

SpaceLab TTC 2.0

Miguel Boing, Gabriel M. Marcelino

2022 August 24

SpaceLab - UFSC

Summary

Project Overview

Hardware

Firmware

Operation

Documentation

Management

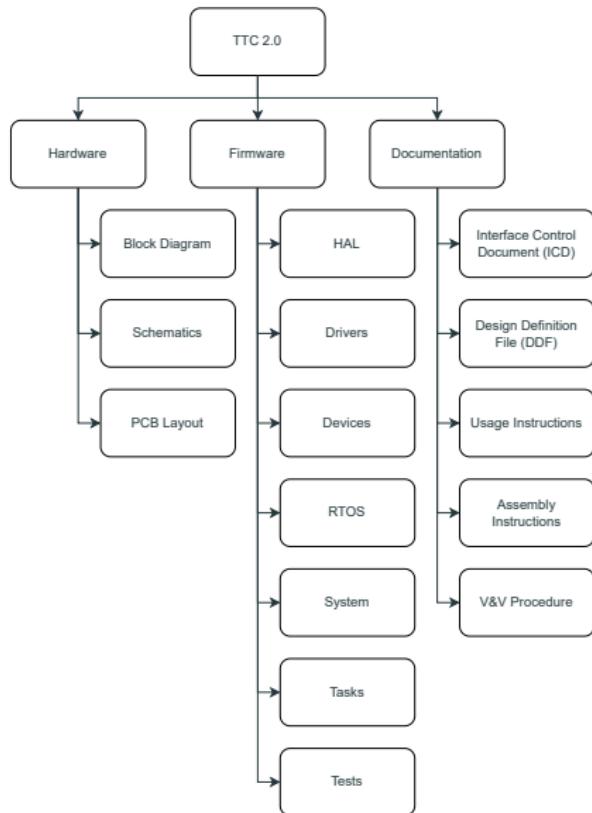
Project Overview

Overview

- Telemetry, Tracking and Command (TTC) module for small satellites like CubeSats
- Project name: “*TTC 2.0*”
- Fully open source
- Based on FloripaSat-1 heritage



Product Tree

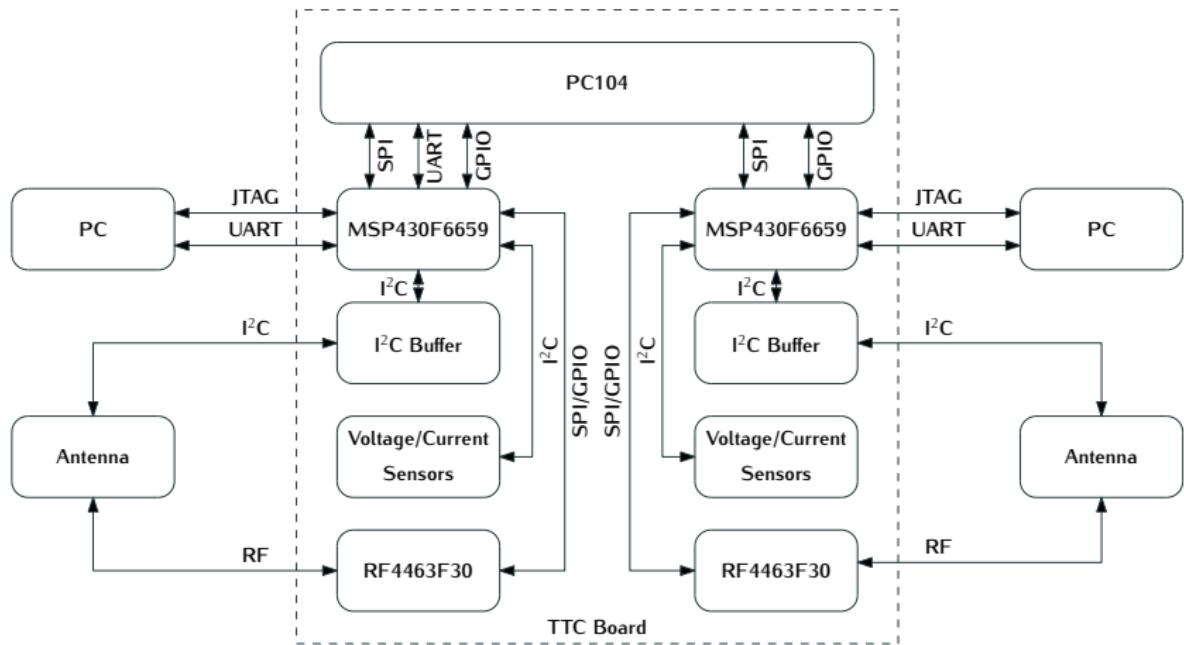


Hardware

Specifications

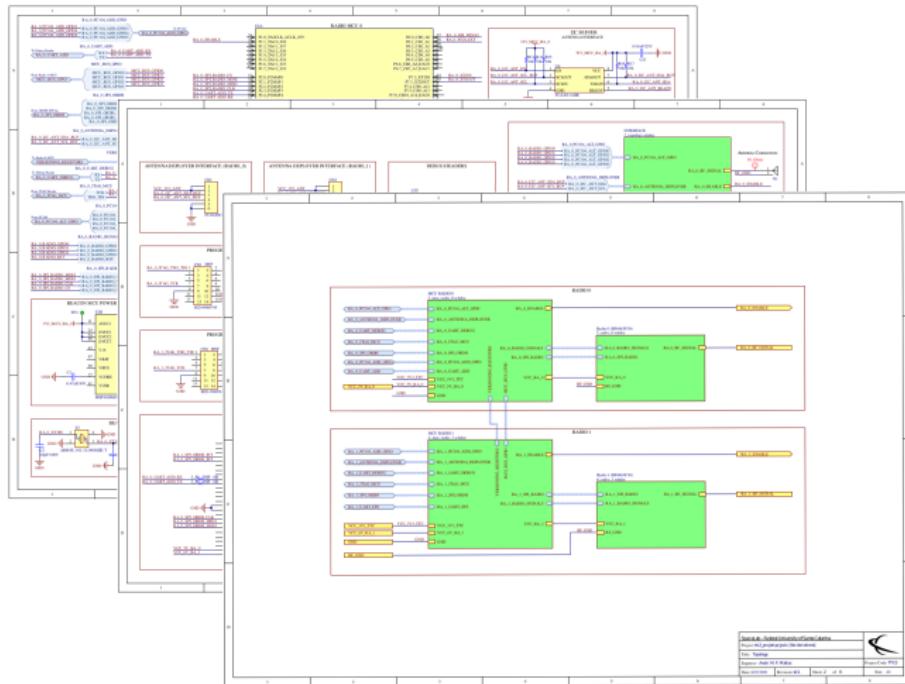
- **Microcontroller:** MSP430F6659/MSP430F5659
- **Clock:** 32 MHz
- **Memories:**
 - *RAM:* 64 kB (SRAM)
 - *Flash:* 512 kB (code)
- **Sensors:** Voltage, current and temperature
- **Modulation:** (G)FSK or (G)MSK
- **Baudrate:** 1200 to 9600 bps
- **Frequency:** 145-146 MHz, 435-438 MHz or 450 MHz bands
- **Protocol:** NGHam
- **Interfaces:** UART, I²C and SPI
- **Mass:** 73 g
- PC-104 compatible

Electrical Block Diagram



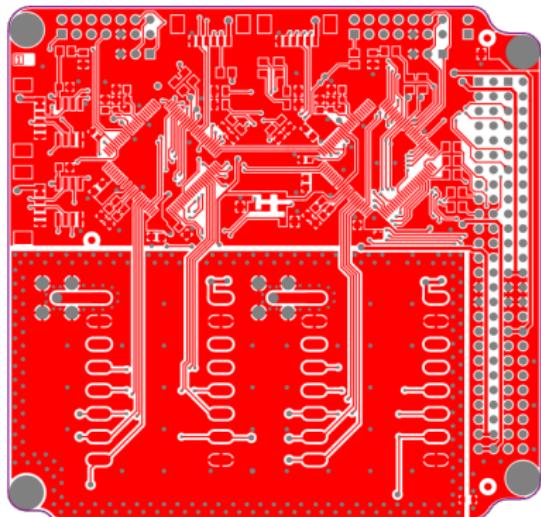
Schematics

Available at: <https://github.com/spacelab-ufsc/ttc2>

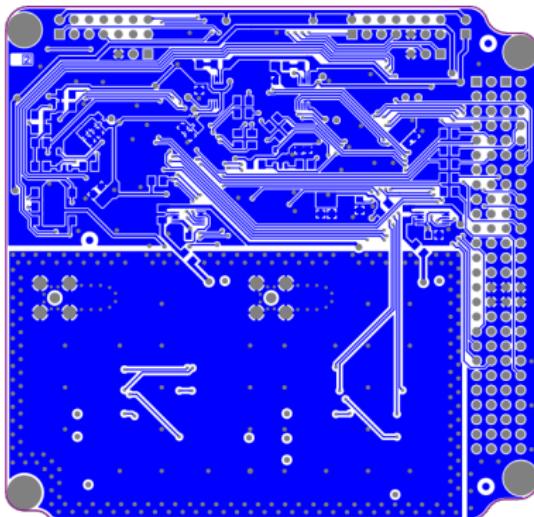


PCB Layout

Available at: <https://github.com/spacelab-ufsc/ttc2>



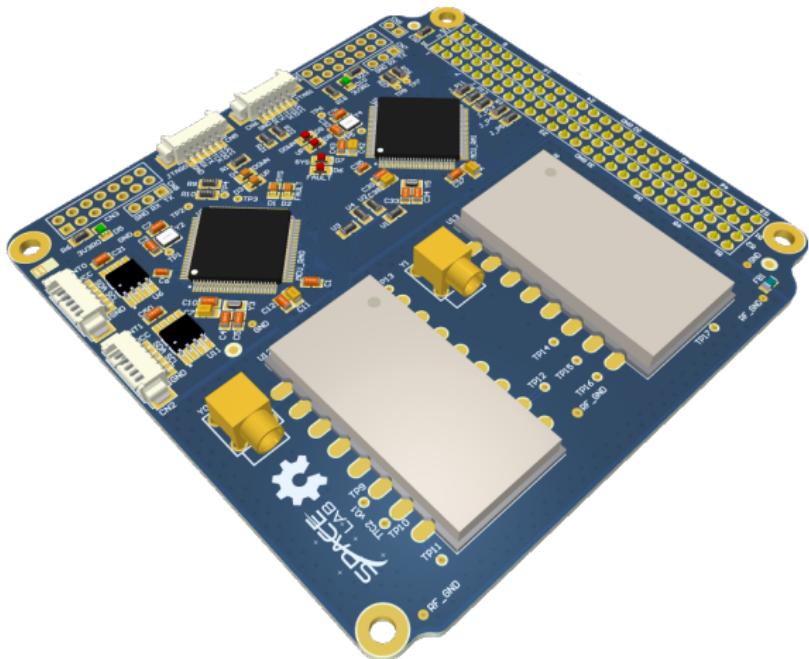
Top side



Bottom side

3D Model

Available at: <https://github.com/spacelab-ufsc/ttc2>



Power Consumption

- Three separated voltage inputs: 3V3, 5V and 6V (microcontrollers and transceivers)
- Power consumption¹:
 - Stand-by mode \cong 83 mW
 - Transmission mode \cong 2800 mW
 - Reception mode \cong 163 mW

¹For one transceiver (there is two in TTC 2.0)

Electrical Interfaces: PC-104

Pin [A-B]	H1A	H1B	H2A	H2B
1-2	-	-	-	-
3-4	-	-	-	-
5-6	-	-	RA_1_UART_RX	-
7-8	GPIO_6	GPIO_7	RA_1_UART_TX	GPIO_0
9-10	RA_1_SPI_INT	RA_1_EN	-	-
11-12	RA_0_SPI_INT	RA_0_EN	RA_1_SPI_MOSI	RA_1_SPI_CLK
13-14	-	-	RA_1_SPI_CS	RA_1_SPI_MISO
15-16	-	-	-	-
17-18	-	-	-	GPIO_1
19-20	-	GPIO_2	-	GPIO_3
21-22	-	-	-	GPIO_4
23-24	-	-	-	-
25-26	-	-	-	-
27-28	-	-	VCC_3V3	VCC_3V3
29-30	GND	GND	GND	GND
31-32	GND	GND	GND	GND
33-34	-	-	-	-
35-36	RA_0_SPI_CLK	-	VCC_3V3_ANT	VCC_3V3_ANT
37-38	RA_0_SPI_MISO	-	-	-
39-40	RA_0_SPI_MOSI	RA_0_SPI_CS	-	-
41-42	-	-	-	GPIO_5
43-44	-	-	-	-
45-46	-	-	-	-
47-48	-	-	-	-
49-50	VCC_5V_RA_0	VCC_5V_RA_0	-	-
51-52	VCC_6V_RA_1	VCC_6V_RA_1	-	-

Other Electrical Interfaces

Connector	Interface	Type	Pins
CN1	I ² C	PicoBlade	3V3 3V3 I2C_SDA I2C_SCL GND GND
CN2	I ² C	PicoBlade	3V3 3V3 I2C_SDA I2C_SCL GND GND

Other Electrical Interfaces

Connector	Interface	Type	Pins
CN3	JTAG	Pin Header	TDO_TDI
			3V3
			None
			TCK
			None
			GND
			None
			UART_TX
			None
			UART_RX

Other Electrical Interfaces

Connector	Interface	Type	Pins
CN4	JTAG	Pin Header	TDO_TDI
			3V3
			None
			TCK
			None
			GND
			None
			UART_TX
			None
			UART_RX

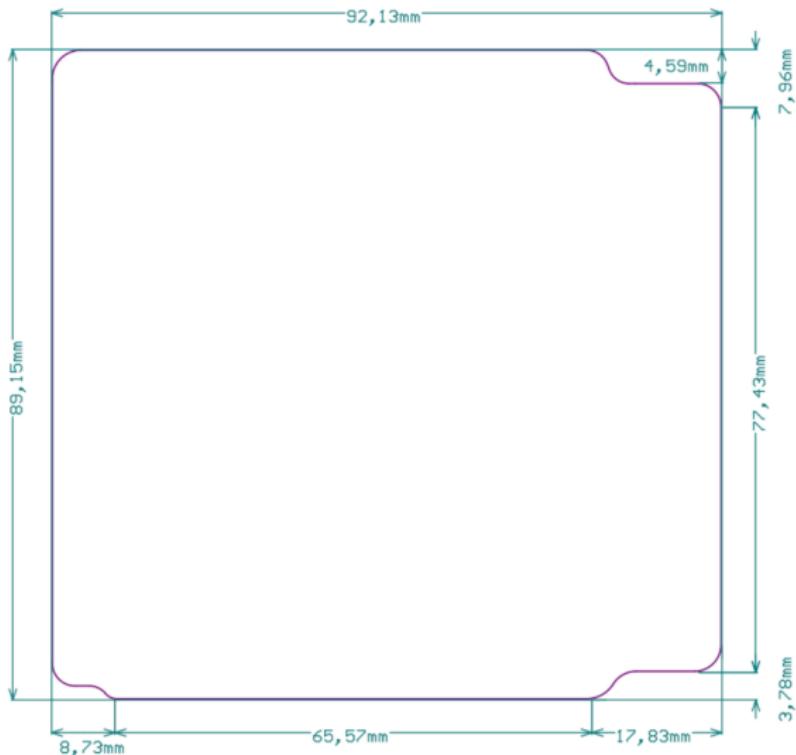
Other Electrical Interfaces

Connector	Interface	Type	Pins
CN5	JTAG	PicoBlade	3V3 TDO_TDI TCK UART_TX UART_RX GND
CN6	JTAG	PicoBlade	3V3 TDO_TDI TCK UART_TX UART_RX GND
CN7	UART	PinHeader	TX RX GND
CN8	UART	PicoBlade	TX RX GND
CN9	Jumper	Pin Header	3V3 -

Other Electrical Interfaces

Connector	Interface	Type	Pins
Y0	RF	MCX	RF_SIGNAL RF_GND
Y1	RF	MCX	RF_SIGNAL RF_GND

Dimensions



Voltage/Current Sensors

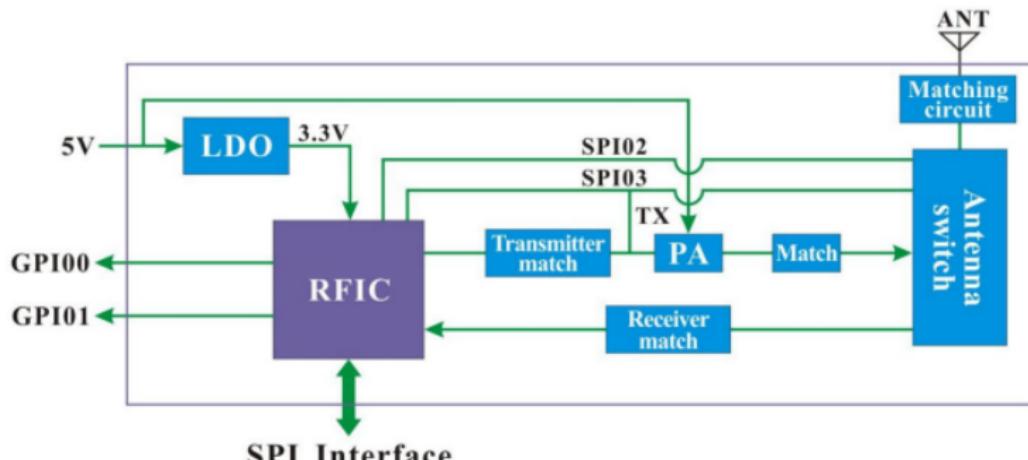
- Voltage, current and power sensor
- Quantity: 4 (one per radio module and one per microncontroller circuit)
- Texas Instruments INA226AQDGSRQ1
- I²C Interface

External Watchdog

- IC: Texas Instruments TPS3823
- Voltage monitor with a watchdog timer feature
- Timeout period: 1600 *ms*
- Reset period: 100 *ms*

Transceiver

- [NiceRF RF4463F30](#)
- Based on [Si4463](#) transceiver
- Integrated power amplifier and RF switch (single antenna)
- Half-Duplex
- Output power: 30 dBm
- SPI Interface



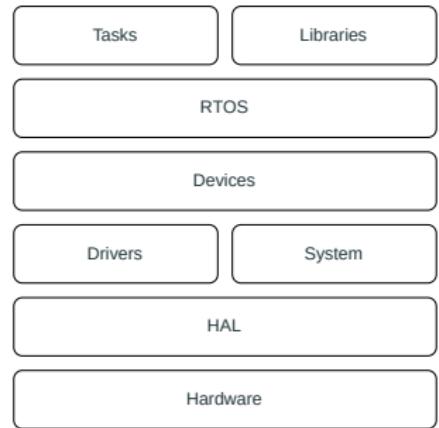
Flight Model Specs. and Preparation

- **PCB specs.:** IPC 6012 Class 3
- **PCB thickness:** 1,6 mm
- **Material:** TG170 FR-4
- **Surface finish:** ENIG
- **Board finish:** Conformal coating application

Firmware

Overview

- Language: C
- OS: FreeRTOS v10.2.1



System Tasks

Name	Priority	Initial delay [ms]	Period [ms]	Stack [bytes]
Command Processing	Highest	0	100	500
Heartbeat	Lowest	0	500	160
Housekeeping	Medium	2000	10000	160
Radio Reset	High	60000	60000	128
Startup	Medium	0	Aperiodic	128
System Reset	Medium	0	36000000	128
Time Control	Medium	1000	1000	128
Uplink	Highest	2000	500	500
Watchdog Reset	Lowest	0	100	150

System Tasks

- **Command Processing:** Process incoming commands (physical interfaces).
- **Heartbeat:** Blinks a status LED at a rate of 1 Hz.
- **Housekeeping:** This task manages the general operation of the TTC.
- **Radio Reset:** Resets the radio at 600 seconds.
- **Startup:** Initializes the TTC 2.0 module.
- **System Reset:** Resets the microcontroller by software at every 10 hours.
- **Time Control:** Manages the system time.
- **Uplink:** Monitors radio for upcoming packages and stores it in memory.
- **Watchdog Reset:** Resets the watchdog timers at every 100 milliseconds.

System Parameters

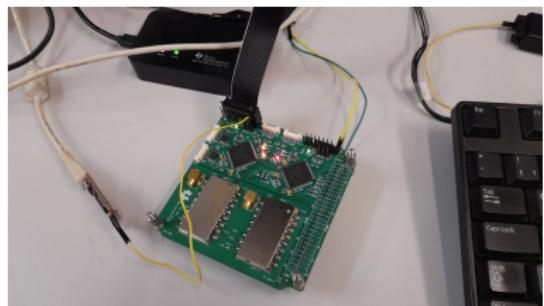
ID	Name/Description	Type
0	Device ID (0xCC2A or 0xCC2B)	uint16
1	Hardware version	uint8
2	Firmware version (ex.: "v1.2.3" = 0x00010203)	uint32
3	Time counter in milliseconds	uint32
4	Reset counter	uint16
5	Last reset cause:	uint8
6	Input voltage of the μ C in mV	uint16
7	Input current of the μ C in mA	uint16
8	Temperature of the μ C in K	uint16
9	Input voltage of the radio in mV	uint16
10	Input current of the radio in mA	uint16
11	Temperature of the radio in K	uint16
12	Last valid command (uplink packet ID)	uint8
13	RSSI of the last valid telecommand	uint16
14	Temperature of the antenna module in K	uint16
15	Antenna module status bits	uint16
16	Antenna deployment status	uint8
17	Antenna deployment hibernation	uint8
18	TX enable	uint8
19	TX packet counter	uint32
20	RX packet counter (valid packets)	uint32
21	TX packets available in the FIFO buffer	uint8
22	RX packets available in the FIFO buffer	uint8
23	Number of bytes of the first available packet in the RX buffer	uint16

Commands

ID	Command	Parameters
1	Read parameter	Parameter ID
2	Write parameter	Parameter ID + Parameter value
3	Transmit packet	Data to transmit (payload of the NGHam packet)
4	Read first available packet	None

Development Environment

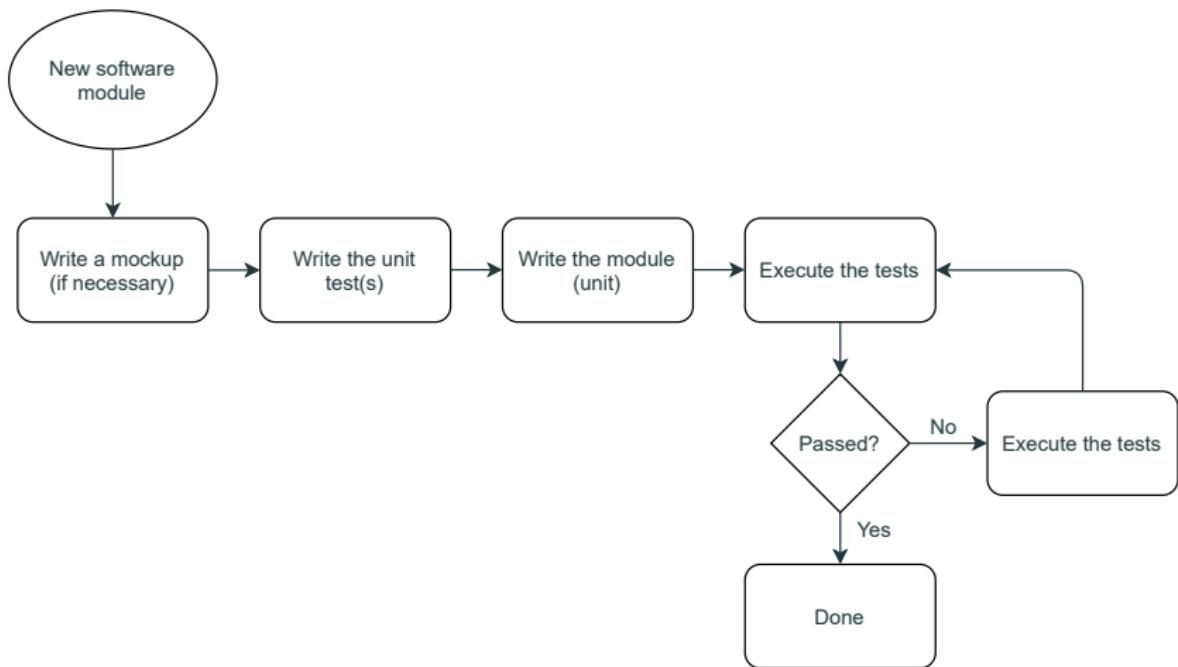
- Hardware: Engineering Model (EM)
- Programmer: Texas Instruments MSP-FET programmer
- IDE/Compiler: Code Composer Studio



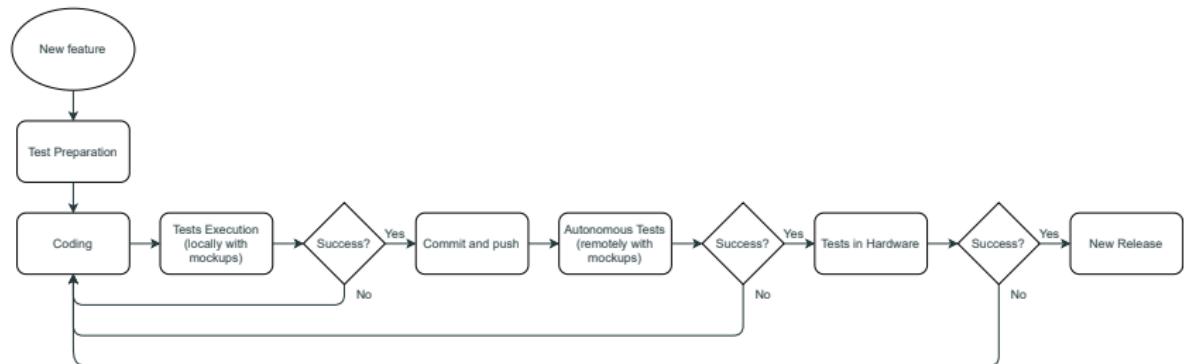
Verification & Validation

- Unit tests framework: [CMocka](#)
- Static analysis tool: [CppCheck](#)
- Code style standard: [MISRA-C 2012](#)

Verification & Validation: TDD Flow



Verification & Validation: Development Flow



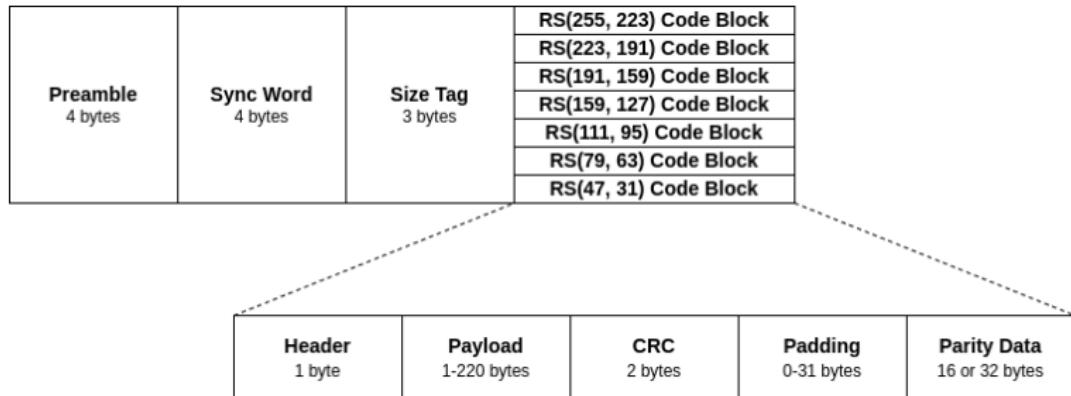
Operation

Operation

- By default, the TTC will always operate in RX mode
- If requested by an external device, a packet is transmitted (TX mode)
- An external device can read or write from/to the internal parameters
- The received packets are stored in a queue with five positions (up to five packets can be stored)
- When requested, an external device can read the decoded packets from the queue

RF Protocol: NGHam

- With FEC (Reed-Solomon with seven different schemes)
- Up to 220 bytes of data per packet



NGHam: Sizes

Size Num.	Tag	RS Config.	Parity Bytes	Max. Data Bytes
1	59, 73, 205	RS(47, 31)	16	28
2	77, 218, 87	RS(79, 63)	16	60
3	118, 147, 154	RS(111, 95)	16	92
4	155, 180, 174	RS(159, 127)	32	124
5	160, 253, 99	RS(191, 159)	32	156
6	214, 110, 249	RS(223, 191)	32	188
7	237, 39, 52	RS(255, 223)	32	220

- **Preamble:** $4 \times 0xAA$
- **Sync. Word:** 0x5D, 0xE6, 0x2A, 0x7E
- **CRC:**
 - *Polynomial:* 0x1021
 - *Initial value:* 0xFFFF
 - *Final XOR value:* 0xFFFF
- **Reed-Solomon:**
 - *Symbol size:* 8
 - *GF polynomial:* 0x187 (coefficients form)
 - *First root of RS code generator polynomial:* 112 (index form)
 - *Primitive element:* 11
 - *Number of roots:* 16 or 32

- Before transmitting a packet, the RS code block is scrambled by making a byte xor operation with a pre-generated table based on the polynomial $x^8 + x^7 + x^5 + x^3 + 1$ (defined in the CCSDS 131.0-B-3 standard)
- When the receiver receives a packet, it also performs the same operation to de-scramble the RS code block and get the original content of the RS part of the packet
- More information: [CCSDS 131.0-B-3](#) (section 8.3)

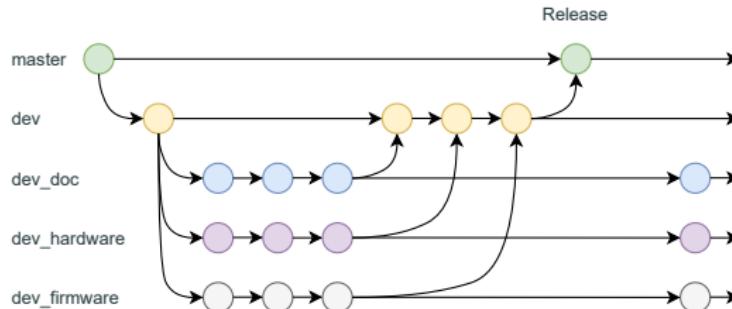
Documentation

- User manual (PDF)
- This presentation
- Schematics
- Firmware: Doxygen

Management

Project Management

- Activities and tasks: Public GitHub issues/project
- Source files and versioning control: **Git repository** with three development branches:
 - *dev_doc*: Documentation
 - *dev_hardware*: Hardware project
 - *dev_firmware*: Firmware project



Thanks!

