensors or devices. The dongle also has a connector for programming and debugging of the CC2531 USB controller.

The dongle comes preprogrammed with firmware such that it can be used as a packet sniffer device.



Antenna

2.4 GHz antenna Titanis from Antenova.

5 Getting started

5.1 Setting up the hardware

After opening the kit, make sure you have all components. Please contact your TI Sales Representative or TI Support [6] if anything is missing.

Start by connecting the antennas to the SMA connector on the RF evaluation boards. Tighten the antenna's screw firmly on to the SMA connector. If not properly connected, you might see reduced RF performance. It is also possible to connect the EM board to RF instruments via coax cables. The EM is designed to match a 50 Ohm load at the SMA connector.

Next, the evaluation modules should be plugged in to the SmartRF05EB. The purpose of the SmartRF05EB is to serve as a general I/O board for testing of the various peripherals of the CC2530 microcontroller. The EB also contains a separate USB controller, which is used as a bridge between the PC and the CC2530 for programming the flash of the CC2530. It is also used for debugging the software running on the CC2530.

The evaluation board can be powered from several different sources:

- 2 x 1.5V AA batteries (included in this kit)
- USB (via the USB connector)
- DC power (4 to 10 Volt) (not included in this kit)
- External regulated power source (not included in this kit)

The power source can be selected using jumper P11 on the SmartRF05EB. The SmartRF05EB User's Guide [8] provides more details.

After assembling the hardware, you now have several options for working with the CC2530:

- **Run the packet error rate (PER) test which is preprogrammed on the CC2530.** The PER test is a quick way to evaluate the range which can be achieved with the radio. Chapter 5.2 will guide you through the PER test.

- **Evaluate and explore the RF capabilities of the CC2530 using SmartRF Studio.** Chapter 5.3 provides the details how to do so.

- **Developing software for the CC2530.** Install IAR Embedded Workbench for 8051 and set up your first software project. Chapter 5.4 explains how.

5.2 Running the Preprogrammed PER Test on the CC2530EM

The CC2530EM comes pre-programmed with a Packet Error Rate (PER) test application. The PER number is the ratio between number of packets being lost and the total number of packets being sent. The PER relates to the more traditional Bit Error Rate (BER) through the formula

$$PER = 1 - (1 - BER)^{packet\_length}$$

A PER value of 1% (when the packet length is 20 bytes) is normally used as the limit for determining the sensitivity threshold of the radio. The sensitivity threshold is the lowest input signal strength at which the receiver can decode the signal with a reasonable degree of correctness.

By using the PER test on the CC2530, it is possible to perform practical range testing. Place the transmitter at a fixed location and place the receiver at a given distance from the transmitter. Then run the PER test to measure packet errors and monitor the signal strength. Read the description below for an explanation how the PER and RSSI values are calculated. Repeat at different distances to get an idea of the range that can be obtained.

To get an idea of the best performance of the device, the test should be performed outdoors on a large field with no other RF sources to avoid fading, reflections, and uncontrolled interference. Alternatively, the range test can be used to see what range is obtainable in the actual environment where the RF system is going to be deployed. See document [15] for considerations and applicable theory for performing open field range measurements.

The CC2530DK Quick Start Guide (www.ti.com/lit/swra273) gives a detailed step-by-step guide for running the PER test. We recommend following the steps in that guide.

Please note the following:

- The most natural power source to use for range testing is batteries. There is a voltage regulator on the SmartRF05EB that regulates the voltage to 3.3V on the board, regardless of the voltage from the batteries. If the low batteries LED (LED D7 below the LCD) on the EB board is turned on, the batteries should be changed.
- Both boards have to be set up to operate on the same channel. The channel is one of the 16 IEEE802.15.4 channels. The first channel (channel number 11, per the IEEE specification) is at 2405 MHz, followed by channels in steps of 5 MHz up to 2480 MHz.
- For the best range, use the highest possible output power on the transmitter.
- The PER value is calculated using the following formula:

  $$PER = \frac{NumPacketsLost + NumPacketsErrors}{NumPacketsOK + NumPacketsLost + NumPacketsErrors}$$

  The software is looking at the sequence number of the received packet to determine how many packets are lost since the last received packet. The PER value on the LCD shows the number per 1000 to avoid time consuming floating point calculations on the controller. That is, if the LCD shows a PER of 6 / 1000, the PER value is 0.6%.
- The RSSI value shown on the LCD is in dBm and represents the average RSSI value from the last 32 received packets. The RSSI value will never be the same for all packets even though the boards are located at the same distance from each other. This is caused by interfering signals, reflections, thermal noise etc.

The source code for the PER test, and a Intel HEX file ready to be programmed on the device, is included in the CC2530 Software Examples, available on the CC2530DK web site [3].

5.3 Evaluate the CC2530 Radio using SmartRF Studio

SmartRF Studio is a PC application developed for configuration and evaluation of many of the RF-IC products from Texas Instruments, including the CC2530. The application communicates with the CC2530 via the USB controller (the CC2511) on the SmartRF05EB board. The USB controller uses the debug interface of the CC2530 to execute commands and to read and write registers.

SmartRF Studio lets you explore the radio on the CC2530, as it gives you full overview and access to the radio registers. The tool has a control interface for running basic radio performance tests from the PC. SmartRF Studio also offers a flexible code export function of radio register settings for software developers.

Before proceeding, please download and install the latest version of SmartRF Studio from the web [9]. By installing Studio, the USB drivers needed for proper interaction between the PC and the hardware of the CC2530DK will also be installed.

In order to use the SmartRF Studio with CC2530, connect the CC2530EM to the SmartRF05EB. Next, connect the SmartRF05EB board to the PC via one of the USB cables included in the kit. If you have installed SmartRF Studio, select automatic installation of driver in the device wizard that appears. The device wizard will only pop up when you turn on the SmartRF05EB and only once for each board. Allow Windows to complete the driver installation before proceeding.

With the board connected to the PC, you can start SmartRF Studio. The following window should appear:

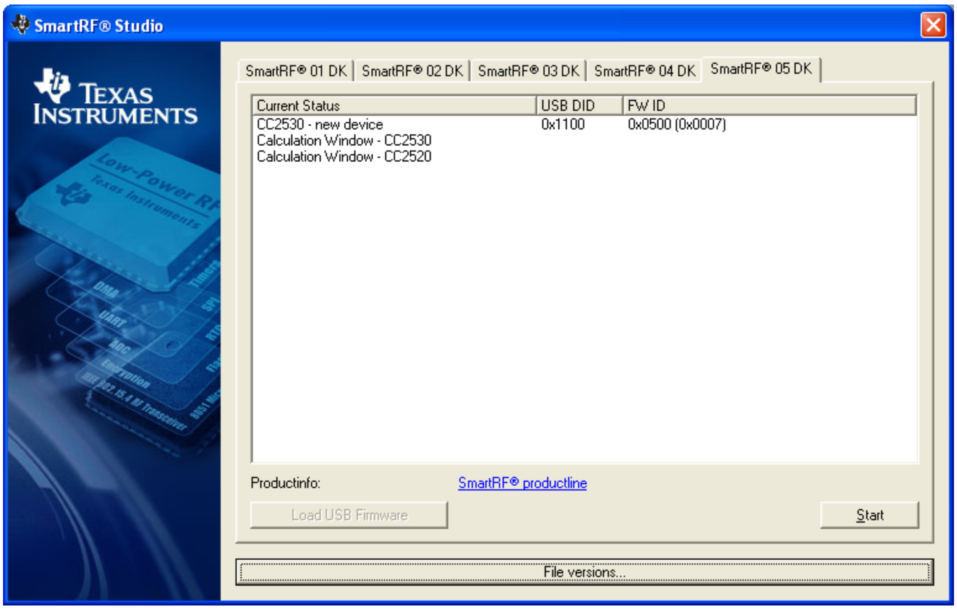


Figure 2 - CC2530 and SmartRF Studio

Make sure you select the tab called SmartRF® 05 DK. The connected evaluation board should be listed, showing that a CC2530 is available. Actually, all connected SmartRF05EB boards will be listed in this window. The list is dynamically updated as you connect or disconnect a board. Double click on the item showing “CC2530 – new device”, and a new window will appear.

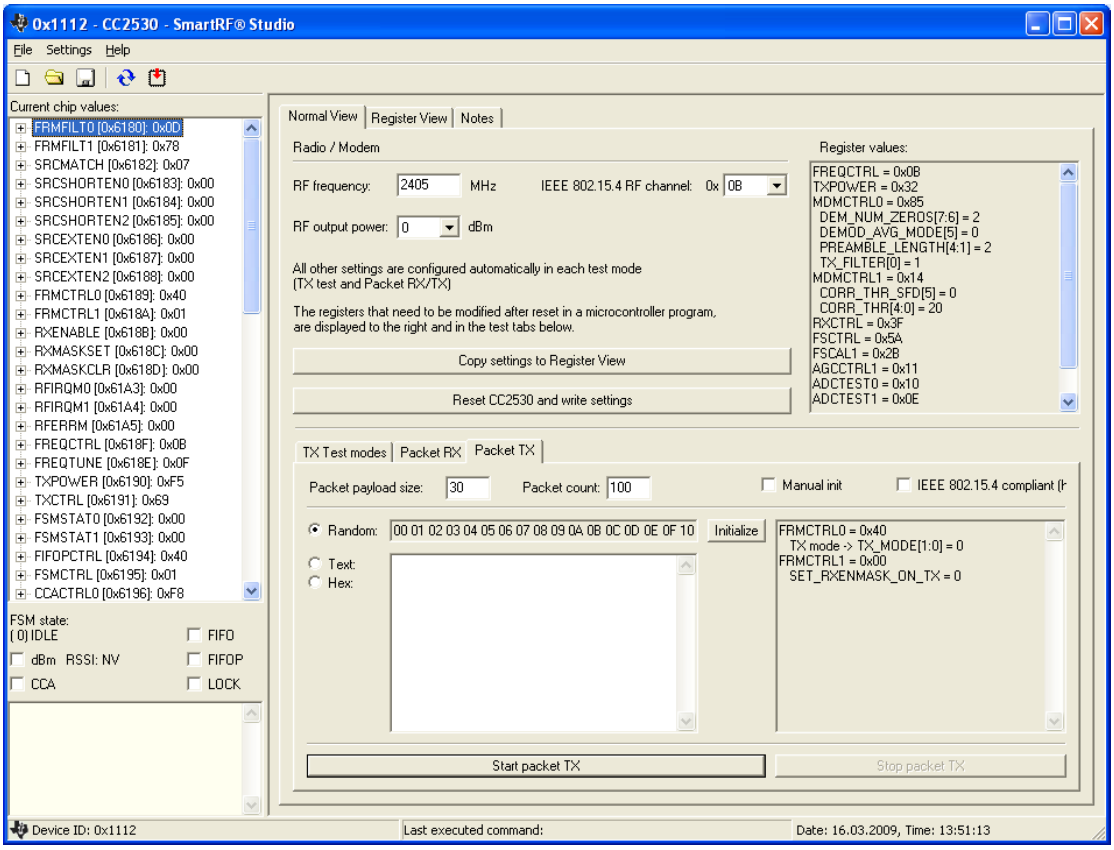


Figure 3 - CC2530 control panel in SmartRF Studio

Figure 3 shows the main control panel for the CC2530. It lets you perform a number of operations:

- Run TX Test modes for testing of RF output power and spectrum; e.g. by connecting a spectrum analyser or a power meter to the CC2530EM SMA connector to perform RF measurements.
- Run Packet TX and RX tests. For this test, you should have two EBs with CC2530EMs connected to the PC.
    - o Double click on both of the devices in the device list in SmartRF Studio (Figure 2), opening two windows, giving control of the two radios at the same time.
    - o Select one device to be the transmitter, by selecting the “Packet TX” tab shown in the lower middle of Figure 3.
    - o On the other device (the receiver), select the “Packet RX” tab.
    - o Set up basic test parameters and press the “Start packet RX” button.
    - o Now you can start transmission by pressing the “Start packet TX” button for the transmitter.
    - o The status window, in the lower right corner of the device control panel, will show the number of packets sent on the transmitter side and the number of received packets and signal strength of the last received packet on the receiver side.
- Read and/or modify registers and common settings, such as RF frequency (or channel) and output power.
- Export RF register values in a user modifiable format by selecting “File → Export CC2530 Code… ”.

The SmartRF Studio User Manual has more details.

5.4 Setting up the Software Development Environment

To develop software and debug an application on the CC2530, it is recommended to use IAR Embedded Workbench. It supports debugging of CC2530 through the SmartRF05EB, so no additional hardware is required.

IAR EW8051 is bundled with all the required files for CC2530 to start development:

- Register definition header file
- Linker command file
- Driver and device description file needed for debugging and programming

Note that other compilers and linkers can be used, but these tools may have limited debugging capabilities.

An evaluation version of IAR Embedded Workbench is included in the Development Kit. To install the software, insert the CD and follow the instructions. You will be asked to register on IAR's web site to get a license key for the product. As the owner of a CC2530 Development Kit, you are entitled to a 60 day evaluation period. The evaluation version in the kit automatically gives you 30 days. Please contact your local IAR sales representative for the additional 30-days evaluation period. For a list of sales offices and distributors in your country, please see this site: http://www.iar.com/contact.

The CC2530 Software Examples User's Guide [5] will take you through the initial steps of starting up IAR, setting up the project and compile and debug the software. Full source code for the software examples can be downloaded from the CC2530DK web page [3].

Appendix A in this document will guide you through the steps of setting up your own project from scratch.

Appendix B gives a brief overview of complete software solutions for CC2530 from Texas Instruments.
.

6 RF Testing

NB! When running RF performance tests, it is recommended to disable all other peripherals on the SmartRF05EB in order to avoid unwanted noise on the on-board voltage. In particular, make sure the RS232 level converter/line driver is disabled.

RF testing can be performed by using SmartRF Studio together with the Development Kit. The basic set-up is described in section 5.3. As described in that chapter, SmartRF Studio can be used to set up basic tests and tune RF registers accordingly.

Since the CC2530 evaluation board is equipped with an SMA connector, both radiated (via antenna) and conducted (via cable) tests can be performed, and it is easy to hook the EM up to RF measurement equipment. The RF equipment may be connected in two different ways.

- To measure radiated performance, connect an appropriate antenna to the spectrum analyzer or power meter and an antenna on the EM board.
- To measure conducted performance, connect a 50 Ohm coaxial cable directly from the EM to the RF equipment.

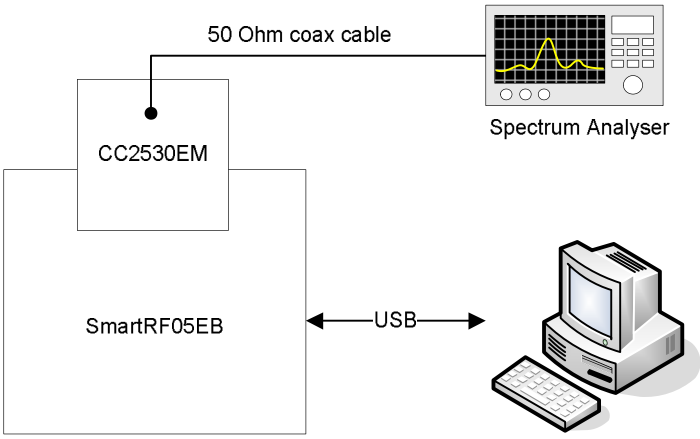


Figure 4 - RF Test Set-Up with a Spectrum analyzer

By using good-quality RF cabling, the loss in the cabling should be negligible. However make sure that the spectrum analyzer is calibrated. If possible, check it against a calibrated instrument such as an RF signal generator. Uncalibrated spectrum analyzers can display errors of several dBs.

6.1 TX Parameter Testing Basics

To investigate the TX performance of the CC2530, you can either use a Spectrum Analyzer or an RF Power Meter. Use the “Simple TX” test mode in SmartRF Studio to set up the device to transmit a signal at the desired frequency. Both a modulated or unmodulated carrier signal can be generated.

Use the RF Power Meter to observe the output power or the spectrum analyzer to observe the spectrum and to measure the error vector magnitude (EVM).

6.2 RX Parameter Testing Basics

To investigate the RX performance of the CC2530, you can use a signal generator or “Packet TX” in SmartRF Studio (with another EB+EM) to generate the packets to receive. The receiver can be configured by using the “Packet RX” test feature in SmartRF Studio.

By adding a jammer (a third node that generates either noise on the same channel or a strong signal on an adjacent channel) it is also possible to measure co-channel rejection and selectivity/blocking performance.

The PER test application, that was described in section 5.2, can be used for simple sensitivity measurements with the CC2530EM and/or with your own prototype hardware. In this case, connect the unit you want to test to a known good transmitter with coaxial cables and attenuators. Add more attenuators until the PER value is 1%. The signal strength at the receiver side is then the sensitivity limit of the system.

For more information regarding sensitivity measurements, refer to “Design Note 2 – Practical Sensitivity Testing” [14].

7 CC2530EM

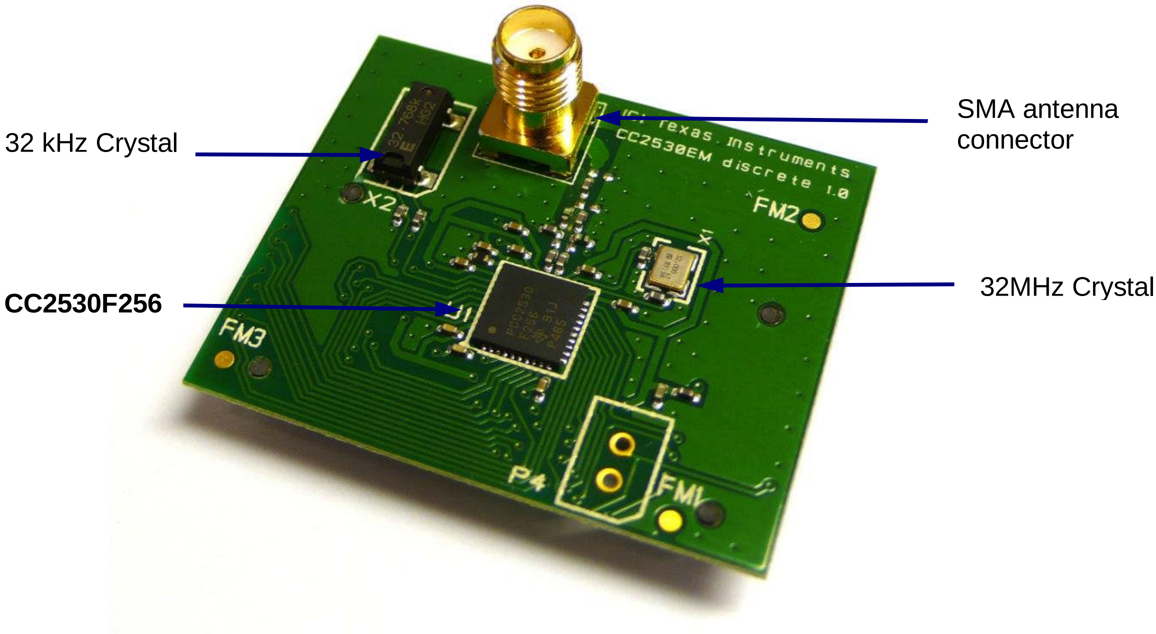


Figure 5 - CC2530 Evaluation Module

The CC2530EM is a complete RF module based on one of the recommended reference designs for the CC2530 radio. The module is equipped with a 32 MHz crystal, a 32.768 kHz crystal, external passive components for the balun and antenna match filter, an SMA connector for the antenna or any other RF instrument connection and general IO headers/connectors.

The table below shows the pin-out from the CC2530 to the two connectors on the backside of the evaluation module.

CC2530 Signal	P1	P1	CC2530 Signal
GND	1	2	NC
P0.4	3	4	P1.3
P0.1	5	6	P1.0
P0.2	7	8	NC
P0.3	9	10	P2.1
P0.0	11	12	P2.2
P1.1	13	14	P1.4
P0.6	15	16	P1.5
P0.7	17	18	P1.6
GND	19	20	P1.7

CC2530 Signal	P2	P2	CC2530 Signal
NC	1	2	NC
NC	3	4	NC
NC	5	6	NC
VDD	7	8	NC
VDD	9	10	NC
NC	11	12	NC
NC	13	14	NC
RESET	15	16	NC
P1.2	17	18	P0.5
P2.0	19	20	NC

Table 1 - CC2530EM pin-out

The part number of the EM connector is SFM-110-02-SM-D-A-K-TR from Samtec. It mates with the TFM-110-02-SM-D-A-K-TR, also from Samtec.

Please refer to the reference design on the web [1] for further details.

8 CC2531 USB Dongle

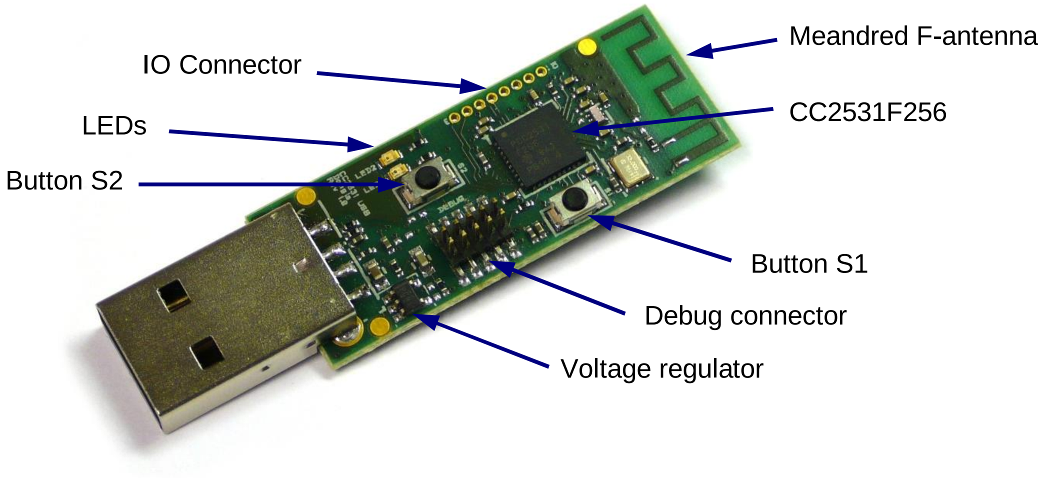


Figure 6 - CC2531 USB Dongle

The USB dongle that is included in the kit comes preprogrammed such that it can be used together with the SmartRF Packet Sniffer [10] to capture packets going over the air. To use the dongle as a sniffer, just install the Packet Sniffer PC application (available on the web [10]), plug in the USB dongle and start capturing packets. The Packet Sniffer User Manual [11] has more information.

The USB dongle can also be used as a general development board for USB and RF software. There is a USB firmware library available from the TI web pages with an implementation of a complete USB framework, including examples showing both HID and CDC ACM. There is a link to this library on the CC2530 DK web pages [3].

Table 2 shows which CC2531 signals are connected to what IO on the dongle.

IO Connector	CC2531
1	P0.2
2	P0.3
3	P0.4
4	P0.5
5	P1.7
6	P1.6
7	P1.5
8	P1.4

Dongle User IO	CC2531
Green LED	P0.0
Red LED	P1.1
Button S1	P1.2
Button S2	P1.3

Table 2 - CC2531 USB Dongle Pinout

In order to debug and program firmware on the CC2531, the CC2531 USB dongle can be connected to the SmartRF05EB as shown in the picture below. The small adapter board and flat cable is included in the development kit.

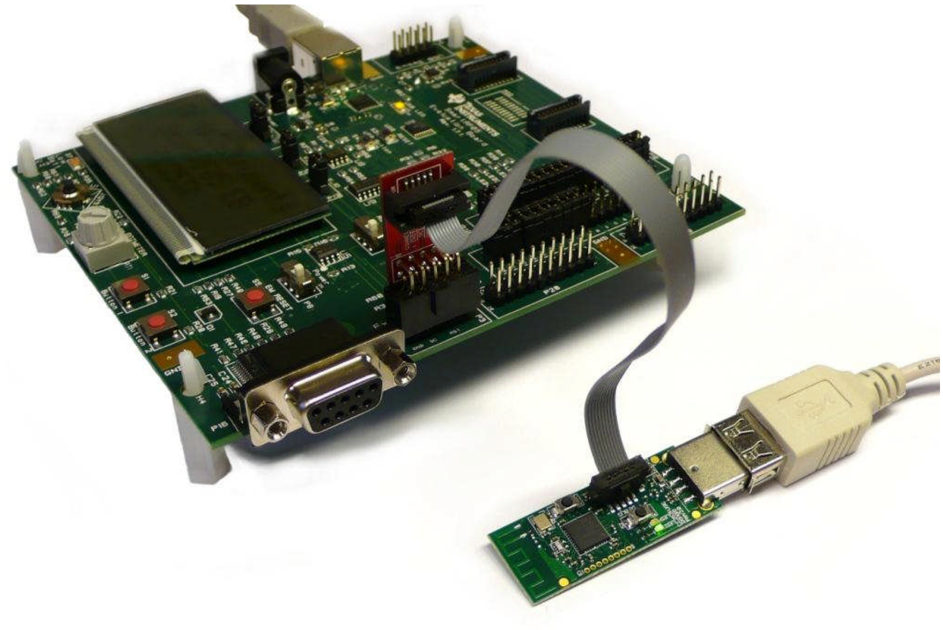


Figure 7 - CC2531 USB Dongle connected to SmartRF05EB

The debug connector on the CC2531 USB Dongle matches the debug connector on the SmartRF05EB (and the CC Debugger). Note that, by default, the CC2531 dongle is not powered through the debug connector, so an external power source must be used while programming. The easiest solution is to connect it to a USB port on the PC. Alternatively, resistor R2 can be mounted. The table below shows the pin out of the debug connector.

Pin #	Connection
1	GND
2	VCC
3	CC2531 P2.2 (DC)
4	CC2531 P2.1 (DD)
5	NC
6	NC
7	CC2531 RESET
8	NC
9	Optional external VCC (R2 must be mounted)
10	NC

Table 3 – CC2531 USB Dongle Debug Connector

Refer to the schematics (in the appendices) and layout (available on the web) for additional details.

9 SmartRF05 Evaluation Board

The SmartRF05 Evaluation Board is thoroughly described in the SmartRF05EB User's Guide [8]. That document will describe the hardware features in detail and provide the schematics for the board.

10 Frequently Asked Questions

Q1 When connecting the SmartRF05EB to my PC via USB, the dialog window below appears. Why? What should I do?

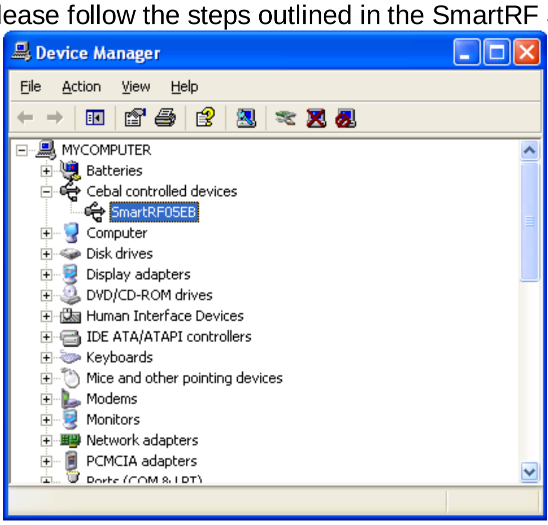
A1 The SmartRF05EB will be recognized as a USB device by the operating system, and it will ask the user to provide information about which USB driver that should be associated with the device.

If you have installed SmartRF Studio, just follow the instructions and select “Automatic installation”. Windows should find the required driver (cebal.sys), as specified in an .inf file. Both files (.inf and .sys) are included in the SmartRF installation.

If you have not installed SmartRF Studio, it is recommended that you do so before proceeding. Both the SmartRF Studio User Manual and SmartRF05EB User’s Guide has more details.

Q2 SmartRF05EB with the CC2530EM is not detected by IAR/SmartRF Studio. Why?

A2 Make sure you have installed SmartRF Studio as described in A2. Then verify that the device is associated with the correct driver by opening the Device Manager on you PC. When the EB is connected, the “Cebal controlled devices” list contains “SmartRF05EB”. If the board is listed as an unknown device, please follow the steps outlined in the SmartRF Studio User Manual.



Q3 How can I measure the current consumption of the CC2530?

A3 The easiest way to measure current consumption of the chip in various modes is to connect the EM directly to the SmartRF05EB and disconnect everything on the board that consumes power by removing all jumpers. The jumper on header P13 should not be removed. Connect the ampere meter between the two terminals on P15. On P10, the jumper for the EM_RESET signal (connector 35-36) should be mounted. On P1, no jumpers are required, but in order to control the SoC from a debugger, mount a jumper between 19-20 (DBG_DD) and 21-22 (DBG_DD). Make sure the RS232 Enable switch is in the “disable” position.

Use SmartRF Studio to set the radio in different modes (RX, TX, etc.), or download an application on the CC2530 setting the device in the preferred state.

Q4 Can I use another compiler than IAR to develop software for CC2530?

A4 Yes, there are several tools available that can be used for CC2530. Any 8051 compiler (e.g. Keil, GCC, and SDCC) can, in theory, be used. Note that these tools may have limited debugging support for CC2530.

11 References

[1] **CC2530 product web site**
http://focus.ti.com/docs/prod/folders/print/cc2530.html

[2] **CC2531 product web site**
http://focus.ti.com/docs/prod/folders/print/cc2531.html

[3] **CC2530DK web site**
http://focus.ti.com/docs/toolsw/folders/print/cc2530dk.html

[4] **CC2530DK Quick Start Guide**
http://www.ti.com/lit/swra273

[5] **CC2530 Software Examples User's Guide**
http://www.ti.com/lit/swru137

[6] **Texas Instruments Support**
http://support.ti.com

[7] **Texas Instruments Low Power RF Online Community**
http://www.ti.com/lprf-forum

[8] **SmartRF05EB User's Guide**
http://www.ti.com/lit/swru210

[9] **SmartRF Studio**
http://www.ti.com/smartrfstudio

[10] **SmartRF Packet Sniffer**
http://focus.ti.com/docs/toolsw/folders/print/packet-sniffer.html

[11] **SmartRF Packet Sniffer User Manual**
http://www.ti.com/lit/swru187

[12] **TIMAC**
http://www.ti.com/timac

[13] **Z-Stack**
http://www.ti.com/z-stack

[14] **DN002 -- Practical Sensitivity Testing**
http://www.ti.com/lit/swra097

[15] **DN018 -- Range Measurements in an Open Field Environment**
http://www.ti.com/lit/swra169

[16] **IAR Embedded Workbench for 8051**
http://www.iar.com

12 Document history

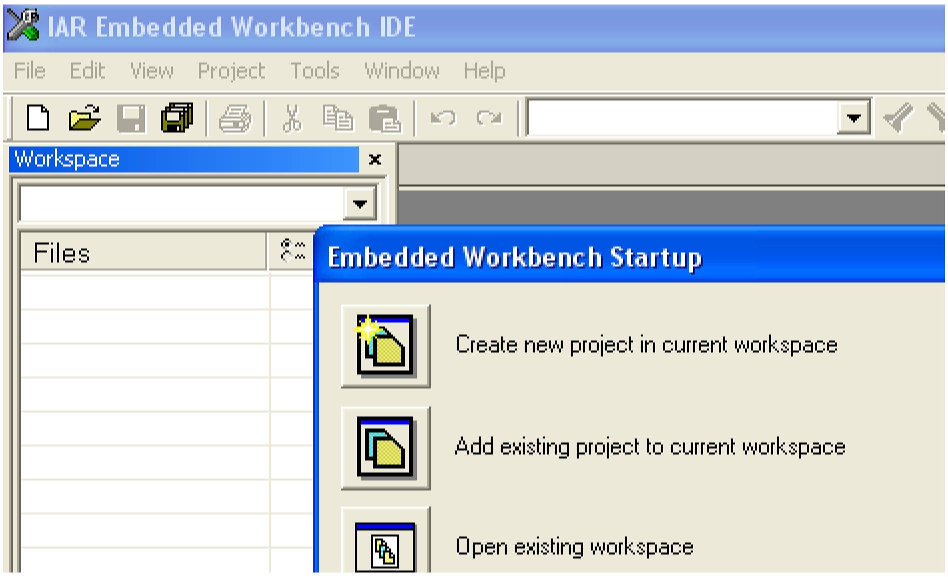
Revision	Date	Description/Changes
A	2009-04-20	Editorial update.
-	2009-04-08	First revision.

Appendix A Setting up the Software Environment

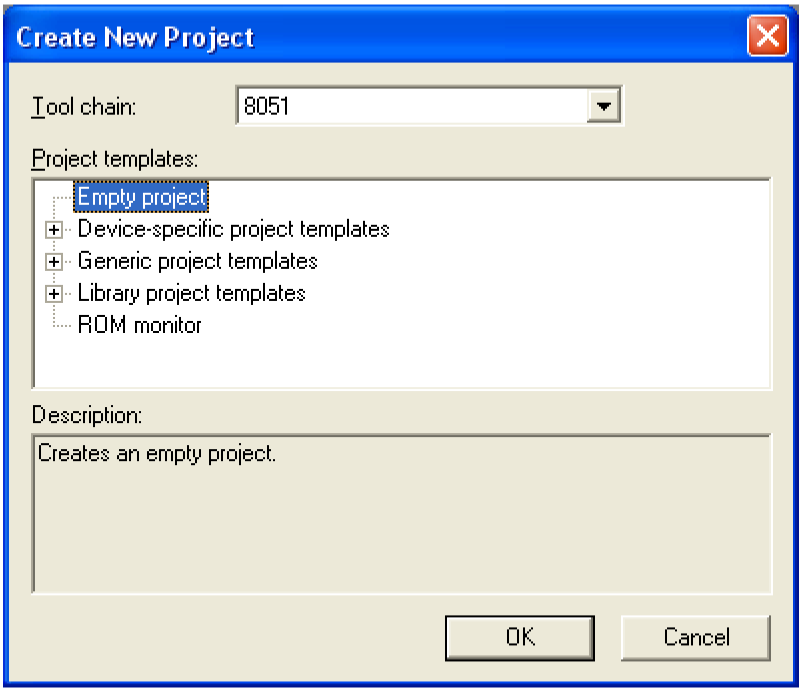
This appendix will guide you through the initial steps of setting up a complete software development environment with IAR Embedded Workbench for 8051. Version 7.51 of the tool supports CC2530 and CC2531 out-of-the-box (no patches required).

A.1 Create the project

After installing IAR EW8051, start the application. The dialog window below should appear:



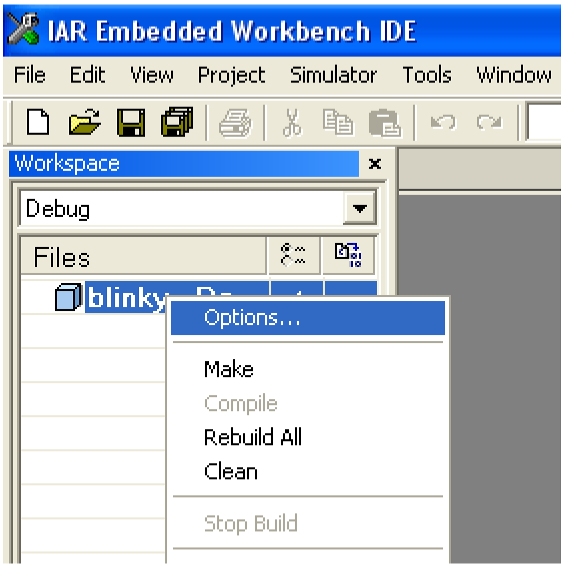
Select “Create new project in current workspace”



Select Empty project and click OK. You will now be asked to save the project. Select an appropriate name for the project and save it. The dialog window will close and the project will be listed in the “workspace” panel at the left side of the IDE.

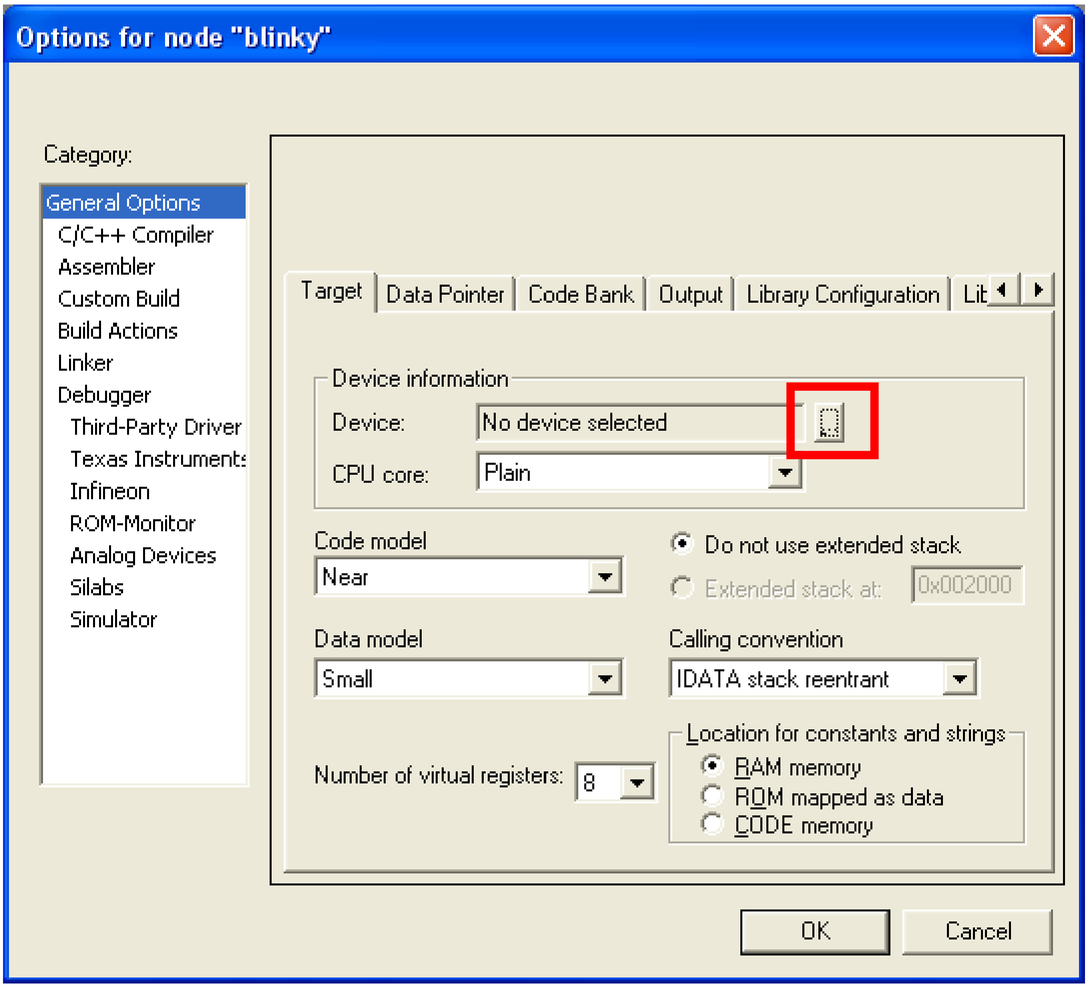
A.2 Project Options

Right click the project to set up the project options.

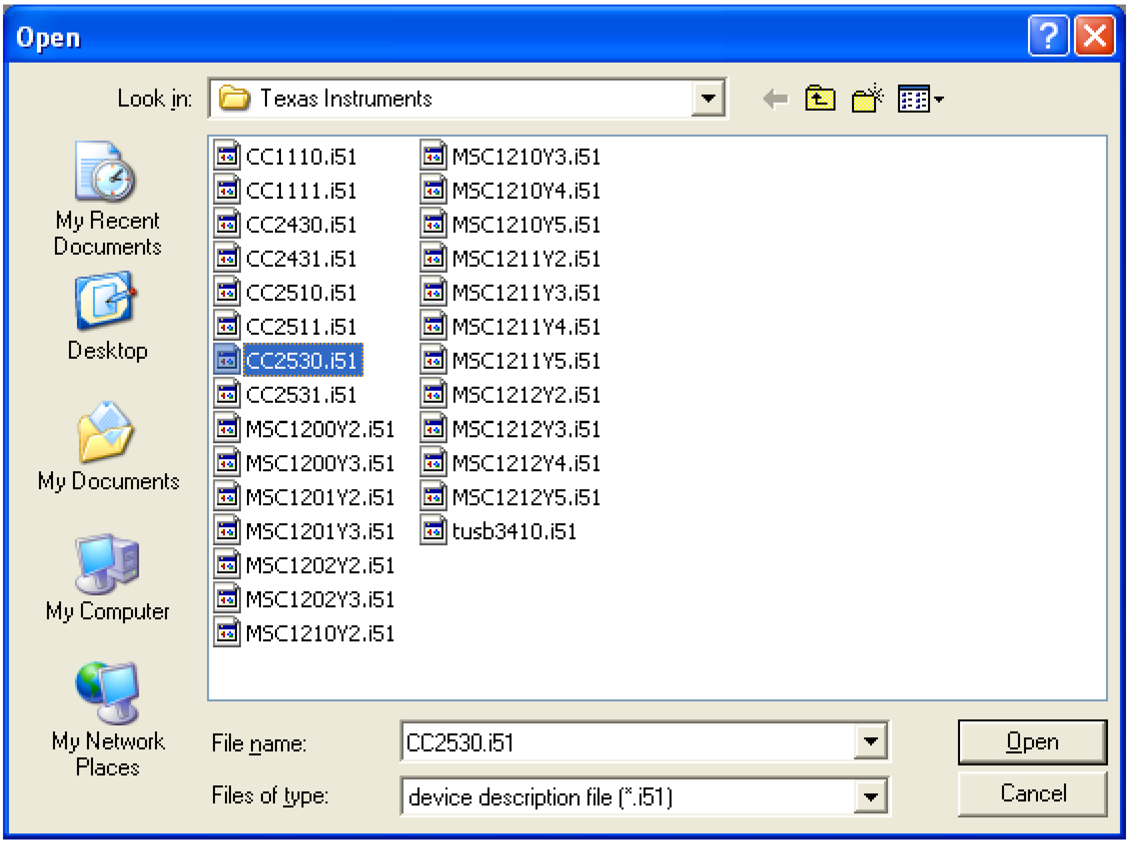


A.3 Select Device

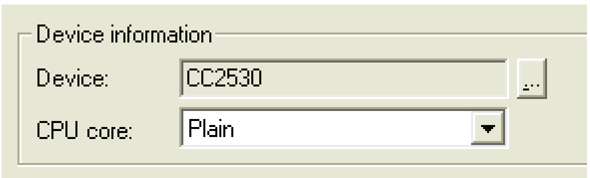
In the dialog window that appears, the first thing that is required is to select the device for which the project is built. Click on the button next to the device field.



A new window will appear, that allows you to browse through the device configuration files to choose an appropriate device. Select the CC2530.i51 file from the *<ew8051_install_dir>\8051\devices\Texas Instruments* folder. This .i51 device description file contains basic information about the chip.



Back in the General Options view, you will see that CC2530 is now the selected device. “CPU core” should be set to Plain.



A.4 Select Code and Memory Model

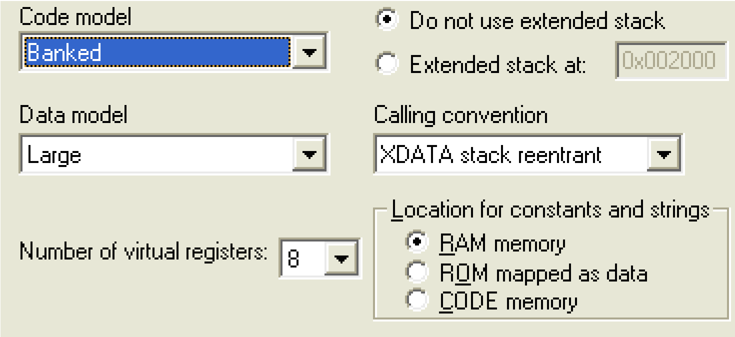
Next, select code model. Either “Near” or “Banked” can be chosen.

“Near” can be used when you don’t need banking support, i.e. when you only need access to 64 kilobytes of the flash memory. This option is suitable for the CC253xF32 and CC253xF64 devices. It is also possible to use this option for the other devices (F128 and F256) when only 64 kB flash is required.

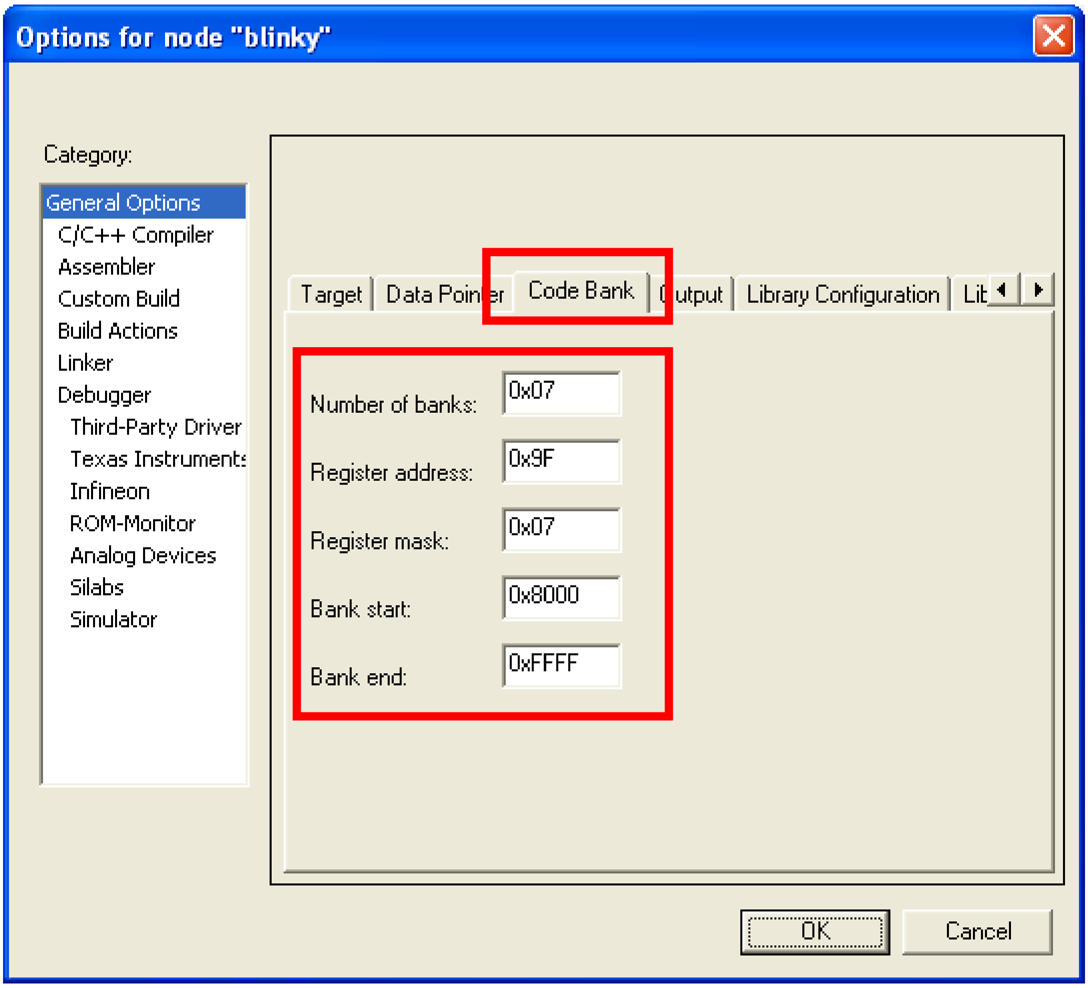
“Banked” should be used for getting access to the whole flash for the CC253xF128 and CC253xF256 devices.

The default data model for the Near code model is Small. For Banked, it is Large. The data model determines how the compiler & linker use the memory of the 8051 for storage of variables. With the small data model, variables are typically stored in the DATA memory space. For the large data model, variables are stored in XDATA. The CC2530 User Guide and IAR 8051 C/C++ Compiler Reference Guide have more information about the various memory spaces. The important thing to remember is that the 8051 core uses different instructions to access the various memory spaces. Access to IDATA is, in general, much quicker than accessing XDATA, but there is normally much more XDATA available than IDATA.

For this example, we use banked code model and large memory model to support the CC2530F256 device included in the development kit. The stack can be placed in XDATA. After setting up the above option, you should have the following settings:



For the Banked code model, some additional settings are required. Select the Code Bank tab in the options window and set up the following parameters:

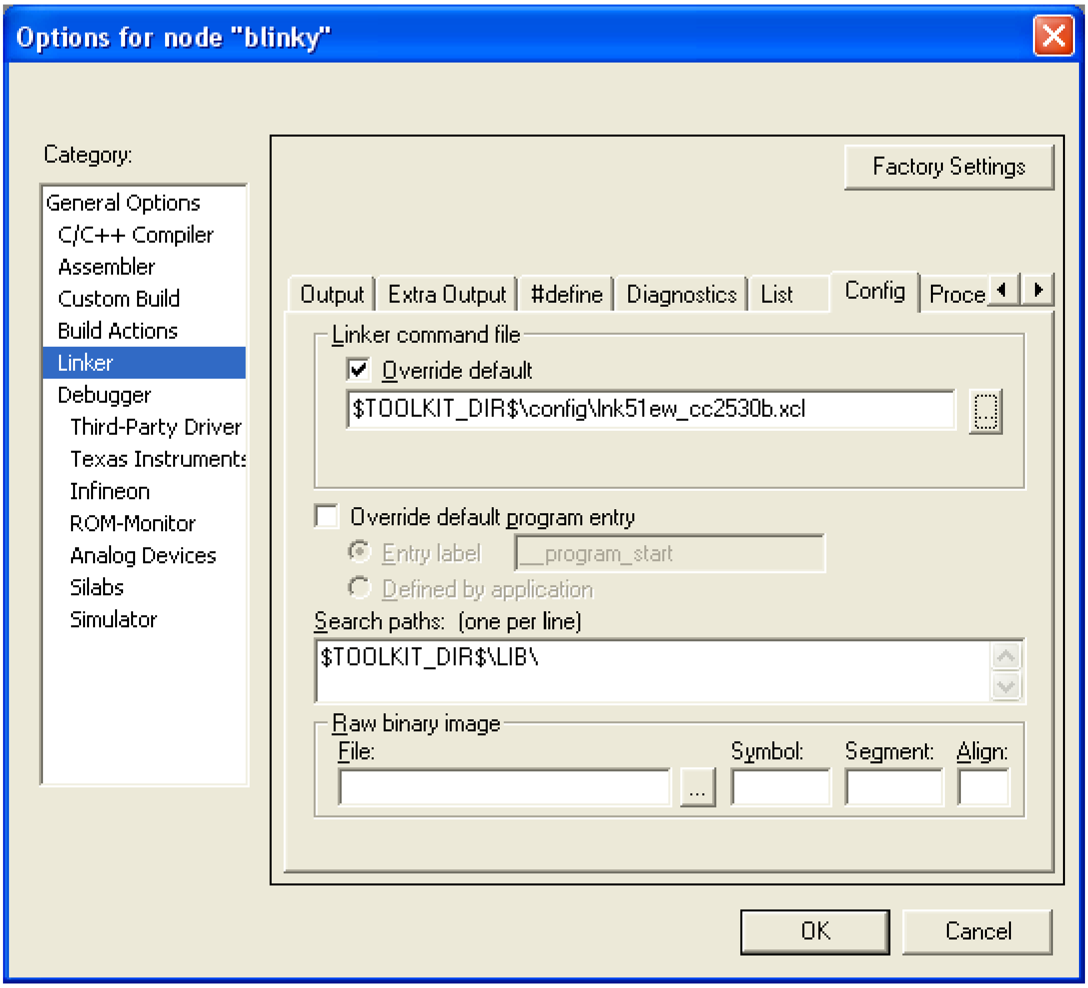


In addition to the common (root) bank, the CC2530 uses 7 code banks in order to access the whole 256 kB of Flash. The number of banks should be set to 0x07 for both F128 and F256. Register address 0x9F is the CC2530 FMAP register, which controls what code bank that is currently mapped into the 8051 address space. The register mask of 0x07 specifies that it is the 3 least significant bits in the FMAP register that are used for selecting banks.

A.5 Configure the Linker

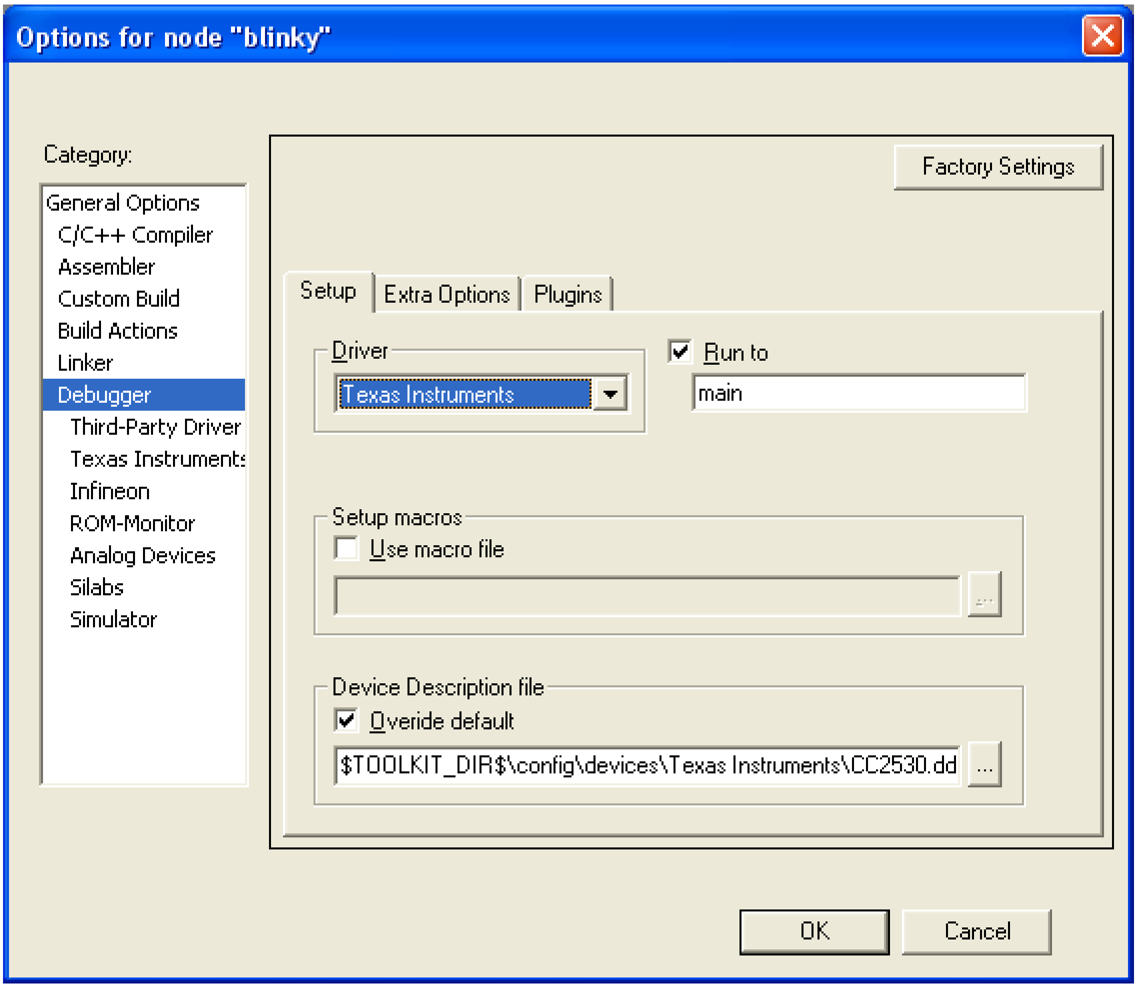
Next, you will need to instruct the IDE what linker command file to use. The linker command file contains information the linker uses in order to place code and variables in ram and flash. Thus, the linker file must match the flash and ram size of device you are working with. Normally, the linker file should be tailor-made to an application for optimum performance, but the default command file will work with most applications.

In the left menu, select “Linker”. Tick the “Override default” in the “Linker command file” section and select the appropriate linker file. For this example, we will use lnk51ew_cc2530**b**.xcl, which is suitable for CC253xF128 and CC253xF256. The **b** indicates banked code model. The other file, lnk51ew_cc2530.xcl, is suitable for CC253xF32 and CC253xF64, or the larger flash variants if banking is not required.



A.6 Configure the Debugger

Finally, in the debugger section, chose “Texas Instruments” for the Driver.



All the other project options can be left as is and you can close the Project Options dialog by clicking OK.

A.7 Write Software

At this point, the project is configured and you can write your first lines of code. We will show a small blinking LED example.

In the project, create a new file that you save as blinky.c. Type the following code:

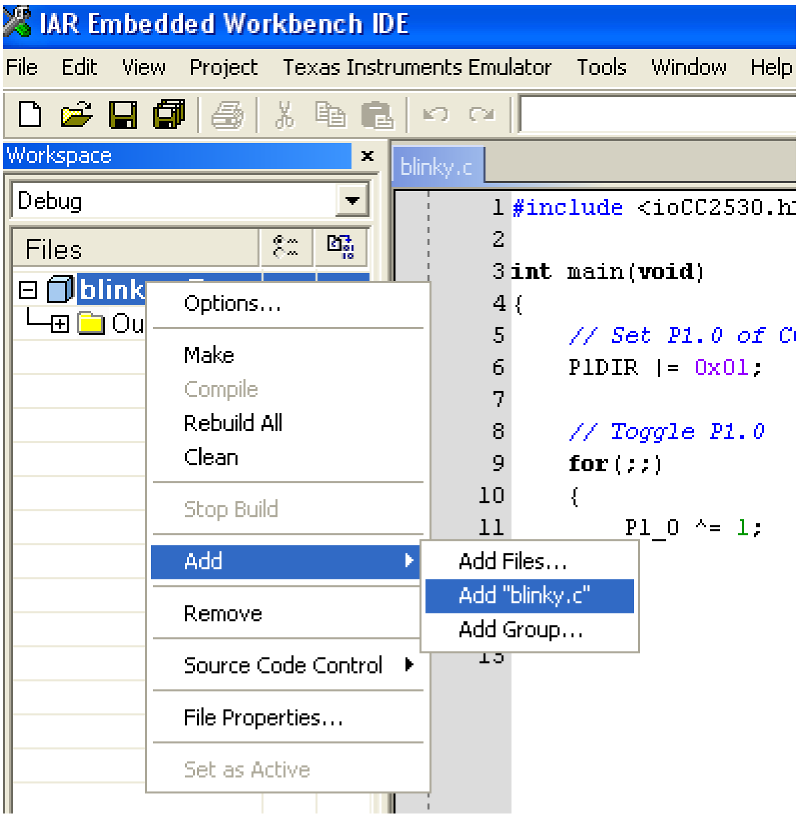
```
#include <ioCC2530.h>

int main(void)
{
    // Set P1.0 of CC2530 as output
    P1DIR |= 0x01;

    // Toggle P1.0
    for(;;)
    {
        P1_0 ^= 1;
    }
}
```

The code will toggle P1.0 (very quickly).

Add the file to the project by right clicking the project and selecting Add “blinky.c”.

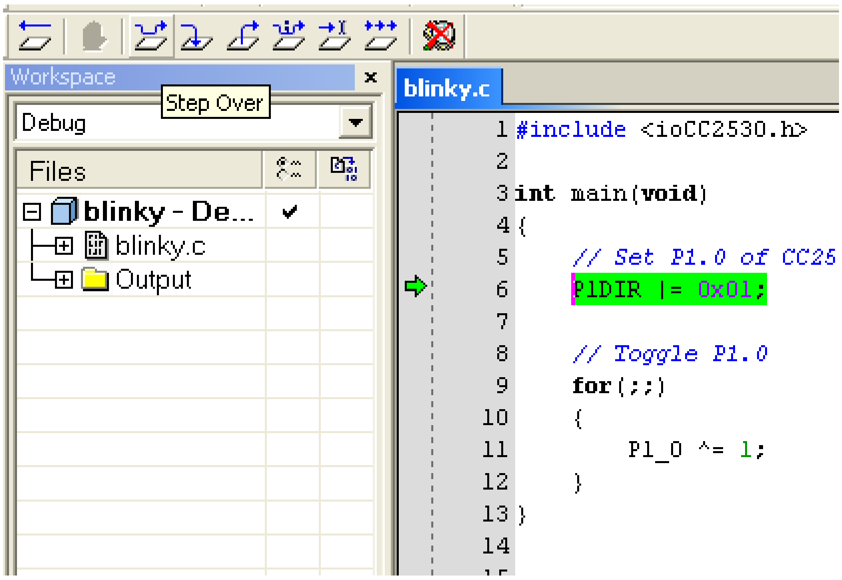


You are now ready to compile and download the code to the target!

A.8 Compile and Debug

Select "Project → Make" from the menu (or press F7) to build the project. The IDE will now compile, assemble and link the files in the project to generate an executable that can be downloaded to the target. A message window at the bottom of the screen should show the progress and indicate that the project was built successfully.

Next, download the application to the target by selecting "Project → Debug" from the menu (or press Ctrl+D). The application will now be downloaded to the target and you can start stepping through the code from main.



A.9 Done!

Congratulations! You have just made your first CC2530 software project in IAR.

Appendix B Software Solutions for CC2530 from TI

B.1 SimpliciTI™ Network Protocol

The SimpliciTI network protocol is a low-power RF protocol (for sub-1 GHz, 2.4 GHz and IEEE 802.15.4 RF ICs) targeting simple, small RF networks. This open-source software is an excellent start for building a network with battery-operated devices using a TI low-power RF System-on-Chip (SoC). The SimpliciTI network protocol was designed for easy implementation and deployment out-of-the-box on several TI RF platforms. It provides several sample applications.

Key Applications

- Alarm and security: occupancy sensors, light sensors, carbon monoxide sensors, glass-breakage detectors
- Smoke detectors
- Automatic meter reading: gas meters, water meters, e-meters
- Active RFID applications

Key Features

- Low power: A TI-proprietary low-power network protocol
- Flexible:
  - Direct device-to-device communication
  - Simple star with access point for store and forward to end device
  - Range extenders to increase range to four hops
- Simple: uses a five-command API
- Low duty cycle
- Ease of use

SimpliciTI is distributed as source code free of charge. For more information about the SimpliciTI network protocol, see the Texas Instruments SimpliciTI network protocol web site www.ti.com/simpliciti.

B.2 TIMAC Software

TIMAC software is an IEEE 802.15.4 medium-access-control software stack for TI's IEEE 802.15.4 transceivers and System-on-Chips.

You can use TIMAC when you:

- Need a wireless point-to-point or point-to-multipoint solution; e.g. multiple sensors reporting directly to a master
- Need a standardized wireless protocol
- Have battery-powered and/or mains-powered nodes
- Need support for acknowledgement and retransmission
- Have low data-rate requirements (around 100-kbps effective data rate)

Features

- Support for IEEE 802.15.4 standard
- Support for beacon-enabled and non-beaconing systems
- Multiple platforms
- Easy application development

The TIMAC software stack is certified to be compliant with the IEEE 802.15.4 standard. TIMAC software is distributed as object code free of charge. There are no royalties for using TIMAC software.

For more information about TIMAC software, see the Texas Instruments TIMAC Web site www.ti.com/timac.

B.3 RemoTI™ Network Protocol

Most existing remote controls use infrared technology to communicate commands to consumer electronics devices. However, radio frequency (RF) remote controls enable non-line-of-sight operation and provide more advanced features based on bidirectional RF communication.

ZigBee Radio Frequency for Consumer Electronics (RF4CE) is the result of a recent agreement between the ZigBee Alliance and the RF4CE Consortium (http://www.zigbee.org/rf4ce) and has been designed to be deployed in a wide range of remotely-controlled audio/visual consumer electronics products, such as TVs and set-top boxes. ZigBee RF4CE key benefits:

- Richer communication and increased reliability
- Enhanced features and flexibility
- Interoperability
- No line-of-sight barrier

The RemoTI network protocol is Texas Instruments' implementation of the ZigBee RF4CE standard. It is a complete solution offering hardware and software support for TI's low-power RF product portfolio. With the RemoTI network protocol we provide:

- An industry leading RF4CE-compliant stack featuring the interoperable CERC profile support, a simple API, easy to understand sample application code, full development kits and reference designs, and much more.
- Operation on our best-in-class IEEE 802.15.4 compliant System-on-Chip, the CC2530, with excellent RF co-existence and RF performance. The four flexible power modes include the lowest current consumption power down mode for long battery in life low duty-cycle applications.
- Extensive worldwide support and tools to ensure that development of ZigBee RF4CE-based products is simple, fast, and can be completed at minimal cost.
- A Golden Unit platform; RemoTI it is used for testing other implementations of the ZigBee RF4CE standard for standard compliance.

For more information on TI's RemoTI network protocol, see the Texas Instruments RemoTI network protocol web site www.ti.com/remoti or contact lprf-rf4ce@ti.com.

B.4 Z-Stack™ Software

The Z-Stack software is TI's ZigBee-compliant protocol stack for a growing portfolio of IEEE 802.15.4 products and platforms. The Z-Stack software stack is compliant with both ZigBee-2006 and ZigBee-2007 specification, supporting both the ZigBee and ZigBee PRO features sets. The Z-Stack software includes implementation of two ZigBee application profiles – Smart Energy and Home Automation. Other application profiles can easily be implemented by the user.

Z-Stack software notables include:

- A fully compliant ZigBee and ZigBee PRO feature set
- A range of sample applications including support for the ZigBee Smart Energy and ZigBee Home Automation profiles
- Over-the-air download and serial boot loader support
- Can be used together with the RF front ends CC2590 and CC2591, which support 10 dBm and 20 dBm output power respectively and improved receive sensitivity.

The Z-Stack software has been awarded the ZigBee Alliance's golden-unit status for both the ZigBee and ZigBee PRO stack profiles and is used by ZigBee developers world wide.

Z-Stack software is well suited for:

- Smart energy (AMI)
- Home automation
- Commercial building automation
- Medical, assisted living, or personal health and hospital care
- Monitoring and control applications
- Wireless sensor networks
- Alarm and security
- Asset tracking
- Applications that require interoperability

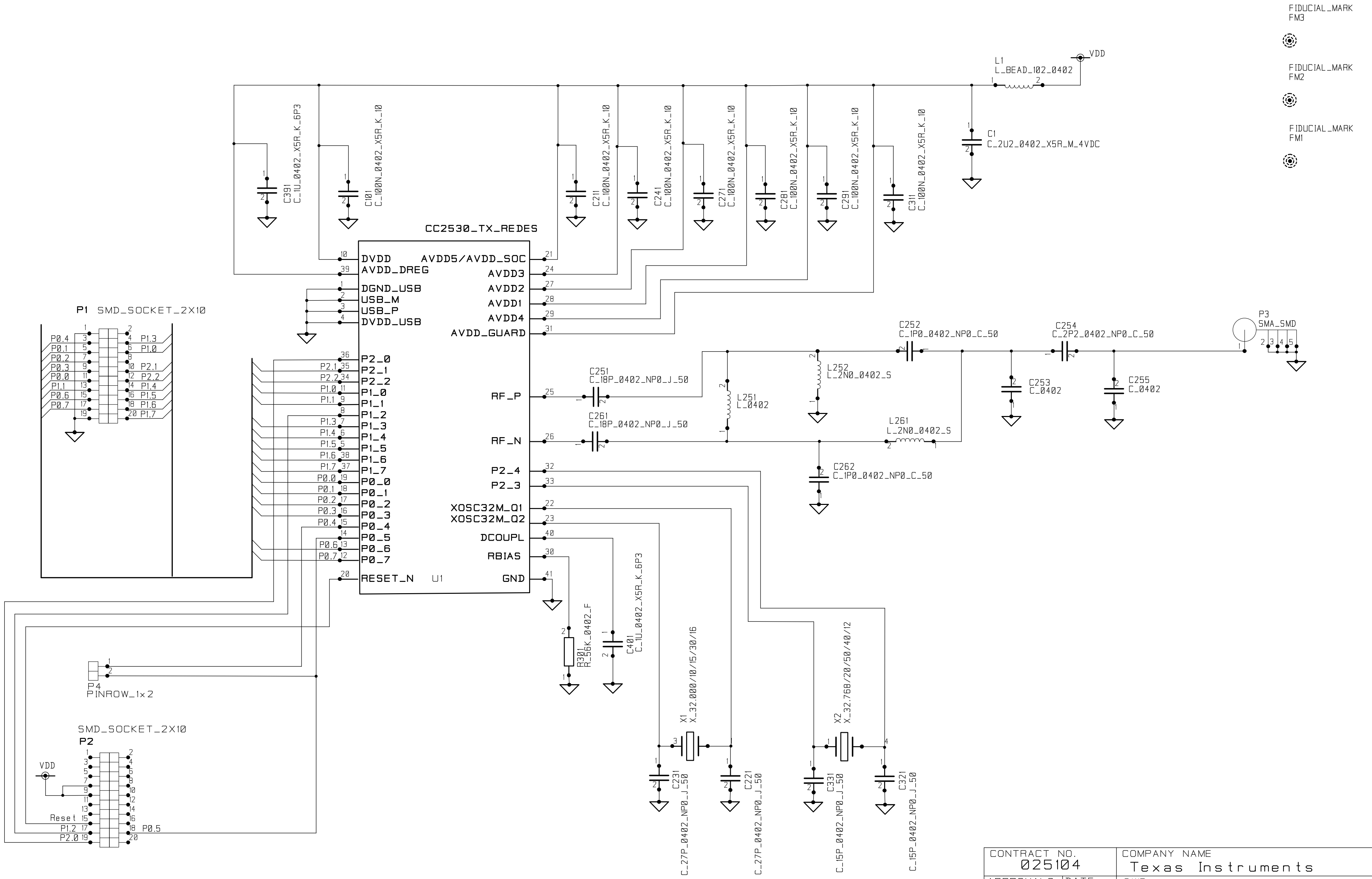
For more information about Z-Stack software, see the Texas Instruments Z-Stack software web site www.ti.com/z-stack.

Appendix C Schematics

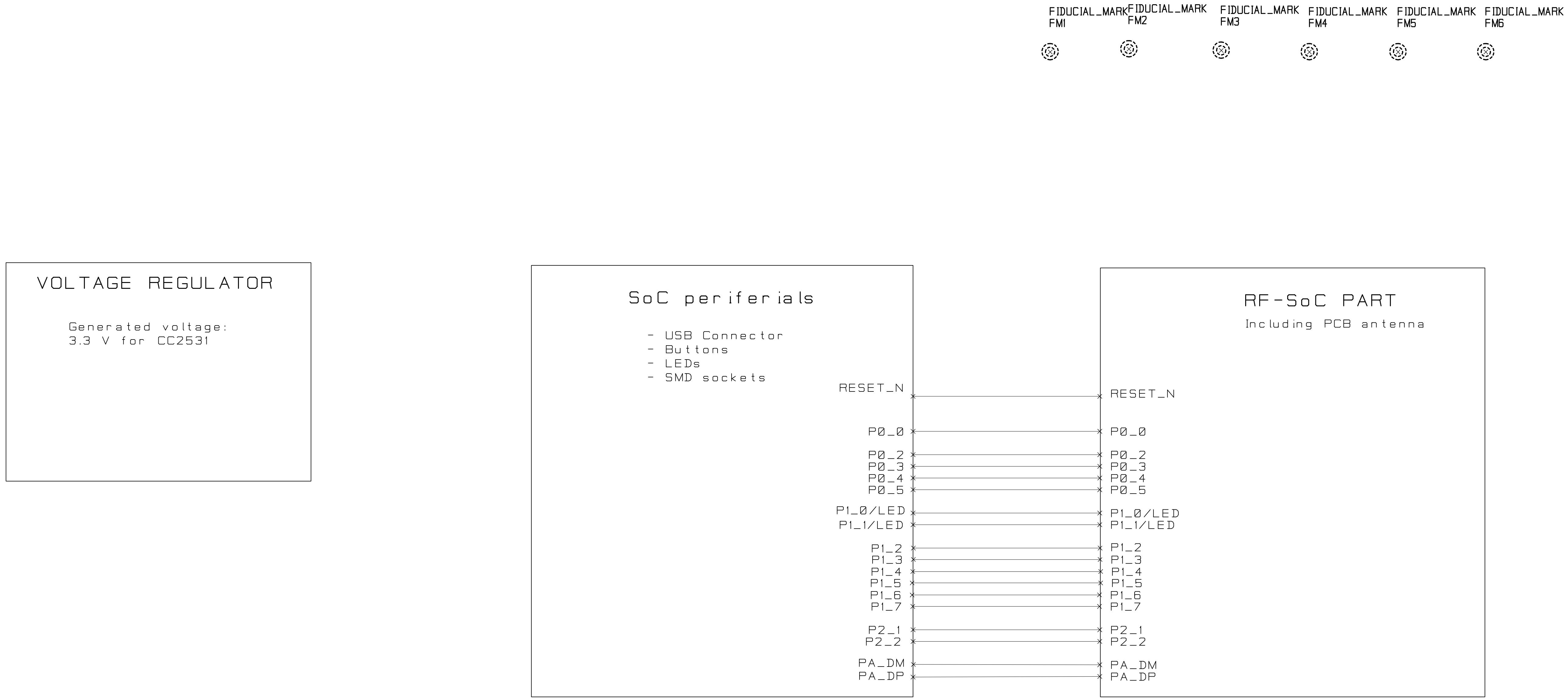
Please refer to the following pages for the schematics for

- CC2530 Evaluation Module
- CC2531 USB Dongle
- SmartRF05 Evaluation Board

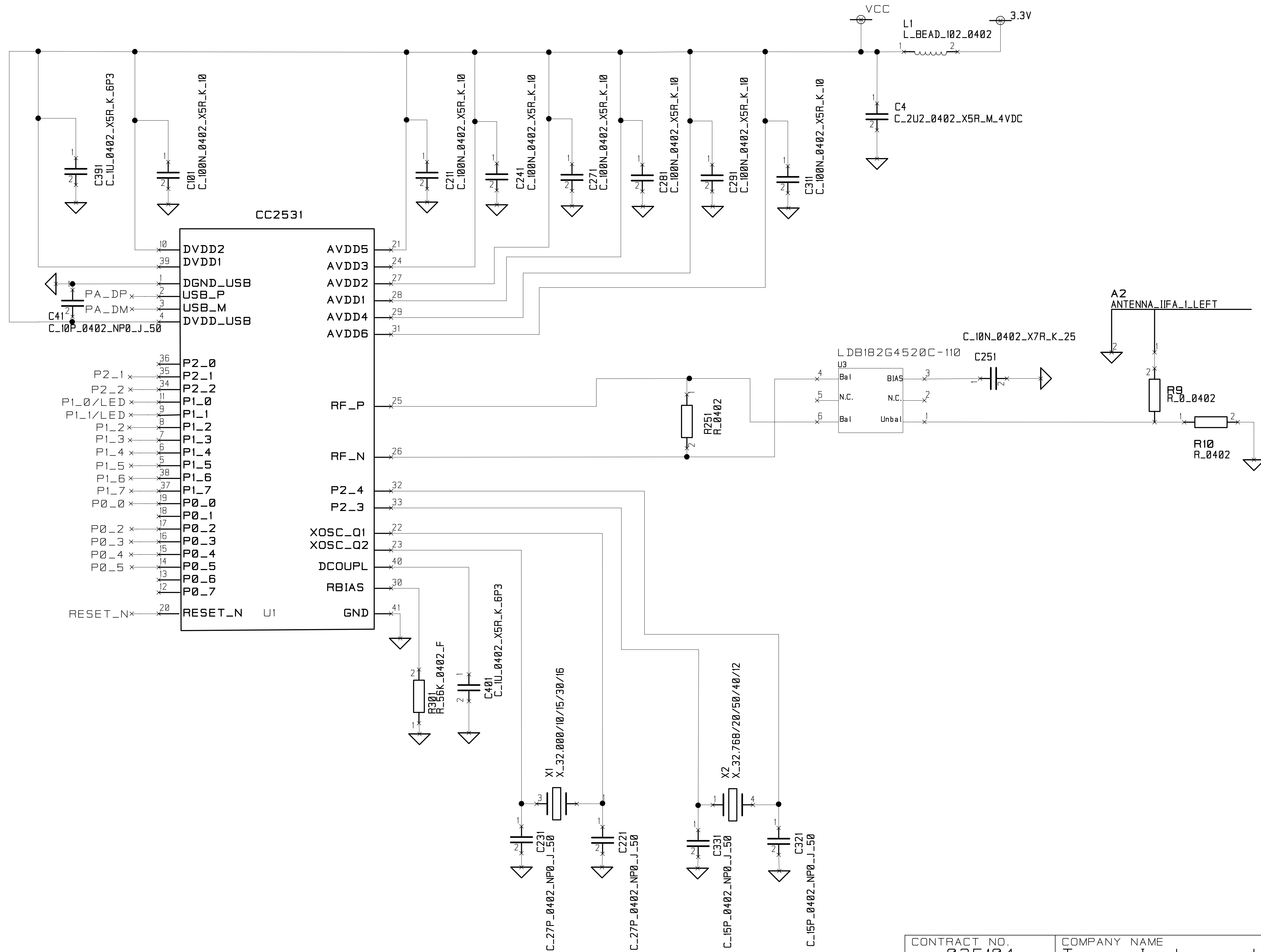
The layout for the evaluation module and USB dongle can be found on the CC2530 [1] and CC2531 [2] web pages respectively.



CONTRACT NO. 025104		COMPANY NAME Texas Instruments			
APPROVALS	DATE	DWG CC2530EM Discrete			
DRAWN	FRS	SIZE A4	FSCM NO.	DWG NO.	REV. 1.1
CHECKED	NN				
ISSUED		SCALE		SHEET 1 (1)	

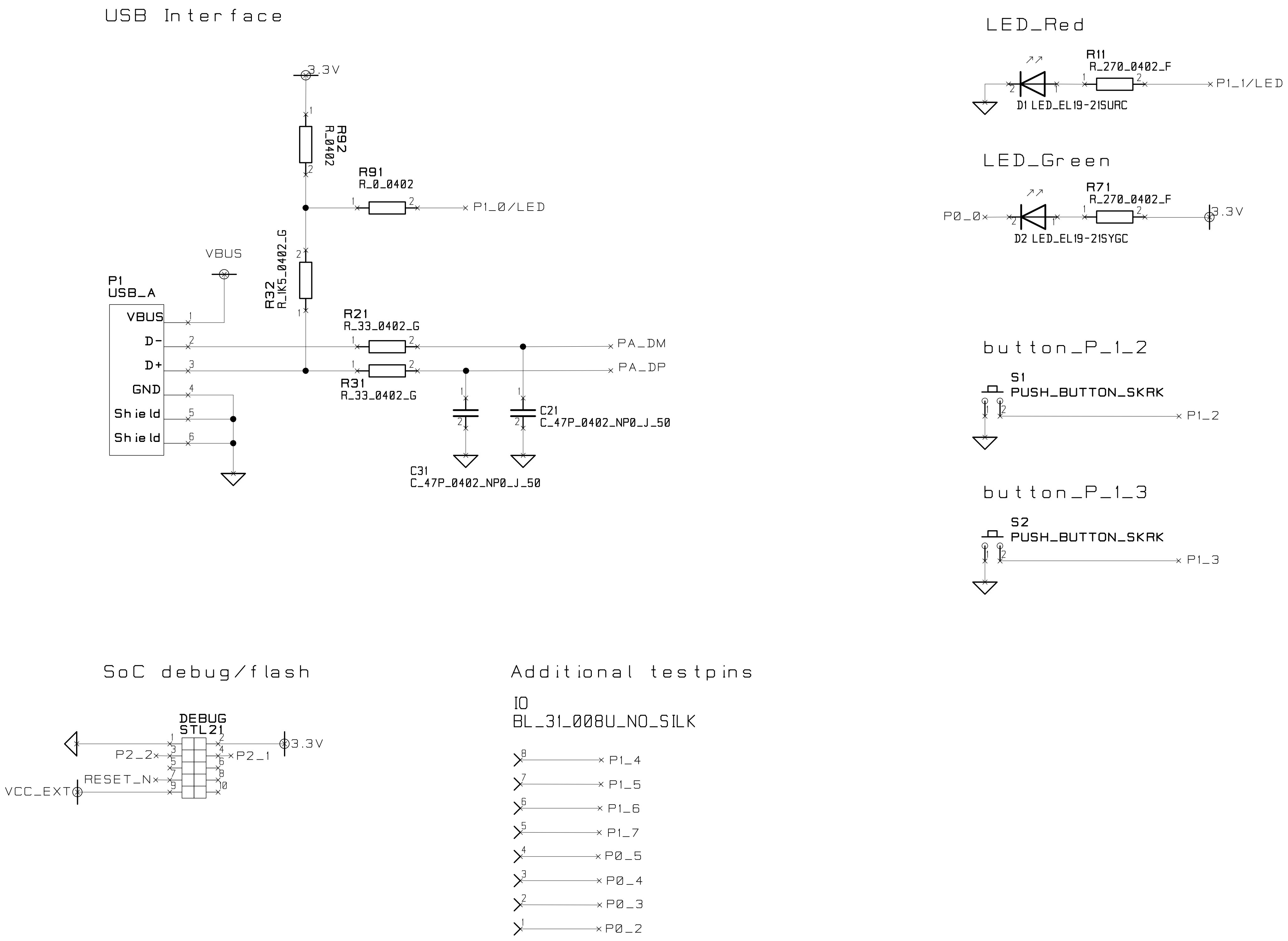


CONTRACT NO. 025104		COMPANY NAME Texas Instruments			
APPROVALS	DATE	DWG CC2531 dongle			
DRAWN	CHS	SIZE A4	FSCM NO.	DWG NO.	REV. 2.0
CHECKED	TIK				
ISSUED		SCALE		SHEET 1(4)	



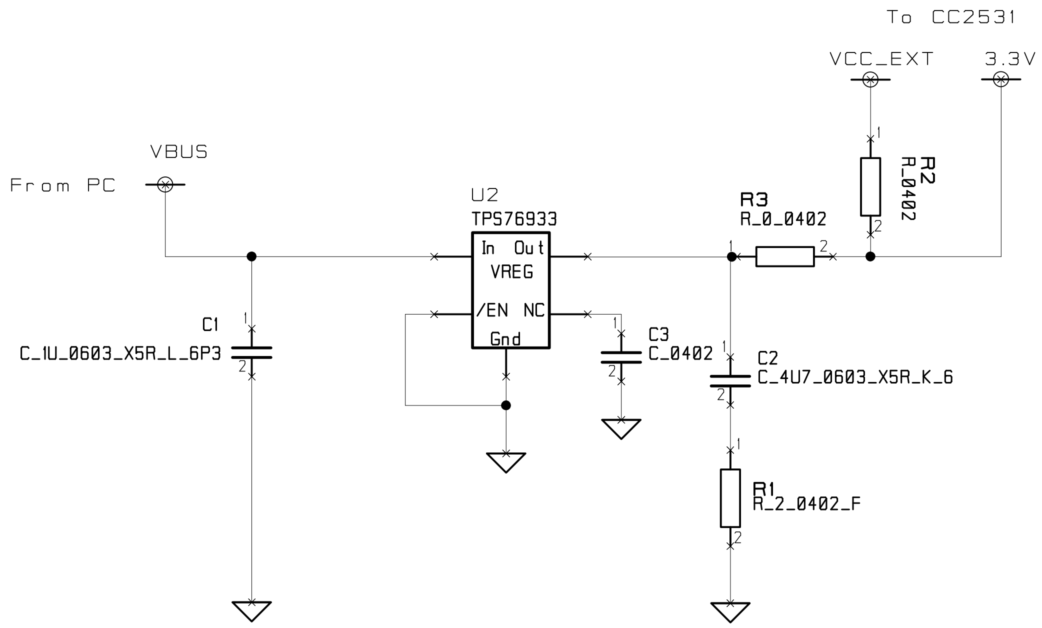
CONTRACT NO. 025104		COMPANY NAME Texas Instruments	
APPROVALS	DATE	DWG CC2531 USB DONGLE RF-PART	
DRAWN	CHS	SIZE A4 · FSCM NO. · DWG NO.	REV. 2.0
CHECKED	TIK		
ISSUED		SCALE	SHEET 3(4)

Not mount: R10,X2,C331,C321



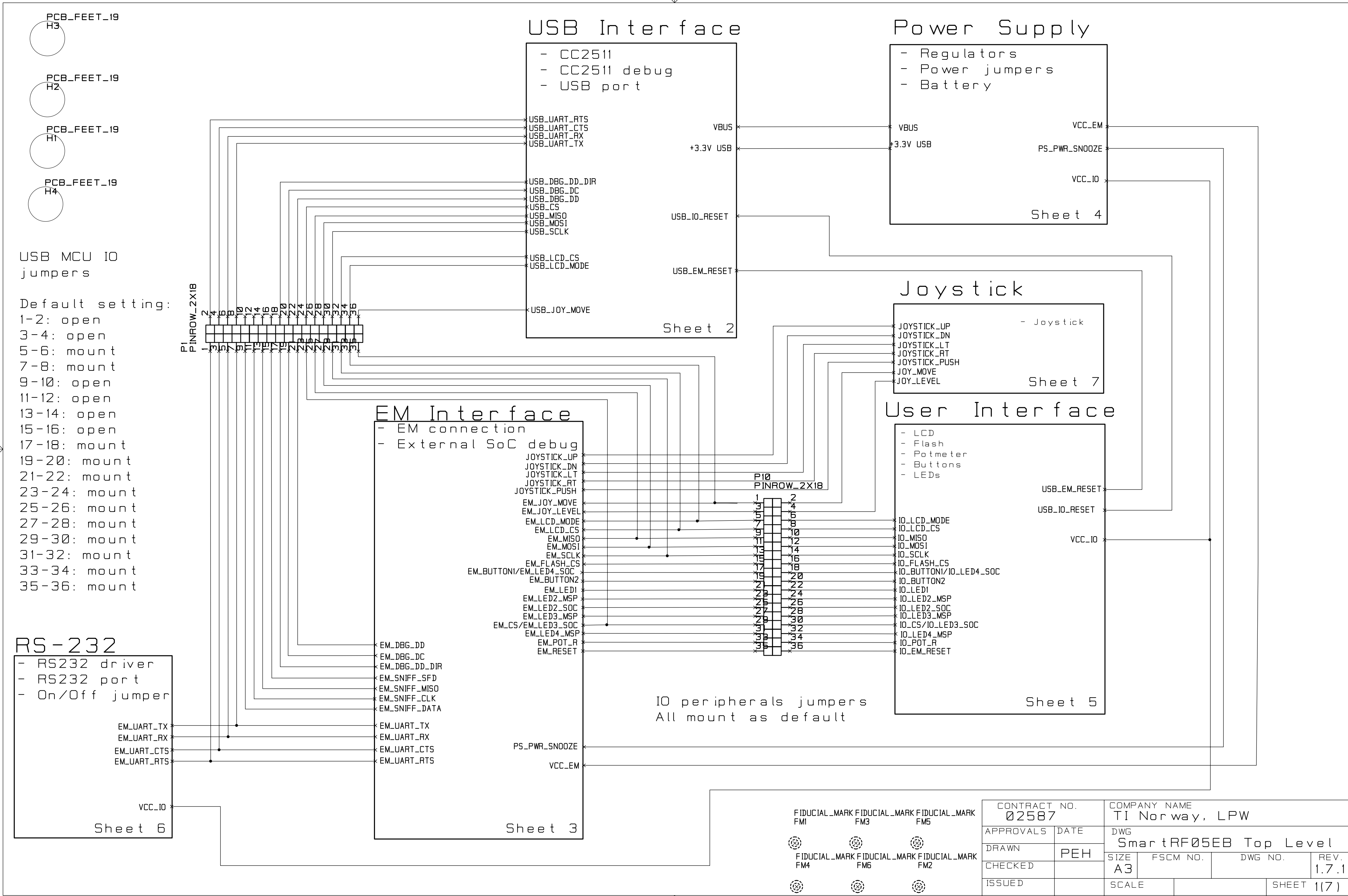
Not mount: R92, IO

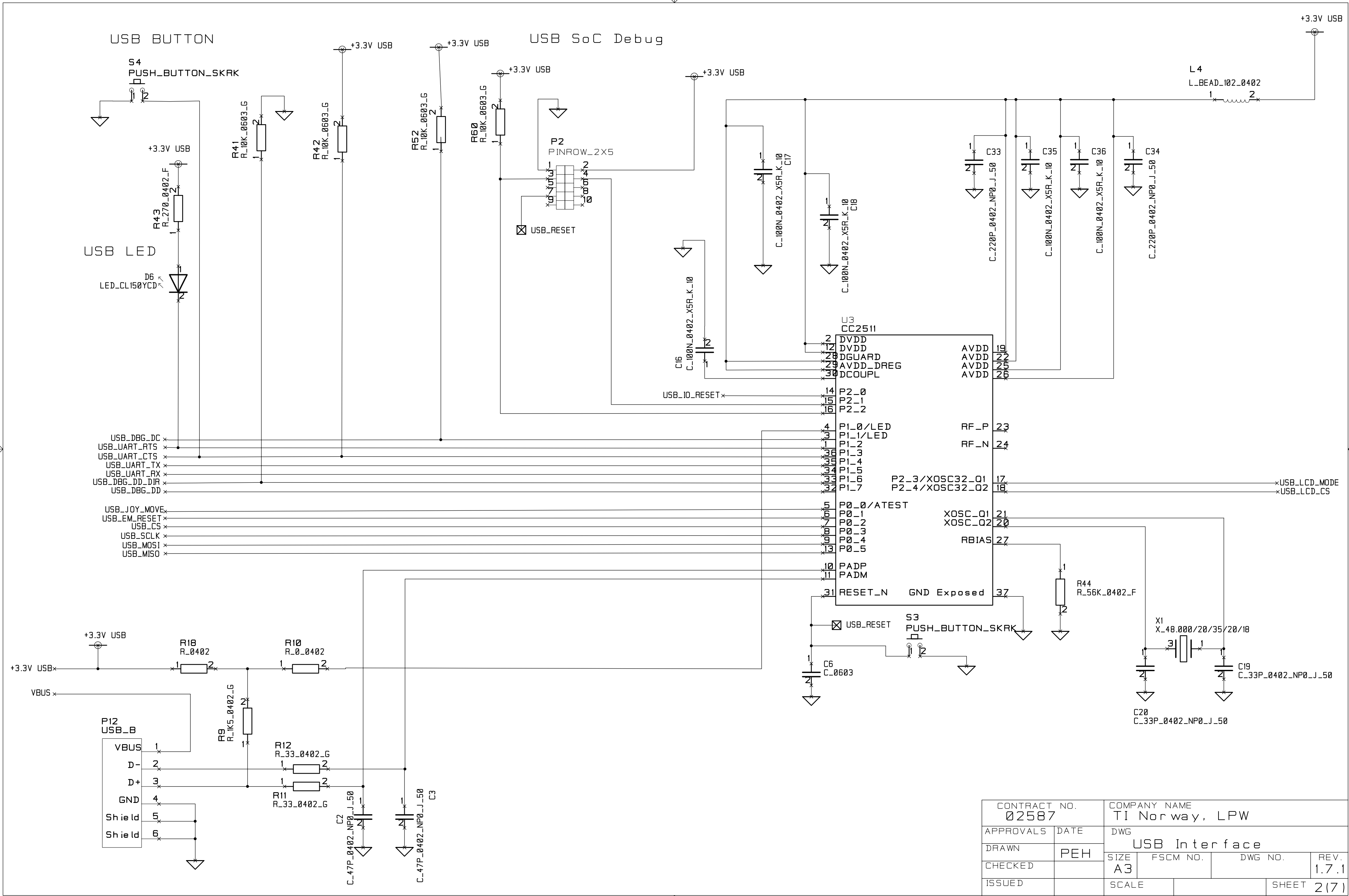
CONTRACT NO. 025104		COMPANY NAME Texas Instruments				
APPROVALS	DATE	DWG CC2531 USB dongle USB circuitry				
DRAWN	CHS	SIZE A4	FSCM NO.	DWG NO.		REV. 2.0
CHECKED	TIK					
ISSUED		SCALE			SHEET 4(4)	



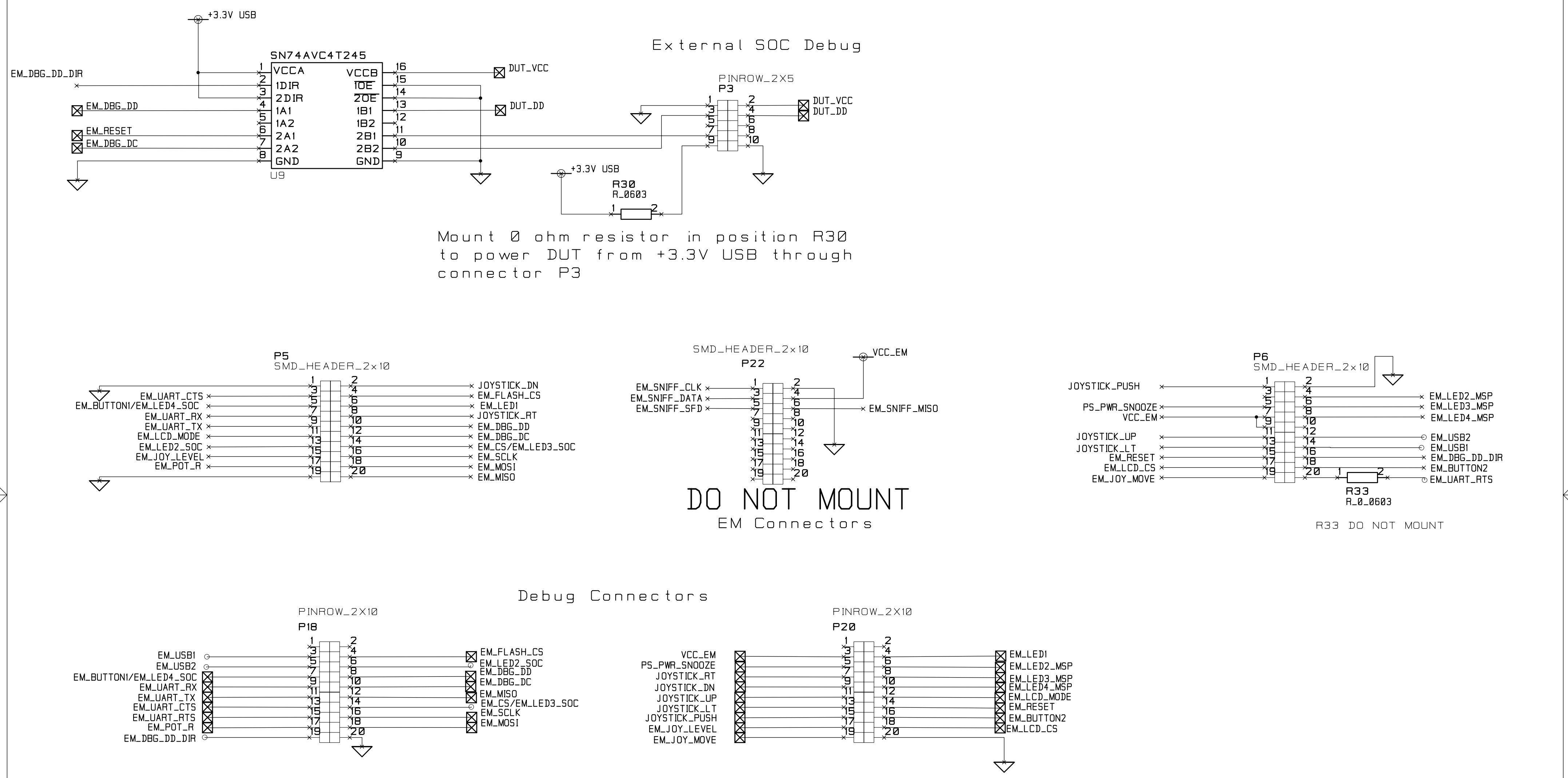
Not mount: C3, R2

CONTRACT NO. 025104		COMPANY NAME Texas Instruments			
APPROVALS	DATE	DWG CC2531 USB DONGLE VOLTAGE REGULATOR			
DRAWN	CHS	SIZE A4	FSCM NO.	DWG NO.	REV. 2.0
CHECKED	TIK				
ISSUED		SCALE		SHEET 2(4)	

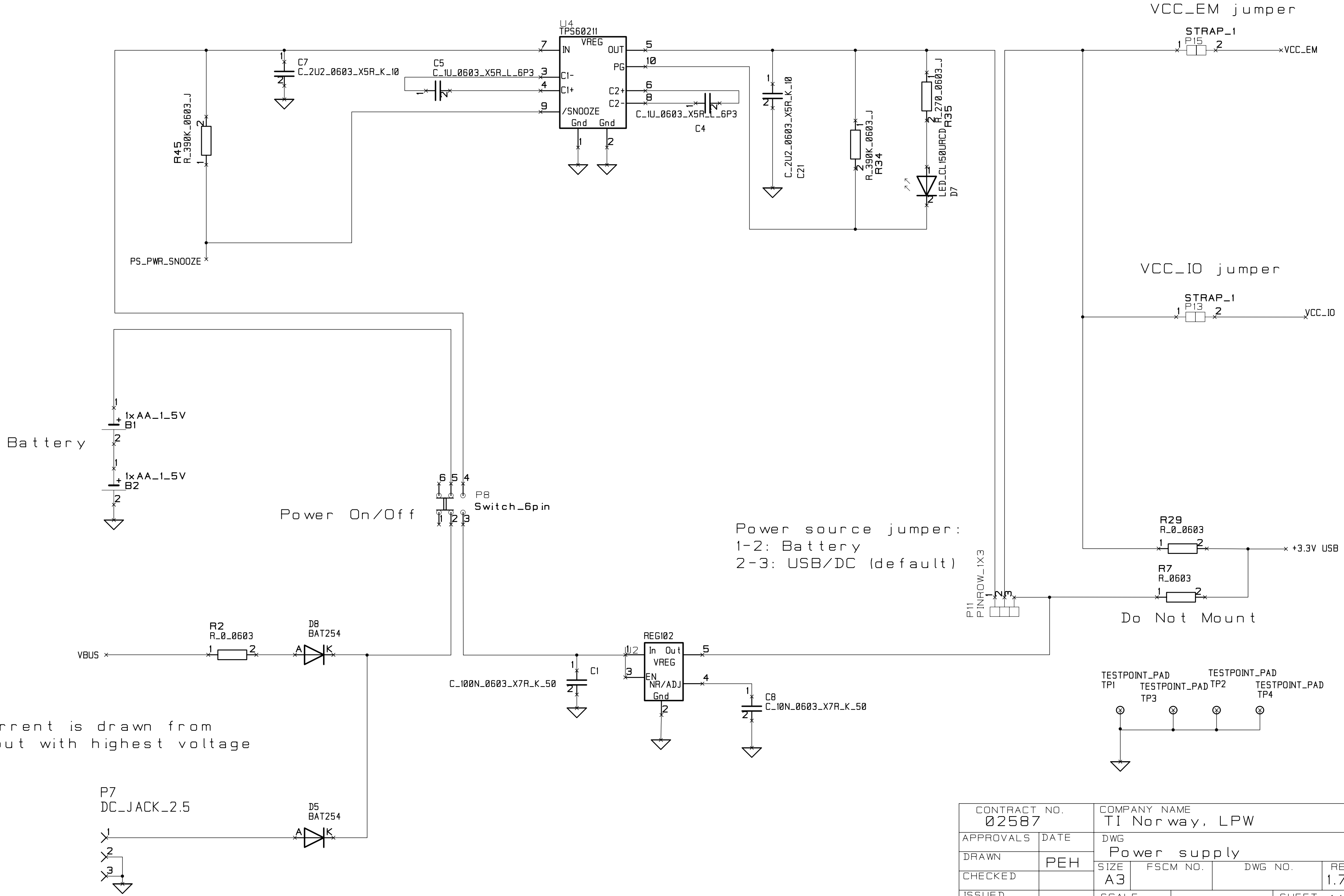

PCB_FEET_19
H3
PCB_FEET_19
H2
PCB_FEET_19
H1
PCB_FEET_19
H4
USB Interface
- CC2511
- CC2511 debug
- USB port
USB_UART_RTS
USB_UART_CTS
USB_UART_RX
USB_UART_TX
USB_DBG_DD_DIR
USB_DBG_DC
USB_DBG_DD
USB_CS
USB_MISO
USB_MOSI
USB_SCLK
USB_LCD_CS
USB_LCD_MODE
USB_JOY_MOVE
VBUS
+3.3V USB
USB_IO_RESET
USB_EM_RESET
Sheet 2
Power Supply
- Regulators
- Power jumpers
- Battery
VBUS
+3.3V USB
VCC_EM
PS_PWR_SNOOZE
VCC_IO
Sheet 4
USB MCU IO
jumpers
Default setting:
1-2: open
3-4: open
5-6: mount
7-8: mount
9-10: open
11-12: open
13-14: open
15-16: open
17-18: mount
19-20: mount
21-22: mount
23-24: mount
25-26: mount
27-28: mount
29-30: mount
31-32: mount
33-34: mount
35-36: mount
P1
PINROW_2X18
Joystick
- Joystick
JOYSTICK_UP
JOYSTICK_DN
JOYSTICK_LT
JOYSTICK_RT
JOYSTICK_PUSH
JOY_MOVE
JOY_LEVEL
Sheet 7
User Interface
- LCD
- Flash
- Potmeter
- Buttons
- LEDs
USB_EM_RESET
USB_IO_RESET
VCC_IO
IO_LCD_MODE
IO_LCD_CS
IO_MISO
IO_MOSI
IO_SCLK
IO_FLASH_CS
IO_BUTTON1/IO_LED4_SOC
IO_BUTTON2
IO_LED1
IO_LED2_MSP
IO_LED2_SOC
IO_LED3_MSP
IO_CS/IO_LED3_SOC
IO_LED4_MSP
IO_POT_R
IO_EM_RESET
Sheet 5
EM Interface
- EM connection
- External SoC debug
JOYSTICK_UP
JOYSTICK_DN
JOYSTICK_LT
JOYSTICK_RT
JOYSTICK_PUSH
EM_JOY_MOVE
EM_JOY_LEVEL
EM_LCD_MODE
EM_LCD_CS
EM_MISO
EM_MOSI
EM_SCLK
EM_FLASH_CS
EM_BUTTON1/EM_LED4_SOC
EM_BUTTON2
EM_LED1
EM_LED2_MSP
EM_LED2_SOC
EM_LED3_MSP
EM_CS/EM_LED3_SOC
EM_LED4_MSP
EM_POT_R
EM_RESET
EM_DBG_DD
EM_DBG_DC
EM_DBG_DD_DIR
EM_SNIFF_SFD
EM_SNIFF_MISO
EM_SNIFF_CLK
EM_SNIFF_DATA
EM_UART_TX
EM_UART_RX
EM_UART_CTS
EM_UART_RTS
PS_PWR_SNOOZE
VCC_EM
Sheet 3
P10
PINROW_2X18
IO peripherals jumpers
All mount as default
RS-232
- RS232 driver
- RS232 port
- On/Off jumper
EM_UART_TX
EM_UART_RX
EM_UART_CTS
EM_UART_RTS
VCC_IO
Sheet 6
FIDUCIAL_MARK FM1
FIDUCIAL_MARK FM3
FIDUCIAL_MARK FM5
FIDUCIAL_MARK FM4
FIDUCIAL_MARK FM6
FIDUCIAL_MARK FM2
CONTRACT NO.
02587
COMPANY NAME
TI Norway, LPW
APPROVALS
DATE
DWG
SmartRF05EB Top Level
DRAWN
PEH
CHECKED
SIZE
A3
FSCM NO.
DWG NO.
REV.
1.7.1
ISSUED
SCALE
SHEET 1(7)



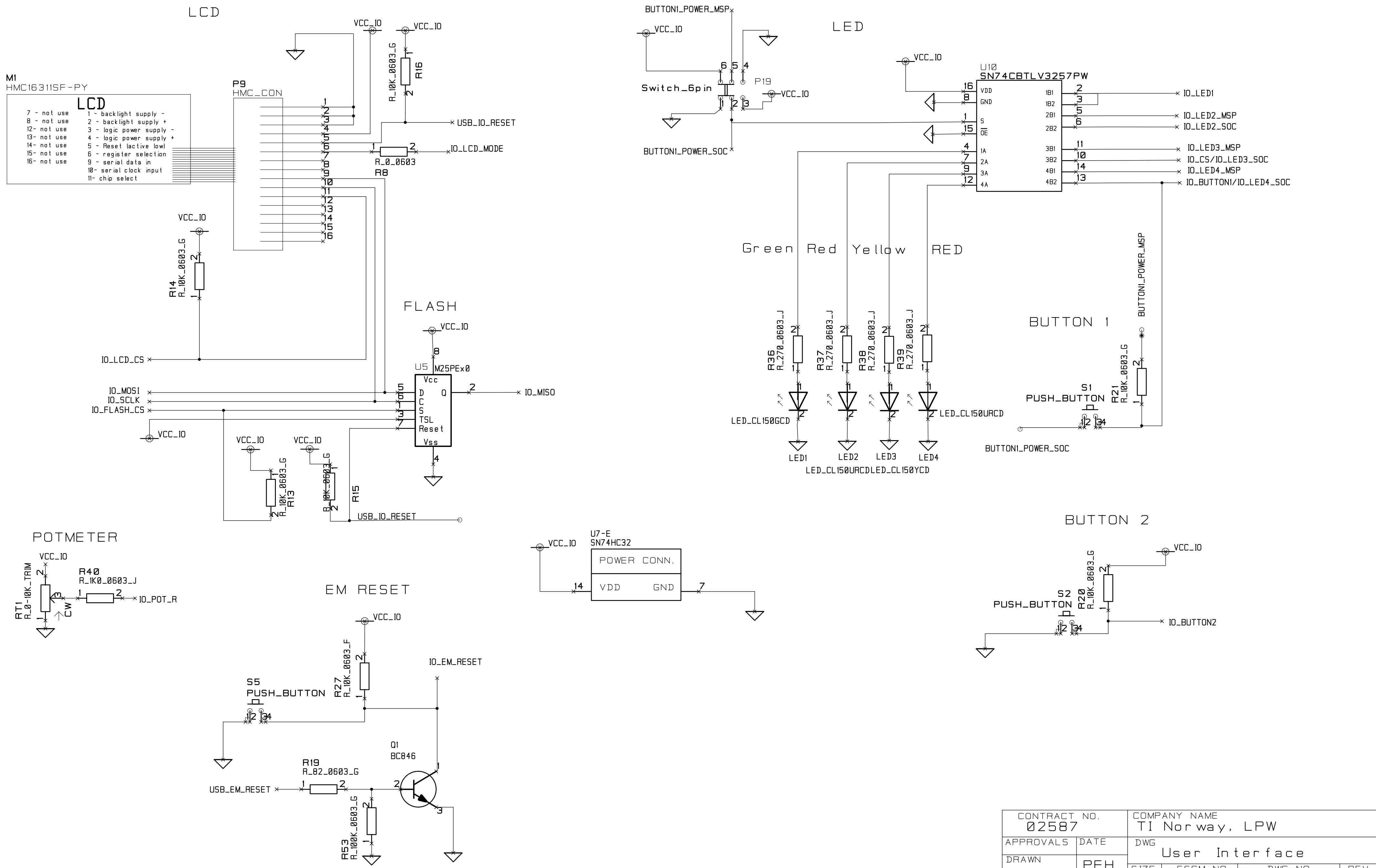
USB BUTTON
S4
PUSH_BUTTON_SKRK
USB SoC Debug
+3.3V USB
R41
R_10K_0603_G
R42
R_10K_0603_G
R52
R_10K_0603_G
R60
R_10K_0603_G
P2
PINROW_2X5
USB_RESET
R43
R_270_0402_F
USB LED
D6
LED_CL150YCD
L4
L_BEAD_102_0402
C17
C_100N_0402_X5R_K_10
C18
C_100N_0402_X5R_K_10
C16
C_100N_0402_X5R_K_10
C33
C_220P_0402_NP0_J_50
C35
C_100N_0402_X5R_K_10
C36
C_100N_0402_X5R_K_10
C34
C_220P_0402_NP0_J_50
U3
CC2511
DVDD
DVDD
DGUARD
AVDD_DREG
DCOUPL
AVDD
AVDD
AVDD
AVDD
USB_IO_RESET
P2_0
P2_1
P2_2
P1_0/LED
P1_1/LED
P1_2
P1_3
P1_4
P1_5
P1_6
P1_7
RF_P
RF_N
P2_3/XOSC32_Q1
P2_4/XOSC32_Q2
P0_0/ATEST
P0_1
P0_2
P0_3
P0_4
P0_5
XOSC_Q1
XOSC_Q2
RBIAS
PADP
PADM
RESET_N
GND Exposed
USB_DBG_DC
USB_UART_RTS
USB_UART_CTS
USB_UART_TX
USB_UART_RX
USB_DBG_DD_DIR
USB_DBG_DD
USB_JOY_MOVE
USB_EM_RESET
USB_CS
USB_SCLK
USB_MOSI
USB_MISO
USB_LCD_MODE
USB_LCD_CS
R44
R_56K_0402_F
S3
PUSH_BUTTON_SKRK
C6
C_0603
X1
X_48.000/20/35/20/18
C19
C_33P_0402_NP0_J_50
C20
C_33P_0402_NP0_J_50
R18
R_0402
R10
R_0_0402
VBUS
R9
R_1K5_0402_G
P12
USB_B
VBUS
D-
D+
GND
Shield
Shield
R12
R_33_0402_G
R11
R_33_0402_G
C2
C_47P_0402_NP0_J_50
C3
C_47P_0402_NP0_J_50
CONTRACT NO.
02587
COMPANY NAME
TI Norway, LPW
APPROVALS
DATE
DWG
USB Interface
DRAWN
PEH
CHECKED
SIZE
A3
FSCM NO.
DWG NO.
REV.
1.7.1
ISSUED
SCALE
SHEET
2(7)



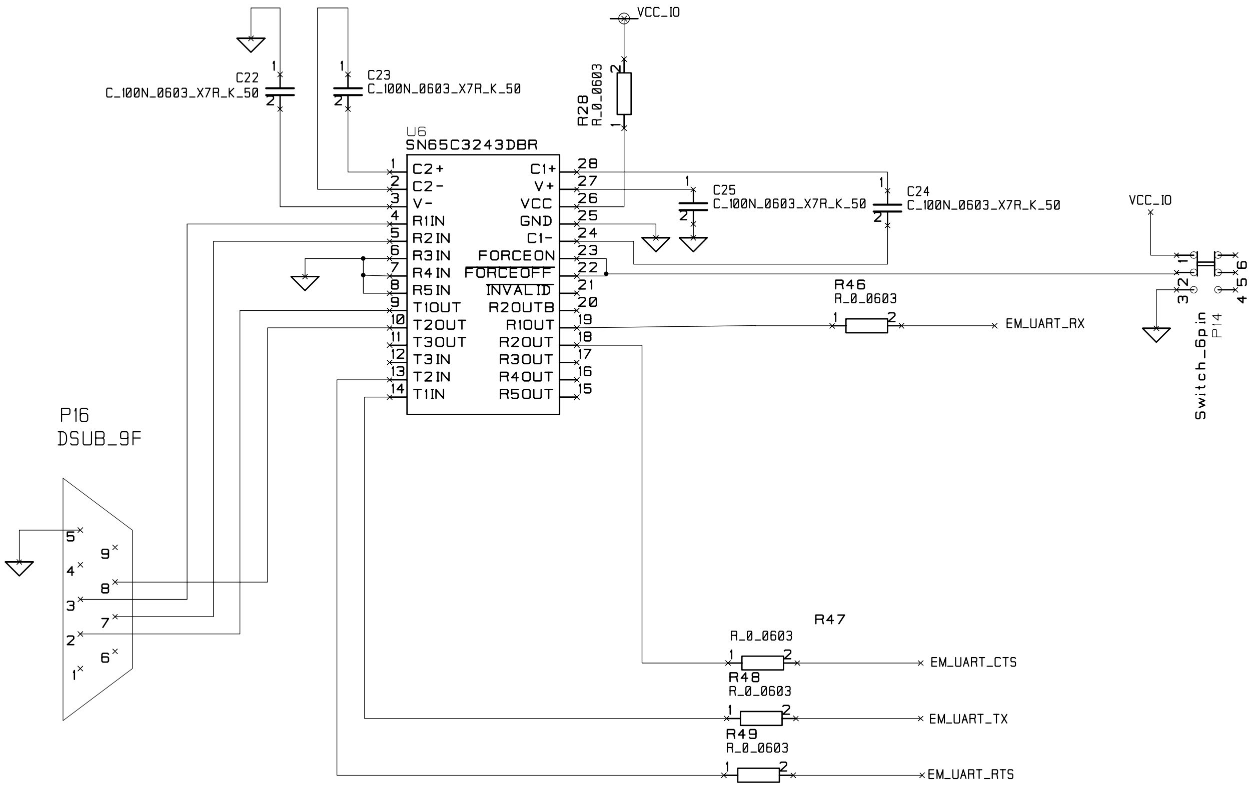
CONTRACT NO. 02587		COMPANY NAME TI Norway, LPW			
APPROVALS	DATE	DWG EM Interface			
DRAWN	PEH	SIZE A3	FSCM NO.	DWG NO.	REV. 1.7.1
CHECKED					
ISSUED		SCALE			SHEET 3(7)



CONTRACT NO. 02587		COMPANY NAME TI Norway, LPW			
APPROVALS	DATE	DWG Power supply			
DRAWN	PEH	SIZE A3	FSCM NO.	DWG NO.	REV. 1.7.1
CHECKED					
ISSUED		SCALE			SHEET 4(7)

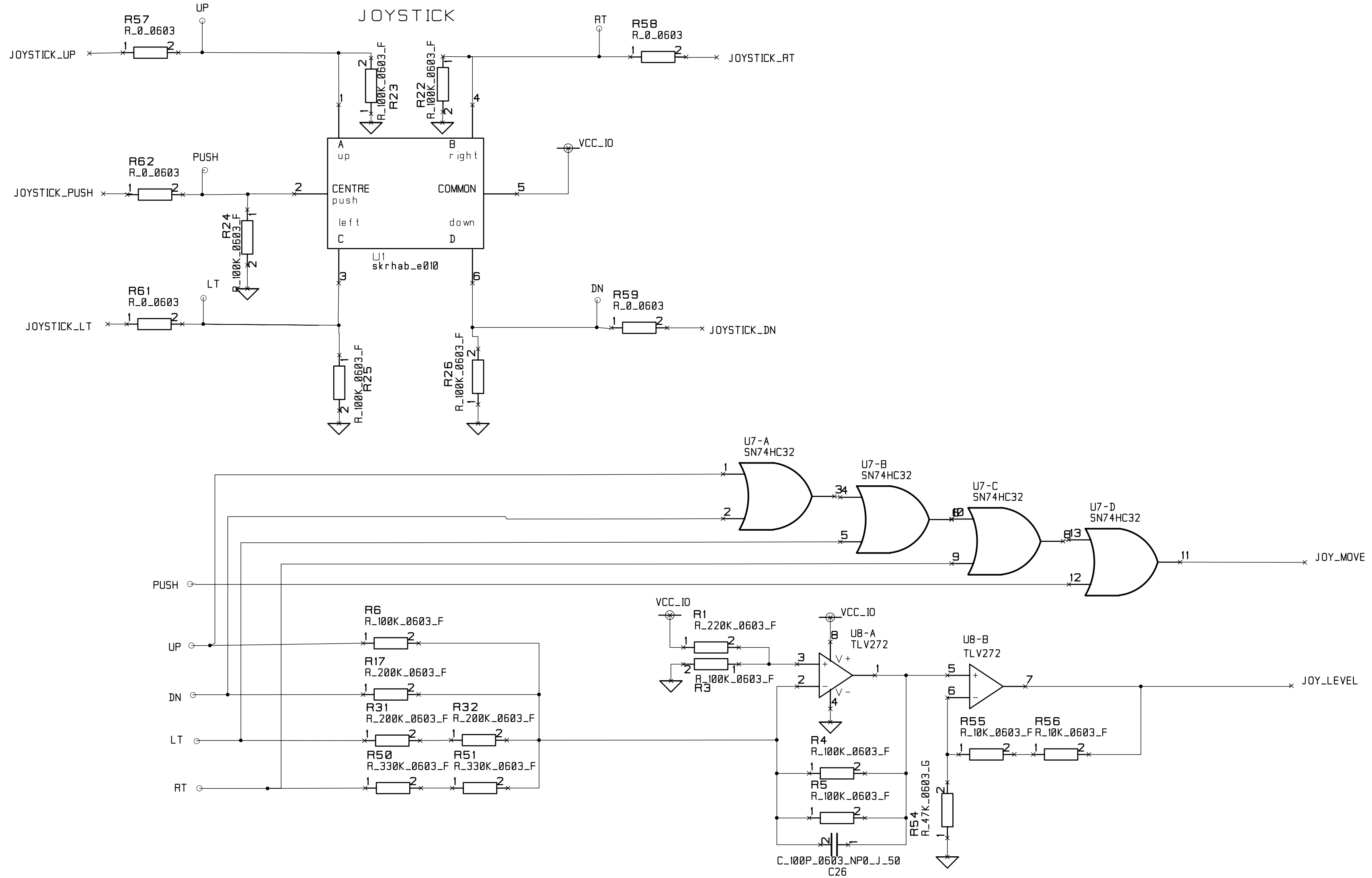


CONTRACT NO. 02587		COMPANY NAME TI Norway, LPW			
APPROVALS	DATE	DWG User Interface			
DRAWN	PEH				
CHECKED		SIZE A3	FSCM NO.	DWG NO.	REV. 1.7.1
ISSUED		SCALE		SHEET 5(7)	



PC RS232-port
2-RXD
3-TXD
5-GND
7-RTS
8-CTS

CONTRACT NO. 02587		COMPANY NAME TI Norway, LPW			
APPROVALS	DATE	DWG RS-232 Interface			
DRAWN	PEH	SIZE A3	FSCM NO.	DWG NO.	REV. 1.7.1
CHECKED					
ISSUED		SCALE		SHEET 6(7)	



CONTRACT NO. 02587		COMPANY NAME TI Norway, LPW			
APPROVALS	DATE	DWG Joystick			
DRAWN	PEH	SIZE A3	FSCM NO.	DWG NO.	REV. 1.7.1
CHECKED					
ISSUED		SCALE			SHEET 7(7)