



CC Debugger User's Guide



SWRU197E

Table of Contents

1	Introduction	3
2	Abbreviations and Acronyms.....	3
3	Box Contents	4
4	Operating Conditions of the CC Debugger.....	4
5	Initial Steps.....	5
5.1	<i>Installing the USB driver</i>	5
5.2	<i>Supported PC Tools.....</i>	5
6	Connecting the CC Debugger to the Device.....	6
6.1	<i>Target Connector Details</i>	6
6.2	<i>Connecting the CC Debugger to a System on Chip</i>	8
6.2.1	Minimum connection for debugging.....	8
6.2.2	Minimum connection for SmartRF Studio	8
6.2.3	Minimum connection for SmartRF Packet Sniffer	9
6.3	<i>Connecting the CC Debugger to a Transceiver</i>	10
6.4	<i>Connecting the CC Debugger to a CC85xx.....</i>	12
7	Using the CC Debugger	13
7.1	<i>Understanding the LED.....</i>	13
8	Updating the Firmware.....	14
8.1	<i>Updating the firmware automatically in SmartRF Studio</i>	14
8.2	<i>Updating the firmware manually in SmartRF Flash Programmer</i>	16
8.3	<i>Forced boot recovery mode</i>	17
8.4	<i>Resurrecting the CC Debugger.....</i>	17
9	Troubleshooting	20
10	Schematics.....	21
11	References	21
12	Document History.....	22

1 Introduction

The CC Debugger is primarily used for debugging and programming the flash on CCxxxx 8051-based System-on-Chip (SoC) devices from Texas Instruments. The PC tools available for these purposes are the SmartRF™ Flash Programmer from Texas Instruments and IAR Embedded Workbench® for 8051 from IAR Systems.

When connected to the debugger, the SoC devices can be controlled directly from SmartRF™ Studio. SmartRF Studio will also be able to control supported CCxxxx RF transceivers (CC2520, CC112x, and CC11xL) when they are connected to the debugger as explained in chapter 6.3.

In addition, CC Debugger is used for configuring the CC85xx devices with PurePath Wireless Configurator.

2 Abbreviations and Acronyms

CSn	Chip Select (active low)
DC	Debug Clock
DD	Debug Data
DUT	Device Under Test
GND	Ground
LED	Light Emitting Diode
MISO	Master In Slave Out
MOSI	Master Out Slave In
RF	Radio Frequency
SCLK	Serial Clock
SoC	System on Chip
SPI	Serial Peripheral Interface
USB	Universal Serial Bus
Vdd	Positive voltage on target

3 Box Contents

- 1 CC Debugger
- 1 USB-A to Mini-B USB cable
- 1 10-pin flat cable with 2x5 2.54 mm connector
- 1 10-pin flat cable with 2x5 1.27 mm connector
- 1 Converter board 2.54mm – 1.27 mm connector
- Documentation

4 Operating Conditions of the CC Debugger

Minimum target voltage:	1.2 Volt
Maximum target voltage:	3.6 Volt
Operating temperature:	0°C to 85°C
Regulated voltage on CC Debugger:	3.3 Volt
Maximum target current draw (*):	200 mA (*)
Supported Operating Systems:	Microsoft® Windows® 2000
	Windows XP SP2/SP3 (32 bit versions)
	Windows Vista® (32 & 64 bit)
	Windows 7 (32 & 64 bit)

(*) Only applicable if the target is powered from the CC Debugger

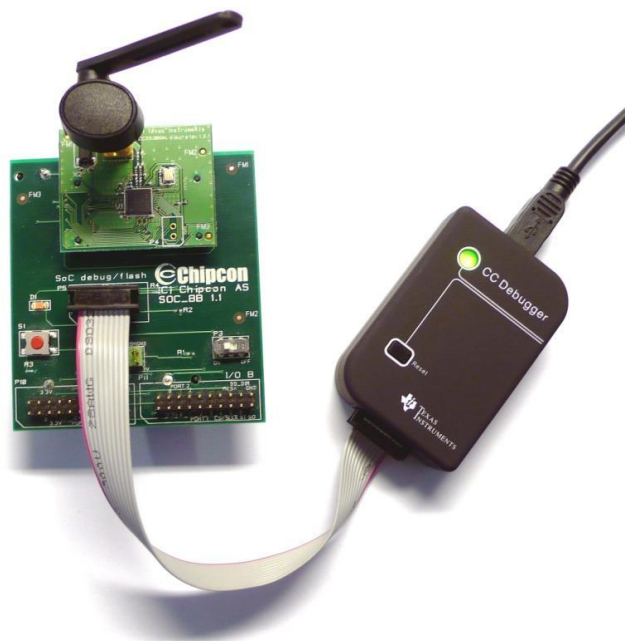


Figure 1 - CC Debugger connected to a SoC Battery Board with a CC2530EM

5 Initial Steps

5.1 Installing the USB driver

To get the required USB driver for the CC Debugger, it is necessary to install one of the tools listed below:

- SmartRF Studio www.ti.com/tool/smartrfstudio
- SmartRF Flash Programmer www.ti.com/tool/flash-programmer
- PurePath Wireless Configurator <http://www.ti.com/tool/purepath-wl-cfg>

Alternatively, you can download “Cebal – CCxxxx Development Tools USB Driver for Windows x86 and x64” [3] which is a standalone installer including only the device driver.

After having installed the driver, connect the CC Debugger to the PC. The USB driver will be installed automatically. You can quickly check that the debugger has been associated correctly with the USB device driver by opening the Windows Device Manager. The debugger should appear as a “Cebal controlled device”.



Figure 2 - Verify correct driver installation

For further details or troubleshooting the driver installation process, please refer to “DN304 – CCxxxx Development Tools USB Driver Installation Guide” [4].

5.2 Supported PC Tools

Currently, the CC Debugger can be used together with the following PC Tools

- **IAR Embedded Workbench for 8051** In circuit debugging of system-on-chips
- **SmartRF Flash Programmer** Flash programming of system-on-chips
- **SmartRF Studio** RF testing of radio devices (transceivers and SoCs)
- **SmartRF Packet Sniffer** Packet sniffing with selected radio devices
- **PurePath Wireless Configurator** Programming of CC85xx devices

The debugger will operate as the interface between the RF device and the tools listed above. Please ensure correct connection between the device and CC Debugger before starting to use the tools.

The connection of the device to the CC Debugger will be covered in the next chapter.

6 Connecting the CC Debugger to the Device

6.1 Target Connector Details

The target connector, located on the lateral side of the debugger, is a 10-pin 2x5 2.54 mm pitch connector with a direction coded plastic guide. A suggested matching (male) surface mounted connector would be BB02-HP from GradConn.

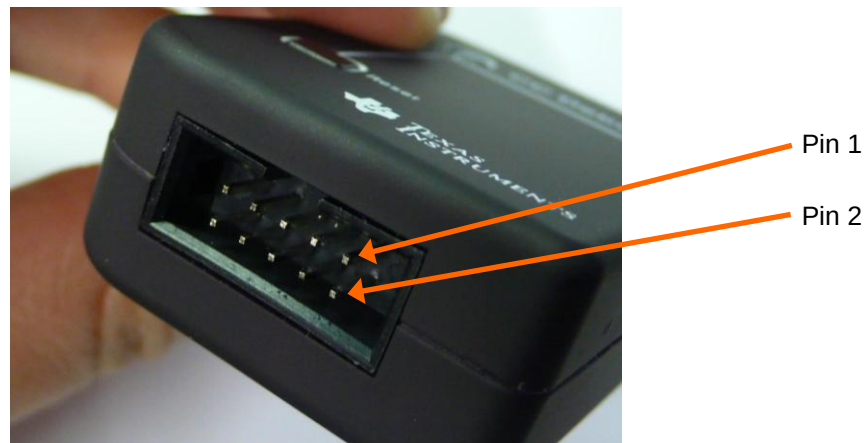


Figure 3 - Placement of Target Connector Pins

The adapter board, which has a 10-pin 2x5 1.27 mm pitch connector, has the same pin placement. A suggested matching (male) surface mounted connector would be STL21-A01VXX from MPE Garry.

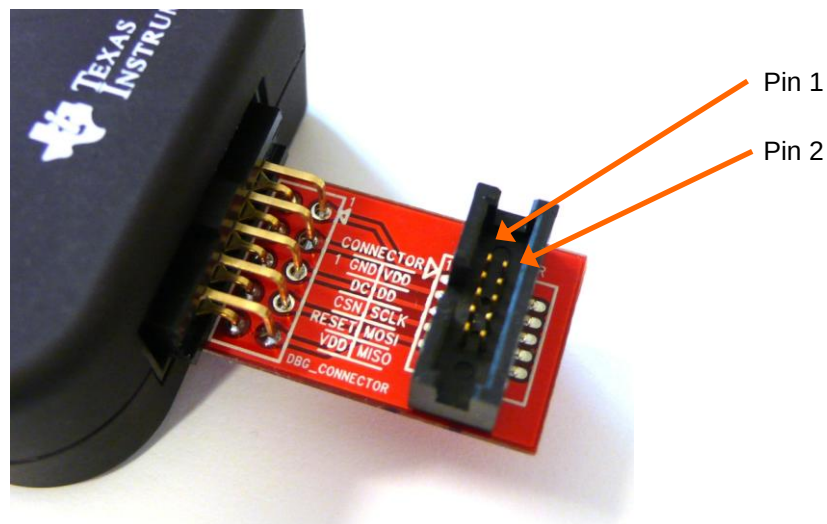


Figure 4 - Placement of Target Connector Pins on Adapter Board

The pin-out of the target connector is shown in Figure 5. Note that not all of these pins need to be connected to the target device for programming and debugging. Only Vdd, GND, DD, DC and RESET are required for System on Chips. The other pins are optional and/or for special features.

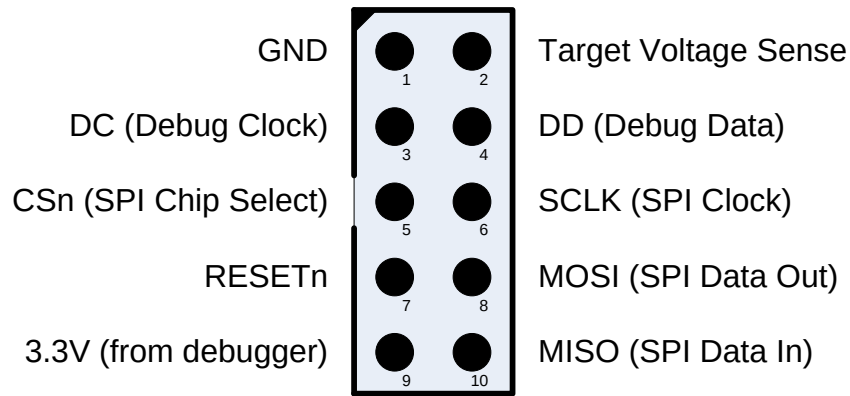


Figure 5 - Target Connector Pin-out

Please note the concept with the target voltage sense signal. This signal is used by the level converters on the CC Debugger to handle different voltage levels on the target board and the debugger. Pin 2 on the target connector must be connected to Vdd on the target board.

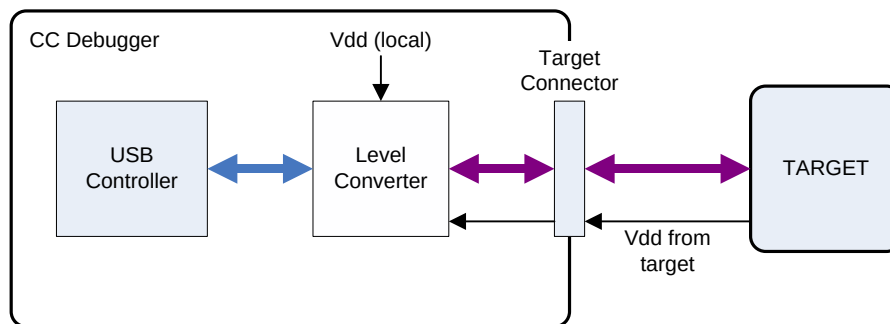


Figure 6 - Voltage from target to CC Debugger

Alternatively, it is possible to power the target by connecting pin 9 to Vdd on the target. In that case, the CC Debugger will supply 3.3V to the target.

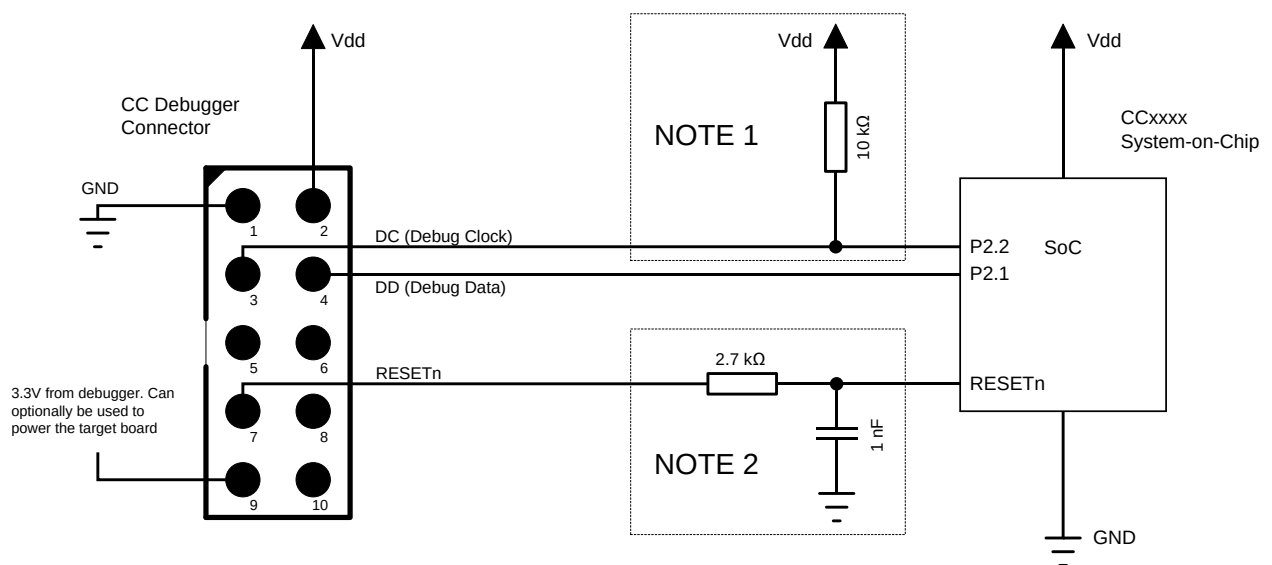
6.2 Connecting the CC Debugger to a System on Chip

6.2.1 Minimum connection for debugging

For successful debugging of a TI 8051-based RF System on Chip, connect the two debug signals Debug Data (DD) and Debug Clock (DC) and the reset signal RESETn to the device. In addition, the CC Debugger must be connected to GND and Vdd on the board. Vdd is used as an input to the level shifters on the CC Debugger, thus allowing a different operating voltage on the target than internally on the debugger.

The DD signal should be connected to pin P2.2 and DC to pin P2.1 on the SoC – with the exception of CC2544, where DD is on P1.3 and DC on P1.2. Please refer to the device data sheets for details on the pin-out.

Note that it is possible to power the target board from the debugger by connecting the 3.3V signal on pin 9 on the connector to the target board.



Note 1: Some early revisions of certain SoCs (CC2430, CC2510 and CC1110) needed an external pull-up to avoid unwanted transitions on the debug clock line during chip reset – thus inadvertently setting the device in debug mode. All new revisions of all SoCs now have an internal pull-up on P2.2, so this external component is not required.

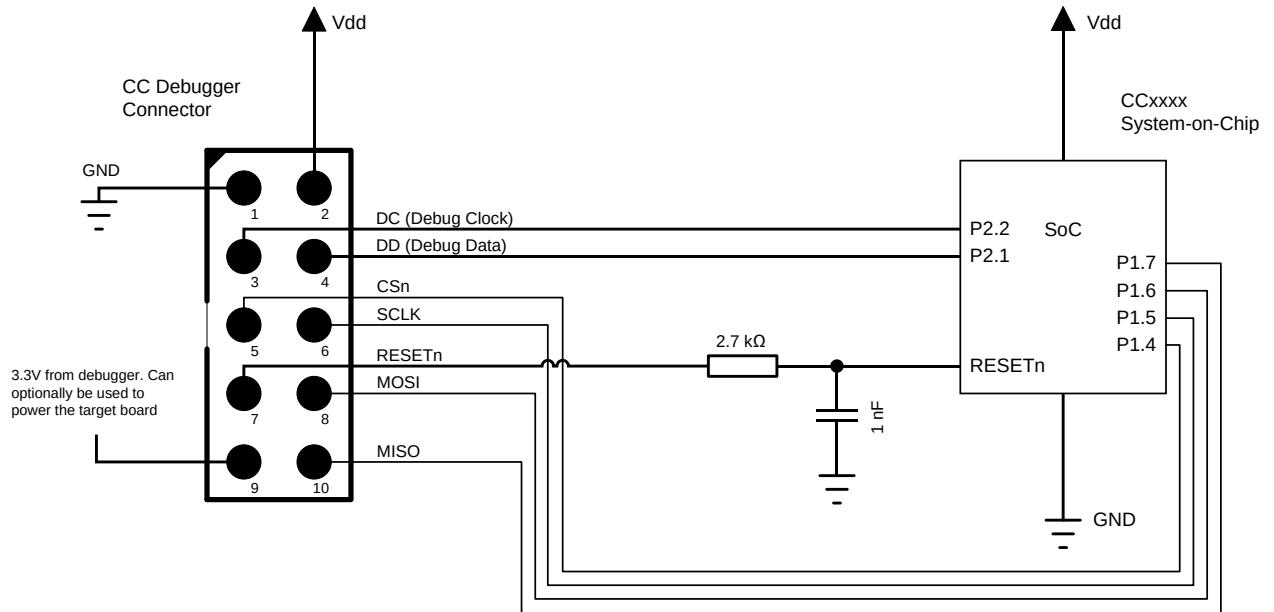
Note 2: The RESETn pin is sensitive to noise and can cause unintended reset of the chip. For reset lines susceptible to noise, it is recommended to add an external RC filter. Please refer to the respective SoC datasheet and reference designs for recommended RESET circuitry. The CC Debugger supports slow transitions on the reset line, using a 2 ms delay between any transition on the RESET line and other transitions on the DC and/or DD lines.

6.2.2 Minimum connection for SmartRF Studio

Use the same connection as for debugging the SoC.

6.2.3 Minimum connection for SmartRF Packet Sniffer

In order to use the packet sniffer capabilities of the CC Debugger, it is also necessary to connect the SPI bus to the SoC. The SPI interface is used by the CC Debugger for reading the captured RF packets from the SoC.



Note that the packet sniffer will overwrite the flash of the SoC with special packet capture firmware.

Note concerning the SPI interface to the SoC

All of the current TI RF SoCs can be configured to operate as SPI slaves, with the SPI signals (CS, SCLK, MISO and MOSI) going to one of the USART peripherals. The packet sniffer application will program the SoC with firmware that configures one of the USART peripherals in order to communicate with the CC Debugger. The firmware can use any of the four possible pin configurations (USART 0 or 1, pin out alternative 1 or 2). However, only a subset is currently supported:

	USART0, alt 1	USART0, alt 2	USART1, alt 1	USART1, alt 2
CC243x	-	-	-	supported
CC253x/CC254x	-	-	-	supported
CC111x	supported	-	-	supported
CC251x	supported	-	-	supported

Table 1 - Supported SPI connections (marked OK)

	USART0, alt 1	USART1, alt 2
SCLK	P0.5	P1.5
CS	P0.4	P1.4
MOSI	P0.3	P1.6
MISO	P0.2	P1.7

Table 2 - USART pin out details

In case of multiple supported interfaces, the Packet Sniffer application will let you choose which interface to use.

6.3 Connecting the CC Debugger to a Transceiver

The SPI interface on the CC Debugger can be used to interface many of the CCxxxx transceivers and control them from SmartRF Studio. The transceivers/transmitters/receivers currently supported are:

- CC1120
- CC1121
- CC1125
- CC1175
- CC110L
- CC113L
- CC115L
- CC2520

Note that the CC Debugger operates as the SPI Master. In a multi master system, it is necessary to make sure the debugger output signals (DC, DD, CSn, SCLK, MOSI and RESETn) do not interfere with the other SPI master on the board. The other SPI master would typically be the microcontroller on the board.

The connection diagrams below show the interconnection between the debugger and the various supported transceivers.

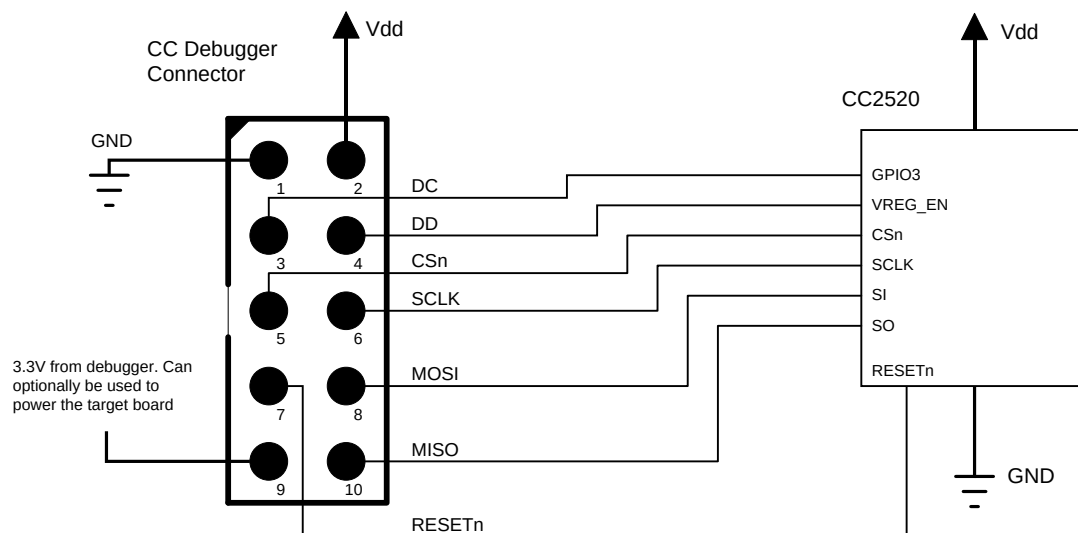


Figure 7 - CC Debugger connected to CC2520

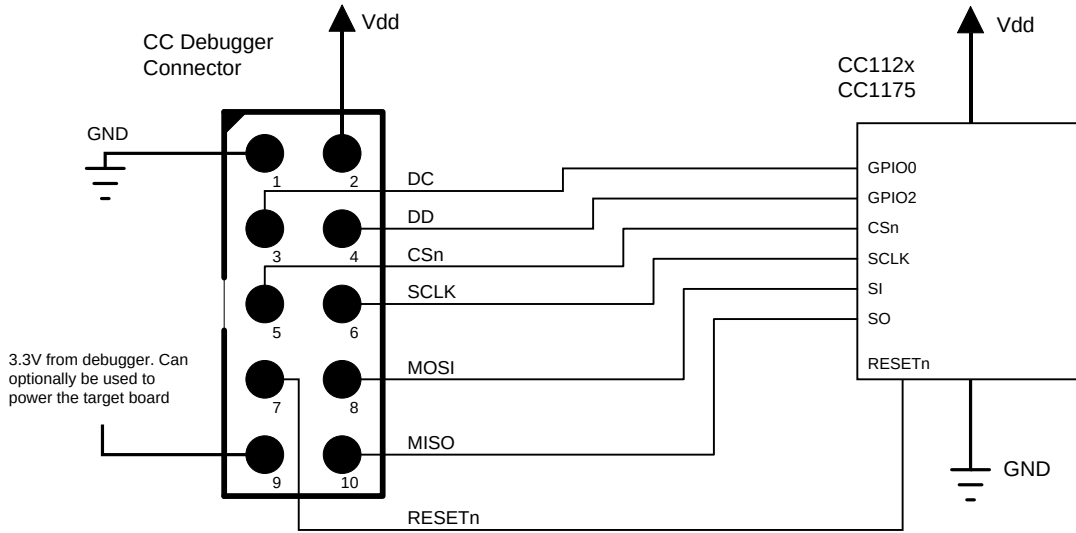


Figure 8 - CC Debugger connected to CC112x/CC1175

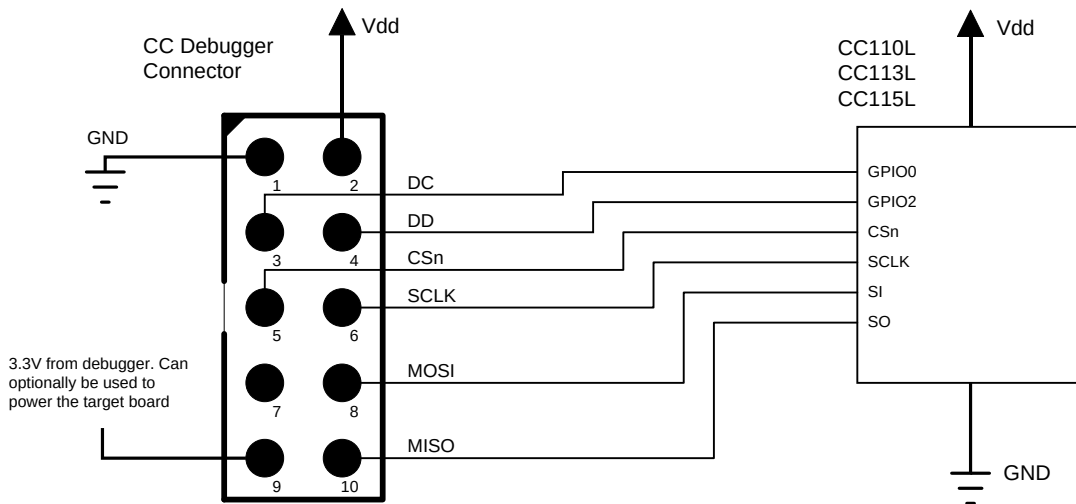


Figure 9 - CC Debugger connected to CC11XL

6.4 Connecting the CC Debugger to a CC85xx

In order to configure the CC85xx devices (i.e. program the flash on the device) with PurePath Wireless Configurator, the device's SPI interface must be connected to the CC Debugger as shown in the figure below.

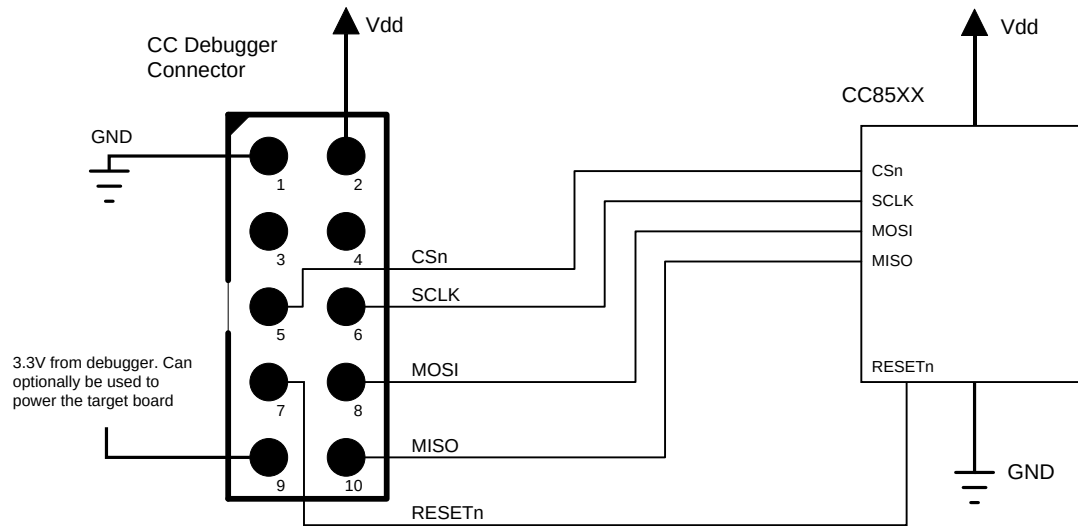


Figure 10 - CC Debugger connected to CC85XX

7 Using the CC Debugger

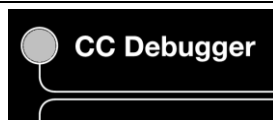
After having connected the debugger to the target device, the debugger can be powered up by plugging in the USB cable.

The debugger will immediately start a device detection process, looking for all known devices. If no devices are detected, the LED will be RED. If a device is detected, the LED will be GREEN.

If the LED is GREEN, it is possible to start using the debugger together with one of the supported PC tools.

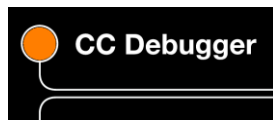
7.1 Understanding the LED

OFF



The debugger has no power or there is no valid firmware on the debugger. Make sure the debugger is properly powered via the USB cable or try to resurrect the debugger using the method described in chapter 8.4.

AMBER (BOTH LEDS ON)



The debugger is powered, but there is no valid firmware. Try to resurrect the debugger using the method described in chapter 8.4.

RED LED BLINKING

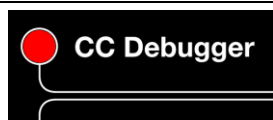


The Debugger is in Boot Recovery Mode.

The debugger will briefly enter this state while the firmware is being upgraded (see chapter 8). The board might also enter this state if the firmware is corrupt or if the user has manually forced the board to start up in the special "boot recovery mode" (section 8.3).

To go out of the state, reset the debugger by pressing the "Reset" button or by power-cycling the device. If the LED is still blinking, reprogram the unit by using the Flash Programmer Application.

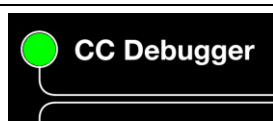
RED LED ON



No device detected. This might be due to old firmware on the CC Debugger. New devices might not be supported with the current firmware on the debugger. Please refer to chapter 8 for the firmware upgrade procedure.

There might also be a problem with the hardware connection. Check the connection to device and make sure the target board is properly powered and that Vdd on the target board is connected to pin 2 on the debug connector. Press and release the reset button to retry the target device detection

GREEN LED ON



The target device has been properly detected. It is possible to start using the supported tools (see chapter 5.2).

8 Updating the Firmware

In order to make sure the CC Debugger works seamlessly with your device, it is important that it has the latest and greatest firmware. This chapter will describe how you can upgrade the firmware automatically from SmartRF Studio or manually from SmartRF Flash Programmer. The chapter will also describe how to resurrect a seemingly broken debugger.

8.1 Updating the firmware automatically in SmartRF Studio

Updating the firmware on the CC Debugger can be done automatically by SmartRF Studio. Please follow the few steps described below.

1. Start SmartRF Studio.
2. Disconnect the debugger from any target board, and connect it to the PC via the USB cable. The debugger will appear in the list of connected devices in the lower part of the SmartRF Studio startup panel.

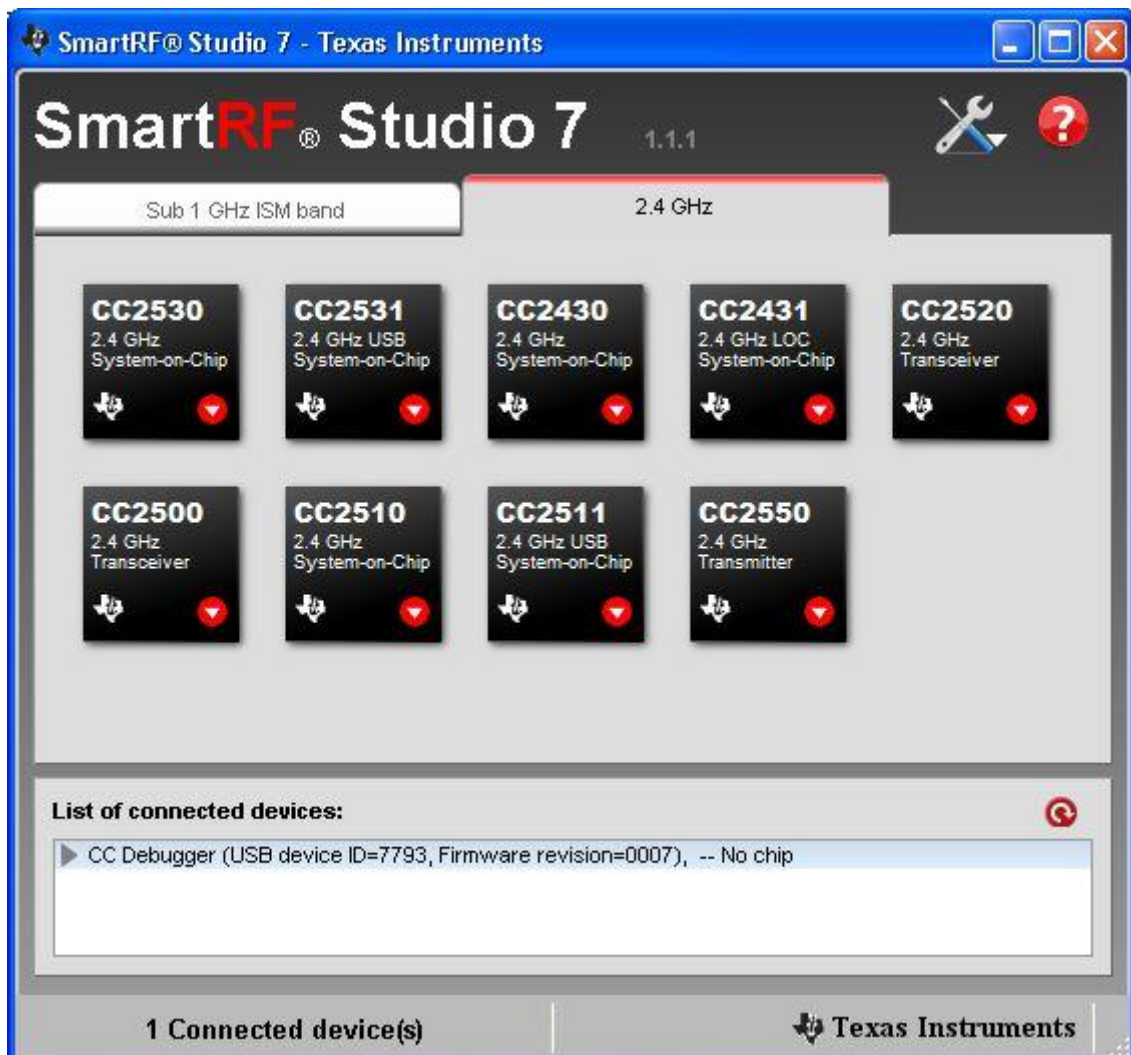


Figure 11 - Auto FW upgrade

3. Double click on the item in the list, and a new window will appear.



Figure 12 - Auto FW upgrade

4. Click "Yes" and let SmartRF Studio do the rest.



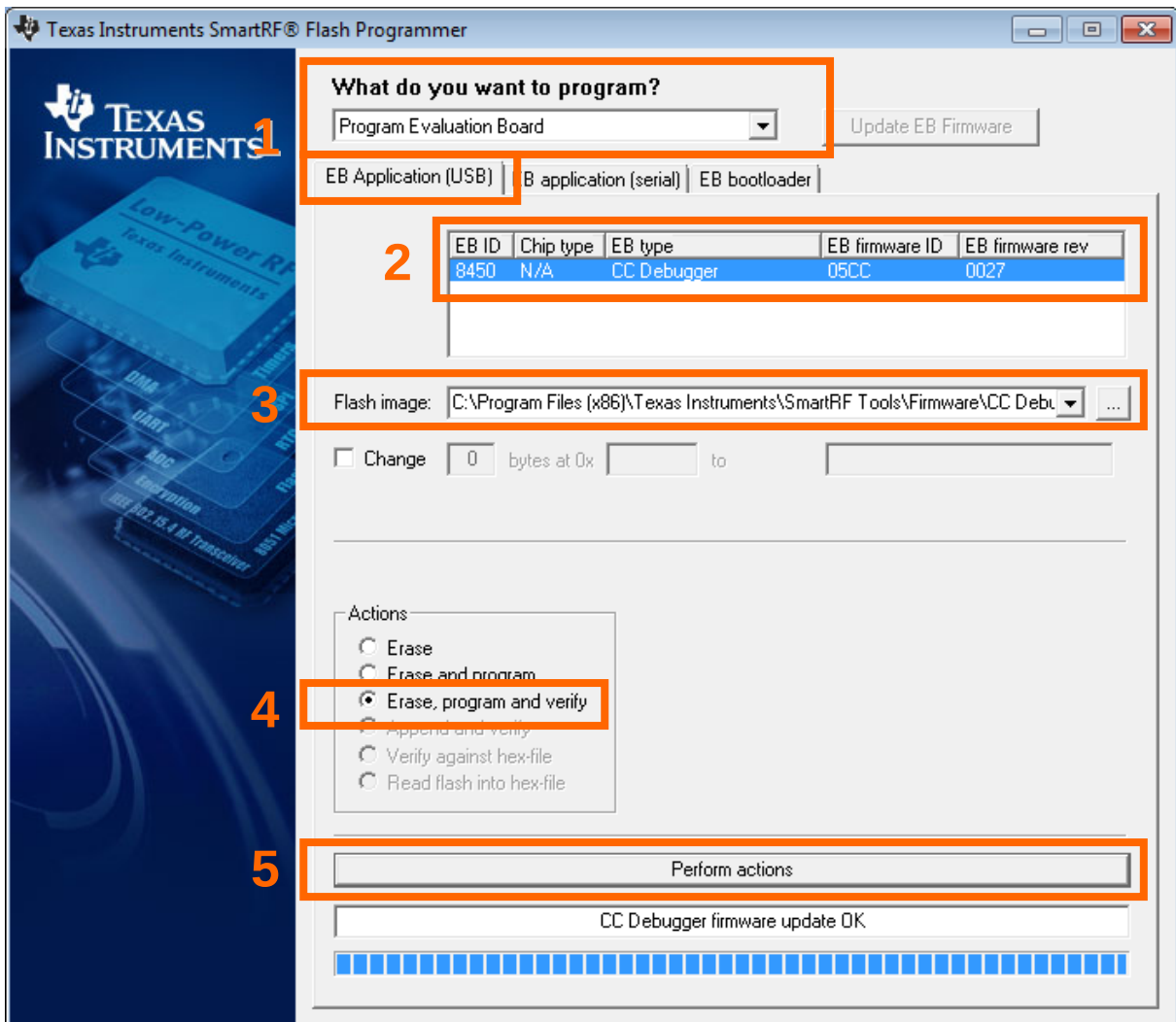
Figure 13 - Auto FW upgrade

5. Click "Done" and you're good to go. The device should appear in the list of connected devices, now showing the new firmware revision.

8.2 Updating the firmware manually in SmartRF Flash Programmer

You can also update the firmware manually using SmartRF Flash Programmer. You can use this method if you like to have full control of the firmware image to be programmed on the controller of the debugger (i.e. programming custom firmware or old firmware revisions).

1. Start SmartRF Flash Programmer and select the tab called “EB application (USB)”. This tab will let you program compatible firmware on the CC Debugger (or evaluation boards) via the USB interface (i.e. no external programming device required).
2. Disconnect the debugger from any target board, and connect it to the PC via the USB cable. The debugger will appear in the list of connected devices. Chip type will be listed as N/A.
3. Select the flash image you want to program on the debugger. Normally, you would select: C:\Program Files (x86)\Texas Instruments\SmartRF Tools\Firmware\CC Debugger\cebal_fw_srf05dbg.hex¹
4. Select the action “Erase, program and verify”
5. Click the “Perform actions” buttons. The programming procedure will start. Note that this will take several seconds.
6. The CC Debugger will reappear in the list of connected devices, now showing the new firmware revision in the device list.
7. Done!



¹ Assuming default installation path of SmartRF Flash Programmer.

8.3 Forced boot recovery mode

If, for some reason, the firmware update fails and the CC Debugger appears to be non responsive, there is a way to force the board to only run the bootloader and stop all further execution. In this mode, no attempts will be made to start the firmware, and the board will only allow the user to perform a new firmware upgrade over USB.

Disconnect the debugger from any power source and open the plastic enclosure.

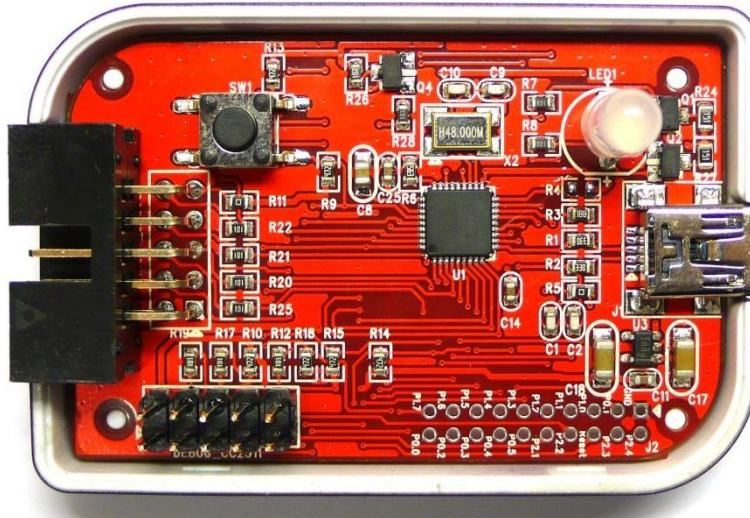


Figure 14 - Internal view of CC Debugger

Short circuit the pins as depicted in Figure 15: P1.6 on the CC2511 must be connected to GND during the power-on reset to enter boot recovery mode.

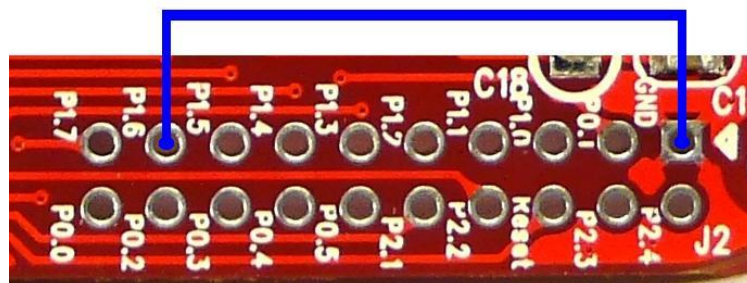


Figure 15 - Short-circuit pins for boot recovery mode

When reconnecting the USB cable, the LED will start to blink with a RED light. This indicates that the bootloader is running and that the debugger is in boot recovery mode.

At this point, follow the same firmware programming steps as describe at the beginning of this chapter.

Please also note that the boot recovery mode can be used as a check to verify that the bootloader on the debugger is working.

8.4 Resurrecting the CC Debugger

If the CC Debugger appears to be completely dead when applying power, there is a way to “unbrick” the board. The method consists of reprogramming the bootloader on the debugger using the debug connector inside the box. This will require an extra programming device.

When opening the box, locate the debug connector header next to the target connector. Connect this header to another CC Debugger (see Figure 16) or to a SmartRF05EB (see Figure 17). When using

SmartRF05EB, connect a 10-pin flat cable from the “Ext SoC Debug” plug (P3) on the EB to the “USB Debug” plug (P2) on the CC Debugger. The dead debugger needs power, so connect the USB cable. Turn on the SmartRF05EB or debugger - it should detect the USB Controller (CC2511) on the debugger.



Figure 16 - Programming the bootloader on the CC Debugger using another CC Debugger

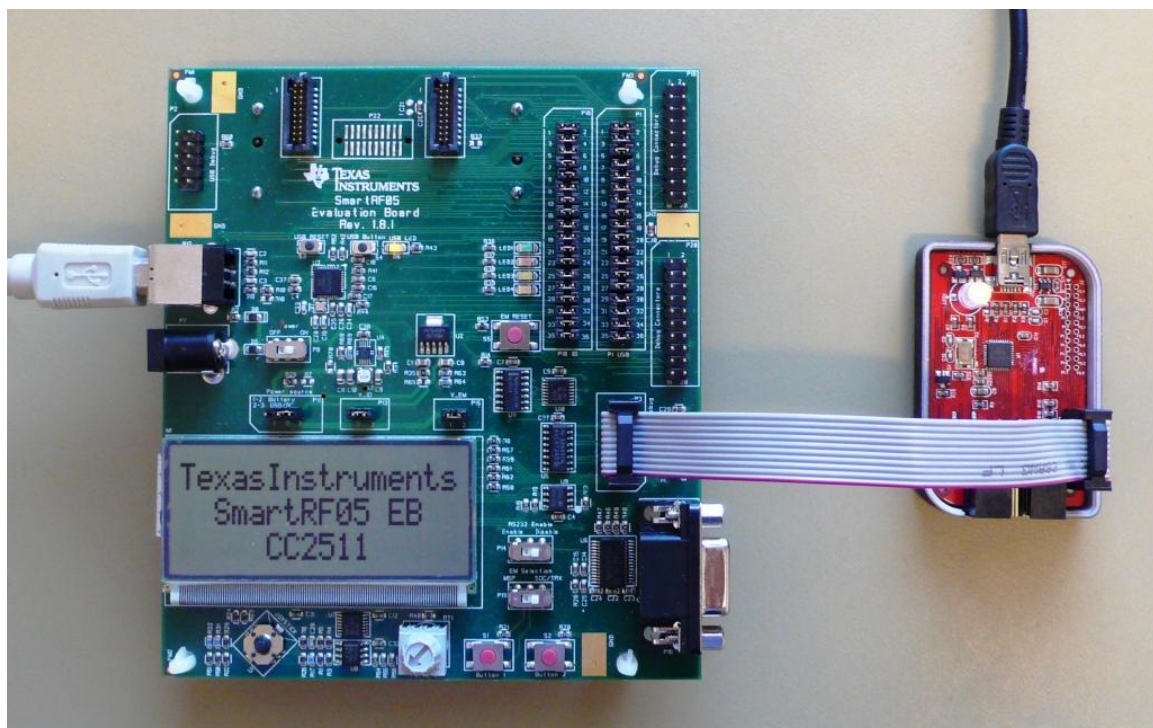


Figure 17 - Programming the bootloader on the CC Debugger using SmartRF05EB

Next, use the SmartRF Flash Programmer to program the bootloader on the debugger. Follow these five steps (illustrated in Figure 18 below):

1. After starting the application, first select "Program Evaluation Board" in the "What do you want to program?" drop down box, then select the "EB Bootloader" tab.
2. In the upper left corner, select device: Use SmartRF05EB regardless of the device being used to program the debugger. I.e. select SmartRF05EB both when you are using a CC Debugger and when you are using a SmartRF05EB for the resurrection.
3. Next, select which flash image to program. The bootloader image is included when installing the flash programmer and it is usually located at "C:\Program Files (x86)\Texas Instruments\SmartRF Tools\Firmware\CC Debugger".
4. It is also necessary to give the debugger a unique ID number – any 4 digit number will work. This number is used by the driver on the PC to uniquely identify devices if more than one debugger is connected at the same time.
5. Select "Erase, program and verify"
6. Press the "Perform Actions" buttons. The firmware upgrade takes a few seconds.

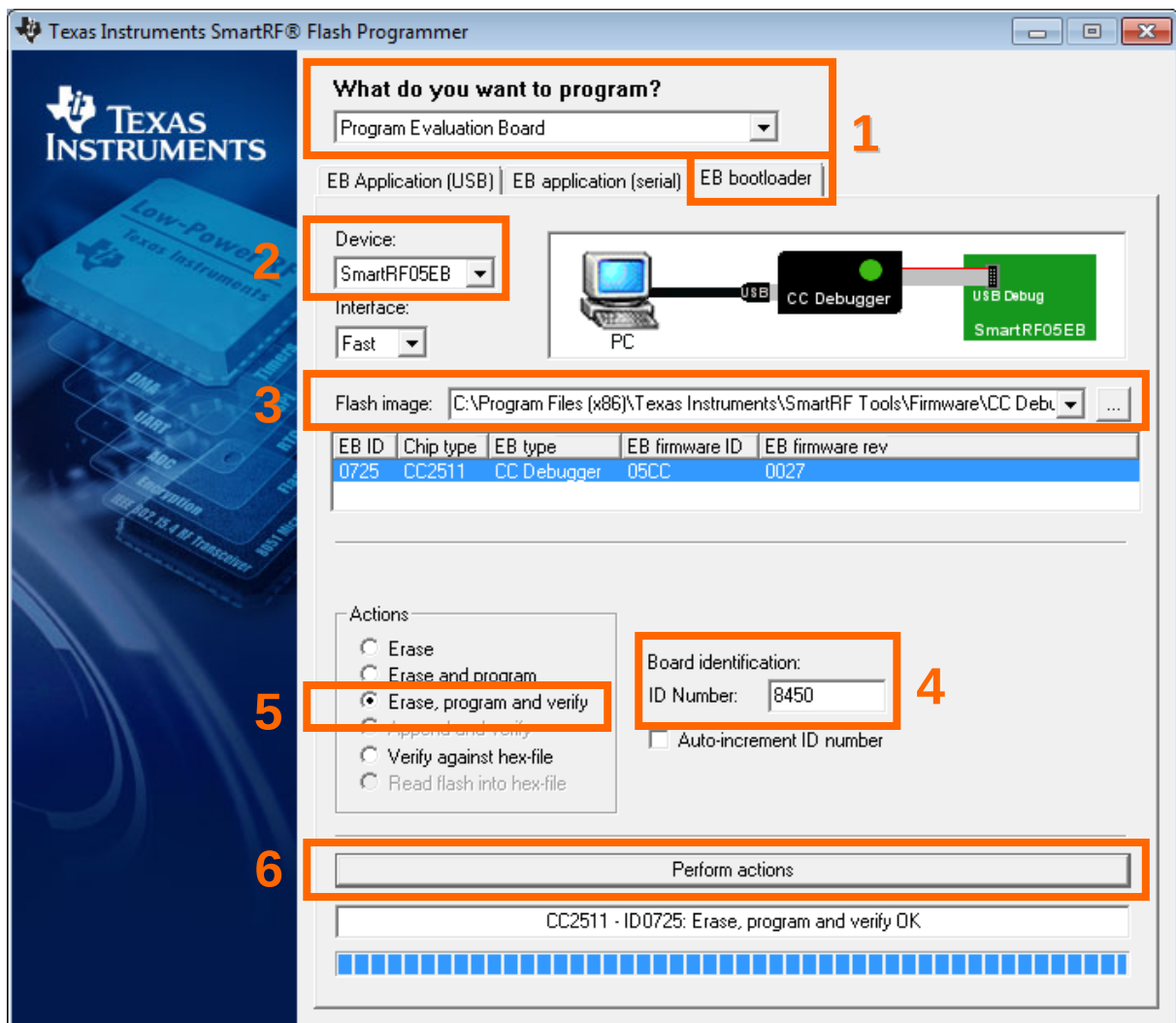


Figure 18 - SmartRF Flash Programmer - Updating the bootloader

Once the bootloader is programmed, you might be asked to install a USB driver on the PC. Just follow the same procedure as when the debugger was connected to the PC the first time (see chapter 5).

The RED LED on the debugger should now be blinking, indicating that the bootloader is running but that no application has been loaded. If the RED LED is off, there is probably something wrong with the hardware. The debugger firmware can now be programmed directly over USB by following the procedure in either chapter 8.1 or 8.2.

9 Troubleshooting

Q1 Help! The debugger does not detect the SoC. What should I do?

A1 There are several things to check.

Upgrade the firmware. Many CC Debuggers have old firmware that will not automatically detect newer devices, like CC2540 and CC2541. These devices need firmware version 13 or newer. Refer to chapter 8 for further instructions.

Check that the cable is oriented correctly and that the pins are connected to the right signals on the debugger.

Check that the debugger gets power from the target (i.e proper connection of the Target Voltage Sense signal). This is required in order for the level converters on the debugger to work.

Check that ground on the target is connected to ground on the debugger. This is normally achieved through the target connector. Note that since the ground planes are the same, please be aware of any adverse effects caused by different ground planes on the target and on the PC (grounded via USB cable).

Check that the cable is not broken. Especially the small flat cable is prone to stop working if handled a lot or being bent and stretched beyond normal operating conditions.

Q2 Does IAR EW8051 support the CC Debugger as debugging device?

A2 Yes – but make sure you have an up to date version of IAR with the new debug driver plug-in from Texas Instruments. You will need version 7.51A or higher.

Q3 Can the debugger be used as an interface to the RF device for packet sniffing?

A3 Yes, this is supported for selected devices. Use the same interconnection as in the diagrams in chapter 6.

Q4 Is there a way to remove the plastic casing without damaging it?

A4 Yes, there is. Hold the bottom piece of the plastic in one hand. With your other hand, take a firm grip on the long lateral sides of the upper part of the plastic and squeeze while moving the upper part away from the bottom. The two parts should separate from each other.

To reassemble the plastic, just click the two pieces together.

Q5 Is this a Mini or a Micro USB plug?

A5 Mini USB type A.

10 Schematics

See next page.

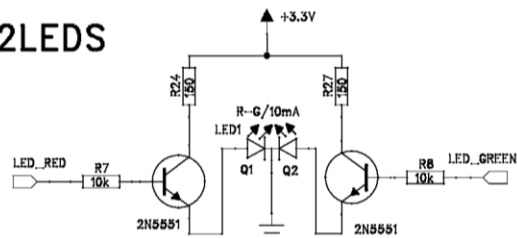
11 References

- [1] CC-Debugger product web site
www.ti.com/tool/cc-debugger
- [2] CC-Debugger Quick Start Guide
www.ti.com/lit/swru196
- [3] Cebal – CCxxxx Development Tools USB Driver for Windows x86 and x64
www.ti.com/lit/zip/swrc212
- [4] DN304 – CCxxxx Development Tools USB Driver Installation Guide
www.ti.com/lit/swra366
- [5] Texas Instruments Support
support.ti.com
- [6] Texas Instruments Low Power RF Online Community
www.ti.com/lprf-forum
- [7] SmartRF Studio
www.ti.com/tool/smartrfstudio
- [8] SmartRF Flash Programmer
www.ti.com/tool/flash-programmer
- [9] SmartRF Packet Sniffer
www.ti.com/tool/packet-sniffer
- [10] SmartRF Flash Programmer User Manual
www.ti.com/lit/swru069
- [11] PurePath Wireless Configurator
www.ti.com/tool/purepath-wl-cfg
- [12] SoC Battery Board product web site
www.ti.com/tool/soc-bb
- [13] IAR Embedded Workbench for 8051
www.iar.com/ew8051

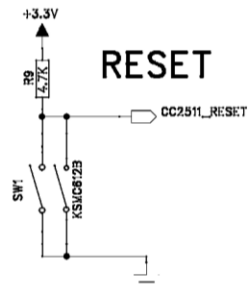
12 Document History

Revision	Date	Description/Changes
E	2012-03-01	Corrected typo in chapter 6.2.1. Special debug pin-out for CC2544, not CC2543.
D	2012-02-22	Added information about connections for programming of CC85xx devices. Updated info about connections for supported transceivers. Updated driver installation information and added more details about firmware upgrade. Describe what it means when the LED is amber. Updated reference links.
C	2010-09-19	Added more information about how to upgrade the firmware.
B	2010-02-25	Fixed erroneous description of interconnection between CC Debugger and CC2520. The VREG_EN signal shall be connected to pin 4 on the target connector, not pin 3.
A	2010-02-11	Added more details about the powering options. Added more information about connection options.
-	2009-05-05	First revision.

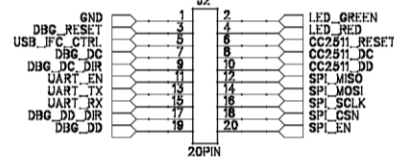
2LEDS



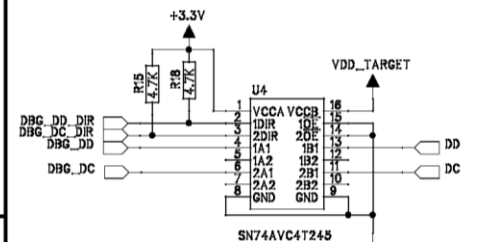
RESET



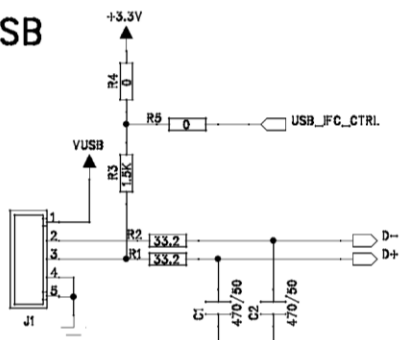
Debug Header



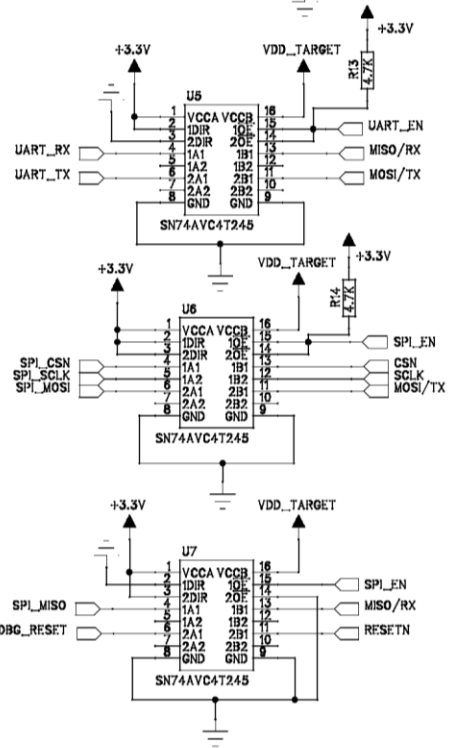
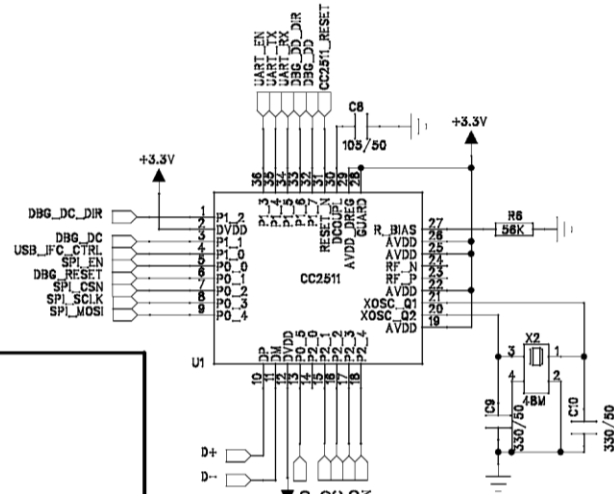
LEVEL CONVERTER



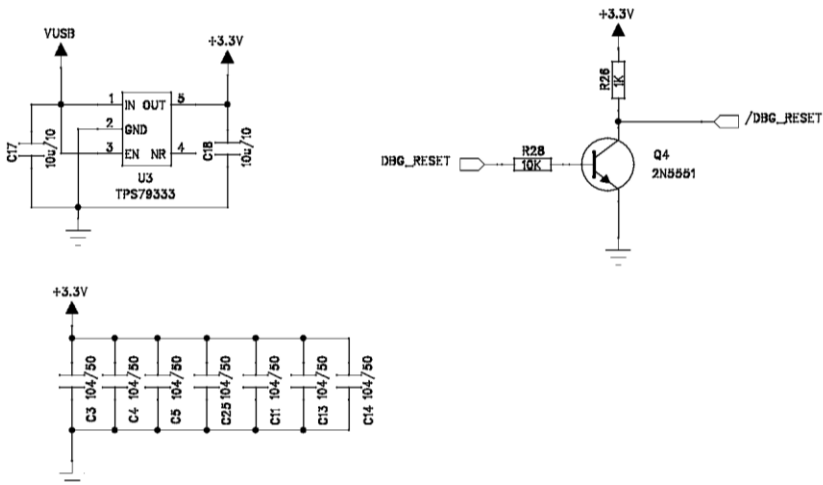
USB



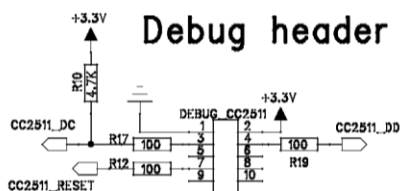
USB controller(CC2511)



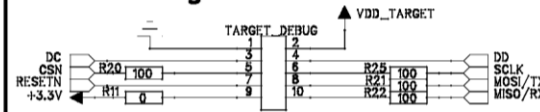
POWER



Debug header



Target Connector



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 TI 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. Information of third parties may be subject to additional restrictions.

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.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products

Audio	www.ti.com/audio
Amplifiers	amplifier.ti.com
Data Converters	dataconverter.ti.com
DLP® Products	www.dlp.com
DSP	dsp.ti.com
Clocks and Timers	www.ti.com/clocks
Interface	interface.ti.com
Logic	logic.ti.com
Power Mgmt	power.ti.com
Microcontrollers	microcontroller.ti.com
RFID	www.ti-rfid.com
OMAP Mobile Processors	www.ti.com/omap
Wireless Connectivity	www.ti.com/wirelessconnectivity

Applications

Automotive and Transportation	www.ti.com/automotive
Communications and Telecom	www.ti.com/communications
Computers and Peripherals	www.ti.com/computers
Consumer Electronics	www.ti.com/consumer-apps
Energy and Lighting	www.ti.com/energy
Industrial	www.ti.com/industrial
Medical	www.ti.com/medical
Security	www.ti.com/security
Space, Avionics and Defense	www.ti.com/space-avionics-defense
Video and Imaging	www.ti.com/video

TI E2E Community Home Page

e2e.ti.com

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2012, Texas Instruments Incorporated