SPACELAB.OBDH2.2019.10.001 REV A. ISSUE 0.1

OBDH 2.0 Documentation

OBDH 2.0 Documentation
SpaceLab, Universidade Federal de Santa Catarina, Florianópolis - Brazil

October 2019

OBDH 2.0 Documentation
October, 2019

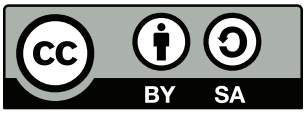
Project Chief:
Eduardo Augusto Bezerra

Authors:
Gabriel Mariano Marcelino

Contributing Authors:

Revision Control:

Version	Author	Changes	Date
0.1	Gabriel Mariano Marcelino	Document creation	10/2019



List of Figures

List of Tables

Contents

CHAPTER 1

Introduction

- Main target: FloripaSat-2
- Improved version of the OBDH from FloripaSat-1
- Open source software (GPLv3 license)
- Open source hardware (GPLv3 license)
- RTOS
- Low power MCU

CHAPTER 2

Requirements

CHAPTER 3

System Overview

3.1 Block Diagram

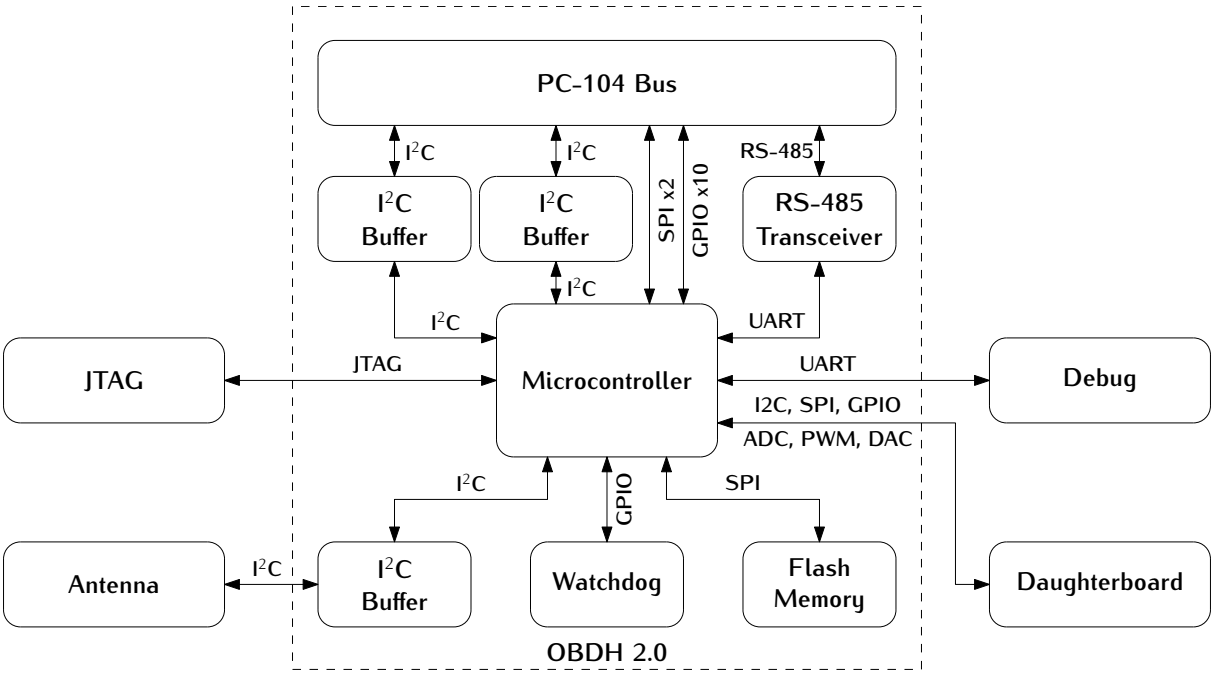


Figure 3.1: OBDH 2.0 Block diagram.

3.2 System Layers

3.3 Operation

3.3.1 Status LEDs

On the development version of the board, there are eight LEDs that indicates some behaviours of the systems. This set of LEDs can be seen on Figure ??.

A description of each of these LEDs are available below:

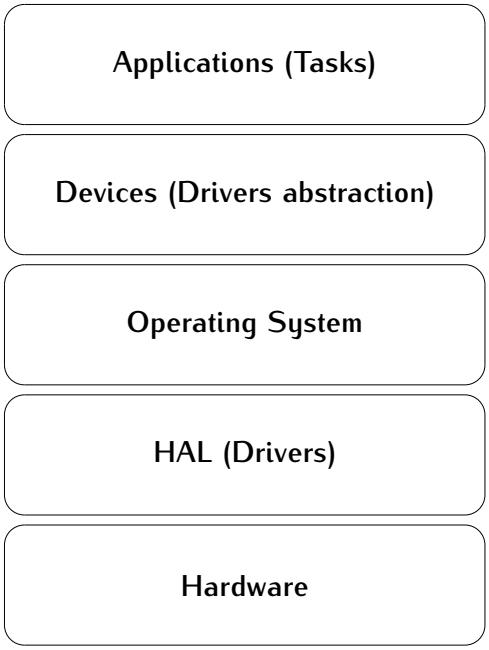


Figure 3.2: System layers.

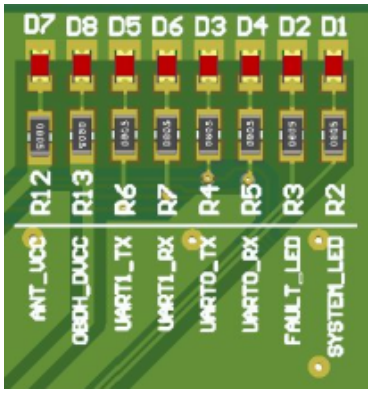


Figure 3.3: Available status LEDs.

- **D1 - System LED**: Heartbeat of the system. Blinks at a frequency of 1 Hz when the system is running properly.
- **D2 - Fault LED**: Indicates a critical fault in the system.
- **D3 - UART0 TX**: Blinks when data is being transmitted over the UART0 port.
- **D4 - UART0 RX**: Blinks when data is being received over the UART0 port.
- **D5 - UART1 TX**: Blinks when data is being transmitted over that UART1 port.
- **D6 - UART1 RX**: Blinks when data is being received over the UART1 port.
- **D7 - Antenna VCC**: Indicates that the antenna module board is being power sourced.
- **D8 - OBDH VCC**: Indicates that the OBDH board is being power sourced.

These LEDs are not mounted in the flight version of the module.

CHAPTER 4

Hardware

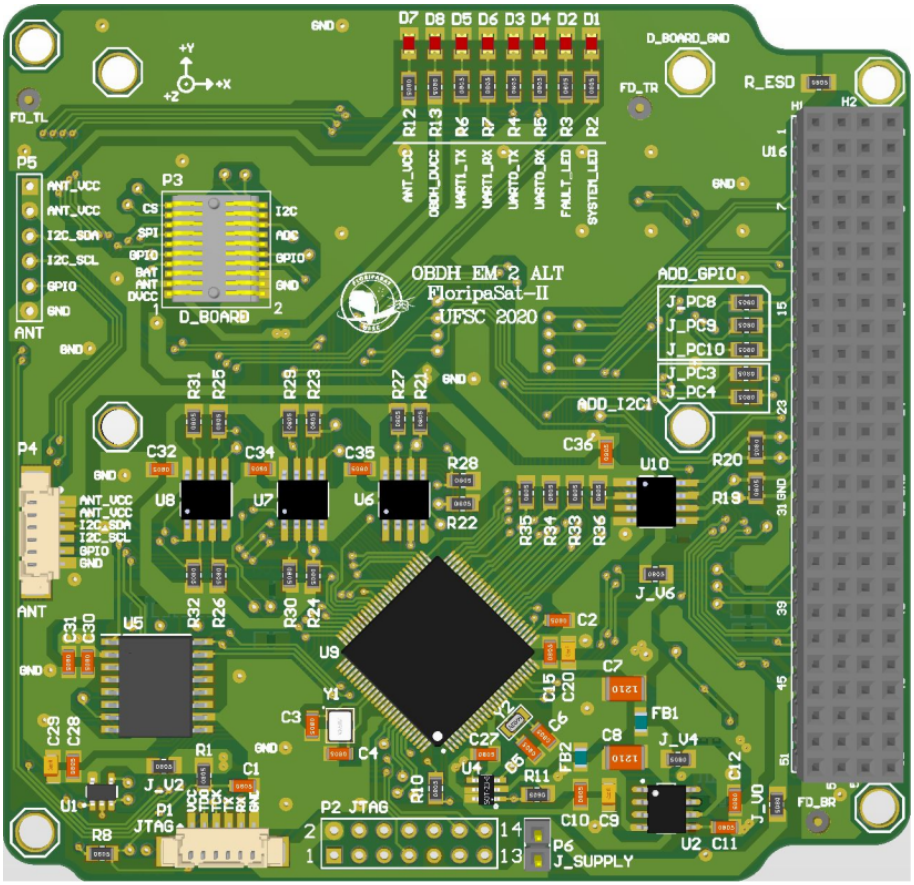


Figure 4.1: Top side of the PCB.

4.1 Interfaces

Currently, "*Payload 1*" and "*Payload 2*" are "*Payload-X*" and "*Payload EDC*" respectively.

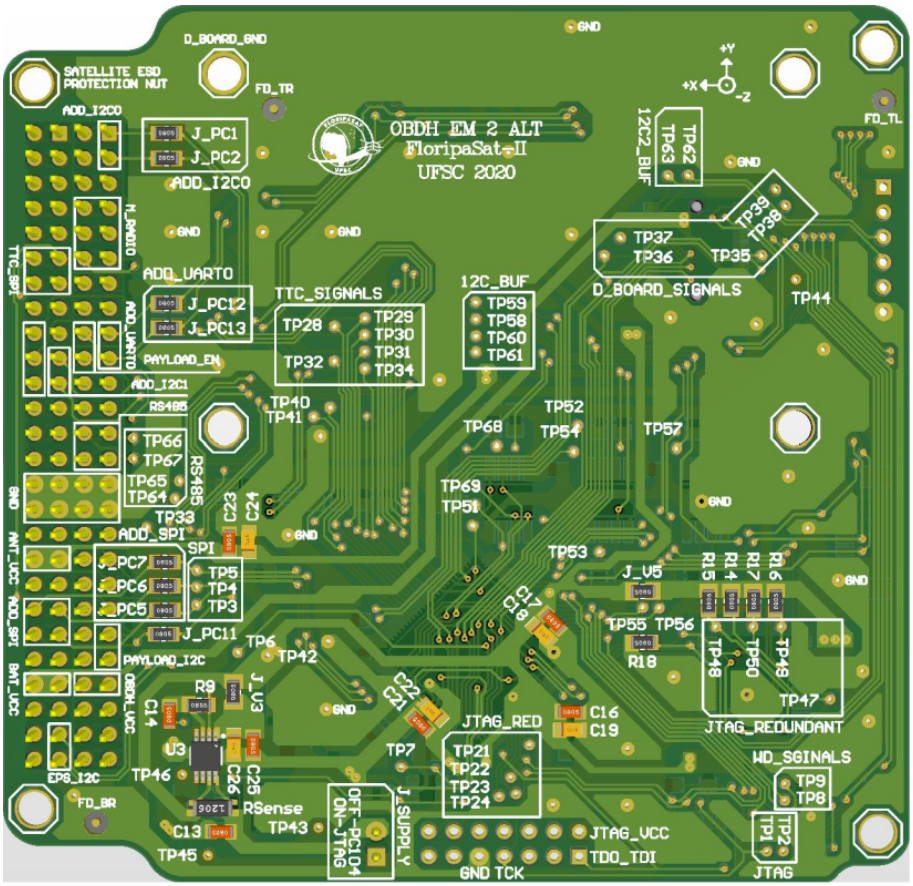


Figure 4.2: Bottom side of the PCB.

4.2 External Connectors

4.2.1 PC-104

4.2.2 Antenna Module

4.2.3 Programmer

4.2.4 Debug Interface

4.2.5 Daughterboard

The daughterboard interface uses the Samtec FSI-110-D connector. An illustration of the connector can be seen in the Figure 4.6. A picture of this connector on the PCB can be seen in Figure 4.7.

The pinout of the daughterboard interface are available in the Table 4.2.

Guidelines

The recommended shape and size of the daughterboard can be seen in the Figure 4.8.

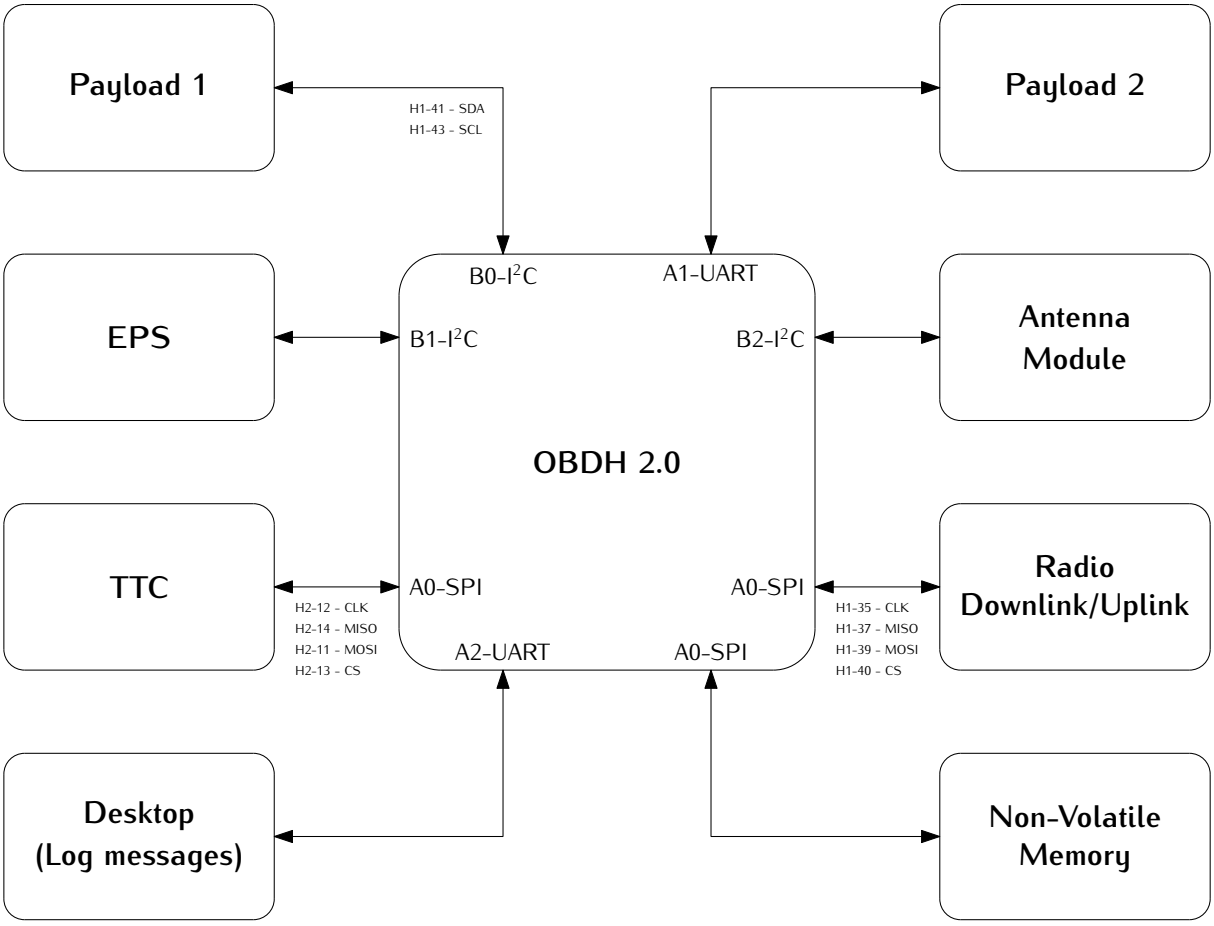
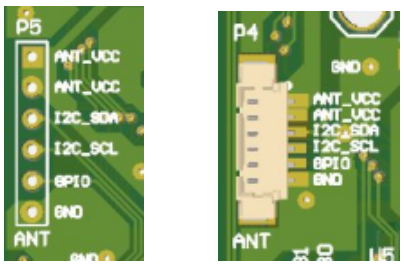


Figure 4.3: Interfaces diagram.



(a) Debug interface of the antenna module.

(b) Main interface of the antenna module.

Figure 4.4: Antenna module conectors.

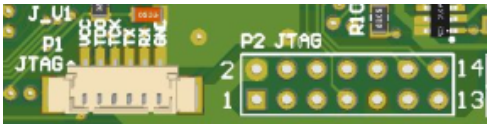


Figure 4.5: JTAG connectors (P1 and P2).

Pin [A-B]	*H1A*	*H1B*	*H2A*	*H2B*
1-2	-	-	-	-
3-4	-	-	-	-
5-6	GPIO_0	GPIO_1	-	-
7-8	GPIO_2	GPIO_3	-	-
9-10	GPIO_4	-	-	-
11-12	GPIO_5	GPIO_6	SPI_0_MOSI	SPI_0_CLK
13-14	GPIO_7	-	SPI_0_CS_1	SPI_0_MISO
15-16	-	-	-	-
17-18	-	GPIO_8	-	-
19-20	-	GPIO_9	-	-
21-22	-	-	-	-
23-24	-	-	-	-
25-26	RS485_TX+	RS485_TX-	-	-
27-28	RS485_RX+	RS485_RX-	-	-
29-30	GND	GND	GND	GND
31-32	GND	GND	GND	GND
33-34	-	-	-	-
35-36	SPI_0_CLK	-	VCC_3V3_ANT	VCC_3V3_ANT
37-38	SPI_0_MISO	-	-	-
39-40	SPI_0_MOSI	SPI_0_CS_0	-	-
41-42	I2C_0_SDA	-	-	-
43-44	I2C_0_SCL	-	-	-
45-46	VCC_3V3	VCC_3V3	VCC_BAT	VCC_BAT
47-48	-	-	-	-
49-50	-	-	-	-
51-52	-	-	-	-

Table 4.1: PC-104 connector pinout.

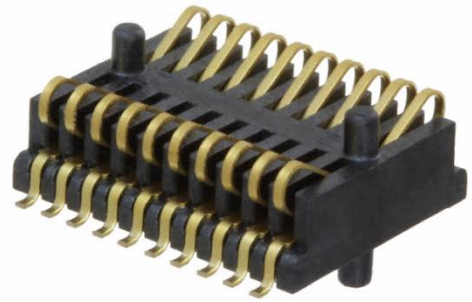


Figure 4.6: Samtec FSI-110-03-G-D-AD connector.

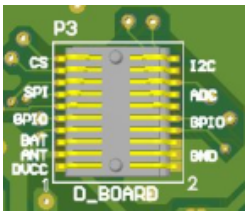


Figure 4.7: Daughterboard connector (P3).

Pin [A-B]	*Row A*	*Row B*
1-2	VCC_3V3	GND
3-4	VCC_3V3_ANT	GND
5-6	VCC_BAT	GND
7-8	GPIO_0	GPIO_1
9-10	GPIO_2	GPIO_3
11-12	SPI_0_CLK	ADC_0
13-14	SPI_0_MISO	ADC_1
15-16	SPI_0_MOSI	ADC_2
17-18	SPI_0_CS_0	I2C_2_SDA
19-20	SPI_0_CS_1	I2C_2_SCL

Table 4.2: Daughterboard connector pinout.

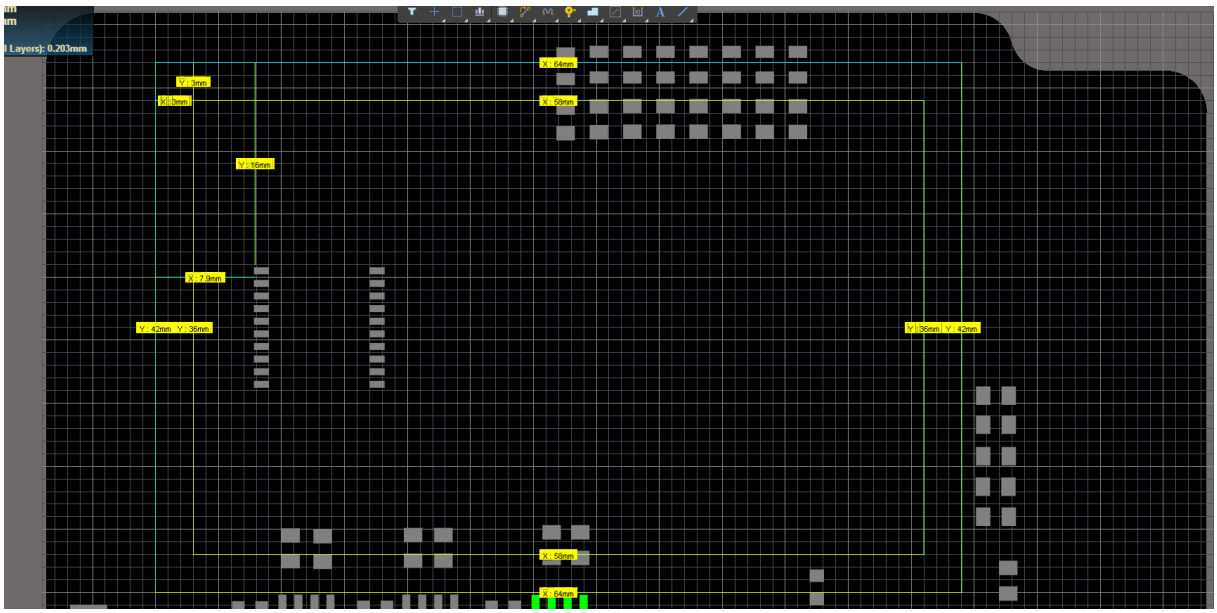


Figure 4.8: Recommended shape and size of the daughterboard.

4.3 Microcontroller

MSP430F6659.

4.3.1 Pinout

Pin Code	*Pin Number*	*Signal*
P1.0	34	MAIN_RADIO_ENABLE
P1.1	35	MAIN_RADIO_GPIO0
P1.2	36	MAIN_RADIO_GPIO1
P1.3	37	MAIN_RADIO_GPIO2
P1.4	38	MAIN_RADIO_RESET
P1.5	39	MAIN_RADIO_SPI_CS
P1.6	40	TTC_MCU_SPI_CS
P1.7	41	-
P2.0	17	SPI_CLK
P2.1	18	I2C0_SDA
P2.2	19	I2C0_SCL
P2.3	20	-
P2.4	21	SPI_MOSI
P2.5	22	SPI_MISO
P2.6	23	-
P2.7	24	-
P3.0	42	I2C0_EN
P3.1	43	I2C1_EN
P3.2	44	I2C2_EN
P3.3	45	I2C0_READY
P3.4	46	I2C1_READY
P3.5	47	I2C2_READY
P3.6	48	PC104_GPIO0
P3.7	49	PC104_GPIO1
P4.0	50	PC104_GPIO2
P4.1	51	PC104_GPIO3
P4.2	52	MEM_HOLD
P4.3	53	MEM_RESET
P4.4	54	MEM_SPI_CS
P4.5	55	-
P4.6	56	-
P4.7	57	-
P5.0	9	VREF
P5.1	10	AGND
P5.2	28	SYSTEM_FAULT_LED
P5.3	31	SYSTEM_LED
P5.4	32	PAYLOAD_0_ENABLE
P5.5	33	PAYLOAD_1_ENABLE
P5.6	16	-
P5.7	88	-
P6.0	97	D_BOARD_ADC0
P6.1	98	D_BOARD_ADC1

P6.2	99	D_BOARD_ADC2
P6.3	100	OBDH_CURRENT_ADC
P6.4	1	OBDH_VOLTAGE_ADC
P6.5	2	D_BOARD_SPI_CS0
P6.6	3	D_BOARD_SPI_CS1
P6.7	4	-
P7.0	-	-
P7.1	-	-
P7.2	84	XT2_N
P7.3	85	XT2_P
P7.4	5	D_BOARD_GPIO0
P7.5	6	D_BOARD_GPIO1
P7.6	7	D_BOARD_GPIO2
P7.7	8	D_BOARD_GPIO3
P8.0	58	-
P8.1	59	-
P8.2	60	UART1_TX
P8.3	61	UART1_RX
P8.4	62	-
P8.5	65	I2C1_SDA
P8.6	66	I2C1_SCL
P8.7	67	ANTENNA_GPIO
P9.0	68	-
P9.1	69	-
P9.2	70	UART0_TX
P9.3	71	UART0_RX
P9.4	72	WDI_EXT
P9.5	73	I2C2_SDA
P9.6	74	I2C2_SCL
P9.7	75	MR_WDOG
PJ.0	92	TP21
PJ.1	93	TP22
PJ.2	94	TP23
PJ.3	95	TP24
-	13	XT1IN
-	14	XT1OUT
-	96	JTAG_TDO_TDI
-	91	JTAG_TCK

Table 4.3: Microcontroller pinout.

4.4 External Watchdog

Additionally to the internal watchdog timer of the microcontroller, to ensure a system reset in case of a software freeze, an external watchdog circuit is being used. For that, the TPS3823 IC from Texas Instruments was chosen. This IC is a voltage monitor with a

watchdog timer circuit.

This circuit works this way: if the WDI pin remains high or low longer than the timeout period, then reset is triggered. The timer clears when reset is asserted or when WDI sees a rising edge or a falling edge.

The watchdog timer task clears the TPS3823 timer by toggling the WDI pin at every 100 ms. If the WDI pin state stays unmodified for more than 1600 ms, the reset pin is cleared and the microcontroller is reseted.

This circuit can be seen in the Figure 4.9.

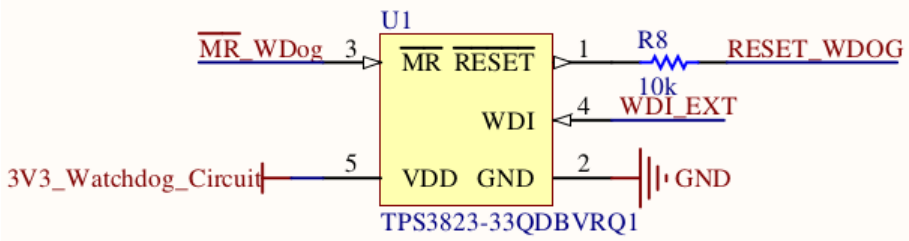


Figure 4.9: External watchdog timer circuit.

4.5 Non-Volatile Memory

The non-volatile memory is composed by a NOR flash memory with 1 Gb of capacity (or 128 MB). The used model is the Micron MT25QL01GBBB.

As can be seen in Figure 4.3, a SPI bus is used to communicate with this peripheral.

CHAPTER 5

Firmware

5.1 Tasks

A list of the firmware tasks can be seen in the Table 5.1.

Name	*Priority*	*Initial delay [ms]*	*Period [ms]*	*Stack [bytes]*
Startup (boot)	Highest	0	Aperiodic	500
Deployment hibernation	Highest	0	Aperiodic	TBD
Antenna deployment	Highest	0	Aperiodic	TBD
Watchdog reset	Lowest	0	100	128
Heartbeat	Lowest	0	500	128
Beacon	Medium	1000	60000	2000
Uplink	Low	1000	10000	500
EPS reading	Medium	5000	60000	TBD
EDC reading	High	5000	1000	TBD
Payload X reading	Medium	5000	5000	TBD
TTC writing	Medium	5000	10000	TBD
Radio periodoc reset	Medium	600000	600000	128
System reset	High	0	36000000	128
Read temperature	Medium	0	60000	128
CSP Server	Lowest	0	500	1024

Table 5.1: Firmware tasks.

All these tasks are better described below.

5.1.1 Startup (boot)

.

5.1.2 Deployment hibernation

.

5.1.3 Antenna deployment

.

5.1.4 Watchdog reset

This task resets the internal and external watchdog timer at every 100 ms. The internal watchdog has a maximum count time of 500 ms, and the external watchdog a maximum of 1600 ms (see chapter 4 for more information about the watchdog timers).

To prevent the system to not reset during an anomaly on some task (like an execution time longer than planned), this task has lowest possible priority: 0.

5.1.5 Heartbeat

The heartbeat task keeps blinking a LED ("*System LED*" in Figure ??) at a rate of 1 Hz during the execution of the system. Its purpose is to give a visual feedback of the execution of the scheduler. This is tasks does not have a specific purpose on the flight version of the module (the flight version of the PCB does not have LEDs).

5.1.6 Beacon

.

5.1.7 Uplink

.

5.1.8 EPS reading

.

5.1.9 EDC reading

.

5.1.10 Payload X reading

.

5.1.11 TTC writing

.

5.1.12 Radio periodic reset

.

5.1.13 System reset

This task resets the microcontroller by software at every 10 hours. This can be useful to cleanup possible wrong values in variables, repeat the antenna deployment routine (limited to n times), cleanup the RAM memory, etc.

5.1.14 Read temperature

This task reads the internal temperature of the microcontroller of the OBDH at every 60 seconds.

5.1.15 CSP Server

.

5.2 Telecommands

Name	*Parameters*	*Access*
Enter hibernation	Hibernation period in seconds	Private
Leave hibernation	None	Private
Activate beacon	None	Private
Deactivate beacon	None	Private
Activate downlink	None	Private
Deactivate downlink	None	Private
Activate EDC	None	Private
Deactivate EDC	None	Private
Get EDC info	None	Private
Activate Payload X	Experiment period in seconds	Private
Deactivate Payload X	None	Private
Set system time	Time value (epoch)	Private
Ping	None	Public
Message broadcast	ASCII message	Public
Request data	Data flags	Public

Table 5.2: System telecomamnds.

5.2.1 Enter hibernation

.

5.2.2 Leave hibernation

.

5.2.3 Activate beacon

.

5.2.4 Deactivate beacon

.

5.2.5 Activate EDC

.

5.2.6 Deactivate EDC

.

5.2.7 Get EDC info

This telecommand request information from the EDC payload. When received, the OBDH transmits the housekeeping and state frames of the EDC module (28 bytes). This telecommand does not requires a key.

5.2.8 Activate Payload X

.

5.2.9 Deactivate Payload X

.

5.2.10 Set system time

.

5.2.11 Ping

.

5.2.12 Message broadcast

.

5.2.13 Request data

.

5.3 Operating System

FreeRTOS 10

5.4 Hardware Abstraction Layer (HAL)

DriverLib

5.5 Protocols

5.5.1 NGHam

NGHam [1], short for Next Generation Ham Radio, is a set of protocols for packet radio communication. Its usage is similar to the existing AX.25 protocol.

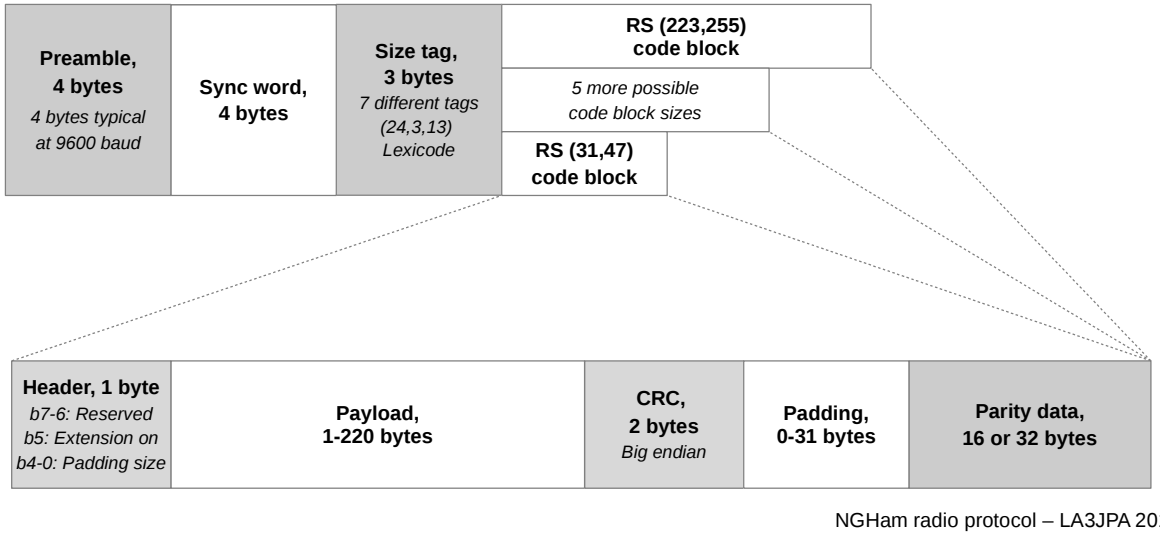


Figure 5.1: NGHam packet structure.

Bibliography

[1] Jon Petter Skagmo. Ngham protocol, 2014. Available at <https://github.com/skagmo/ngham>.