



AN012: FIRMWARE UPGRADING RADIOCRAFTS MODULES

APPLICATION NOTE





Firmware Upgrading Radiocrafts modules

By T.A.Lunder

Introduction

The Radiocrafts Demo Boards and USB-sticks include an on-board firmware upgrade connector compatible with different flash programming adapters. It is also recommended that PCB layouts include a connector for firmware upgrade. This Application Note describes how to upgrade the firmware of any Radiocrafts RF-module using either a CC Debugger or the FlashPro-CC tools.

Firmware upgrade should only be done after approval by Radiocrafts. The warranty of the module will be void after reprogramming outside our factory, and should only be done in special cases under agreement with Radiocrafts.

Supported modules

This Application Note is applicable for the following Radiocrafts module families: RC11xx, RC11xxHP, RC25xx, RC25xxHP, RC17xx and RC17xxHP.

Preparations

Before the upgrade can start you need the following:

- 1. RCxxxx(HP)-DB with USB connector or USB-stick. Demo boards with RS232 connector do not support firmware upgrades.
 - Alternatively your own PCB with firmware upgrade connector.
- 2. Module firmware file (*.hex) from Radiocrafts. This is available under a NDA that ensure that that the file only can be used on Radiocrafts modules.
- 3. The FlashPro-CC Flash Programming Adapter. See <u>www.elprotronic.com</u>.
- FlashPro-CC PC software which can downloaded for free at <u>www.elprotronic.com</u>. See next page for download details. Or:
- 5. CC debugger USB dongle: http://www.ti.com/graphics/tool/cc_debugger_800.jpg
- 6. SmartRF Flash Programmer which can be downloaded at: <u>www.ti.com/lit/zip/swrc044</u> (SmartRF Studio can be used as well if this is installed from before).

Important Notice on calibration values

RC11xx modules with date code 201014 or newer supports initial tolerance calibration during production, resulting in a Slave lifetime of >> 30 years. The calibration value is stored in configuration memory address 0x39 (Please check with latest Data Sheet / User Manual). If firmware upgrades are to be done, this value must be read, and written back in the module to keep the frequency calibration after firmware upgrade.

For Development Kits and evaluation purposes, this procedure is not mandatory.

Connecting RCxxxx(HP)DB (Demo board) to FlashPro-CC / CC Debugger

The Demonstration Kit includes a connector (P8) that allows the end user to firmware upgrade the module. Figure 1 illustrates how to connect the RC11xx Demo Board to the FlashPro-CC via the 14 to 10 pin adaptor board. The CC debugger connects via the same connector. The USB stick has a 2x5 pin connector with 1.27 mm pitch so the adaptor following the CC Debugger must be used.

Page **2** of **7**



APPLICATION NOTE: AN012



Figure 1: FlashPro-CC connected to RC1180DB-USB

FlashPro-CC software

The FlashPro-CC software form Elprotronic can be downloaded from:

http://www.elprotronic.com/download.html

Download the version valid for CCxx series devices. FlashPro-CC Revision 1.32 or newer supports the RC1180-MBUS module from Radiocrafts.

The installation will also install a manual for the FlashPro-CC, and you should use this manual if you need additional help. The software program with RC11XX setup is illustrated in figure 2.

FlashPro-CC (Flash Programmer for the CCxxxx Chipcon Products from TI) - Elprotronic Inc.				
File View Setup IEEE/Serialization About/Help				
Open Code File -> RC1180-MBUS3_3_14.hex	path: G:\02_Internal_projects\02021_A	Automatic_Test_Station-ATS\02021_6_Proc		
Programming and Pairing Master and Slave MCU's Slave MCU Code → ✓ 32-kB Banked Placement Code IEEE/SN File SoC Device Type RC11xx-32kB Tarret DOt1 + 2000 Total: Total:	Check Sum (Code) Source: 0x179C1E38 Memory: 0x179C1E38 ✓ Power Device from Adapter 3.2 ✓ 100mA ☞ Enable POWER ON/OFF RESET 0.45 ∨	Lock Protection Bits Virite Lock Bits Enable Protected: None Debug - blocked BOOT not protected Virite LOCK Bits MCU Frequency [MHz]		
Balance: 0 Selected Device Information RAM - 4096 bytes: FLASH - 32 kB;	IEEE address (hex)	Image: Market State Market State (RC gen.) Device Action Image: Reload Code File M S Image: Reload Code File		
Report	Read IEEE			
Reading Code File done done Code size = 0x0000 (0) bytes SoC communication initialization OK Erasing memory done SoC communication initialization OK All Memory Blank checking OK Elach programming done Elach programming done Elach programming done Elach programming done Elach programming done E	Device Serialization Read SN Next Model-Group-Revision:	Image: Second		
Verifying check sum OK Writing Lock bits One D O N E - (run time = 1.6 sec.)	00000000 Write SN Format: yyyy1234	VERIFY FLASH READ / COPY		
Adapter: (CC-xx) FlashPro-CC STD Interface: Communication up to 3 Mb/s	Erase / Write memory option: *All Memory*	AUTO PROGRAM NEXT (F5)		

Figure 2: FlashPro-CC software

According to the firmware NDA the user must ensure that the hex file is not available for others. For this reason the debug interface must be blocked after firmware update to ensure that no one can read back the memory. This is handled by the FlashPro-CC in the

Setup->Memory Option menu using the following setting:

Memory Options		
Memory Erase/Write/Verify Ad	ddress Range	Read Address Range
Unblock debug	er Defined	All Memory User Defined
Enable if blocked	Start Address: 0x000	Start Address: 0x0000
Whole memory will	Stop Address: 0x1FF	F Stop Address: 0x1FFFF
be erased !!! No retain data etc.	Start Address: 0x000	Ouser Defined Start Address: 0x0000
	B2 Stop Address: 0x1FF	F Stop Address: 0x1FFFF
O Append Only		
	B3 Start Address: 0x000	B3 Start Address: 00000
(Air Memory	Stop Address: 0x1HH	F Stop Address: 0x1FFFF
O Used by Code File	B4 Start Address: 0x000	Start Address: 0x0000
O User Defined	Stop Address: 0x1FF	F Stop Address: 0x1FFFF
Retain Flash Data (Erase and Autoprogram) Specified data are read before flash erase and downloaded to flash on the end. No effect if debug commands are blocked. Enable Start Address: Max size - 2048 bytes Stop Address: Write Verification Stanta (Write, Verify CS) [Recommended] Standard (Write, Read and Verify Byte by Byte) Code Memory Start None Flash lock after "Autoprogram"		Plash lock after "Autoprogram" Write protect: No protection Write Protect BOOT block Block debug commands (ind. read access). Note: Cannot Retain, Append or Verify" when set!. About Device Code Memory Start Addr: 0x0000 Selected Device: RC11XX-32KB Code Memory End Addr: 0x7FFF Flash Segment Size: 0x0400
	ОК	Cancel

Figure 3: FlashPro-CC Memory options for block Debug interface

Step-by-step guide

- 1. Run the downloaded setup file to install FlashPro-CC on your PC
- 2. Ensure that you have all jumpers set to default position. Se Development Kit user manual for details.
- 3. Connect The Flash-Pro-CC to P8 as illustrated in figure 1.
- 4. Connect the FlashPro-CC to an available USB port. If this is the first time you connect the FlashPro-CC to your PC, you need to follow the *Found new Hardware wizard* for installing the USB driver in windows. See Manual from elprotronic for details.
- 5. Open FlashPro-CC software after the USB driver is installed
- 6. Click on the "Open Code File ->" button and select the module firmware file (*.hex) you have received from Radiocrafts under NDA.
- 7. Select "RC11XX-32KB" as your SoC Device Type
- 8. "Enable" power Device from Adapter
- 9. "Disable" IEEE address
- 10. "Enable" Lock Protection Bits
- 11. Block debug commands in the Memory Options
- 12. Click on the "Auto Program" button
- 13. Verify that the status gives a green "PASS" sign
- 14. Disconnect Flashpro-CC from P8 on RC11XXDB.

The module is now updated. You can verify that you have an updated version by reading the FW revision from the RCTools software from Radiocrafts.

Page 4 of 7

APPLICATION NOTE: AN012

SmartRF Flash Programmer

🔻 Texas Instruments SmartRF® Flash Programmer 🛛 📃 🗆 🔀				
TEXAS INSTRUMENTS	What do you want to program? Program CCxxxx SoC or MSP430			
	System-on-Chip MSP430 EB ID Chip type EB typ 2561 CC2510 CC De Interface: Flash image: Change 0 bytes at 0x	e EB firmware ID EB firmware rev bugger 05CC 0029	-	
	Actions C Erase Erase and program Erase, program and verify Append and verify Verify against hex-file Read flash into hex-file N	ash lock (effective after program/append): rite protect: No upper pages Write protect boot block Block debug commands fincl. read access) 3: Cannot "Append and verify" when set! Perform actions		

Figure 4: SmartRF Flash Programmer and default settings (do not tick different)

Step-by-step guide

- 1. Run the downloaded setup file to install SmartRF Flash programmer
- 2. Ensure that you have all jumpers set to default position. Se Development Kit user manual for details.
- 3. Connect to connector P8 and provide power to the board (via USB connector)
- Press Reset-button on CC Debugger and the green LED shall lit. If not, turn the programming connector 180 degrees and try again. If still no Red light, check power connection to the Demonstration board
- 5. Import the .hex-file from the stored location
- 6. Tick exactly the buttons as described in the picture above, otherwise the re-flashing will fail
- 7. Press "Perform actions"
- 8. Disconnect the CC debugger

Page 5 of 7

Document Revision History

Document Revision	Changes
1.0	First release
1.1	- Memory option to block debug commands included
	- Notice about keeping frequency tolerance calibration value
1.2	Added CC Debugger information
1.21	Update figure 4.
1.22	"Enable" Lock Protection Bits in page 3; Include info on all supported modules
1.3	Design Update

Disclaimer

Radiocrafts AS believes the information contained herein is correct and accurate at the time of this printing. However, Radiocrafts AS reserves the right to make changes to this product without notice. Radiocrafts AS does not assume any responsibility for the use of the described product; neither does it convey any license under its patent rights, or the rights of others. The latest updates are available at the Radiocrafts website or by contacting Radiocrafts directly.

As far as possible, major changes of product specifications and functionality, will be stated in product specific Errata Notes published at the Radiocrafts website. Customers are encouraged to check regularly for the most recent updates on products and support tools.

Trademarks

RC232[™] is a trademark of Radiocrafts AS. The RC232[™] Embedded RF Protocol is used in a range of products from Radiocrafts. The protocol handles host communication, data buffering, error check, addressing and broadcasting. It supports point-to-point, point-to-multipoint and peer-to-peer network topologies.

All other trademarks, registered trademarks and product names are the sole property of their respective owners.

Life Support Policy

This Radiocrafts product is not designed for use in life support appliances, devices, or other systems where malfunction can reasonably be expected to result in significant personal injury to the user, or as a critical component in any life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness. Radiocrafts AS customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Radiocrafts AS for any damages resulting from any improper use or sale.

© 2017, Radiocrafts AS. All rights reserved.





For More Information, Please Visit Our Website!

www.radiocrafts.com Email: sales@radiocrafts.com Tel: +47 4000 5195