FLORIPASAT - TTC-DOC - REV1

Telemetry, Tracking and Command Module of the FloripaSat Project

Module Documentation

GSE, Federal University of Santa Catarina, Florianópolis - Brazil

December 2017

FloripaSat Project, Telemetry, Tracking and Command Module Documentation
December, 2017

Project Manager:
Eduardo Augusto Bezerra

Author:
Gabriel Mariano Marcelino

Contributing Authors:
Marcelo Daniel Berejuck
Sara Vega Martinez

Layout:
Gabriel Mariano Marcelino

Abstract

This document...

Keywords: Cubesats. Embedded systems. Telecomunications.

Contents

List of Figures

List of Tables

Nomenclature

ADC	Analog-To-Digital Converter.
BSL	Bootstrap Loader.
CPU	Central Processing Unit.
DMA	Direct Memory Access.
GPIO	General Purpose Input/Output.
HAL	Hardware Abstraction Layer.
I^{2}C	Inter-Integrated Circuit.
ISR	Interruption Service Routine.
PCB	Printed Circuit Board.
RAM	Random Access Memory.
SPI	Serial Peripheral Interface.
TTC	Telemetry, Tracking and Command.
UART	Universal Asynchronous Receiver/Transmitter.
USB	Universal Serial Bus.

CHAPTER 1

Introduction

INTRODUCTION...
[1].
[2]
The beacon executes the following tasks during its execution:

- 145 MHz band antenna deployment.
- One-way communication (RX) with the EPS module, using the FSP protocol.
- Two-way communication (TX and RX) with the OBDH module, using the FSP protocol.
- Transmission of the beacon packets (In two protocols: NGHam and AX.25), containing the data from the EPS or the OBDH module and the satellite ID ("FLORIPASAT").
- When required by the OBDH module, the transmissions are stopped for an hibernation period (shutdown).
- In case of an critical failure of the OBDH module, or the uplink channel, the beacon activates its reception and is capable of make its own hibernation (shutdown).

Module Requirements

In the list below, the TTC module requirements for the mission are described. These requirements are nominated as TMR, or Telemetry, Tracking and Command Module Requirements.

TMR 1 The FloripaSat shall have a physical device to inhibit radio frequency (RF) transmission.

Compliance with CDS 3.3.9: The use of three independent inhibits is highly recommended and can reduce required documentation and analysis.

TMR 2 The CubeSat will have the RF power output to the transmitting antenna input no greater than 1,5 W.

Compliance with CDS 3.3.9.1.

TMR 3 The CubeSat will have the RF power output to the transmitting antenna input no less than 1,0 W (or 30 dBm).

Defined by team analysis.

TMR 4 No CubeSats shall generate or transmit any RF signal from the time of integration into the P-POD through 45 minutes after on-orbit deployment from the P-POD.

Compliance with CDS.

TMR 5 TTC transceiver shall transmit and receive on the frequency of 437,9 Mhz.

Defined by the team, based on available spectrum allocation to Amateur communication.

TMR 6 TTC beacon shall transmit on the frequency of 145,9 Mhz.

Defined by the team, based on available spectrum allocation to Amateur communication.

TMR 7 TTC shall modulate and demodulate information using GFSK.

Defined by the team.

TMR 8 TTC Beacon must transmit periodic beacon messages at an interval of 10 seconds, except when in hibernation or shutdown mode. Allows ground stations to track and receive satellite data even if telecommand was not sent to the satellite.

TMR 9 TTC transceiver must receive signals from ground stations and demodulate them.

TMR 10 TTC must interface with OBDH, exchanging encoded raw data received or to be transmitted.

TMR 11 TTC must interface with OBDH using the SPI protocol (@2 KHz).

Defined by the team.

TMR 12 TTC radio must modulate raw data received from OBDH using GFSK, prior to transmission, and demodulate received data and forward raw data to OBDH.

TMR 13 TTC transceiver shall transmit and receive data at a baud rate of 2400 bps. Defined by the team, based on link budget analysis.

TMR 14 TTC beacon shall transmit data at a baud rate of 1200 bps.

Defined by the team, based on link budget analysis.

TMR 15 TTC beacon shall transmit packets using the NGHam and AX.25 protocols.

Defined by the team.

TMR 16 A same beacon packet must be transmitted in both NGHam and AX.25 protocols.

Defined by the team.

TMR 17 TTC must receive the batteries voltages from the EPS module at every 10 seconds.

Defined by the team.

TMR 18 The payload from the packets transmitted by the beacon must contain at least the satellite ID ("FLORIPASAT") and batteries voltages (received from the EPS module at every 10 seconds).

TMR 19 TTC uC must perform the antenna deployment of the VHF band antenna.

Defined by the team.

TMR 20 Between the beacon packets transmissions, the beacon MCU and radio must operate in low power mode, to save energy.

Defined by the team.

TMR 21 TTC PAs (Power amplifiers) must just only be activated during transmissions. When they are not in operation, they must be turned off.

Defined by the team.

TMR 22 All the beacon critical data, like time control and antenna deployment status, must be stored in a non-volatile memory.

Defined by the team.

TMR 23 TTC beacon must be able to receive a 24 hour shutdown command from the OBDH module.

Compliance with AMSAT/IARU regulations.

CHAPTER 2

Hardware

The TTC board is composed by the following main components:

- MSP430F6659, as the beacon microcontroller.
- RF4463F30, as the radio module for the beacon and the telemetry link.

In the figure 2.1, ...

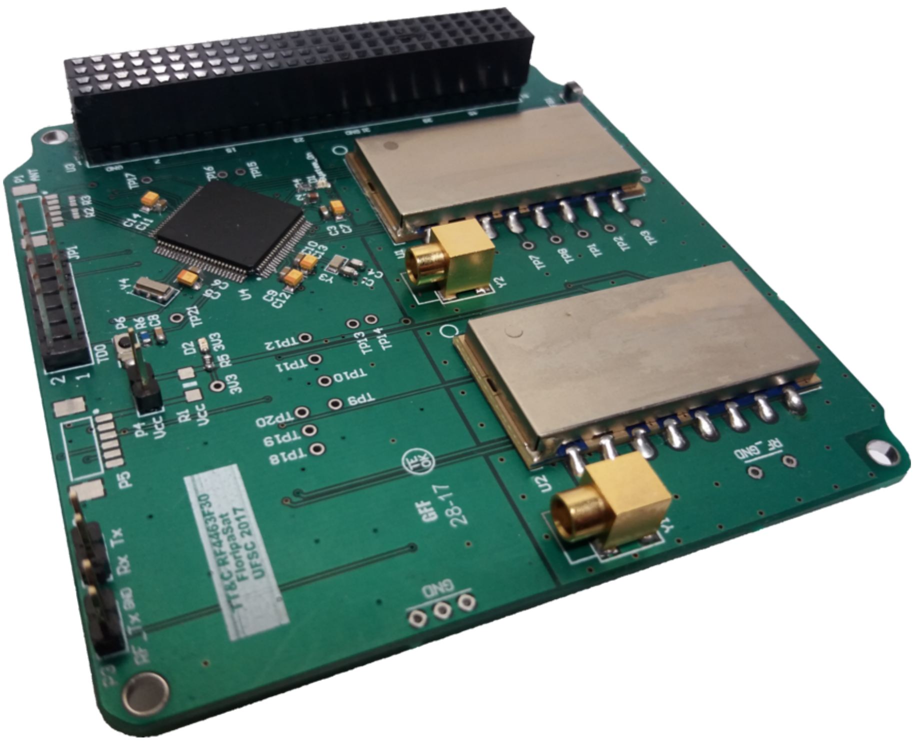


Figure 2.1: TTC PCB.

General Diagram

In the figure 2.2, a general hardware diagram can be seen.

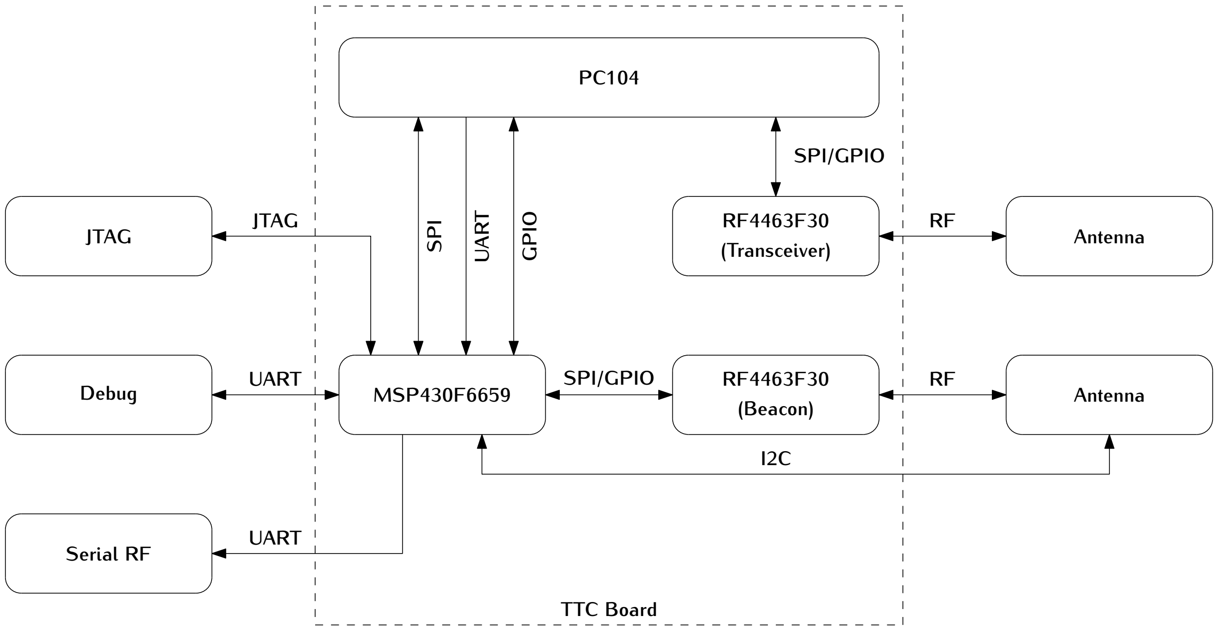


Figure 2.2: Hardware diagram of the TTC module.

Main Components

M...

Microcontroller

The beacon microcontroller is the MSP430F6659IPZR [3]. Its main characteristics can be found in the table 2.1.

Radio Modules

The NiceRF RF4463F30 [4] is a transceiver module based on the Silicon Labs Si4463 [5] radio. This module also contains a PA module to increase the output power up to 31 dBm.

Si4463

External Connections

This section describes the external available connections of the TTC module.

In the figure 2.3, all the external connections are enumerated.

A brief description of each connection is presented in the table 2.3.

The connections 1, 2, 4 and 6 were designed to be used during the software development stage, and not during the satellite operation.

Characteristic	*Value*
CPU	MSP430
Frequency	Up to 20 MHz
Non-volatile memory	512 kB
RAM	66 kB
GPIO pins	74
I^2C	3
SPI	6
UART	3
DMA	6
ADC	ADC12-12ch
Comparators	12 inputs
Timers - 16-bit	4
Multiplier	32×32
BSL	USB
Min V_{cc}	1,8 V
Max V_{cc}	3,6 V
Active Power	360 $\mu A/MHz$
Standby Power (LMP3)	$2,6$ μA
Wakeup Time	3 μs
Operating Temperature Range	-40 to 80 $°C$

Table 2.1: MSP430F6659 features.

Characteristic	*Value*	*Unit*
Frequency range	119-1050	MHz
Receiver sensitivity	-126	dBm
Modulation	(G)FSK, 4(G)FSK, (G)MSK and OOK	-
Max. output power	+20	dBm
PA support	+27 to 30	dBm
Ultra low current powerdown modes	30 (shutdown), 50 (standby)	nA
Data rate	100 bps to 1 Mbps	-
Power supply	1,8 to 3,6	V
TX and RX FIFOs	64 bytes for each or 129 bytes shared	-

Table 2.2: Si4463 features.

PCI-104 Pins

The table 2.4 describes the PCI-104 connector used pins. The first column is the row number of the connector, and the remaining columns are the respective columns (Named as H1A, H1B, H2A and H2B respectively). If the pin has no description, it is not connected to the TTC board.

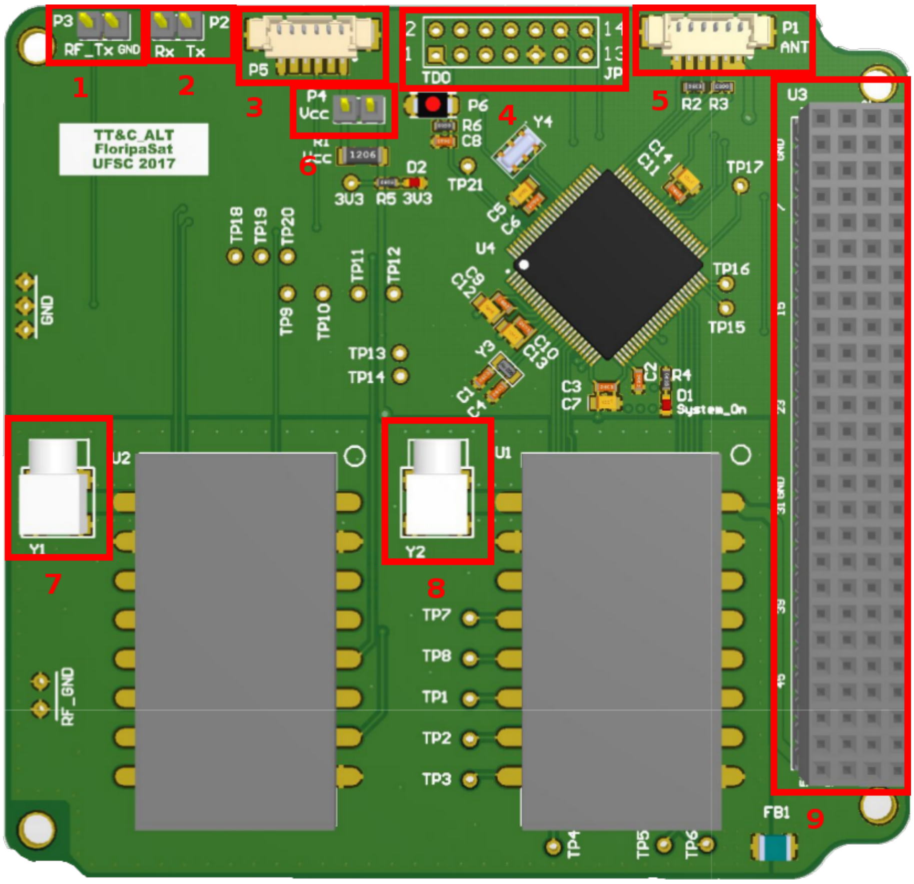


Figure 2.3: External connections on the board.

PCB

The PCB (Printed Circuit Board) of the TTC module has basically the MSP430F6659 ic, the RF4463F30 module and all the external connectors.

Some characteristics of this PCB are described bellow:

- The components and traces of the board are distributed over two layers.
- The RF modules are placed in a isolated ground plane (Connected to the main GND using a ferrite bead).
- The antenna output of the RF4463F30 modules are connected to their respective connector over a 50 Ohm coupled trace.

Number	*Connector*	*Description*
1	Male pin header (1 × 2)	UART TX @4800 bps. These pins transmit the beacon packets over a serial connection (It is enable in the configuration file, setting the BEACON_RADIO variable as UART_SIM).
2	Male pin header (1 × 2)	Debug UART TX/RX @115200 bps. These pins transmit a description of the main events of the beacon software during it's execution. This feature is only available in DEBUG_MODE.
3	Male PicoBlade^{TM} (×6)	JTAG and Debug. This connection contains the relevant pins of the connectors 2 and 4.
4	Male pin header (2 × 2)	MSP430 JTAG. This connection is for programming the uC code, using a MSP-FET debugger.
5	Male PicoBlade^{TM} (×6)	Antenna I2C. I2C bus for a communication channel with the antenna module.
6	Male pin header (1 × 2)	Power supply jumper. With a jumper, the beacon microcontroller power source comes from the JTAG connector. Without a jumper, the uC power supply comes from a pin of the PC104 connector.
7	Female Angled MCX	437 MHz band RF signal (Goes to the antenna module).
8	Female Angled MCX	145 MHz band RF signal (Goes to the antenna module).
9	Male/Female PCI-104	PCI-104. Power supply and communication buses with others stacked up modules.

Table 2.3: External connections description.

Row	*H1A*	*H1B*	*H2A*	*H2B*
1	GND	GND	GND	GND
2	GND	GND	GND	GND
3	-	-	UART RX @4800 bps from the EPS module.	-
4	Telemetry radio GPIO0	Telemetry radio GPIO1	-	-
5	Telemetry radio GPIO2	Enable beacon radio power supply	-	-
6	Telemetry radio SDN	-	OBDH communication (SPI MOSI)	OBDH communication (SPI clock)
7	-	-	OBDH communication (SPI chip select)	OBDH communication (SPI MISO)
8	-	-	-	-
9	-	-	-	-
10	-	-	-	-
11	-	-	-	-
12	-	-	-	-
13	-	-	-	-
14	-	-	Beacon uC power supply (3,3 V/50 mA)	3,3 V beacon uC power supply (3,3 V/50 mA)
15	GND	GND	GND	GND
16	GND	GND	GND	GND
17	-	-	-	-
18	Telemetry radio SPI clock	-	-	-
19	Telemetry radio SPI MISO	-	-	-
20	Telemetry radio SPI MOSI	Telemetry radio SPI chip select	-	-
21	-	-	-	-
22	-	-	-	-
23	-	-	-	-
24	-	-	-	-
25	Telemetry radio power supply (5 V/500 mA)	-	-	-
26	Beacon radio power supply (5 V/500 mA)	-	-	-

Table 2.4: PCI-104 connector reference.

CHAPTER 3

Software

SOFTWARE...

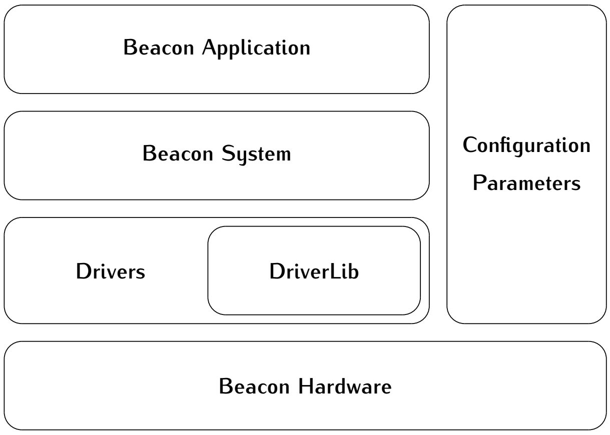


Figure 3.1: Beacon software stack-up.

Flowcharts

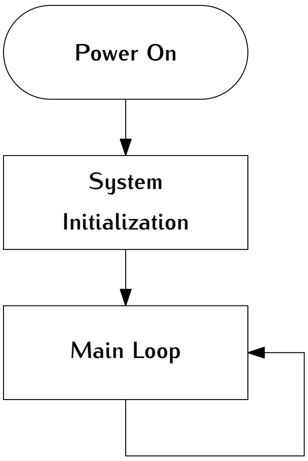
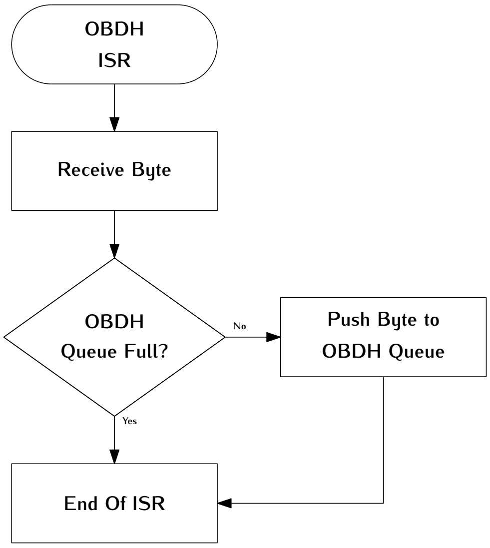
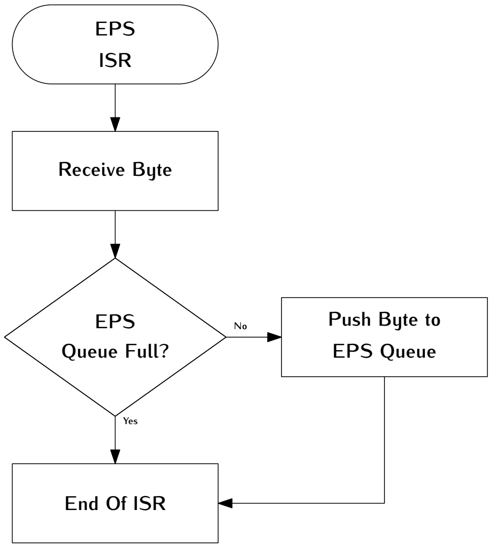


Figure 3.2: Main flowchart of the beacon software.



(a) OBDH communication ISR flowchart.



(b) EPS communication ISR flowchart.

Figure 3.3: OBDH and EPS modules comunication ISRs routines.

Tasks

In the table 3.1, the beacon tasks are listed with a brief description and its period of execution.

Task	*Description*	*Period*
System initialization	Submodules initialization (CPU, memory, radio, etc.)	Aperiodic (Executed only at initializations)
Antenna deployment	145 MHz band antenna deployment after the satellite launch	Aperiodic (Executed once)
NGHam packet transmission	Transmission of the beacon packets with a basic telemetry data using the NGHam protocol	10, 20 or 30 seconds (EPS energy level dependent)
AX.25 packet transmission	Transmission of the beacon packets with a basic telemetry data using the AX.25 protocol	10, 20 or 30 seconds (EPS energy level dependent)
OBDH data processing	The OBDH module can send telemetry data or commands	Aperiodic (OBDH module dependent)
EPS data processing	The EPS modules sends telemetry data	Aperiodic (EPS module dependent)
Radio data processing	Processing of an incoming packet from the beacon radio	Aperiodic (Only when the OBDH fails and an hibernation is required)
Beacon radio reception activation	When a critical failure occur in the OBDH module, the beacon activates its reception between the beacon transmissions	Aperiodic (Only when the OBDH module fails)
Radio reset	Beacon radio reset	10 minutes
Beacon reset	Beacon system reset	12 hours

Table 3.1: Beacon software tasks.

CHAPTER 4

Tests

THIS...

RF Signal Power

P...

CHAPTER 5

Conclusion

CONCLUSION...

Bibliography

[1] Rafael P. Alevato. <*http://floripasat.space/*>. FloripaSat Project, December 2017.

[2] FloripaSat Project. <*https://github.com/floripasat/ttc*>, December 2017.

[3] Texas Instruments Inc. *MSP430x5xx and MSP430x6xx Family User's Guide*, October 2016.

[4] NiceRF Wireless Technology Co., Ltd. *RF4463F30 1W High Sensitivity Wireless Transceiver Module Product Specification*, rev 2.1 edition, June 2016.

[5] Silicon Laboratories Inc. *Si4464/63/61/60 High-Performance, Low-Current Transceiver*, rev 1.2 edition, 2016.

Appendices

APPENDIX A

Radio Configuration

This appendix is a tutorial with the purpose of generate a source code file with basic configuration parameters of the radio module. For it, the WDS software from Silicon Labs will be used (version 3.2.11.0).

Unfortunately, the software in only available for Windows platforms.

Steps

After the installation of the software, the procedures to configure the beacon radio are described bellow (When a parameter to configure the telemetry link radio differs from the beacon radio, a note describes the difference).

Step 1

1. Open the WDS software.
2. The following box will appear in the center of the window.
3. Click in "Simulate radio" and go to the next step.

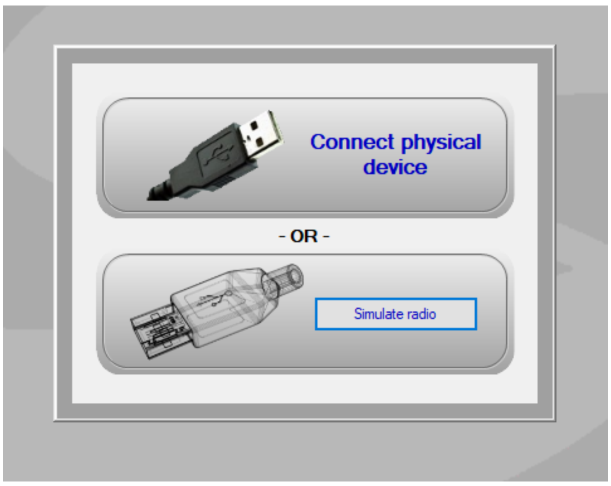


Figure A.1: Step 1 of the radio configuration.

Step 2

1. In the list of radios that appeared on the new window, select the chip type “Si4463”.
2. In the revision column, select “B1”.
3. Click on “Select Radio” to go to the next step.

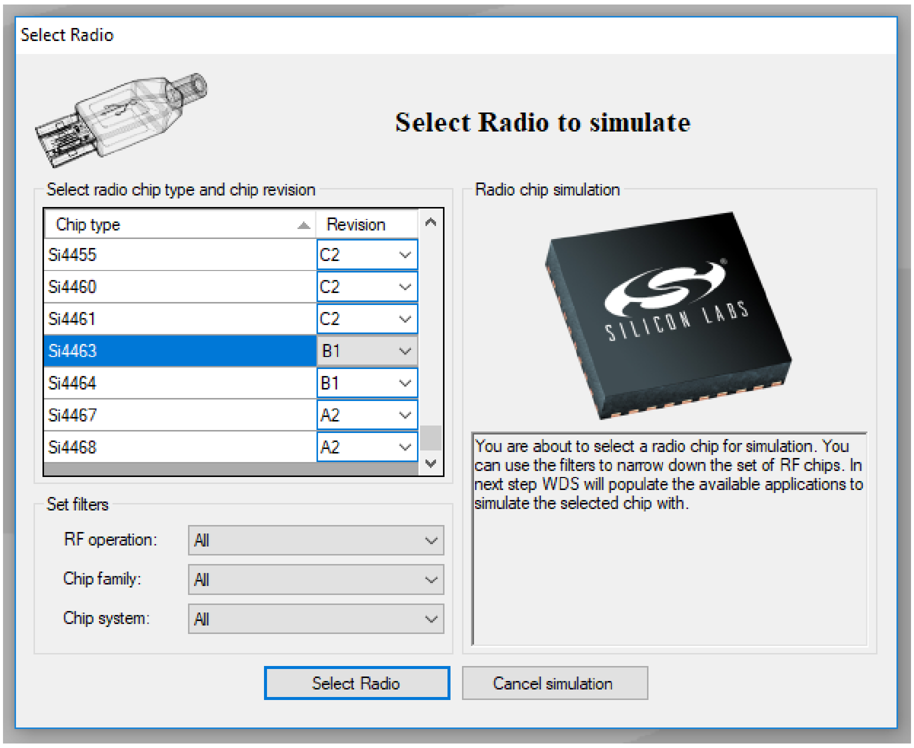


Figure A.2: Step 2 of the radio configuration.

Step 3

1. Select “Radio Configuration Application”.
2. Click on “Select Application” to go to the next step.

Step 4

1. In the “Frequency and power” tab, change the base frequency to 145,9 MHz.
2. Change the channel spacing to 0 kHz.
3. Change the crystal tolerance to 10,0 ppm (Both RX and TX).
4. Go to the next step.

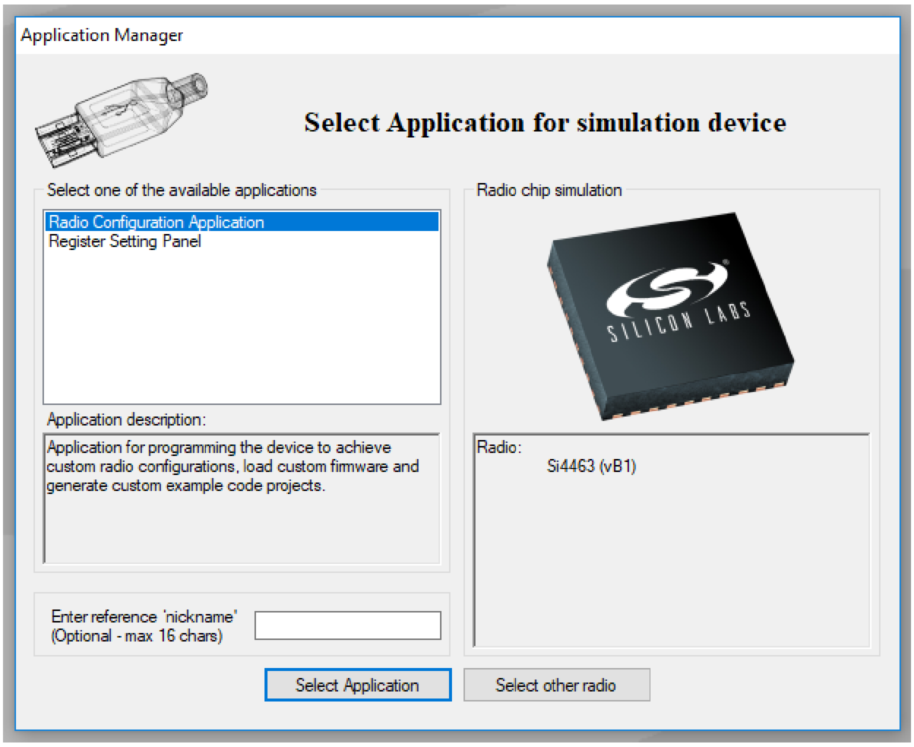


Figure A.3: Step 3 of the radio configuration.

Step 5

1. In the “RF parameters” tab, change the modulation type to “2GFSK”.
2. Change the the data rate to 1,2 kbps.
3. Change the deviation to $2,5\ kHz$.
4. Go to the next step.

Step 6

1. In the “Packet” tab, many subtabs will appear. In “Preamble” change the “Preamble TX length” to 4 bytes.
2. Again, in “Preamble”, change the “Preamble pattern” to “Std. 1010 pattern ($>=$ 32 and $<$ 40 bits)”.
3. Go to the next step.

Step 7

1. In the “Sync Word” tab, change the sync word length field to “4 bytes”.
2. In the “Sync Word (on air int.)”, enter the following sequence: 5D E6 2A 7E. This sequence is the sync word used by the NGHam protocol.
3. Go to the next step.

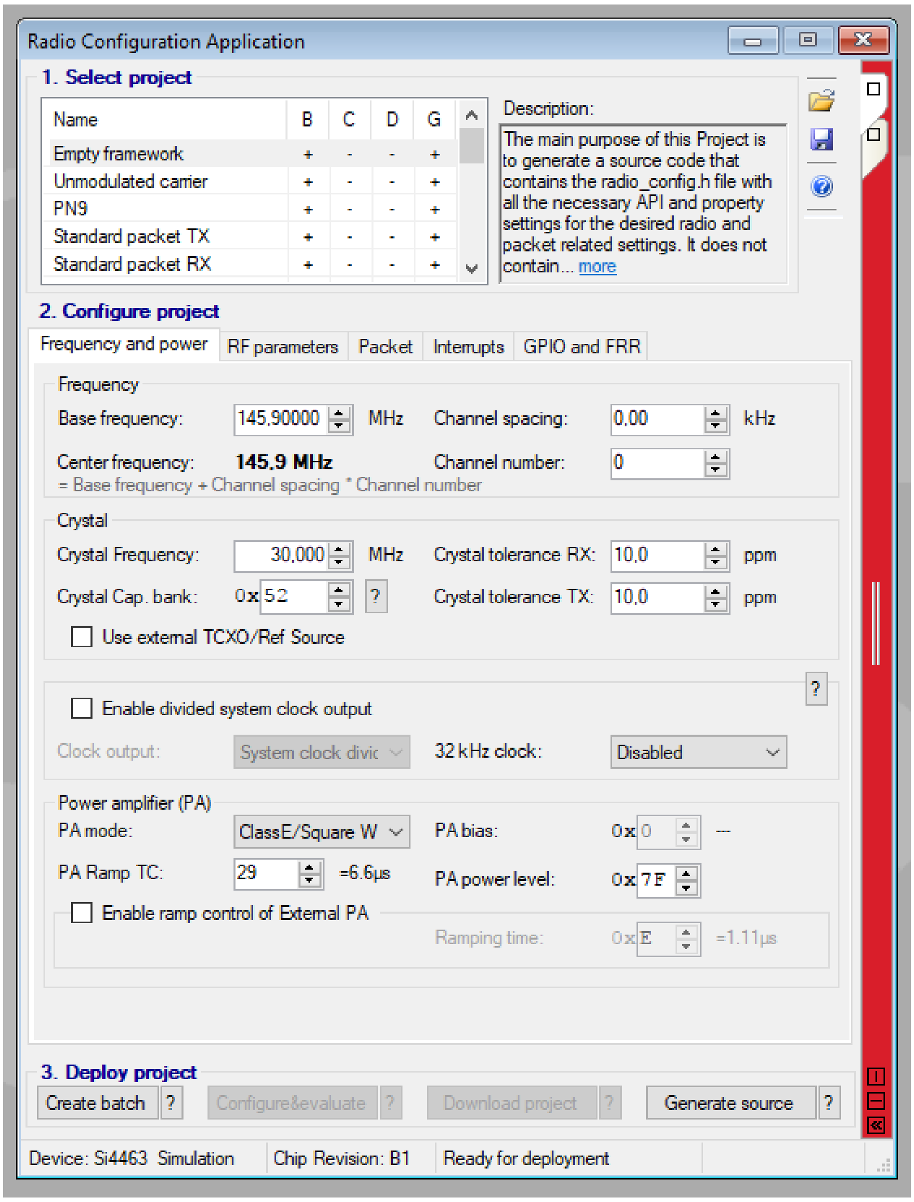


Figure A.4: Step 4 of the radio configuration.

Step 8

1. In the “Field 1” tab, change the “Field length” to 50 bytes.

2. Go to the next step.

Step 9

1. In the “Variable length config” tab, there is no values to change.

2. Go to the next step.

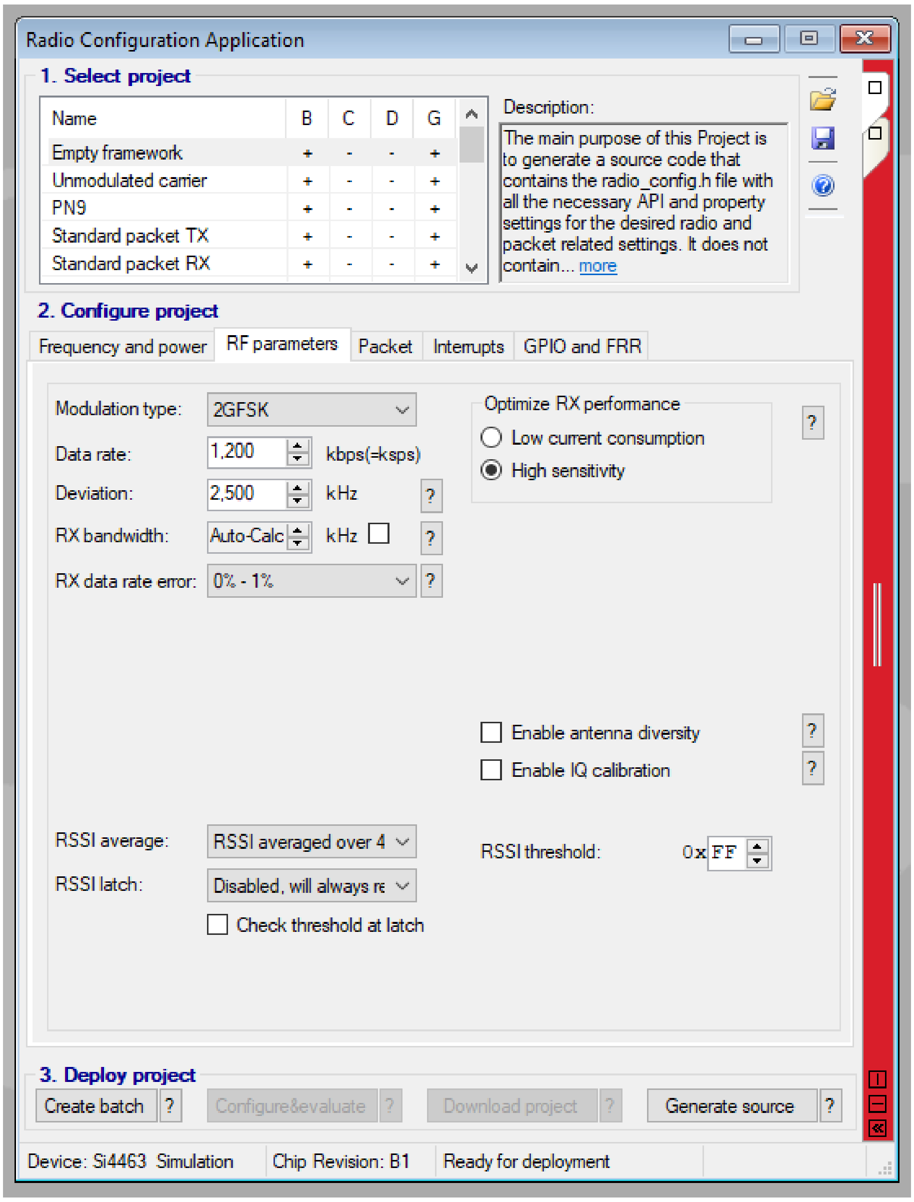


Figure A.5: Step 5 of the radio configuration.

Step 10

1. In the “CRC config” tab, choose “No CRC.” in “CRC polynomial”.
2. Go to the next step.

Step 11

1. In the “Whitening config” tab, there is no values to change.
2. Go to the next step.

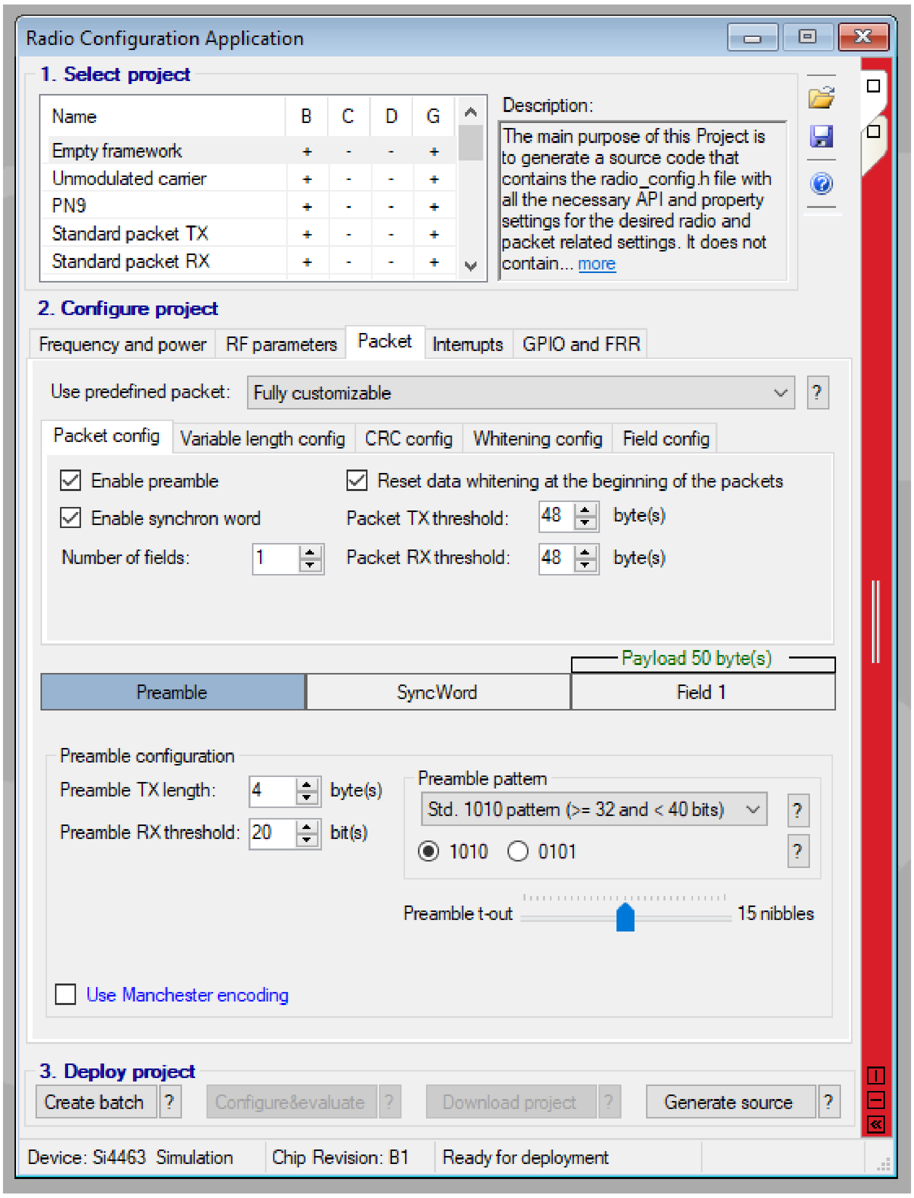


Figure A.6: Step 6 of the radio configuration.

Step 12

1. In the “Field config” tab, there is no values to change.
2. Go to the next step.

Step 13

1. In the “Interrupts” tab, there is no values to change.
2. Go to the next step.

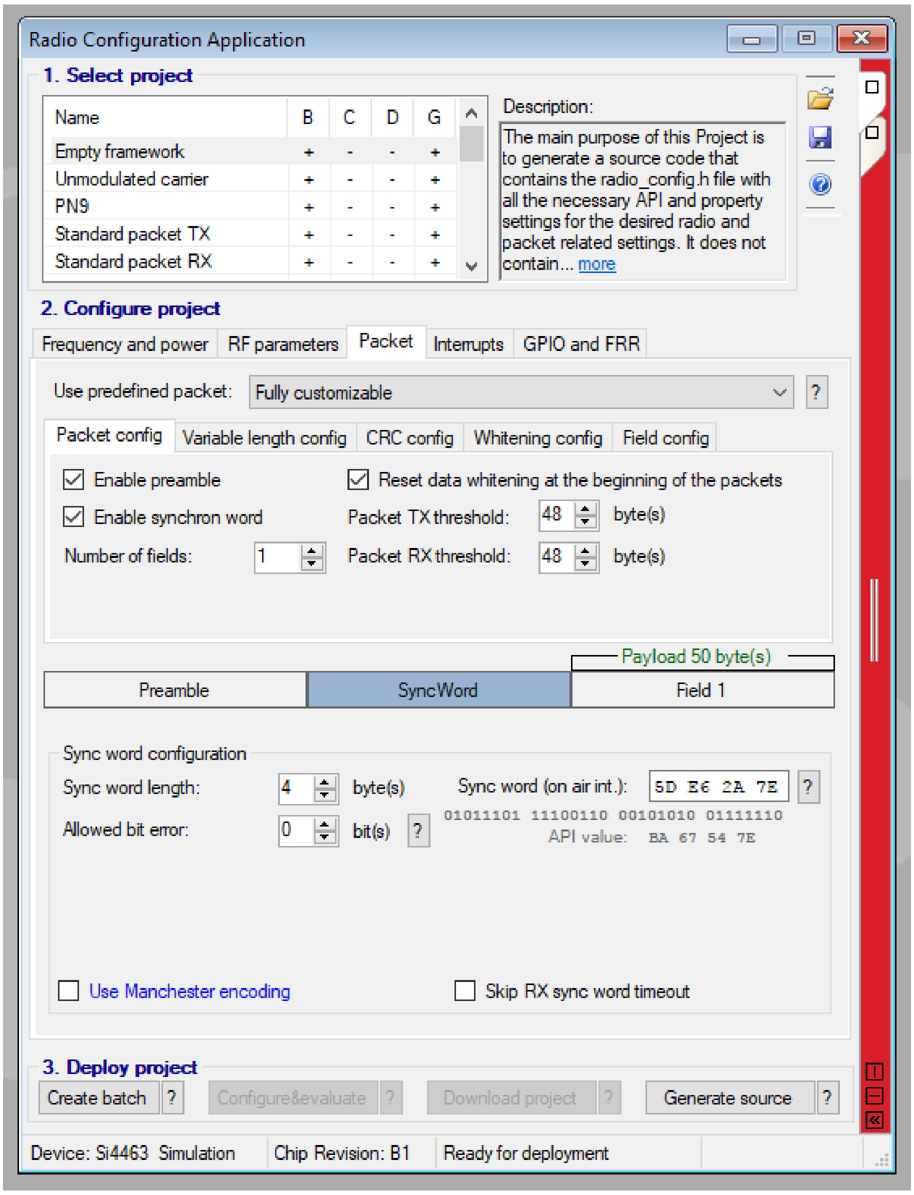


Figure A.7: Step 7 of the radio configuration.

Step 14

1. In the "GPIO and FRR" tab, enable pullup in GPIO1 and choose "TX_FIFO_EMPTY - This output is..." as functionality.

2. Enable pullup in GPIO2 and choose "RX_STATE - This output is..." as functionality.

3. Enable pullup in GPIO3 and choose "TX_STATE - This output is..." as functionality.

4. Enable pullup in NIRQ and choose "Active low interrupt signal" as functionality.

5. Enable pullup in SDO and choose "SDO - Output SPI Serial data out." as functionality.

6. Select "Global status" for the "Fast Response Register A".

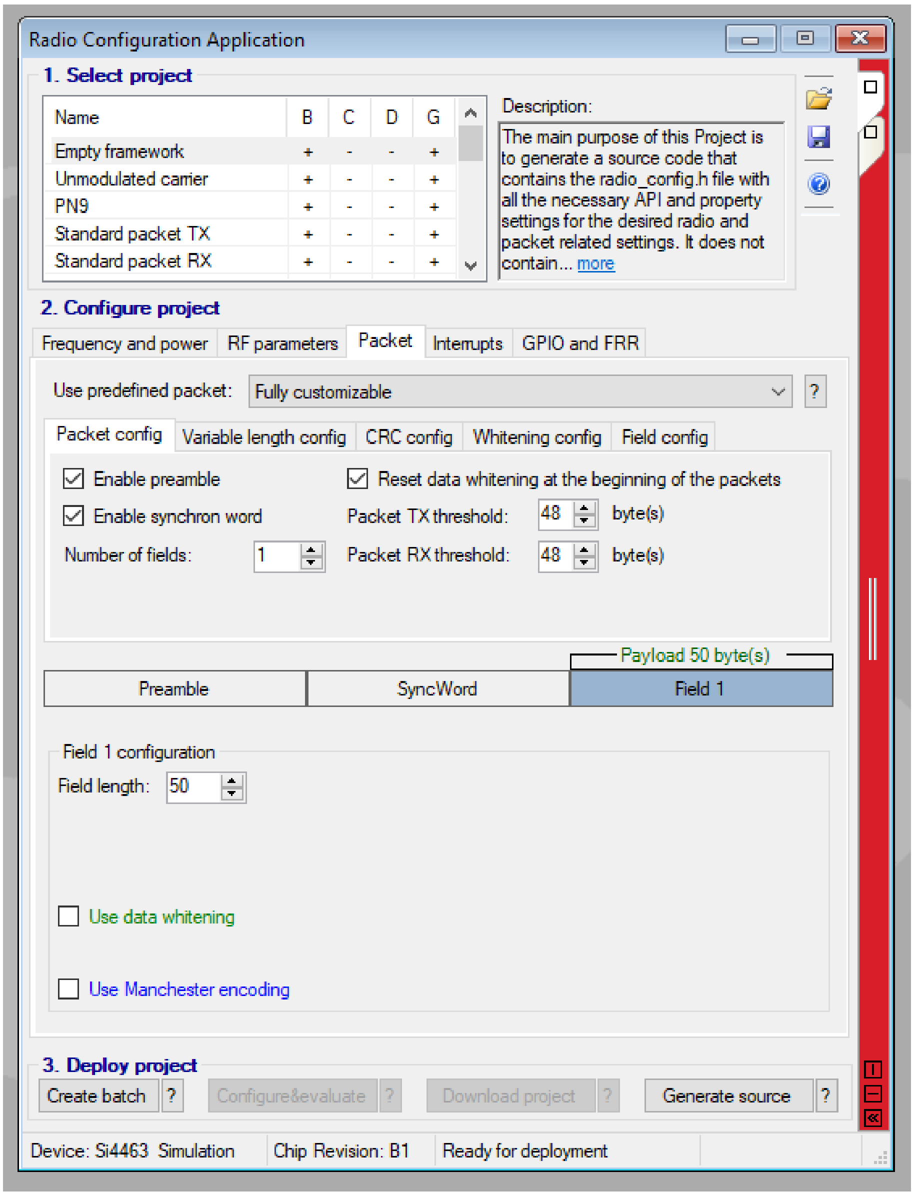


Figure A.8: Step 8 of the radio configuration.

7. Select "Global interrupt status" for the "Fast Response Register B".
8. Select "Packet Handler status" for the "Fast Response Register C".
9. Select "Chip status status" for the "Fast Response Register D".
10. Go to the next step.

Step 15

1. Click in "Generate Source" and select the ".h" type of source.
2. The software will ask where to save the generated file.
3. The generated *.h file must be copied to the directory of the rf4463 driver.

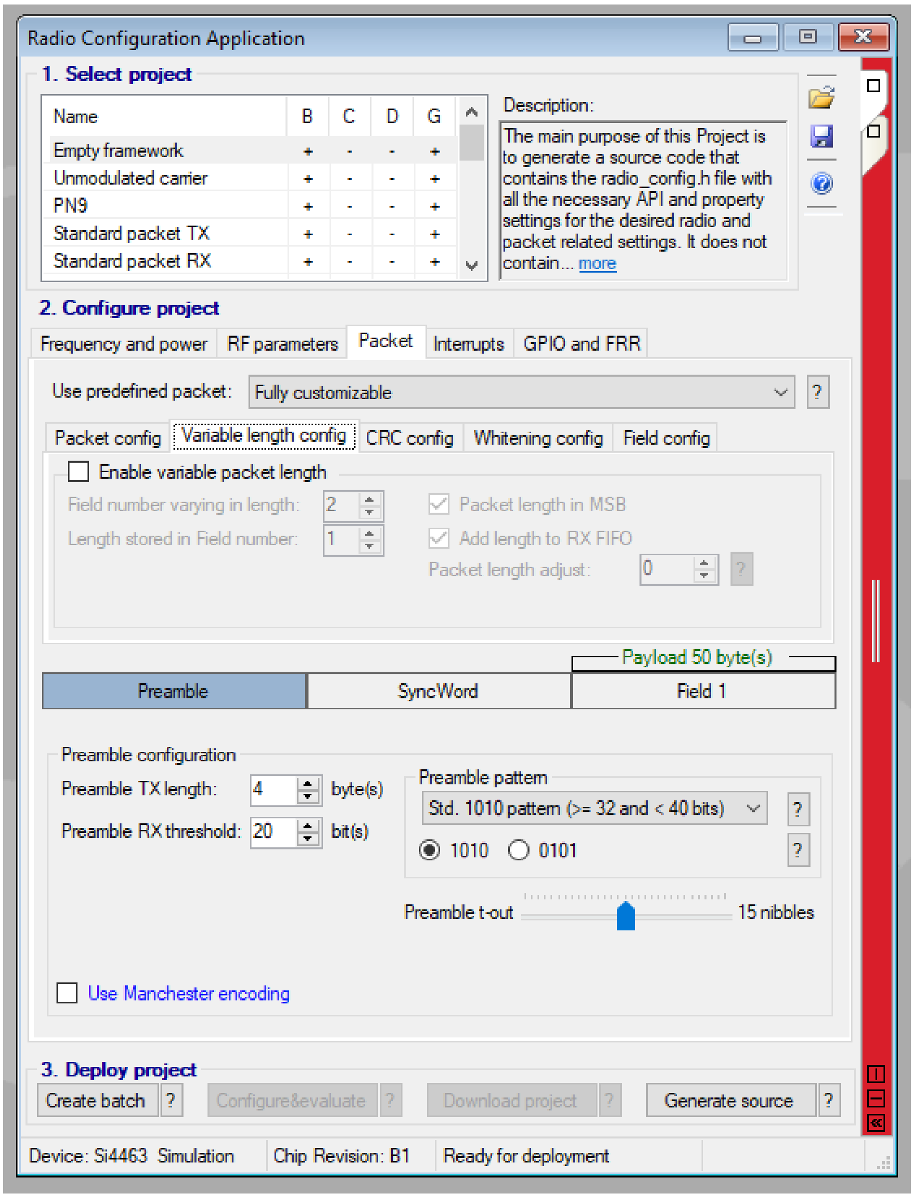


Figure A.9: Step 9 of the radio configuration.

Final Remarks

This tutorial has the objective of generate a basic configuration parameters of the radio, some functionalities of the radio are not covered by the WDS software, and so, must be configured/controlled in the device driver.

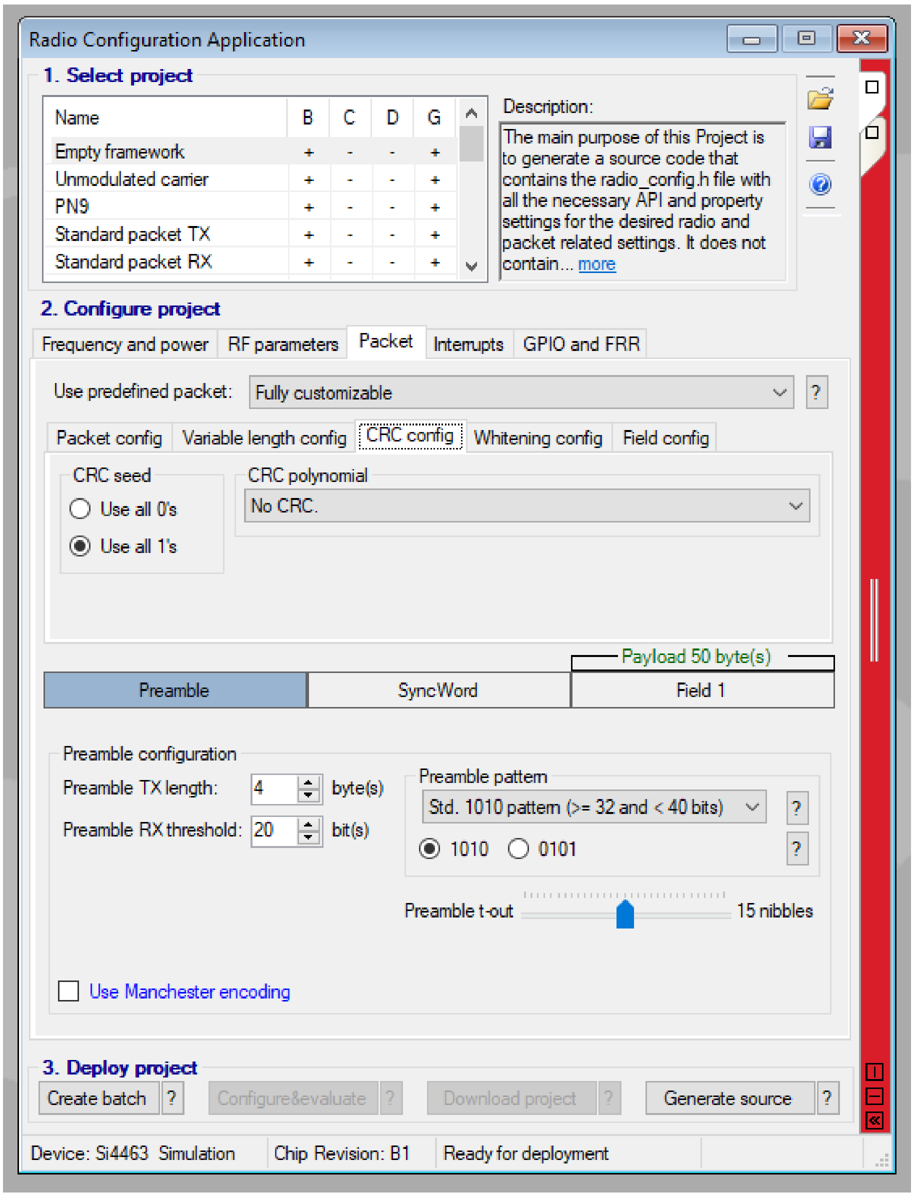


Figure A.10: Step 10 of the radio configuration.

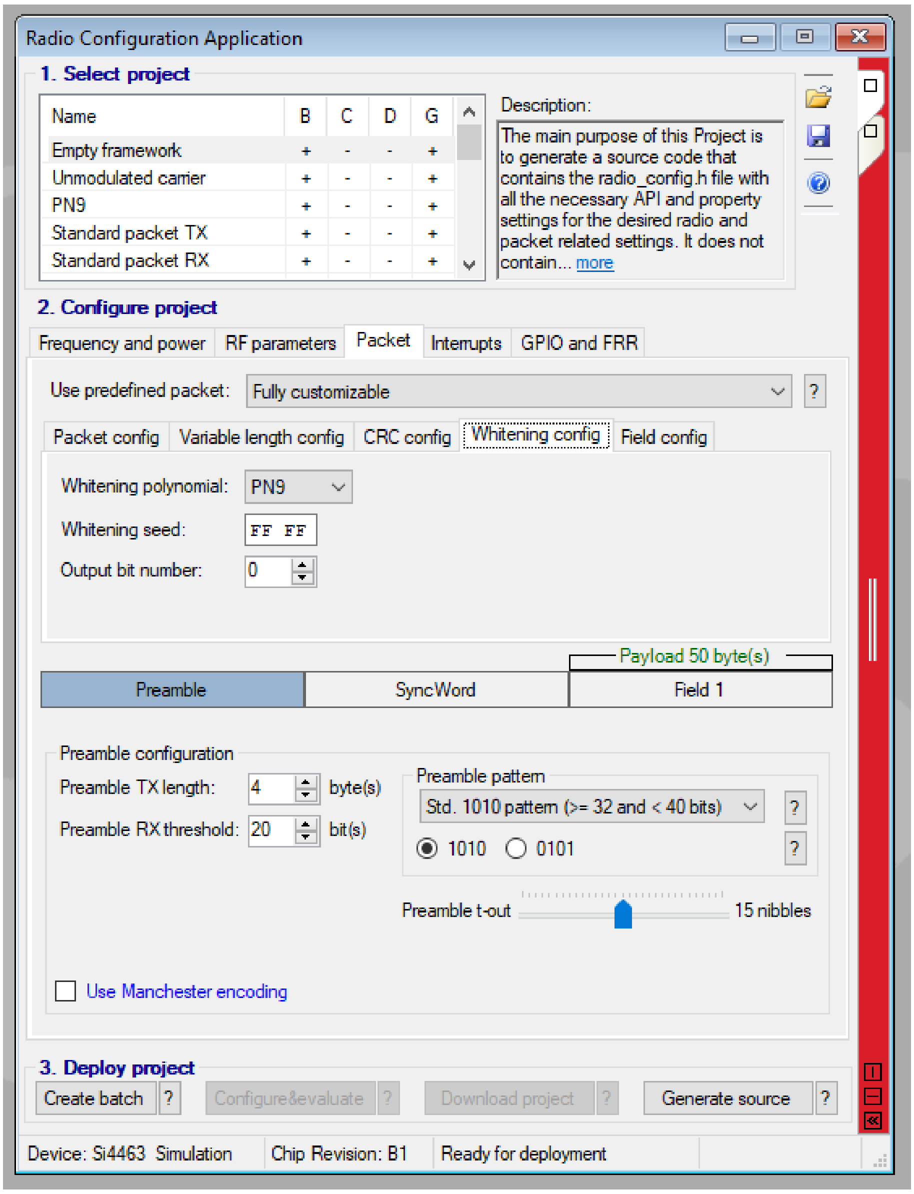


Figure A.11: Step 11 of the radio configuration.

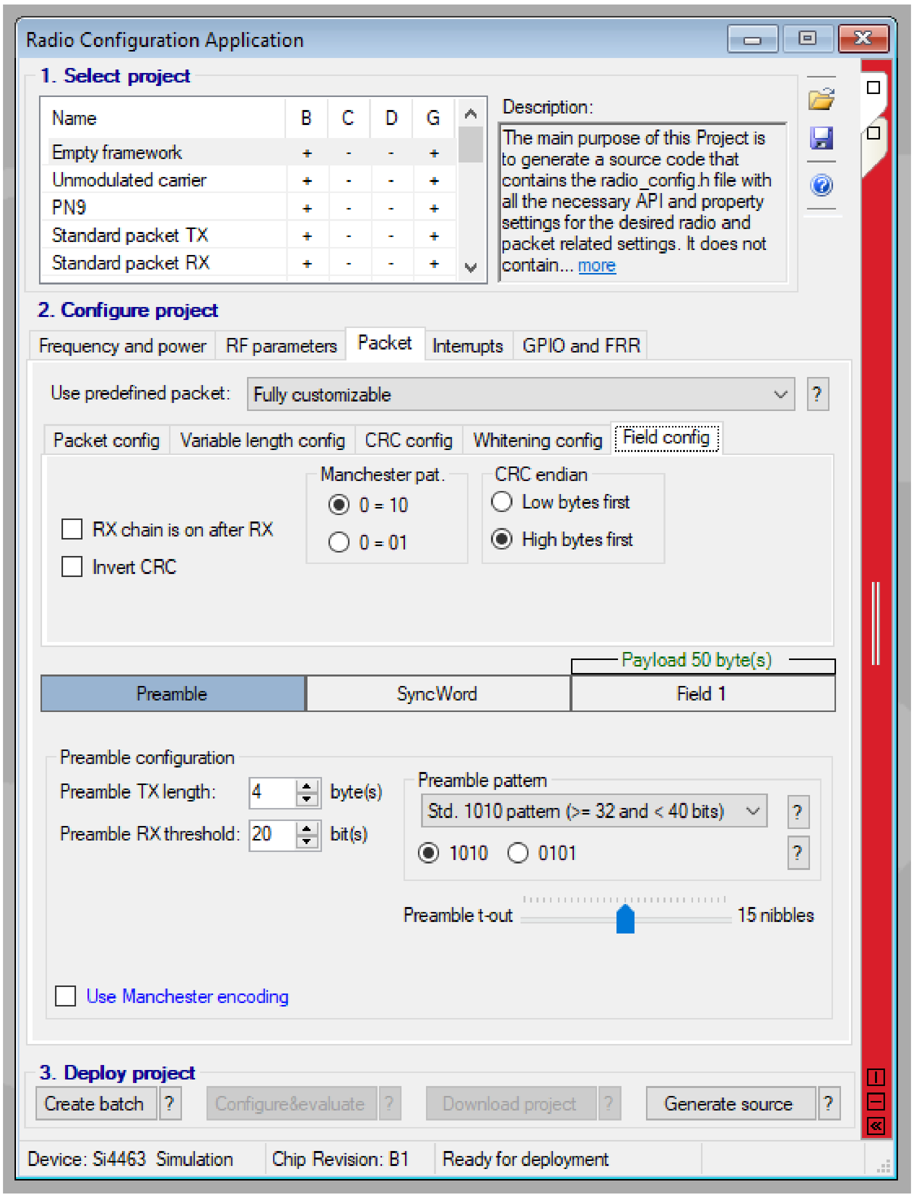


Figure A.12: Step 12 of the radio configuration.

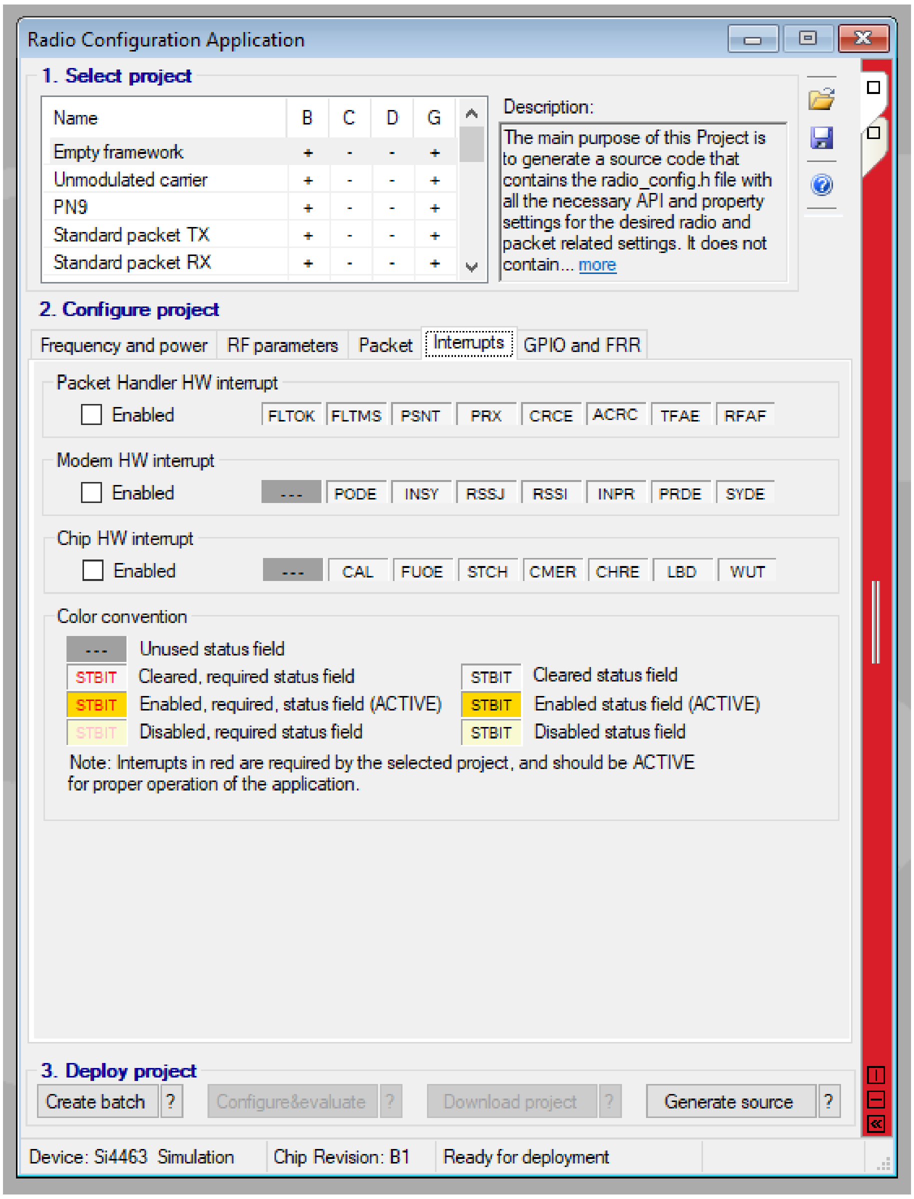


Figure A.13: Step 13 of the radio configuration.

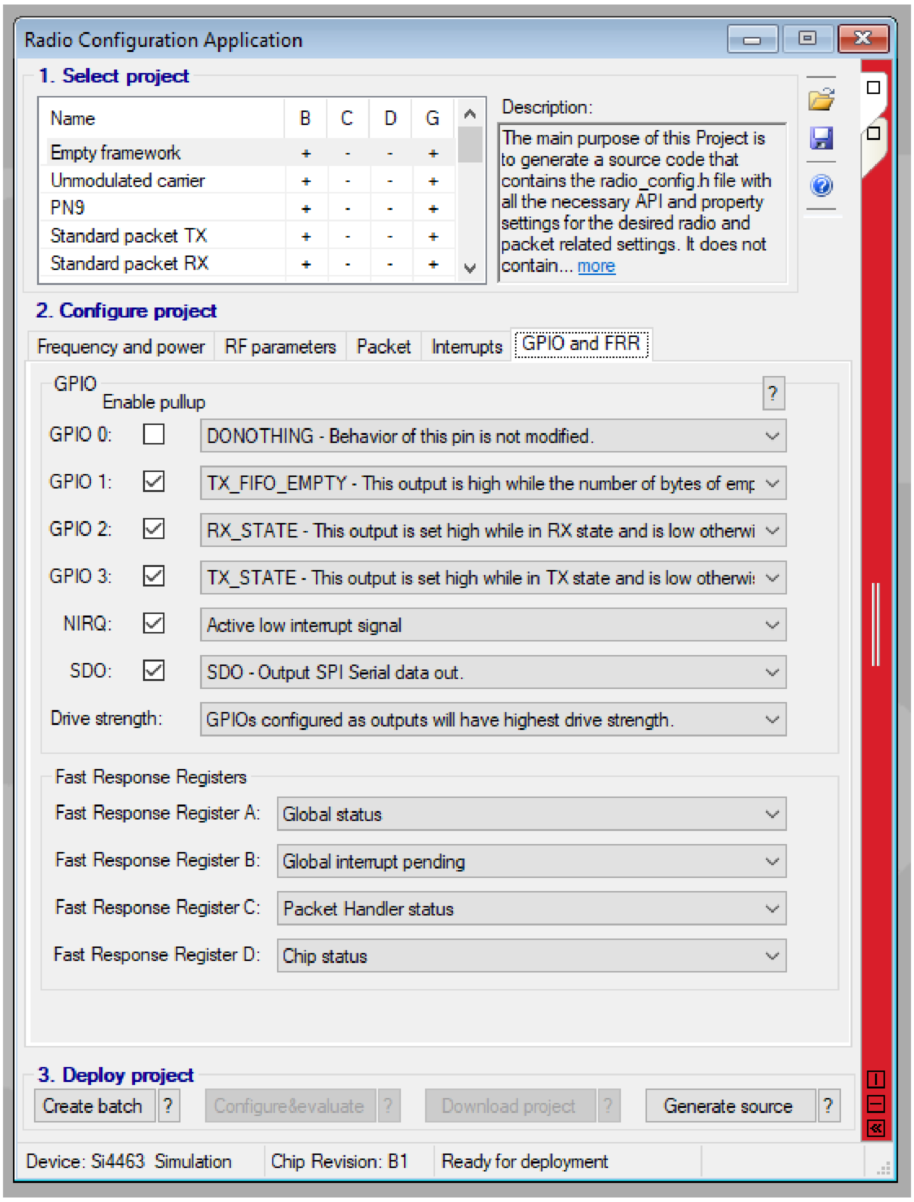


Figure A.14: Step 14 of the radio configuration.

APPENDIX B

Compiling and Building the Beacon Firmware

This tutorial is a reference to compile, build and flash the firmware of the beacon of the TTC module. All the software development was made using the Code Composer Studio (CCS) IDE, version 7.3.0. To load the code into the TTC MCU, the MSP-FET can be used.

Creating the Project

After the download and installation of the CCS, open it and create a new project with following steps:

1. Go to “File” -> “New” -> “CCS Project”.
2. The window from the figure B.1 will appear on the screen.
3. On “Target”, select MSP430F6659.
4. On “Project name”, enter the name of the project (It can be “beacon”, “beacon_firmware”, or whatever name you want).
5. On “Project templates and examples” box, select “Empty project”.
6. Click on “Finish”.

Now the beacon project with the correct parameters has been created, and the project source code files must be moved to the project directory. Copy the files inside the “beacon” folder from the TTC git repository to the project folder.

Compiling and Building

To compile and build the firmware, click on ”Project” -> ”Build Project”. If the process finish with no errors, the code is ready to go to the MCU of the beacon.

Flashing

To load the code into the board, connect the MSP-FET in the computer and follow the power-on tutorial.

With the board turned on and the MSP-FET connected, click on “Run” -> “Debug” or just press F11. If no errors occur, the firmware was loaded succesfully to the board.

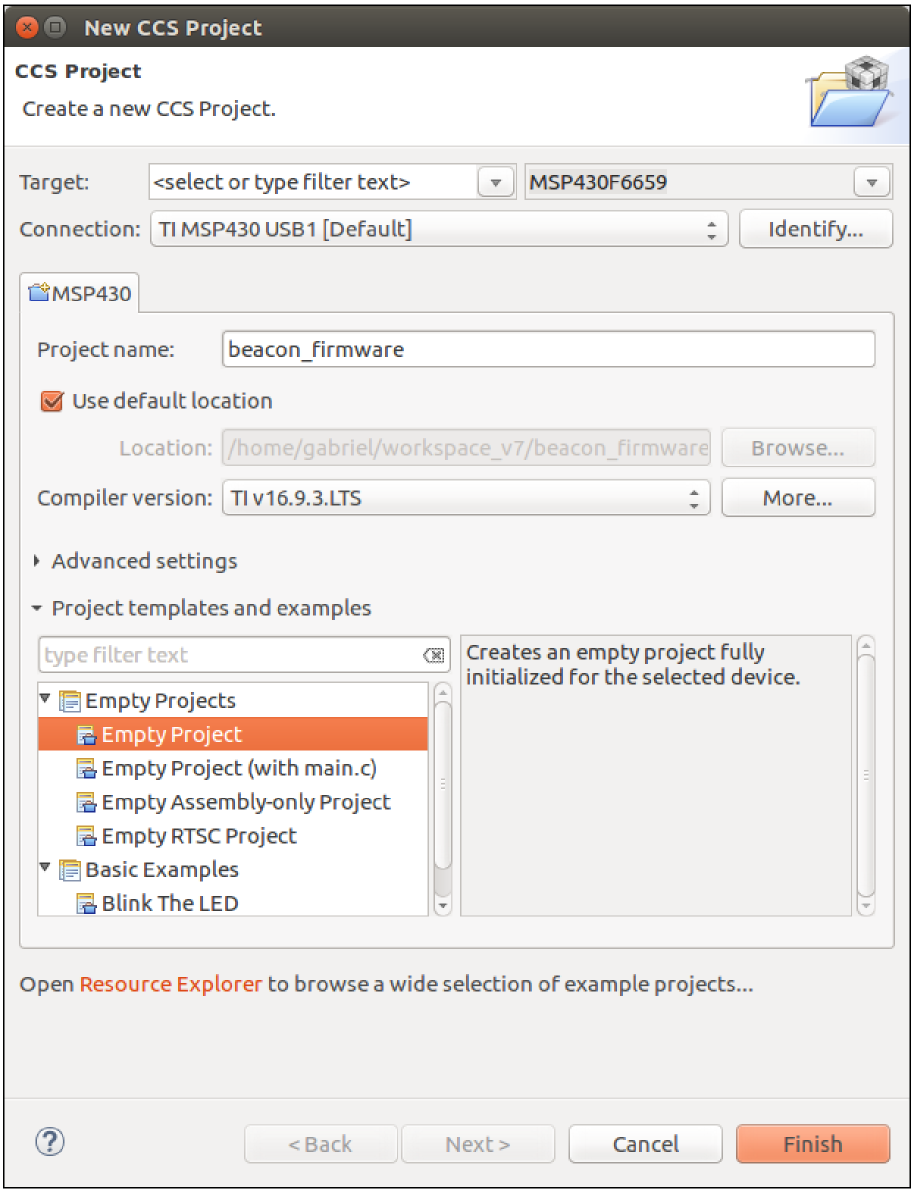


Figure B.1: New CCS project window.

APPENDIX C

Power-On the TTC Module

To power-on the TTC module, there are three different possibilities:

1. Using three different power sources: one for the MCU, one for the beacon radio module and one for the telemetry radio module.

2. Using two different power sources: one for each radio module (in this case, the beacon MCU is powered using the JTAG bus).

3. Using just one power source to all components (THIS METHOD IS NOT SAFE: there is no control over the power consumption of each component).

To test just the beacon, there is no need to power-on the telemetry radio module. Turning on the telemetry radio only makes sense if an external module is connected and controlling it (Like an OBC or an OBDH).

The connections reference of the TTC board can be found in External Connections.

Using an External Power Source for the Beacon MCU

Power-On the Beacon MCU

1. Set an output of a channel of the power source to $3,3\ V$ and $30\ mA$ (Cut-off current).

2. Connect the positive cable to the H2A-14 or H2B-14 pin of the PCI-104 connector.

3. Connect the ground cable to any GND pin of the PCI-104 connector.

4. Turn on this channel of the power source.

Power-On the Beacon Radio Module

1. Set another output of a channel of the power source to $5,0\ V$ and $500\ mA$ (Cut-off current).

2. Connect the positive cable to the H1A-26 pin of the PCI-104 connector.

3. Connect the ground cable to any GND pin of the PCI-104 connector.

4. Turn on this channel of the power source.

Power-On the Telemetry Radio Module

1. Set another output of a channel of the power source to $5,0\ V$ and $500\ mA$ (Cut-off current).
2. Connect the positive cable to the H1A-25 pin of the PCI-104 connector.
3. Connect the ground cable to any GND pin of the PCI-104 connector.
4. Turn on this channel of the power source.

Using the JTAG as Power Source for the Beacon MCU

In this case, to power on the beacon MCU, just connect a jumper in the P4 connector of the board and after, connect a MSP-FET debbuger to the JTAG connector.

The procedure to power-on the radio modules is the same as above.

APPENDIX D

Receiving the Beacon Data

Required Softwares

- GQRX (For just receive the signal).
- RTL-SDR drivers (The GRQX will install all the required drivers).
- FloripaSat GRS (For receive the signal and data).

Supported SDRs

- RTL-SDR
- FunCube Dongle Pro+

Receiving the Beacon Signal

In this method, you will not be able to see the beacon data in real time. This only shows the presence of the beacon signals in the air.

The instructions to receive the beacon signal in the GQRX are available bellow:

1. Open the GQRX software.
2. Configure it for your SDR model.
3. Press the power button on the upper left corner.
4. Select the frequency to 145,9 MHz.
5. Wait the spectrum of the beacon signal appear.

The figure D.1 illustrates the reception of the beacon signal in the GQRX software.

Receiving the Beacon Data

Using this method, you will be able to see the beacon (and the telemetry) data in real time on the computer screen. For that, the FloripaSat GRS sofware is required.

1. Open the FloripaSat GRS software.

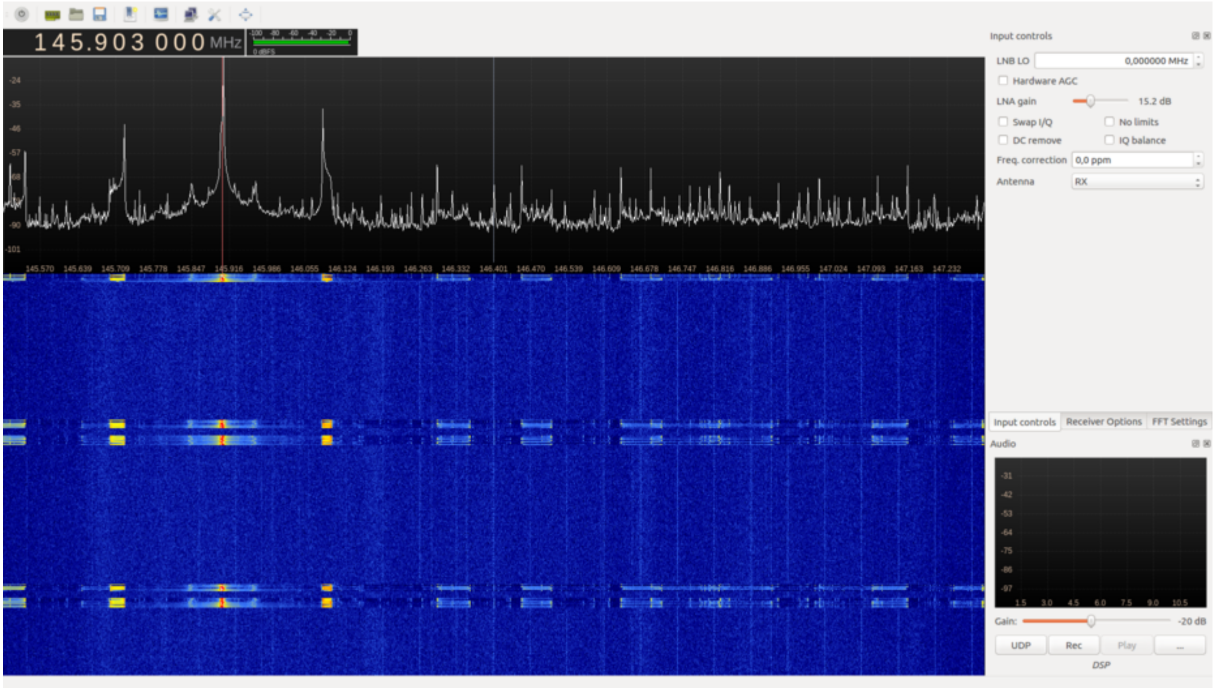


Figure D.1: Beacon signal in GQRX.

2. Connect a the SDR device.
3. Select the source of the signal.
4. Press the play button in the beacon frame.
5. A new window will appear in the screen, with the FFT and Waterfall plot of the received signal (in real time) of the connected SDR.
6. Power on the beacon module of the TTC board.
7. Adjust the frequency of the receive to the center frequency of the beacon signal (Use the FFT and the Waterfall plot for that).
8. With the correct frequency sinthonization, the beacon data should appear in the main window of the FloripaSat GRS softwate.

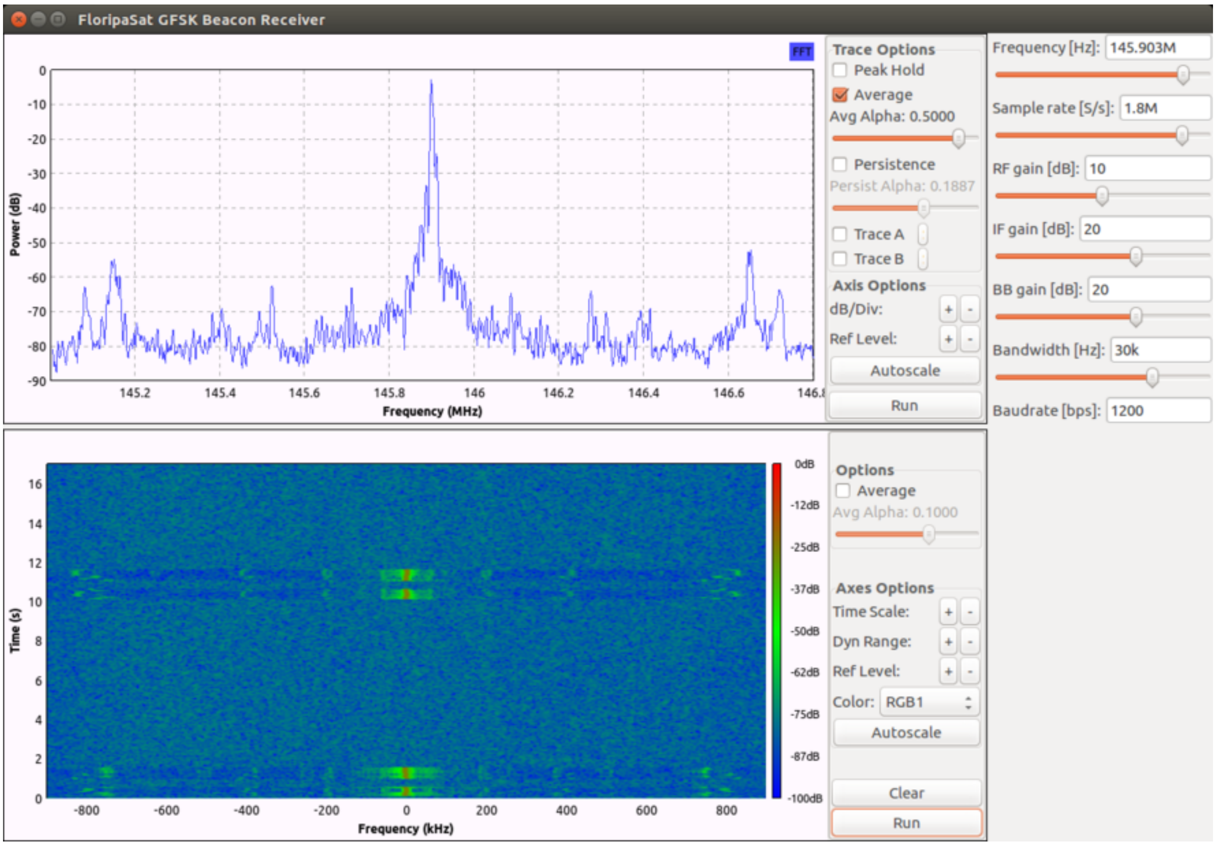


Figure D.2: Beacon signal in the GNURadio receiver from the FloripaSat GRS.

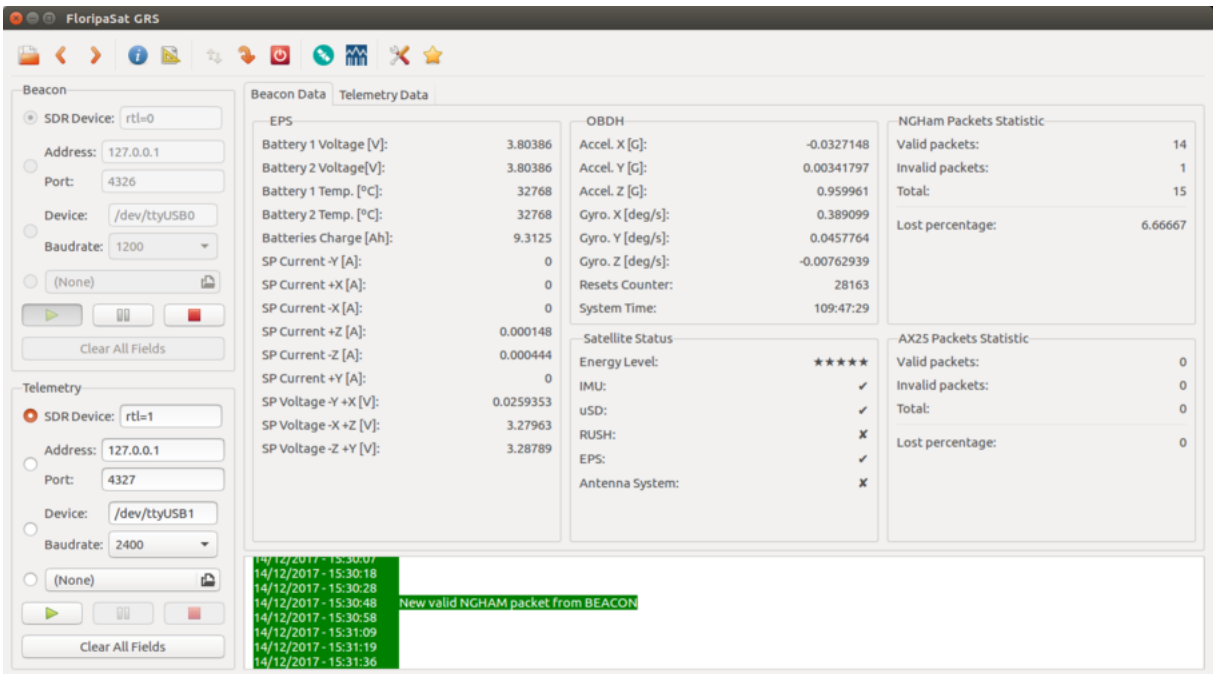


Figure D.3: Beacon data in the FloripaSat GRS.