SLB-EPS2-DOC-v0.3

EPS 2.0 Documentation

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

March 2022

EPS 2.0 Documentation
March, 2022

Project Chief:
Eduardo Augusto Bezerra <eduardo.bezerra@spacelab.ufsc.br>

Authors:
André Martins Pio de Mattos <andre.mattos@spacelab.ufsc.br>
Gabriel Mariano Marcelino <gabriel.marcelino@spacelab.ufsc.br>
Yan Castro de Azeredo <yan.azeredo@spacelab.ufsc.br>
João Cláudio Elsen Barcellos <joao.claudio.barcellos@spacelab.ufsc.br>
Ramon Borba
Augusto Cezar Boldori Vassoler
Vinicius Pimenta Bernardo

Contributing Authors:
Leonardo Kessler Slongo
Sara Vega Martinez
Bruno Vale Barbosa Eiterer
Túlio Gomes Pereira

Revision Control:

Version	Author	Changes	Date
0.1	G. Marcelino	Document creation	2020/11/05
0.2	Y. C. Azeredo, G. Marcelino	First stable hardware	2021/06/28
0.3	G. Marcelino, J.C.E. Barcellos, R. Borba	Adding task sections and updating the tests and hardware sections	TBD

© 2022 by SpaceLab. EPS 2.0 Documentation. This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/4.0/.

List of Figures

1.1 3D view of the EPS 2.0 PCB. 1

2.1 Product tree of the EPS 2.0 module. 4
2.2 EPS 2.0 MCU Block diagram. 5
2.3 EPS 2.0 Power Block diagram. 5
2.4 System layers. 6
2.5 Simplified representation of the EPS2'S execution flow. 7
2.6 Available status LEDs. 8

3.1 Top side of the PCB. 9
3.2 Bottom side of the PCB. 10
3.3 Side view of the PCB. 10
3.4 EPS interfaces diagram. 11
3.5 Microcontroller internal diagram. 12
3.6 Voltage reference circuit for EPS MCU schematic circuit. 13
3.7 Voltage reference circuit for EPS MCU on the PCB. 13
3.8 Microcontroller pinout positions. 16
3.9 EPS2 and BAT4C 3D models assembled. 16
3.10 EPS2 battery connectors. 17
3.11 Solar panels connection to EPS2. 18
3.12 I2C buffer circuit. 18
3.13 One of the MPPT boosters schematic circuit. 19
3.14 MPPT boost converters circuit on the PCB. 19
3.15 Solar panels −Y and +X input circuit schematic. 20
3.16 Solar panels current measurement circuits on the PCB. 21
3.17 Solar panels −Y and +X voltage measurement circuit schematic. 21
3.18 Battery monitor circuit schematic. 22
3.19 Battery monitor circuit on the PCB. 23
3.20 ADS1248 circuit on the PCB. 23
3.21 Heater drivers circuit schematic. 24
3.22 Heater drivers circuit on the PCB. 24
3.23 EPS/TTC regulator circuit on the PCB. 25
3.24 OBDH regulator circuit on the PCB. 25
3.25 Antenna deployer regulator circuit on the PCB. 26
3.26 Main radio transceiver regulator circuit on the PCB. 26
3.27 Beacon radio transceiver regulator circuit on the PCB. 27
3.28 Payloads regulator circuit on the PCB. 27
3.29 Bottom view of PC-104 and simplified labels. 28
3.30 Solar panels power input connectors on the PCB. 31

3.31 Kill-switch spring button mechanism. 31
3.32 Kill-switches PicoBlade connectors on the PCB. 32
3.33 RBF PicoBlade connector on the PCB. 32
3.34 RTDs PicoBlade connectors on the PCB. 32
3.35 Heaters PicoBlade connector on the PCB. 32
3.36 External batteries charger PicoBlade connector on the PCB. 34
3.37 JTAG PicoBlade connector on the PCB. 34
3.38 Debug UART pin header connector on the PCB. 34

4.1 Product tree of the firmware of the EPS 2.0 module. 36

A.1 Top view of the EPS 2.0 v0.1 board. 53
A.2 Bottom view of the EPS 2.0 v0.1 board. 54
A.3 Electrical test setup of the power-up sequence and output power supply channels. 56
A.4 Power consumption during standby with any intensive firmware task. . . . 56
A.5 Payload step-down regulator power characterization. 57
A.6 Antenna step-down regulator power characterization. 58
A.7 Radio 0 step-down regulator power characterization. 59
A.8 Radio 1 step-down regulator power characterization. 60
A.9 OBDH step-down regulator power characterization. 61
A.10 EPS/TTC step-down regulator power characterization. 62
A.11 Setup used for the first firmware boot 63
A.12 Log messages during the first boot. 63

B.1 Top view of the EPS 2.0 v0.2 board. 66
B.2 Bottom view of the EPS 2.0 v0.2 board. 66
B.3 Measuring the voltage of the power buses without a load. 68
B.4 Setup to evaluate the 5V and 6V buses with load. 69
B.5 Setup to evaluate the 3,3V buses with load. 70
B.6 Setup used for the first firmware boot 70
B.7 Log messages during the first boot. 71
B.8 Communication between the TTC and EPS 72
B.9 Communication between the OBDH and EPS. 73
B.10 Values of temperature during the evaluation of heater's on-off controller. . 74
B.11 First part of the log. 75
B.12 Second part of the log. 75
B.13 Last part of the log. 76
B.14 Temperature of the batteries being controlled. 76
B.15 PWM signal, at 50kHz, generated by the MPPT controller. 77
B.16 PWM signal, at 100kHz, generated by the MPPT controller. 77

List of Tables

2.1 Hard code versioning table. 8

3.1 Boards interfaces. 10
3.2 Microcontroller features summary. 11
3.3 USCI configuration. 12
3.4 Microcontroller pinout and assignments. 15
3.5 Battery connector pinout. 17
3.6 PC-104 connector pinout. 29
3.7 PC-104 bus signal description. 30
3.8 Solar panels PicoBlades pinout. 30
3.9 Kill-switches PicoBlades pinout. 30
3.10 RBF PicoBlade pinout. 32
3.11 RBF PicoBlade pinout. 33
3.12 Heater PicoBlade pinout. 33
3.13 External batteries charger PicoBlade pinout. 33
3.14 JTAG PicoBlade pinout. 34
3.15 Debug UART pin header pinout. 34

4.1 FreeRTOS main configuration parameters. 35
4.2 Firmware tasks. 37
4.3 Variables and parameters of the EPS 2.0. 40

5.1 PCB fabrication specifics. 41
5.2 PCB stack up. 42

7.1 Test workflow table. 46
7.2 Visual Inspection test steps. 47
7.3 Mechanical Inspection test steps. 47
7.4 Integration Inspection test steps. 48
7.5 Electrical Inspection test steps. 48
7.6 Electrical Testing steps. 49
7.7 Functional Testing steps. 49
7.8 Module Testing steps. 50

Nomenclature

ADC	*Analog-to-Digital Converter.*
BAT4C	*Battery Module 4 Cells.*
EPS	*Electric Power System.*
GPIO	*General Purpose Input/Output.*
HAL	*Hardware Abstraction Layer.*
I2C	*Inter-Integrated Circuit.*
IDE	*Integrated Development Environment.*
isr	*interrupt service routine.*
JTAG	*Joint Test Action Group.*
LED	*Light-Emitting Diode.*
MCU	*Microcontroller.*
MPPT	*Maximum Power Point Tracking.*
NC	*Normally closed.*
NO	*Normally open.*
PCB	*Printed Circuit Board.*
PWM	*Pulse Width Modulation.*
RBF	*Remove Before Flight.*
RTD	*Resistance Temperature Detector.*
RTOS	*Real Time Operating System.*
SPI	*Serial Peripheral Interface.*
UART	*Universal Asynchronous Receiver/Transmitter.*
XCVR	*Transceiver.*

Contents

List of Figures vi

List of Tables vii

Nomenclature ix

1 Introduction 1

2 System Overview 3
- 2.1 Product tree . . . 3
- 2.2 MCU Block Diagram . . . 3
- 2.3 Power Block Diagram . . . 5
- 2.4 System Layers . . . 6
- 2.5 Operation . . . 6
  - 2.5.1 Execution Flow . . . 6
  - 2.5.2 Status LEDs . . . 7
- 2.6 Hard Code Versioning . . . 8

3 Hardware 9
- 3.1 Interfaces . . . 9
- 3.2 Microcontroller . . . 11
  - 3.2.1 Interfaces Configuration . . . 11
  - 3.2.2 Voltage Reference . . . 12
  - 3.2.3 Clocks Configuration . . . 12
  - 3.2.4 Pinout . . . 13
- 3.3 Batteries DaughterBoard . . . 15
- 3.4 Solar Panels . . . 15
- 3.5 I2C Buffer . . . 17
- 3.6 MPPT Subsystem . . . 18
  - 3.6.1 MPPT Boost Converters . . . 18
  - 3.6.2 Solar Panels Current Measurement . . . 19
  - 3.6.3 Solar Panels Voltage Measurement . . . 20
- 3.7 Batteries Management Subsystem . . . 21
  - 3.7.1 Boost Converters Output Voltage Measurement . . . 21
  - 3.7.2 Kill-Switches and Remove Before Flight . . . 22
  - 3.7.3 Battery Monitor Circuit . . . 22
  - 3.7.4 Main Power Bus Voltage Measurement . . . 23
  - 3.7.5 External ADC . . . 23
  - 3.7.6 Heaters Drivers . . . 24

3.8 Power Converters Subsystem . . . 24
3.8.1 EPS/TTC Regulator . . . 25
3.8.2 OBDH Regulator . . . 25
3.8.3 Antenna Deployer Regulator . . . 25
3.8.4 Radio 0 Transceiver Regulator . . . 26
3.8.5 Radio 1 Transceiver Regulator . . . 26
3.8.6 Payloads Regulator . . . 27
3.9 External Connectors . . . 27
3.9.1 PC104 . . . 28
3.9.2 Solar Panels PicoBlades . . . 28
3.9.3 Kill-Switches PicoBlades . . . 28
3.9.4 RBF PicoBlade . . . 28
3.9.5 RTDs PicoBlade . . . 29
3.9.6 Heater PicoBlade . . . 29
3.9.7 External Batteries Charger PicoBlade . . . 29
3.9.8 JTAG PicoBlade . . . 30
3.9.9 Debug UART Pin Header . . . 32

4 Firmware **35**
4.1 Product tree . . . 35
4.2 Operating System . . . 35
4.3 Hardware Abstraction Layer (HAL) . . . 35
4.4 Tasks . . . 37
4.4.1 Startup (boot) . . . 37
4.4.2 Watchdog reset . . . 37
4.4.3 Heartbeat . . . 37
4.4.4 System reset . . . 37
4.4.5 Heater controller . . . 38
4.4.6 Read sensors . . . 38
4.4.7 MPPT . . . 38
4.4.8 Parameter Server . . . 38
4.4.9 Time Control . . . 39
4.5 Variables and Parameters . . . 39

5 Board Assembly **41**
5.1 PCB Fabrication . . . 41

6 Usage Instructions **43**
6.1 Powering the Board . . . 43
6.1.1 Powering through JTAG Interface . . . 43
6.1.2 Powering through Power Supply . . . 43
6.1.3 Powering through Batteries . . . 44
6.2 Log Messages . . . 44

7 Test Procedures **45**
7.1 Visual Inspection . . . 45
7.2 Mechanical Inspection . . . 45
7.3 Integration Inspection . . . 45
7.4 Electrical Inspection . . . 45

7.5 Electrical Testing 46
7.6 Functional Testing 47
7.7 Module Testing 47

References **51**

Appendices **53**

A Test Report of v0.1 Version **53**
A.1 Visual Inspection 53
A.2 Mechanical Inspection 54
A.3 Integration Inspection 55
A.4 Electrical Inspection 55
A.5 Electrical Testing 55
A.6 Functional Testing 62
A.6.1 Firmware Programming 62
A.7 Conclusion 64

B Test Report of v0.2 Version **65**
B.1 Visual Inspection 65
B.2 Mechanical Inspection 67
B.3 Integration Inspection 67
B.4 Electrical Inspection 67
B.5 Electrical Testing 68
B.6 Functional Testing 70
B.6.1 Firmware Programming 70
B.6.2 Communication Buses 71
B.6.3 Sensors 73
B.6.4 Peripherals 74
B.6.5 Battery monitor 74
B.6.6 Heater's on-off controller 75
B.6.7 MPPT controller 77
B.7 Conclusion 78

CHAPTER 1

Introduction

The EPS 2.0 is a PCB designed to harvest, store and distribute energy for a nanosatellite. It is one of the service modules developed for FloripaSat-2 CubeSat Mission [1]. The energy harvesting system is based on solar energy conversion through ten solar panels attached to the 2U CubeSat structure. The EPS 2.0 is designed to operate the solar panels at their maximum power point (MPPT). The board is also responsible for measuring solar panels current, voltage and the temperature of the panels and batteries. The harvested solar energy is stored in the Battery Module 4C [2] connected to the EPS. The energy distribution is done by several integrated buck DC-DC converters. The full EPS system is composed of the solar panels, the EPS 2.0 PCB and the battery module. A general view of the EPS 2.0 board can be seen in Figure 1.1.

The module is a direct upgrade from the EPS of FloripaSat-1 [3], which grants a flight heritage rating. The improvements focus on providing a cleaner and more generic implementation in comparison with the previous version, more reliability in software, and adaptations for the new mission requirements. All the project, source and documentation files are available freely on a GitHub repository [4] under its respective licenses.

Figure 1.1: 3D view of the EPS 2.0 PCB.

CHAPTER 2

System Overview

The board has a MSP430 low-power MCU that runs the firmware application intended to control and communicate with its peripherals, subsystems and other modules. The programming language used is C and the firmware was developed using the Code Composer Studio IDE (a.k.a. CCS) for compiling, programming and testing. The module has many tasks, such as interfacing internal peripherals and communicating with other boards, over distinct protocols and time requirements. So, in order to improve predictability, a Real Time Operating System (RTOS) is used to ensure that the deadlines are observed, even under a fault situation in a routine. The RTOS chosen is the FreeRTOS (v10.2.1), since it is designed for embedded systems applications and it was already validated in space applications. The firmware architecture follows an abstraction layer scheme to facilitate higher level implementations and allow more portability across different hardware platforms, see section 2.4 for more details.

The EPS 2.0 is compatible with GOMspace Solar Panels or with panels of similar characteristics. Algorithms are implemented for MPPT improving power generation, also through measurements the load output can be regulated for a more efficient power distribution to the nanosatellite.

2.1 Product tree

The product tree of the EPS 2.0 module is available in Figure 2.1.

2.2 MCU Block Diagram

The Figure 2.2 presents a simplified view of the module subsystems and interfaces though the microcontroller perspective. The MCU has a programming JTAG, a dedicated UART debug interface and 4 communication buses, divided in 4 different protocols (I2C, SPI and UART).

There is a I2C buffer to allow secure and proper communication with the OBDH 2.0 module [5]. The SPI protocol is used for controlling and retrieving data from a additional ADC IC that measures temperature sensors (RTDs) on the batteries board and solar panels. Several parameters from the Batteries Management Subsystem are sent to the EPS 2.0 MCU via I2C protocol. The UART bus that goes to the PC/104 is used for basic telemetry to be sent to the beacon microcontroller within the TTC module. Besides this channels, there are GPIO connections for enabling and disabling power buses, for hard code PCB

Figure 2.1: Product tree of the EPS 2.0 module.

versioning and some optional GPIOs that can be added and used though the PC/104 interface.

The MCU reads the measurements of current and voltage of the solar panels from its ADC ports for the MPPT Subsystem, also from this data the MPPT is controlled by the microcontroller through PWM signals.

A external charger is used for charging the batteries and kill-switches for powering off the EPS 2.0 module during test phase, for flight the kill-switches are also connected to the button switches present on a CubeSat structure.

Figure 2.2: EPS 2.0 MCU Block diagram.

2.3 Power Block Diagram

The Figure 2.3 presents a more detailed view of the power subsystems that complements the MCU Block Diagram. More details and descriptions about these hardware components and interfaces are provided in the chapter 3.

Figure 2.3: EPS 2.0 Power Block diagram.

2.4 System Layers

As herein mentioned, the system is divided in various abstraction layers to favor high level firmware implementations. The Figure 2.4 shows this scheme, which is composed of third-part drivers at the lowest layer above the hardware, the operating system as the base building block of the module, the devices handling implementation, and the application tasks in the highest layer. More details are provided in the chapter 4.

Figure 2.4: System layers.

2.5 Operation

The system operates through the sequential execution of routines (tasks in the context of the operating system) that are scheduled and multiplexed through time. Each routine has a priority and a periodicity, which determine the following execution, the set of functionalities currently running, and the memory usage management.

Besides this deterministic scheduling system, the routines have communication channels with each other through the usage of queues, which provides a robust synchronization scheme. In the chapter 4 the system operation and the internal nuances are described in detail. This section is a top view, user perspective, to describe the module's operation.

2.5.1 Execution Flow

To better understand how the EPS2 is going to operate during the mission, a simplified representation of its operation is presented in Figure 2.5. Overall, the EPS2 should control the temperature of the battery and its energy harvesting system operation, as well as obtaining the measurements of the sensors periodically. The OBDH or TTC modules

can communicate with the EPS2, which should receive or transmit data as requested. Furthermore, every 10 hours the EPS2 will reset.

Figure 2.5: Simplified representation of the EPS2'S execution flow.

2.5.2 Status LEDs

On the development version of the board, there are ten LEDs that indicates some behaviors of the systems. This set of LEDs can be seen on Figure 2.6.

A description of each of these LEDs are available below:

- **6V_RADIO_1_LED**: Indicates that the radio 1 transceiver 6V power is being sourced.
- **5V_RADIO_0_LED**: Indicates that the radio 0 transceiver 5V power is being sourced.
- **SYSTEM_ON**: Heartbeat of the system. Blinks at a frequency of 1 Hz when the system is running properly.

Figure 2.6: Available status LEDs.

- **SYSTEM_FAULT**: Indicates an error has occurred during the running firmware of the board.
- **3V3_ANT**: Indicates that the antenna deployer 3.3V power is being sourced.
- **3V3_MCU**: Indicates that the EPS2 MCU 3.3V power is being sourced.
- **3V3_OBDH**: Indicates that the OBDH module 3.3V power is being sourced.
- **3V3_EPS_BEACON**: Indicates that the EPS2 board and beacon MCU 3.3V power is being sourced.
- **5V_PAYLOADS_LEDS**: Indicates that the payloads 5V power is being sourced.
- **VBUS_LED**: Indicates that the main power bus from the batteries is being sourced.

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

2.6 Hard Code Versioning

On the EPS2 board there are 2 GPIOs dedicated for hard code versioning. The on-board firmware can read these pins to identify the correct version of the hardware project. Each line can be either pulled to VCC or ground, representing in binary as 1 and 0 respectively. The Table 2.1 shows the representation of the versioning up to the latest revision of the project.

Version	P3.5 (pin code) / 47 (pin number)	P3.4 (pin code) / 46 (pin number)
v0.1	0	0
v0.2	0	1

Table 2.1: Hard code versioning table.

CHAPTER 3

Hardware

The EPS2 is a 4 layer 1.6mm thick PCB with FR-4 dielectric. The module doesn't have any impedance control requirements, for this reason the layer stackup has 1oz (0.0347mm) thickness in inner and outer copper layers. In the following sections, the hardware design, interfaces, and standards are described in detail. Section are divided by subsystem blocks, following the diagrams present on Figure 2.2 and Figure 2.3. The Figure 3.1, Figure 3.2 and Figure 3.3 presents the 3D rendered images of the top, bottom and side views of the board, respectively.

Figure 3.1: Top side of the PCB.

3.1 Interfaces

The Figure 3.4 presents the board interfaces, which consists of communication with other modules, debug access points, and internal peripherals. From the perspective of the microcontroller, there are 5 individual communication buses and the JTAG interface (Spy-Bi-Wire), in the Table 3.1: A1-SPI (dedicated for RTD analog readings with ADS1248);

Figure 3.2: Bottom side of the PCB.

Figure 3.3: Side view of the PCB.

A0-UART (dedicated for Beacon Radio); A2-UART (dedicated for debug); B0-I2C (dedicated for DS2777); B2-I2C (dedicated for OBDH); From the External Interface (IIP) can be acquired UART log messages for debugging via USB without the use of an external UART to USB converter. The SPI communication bus is actually an dedicated internal channel for the EPS MCU (master) to the ADS1248 (slave) ADC IC, the analog readings from BAT4C module (a.k.a Battery DaughterBoard) were also represented to show where the RTDs readings come from. Table 3.3 shows the interfaces configuration.

Peripheral	USCI	Protocol	Comm. Protocol
ADS1248	A1	SPI	-
Beacon Radio	A0	UART	-
PC (log messages)	A2	UART	ANSI messages
DS2777	B0	I^2C	-
OBDH	B2	I^2C	FSP
External Interface	-	JTAG	Spy-Bi-Wire

Table 3.1: Boards interfaces.

Figure 3.4: EPS interfaces diagram.

3.2 Microcontroller

The MCU consists of a CPU, RAM Memory and Flash Memory (used for program storage and non-volatile status registers). The chosen MCU is a low power 16-bit RISC (*MSP430F6659IPZR*) from Texas Instruments[6]. The Table 3.2 presents a summary of the main available features and Figure 3.5 shows the internal subsystems, descriptions, and peripherals. The microcontroller interfaces, configurations, and auxiliary components are described in the following topics.

Flash	*SRAM*	*Timers*	*USCI*	*ADC*	*DAC*	*GPIO*
512KB	64KB	2	6 (SPI / I2C / UART)	12	2	74

Table 3.2: Microcontroller features summary.

3.2.1 Interfaces Configuration

The microcontroller has 6 Universal Serial Communication Interfaces (USCI) that can be configured to operate with different protocols and parameters. The Table 3.3 describes each interface configurations.

Figure 3.5: Microcontroller internal diagram.

Interface	*Protocol (Index)*	*Mode*	*Word Length*	*Data Rate*	*Configuration*
USCI_A0	UART0	-	8 bits	9600 bps	Stop bits: 1 Parity: None
USCI_A1	SPI	Master	8 bits	TBD	Phase: High Polarity: Low
USCI_A2	UART1	-	8 bits	9600 bps	Stop bits: 1 Parity: None
USCI_B0	I2C0	Master	8 bits	100 kbps	Adr. len: 7 bits
USCI_B2	I2C1	Slave	8 bits	100 kbps	Address value: 0x36

Table 3.3: USCI configuration.

3.2.2 Voltage Reference

To generate the 3 volts reference for the MCU internal ADC the EPS uses a *595-REF5025AQDRQ1* chip. Its circuit schematic can be seen in Figure 3.6 and location on the PCB in Figure 3.7.

3.2.3 Clocks Configuration

Besides the internal clock sources, the microcontroller has two dedicated clock inputs for external crystals: the main clock and the auxiliary clock inputs. There are a 32MHz (*ABM8X-102-32.000MHZ-T*) and a 32.769kHz (*ECS-.327-12.5-34S-TR*) crystals connected to these inputs, respectively. The first source is used for generating the Master Clock (MCLK) and the Subsystem Master Clock (SMCLK), which are used by the CPU and the internal peripheral modules. The second source is used for generating the Auxiliary Clock (ACLK) that handles the low-power modes and might be used for peripherals.

Figure 3.6: Voltage reference circuit for EPS MCU schematic circuit.

Figure 3.7: Voltage reference circuit for EPS MCU on the PCB.

3.2.4 Pinout

An illustration of the microcontroller pinout positions can be seen in the Figure 3.8. The Table 3.4 presents all the EPS 2.0 microcontroller pins assignment.

Pin Code	*Pin Number*	*Signal*
P1.0	34	-
P1.1	35	-
P1.2	36	-
P1.3	37	EN_5V_PAYLOADS
P1.4	38	-
P1.5	39	BAT_GPIO1
P1.6	40	BAT_GPIO2
P1.7	41	-

P2.0	17	PIO
P2.1	18	I2C0_SDA
P2.2	19	I2C0_SCL
P2.3	20	PC104_GPIO2
P2.4	21	UART_EPS_TX_BEACON_RX
P2.5	22	UART_EPS_RX_BEACON_TX
P2.6	23	PC104_GPIO1
P2.7	24	PC104_GPIO0
P3.0	42	-
P3.1	43	-
P3.2	44	HEATER2_PWM
P3.3	45	HEATER1_PWM
P3.4	46	VERSION_BIT0
P3.5	47	VERSION_BIT1
P3.6	48	-
P3.7	49	-
P4.0	50	-
P4.1	51	MPPT_PWM_1
P4.2	52	MPPT_PWM_2
P4.3	53	MPPT_PWM_3
P4.4	54	-
P4.5	55	-
P4.6	56	-
P4.7	57	-
P5.0	9	VREF
P5.1	10	AGND
P5.2	28	-
P5.3	31	SYSTEM__FAULT_LED
P5.4	32	SYSTEM_ON
P5.5	33	-
P5.6	16	-
P5.7	88	-
P6.0	97	ADC1_+Y_SOLAR_PANEL_CURRENT
P6.1	98	ADC2_+X_SOLAR_PANEL_CURRENT
P6.2	99	ADC3_−X_SOLAR_PANEL_CURRENT
P6.3	100	ADC4_+Z_SOLAR_PANEL_CURRENT
P6.4	1	ADC5_−Z_SOLAR_PANEL_CURRENT
P6.5	2	ADC6_+Y_SOLAR_PANEL_CURRENT
P6.6	3	ADC7_EPS_TTC_XCVR_CURRENT
P6.7	4	ADC_MAIN_POWER_BUSS_VOLTAGE
P7.2	84	XT2_N
P7.3	85	XT2_P
P7.4	5	ADC1_−Y_+X_SOLAR_PANEL_VOLTAGE
P7.5	6	ADC2_−X_+Z_SOLAR_PANEL_VOLTAGE
P7.6	7	ADC3_−Z_+Y_SOLAR_PANEL_VOLTAGE
P7.7	8	ADC_SOLAR_PANELS_TOTAL_VOLTAGE

P8.0	58	-
P8.1	59	RTD_SCLK
P8.2	60	RTD_DIN
P8.3	61	RTD_DOUT
P8.4	62	RTD_RESET
P8.5	65	RTD_CS
P8.6	66	RTD_START
P8.7	67	RTD_DRDY
P9.0	68	-
P9.1	69	-
P9.2	70	UART0_TX
P9.3	71	UART0_RX
P9.4	72	I2C1_EN
P9.5	73	I2C1_SDA
P9.6	74	I2C1_SCL
P9.7	75	I2C1_READY
PJ.0	92	-
PJ.1	93	-
PJ.2	94	-
PJ.3	95	-
-	13	XT1IN
-	14	XT1OUT
-	96	JTAG_TDO_TDI
-	91	JTAG_TCK

Table 3.4: Microcontroller pinout and assignments.

3.3 Batteries DaughterBoard

Due to size restrictions the 4 cell batteries of the EPS2 were allocated to a dauhterboard named Battery Module 4 Cells, a.k.a BAT4C[2]. Both boards 3D models are assembled together in a EDA tool as seen in Figure 3.9. BAT4C is connected below EPS2 in a board-to-board connector, the female counterpart (*BAT4CIPS1-105-01-S-D*) present on the EPS is seen in Figure 3.10 with its pinout present on Table 3.5.

For compatibility with the older version of the battery module, the same connector pads are present near the middle section of the PCB. If the BAT4C is to be used, the compatibility connector must not be soldered, more detail on chapter 5. Also external connectors are used for temperature measurement and control with RTDs and heaters, more details can be seen on subsection 3.9.5 and subsection 3.9.6.

3.4 Solar Panels

The energy harvesting system is based on solar energy conversion through ten solar panels attached to a 2U CubeSat structure. The solar panels are connected through six 4 pin PicoBlade connectors *0533980471*. Because the EPS2 module has only six input connectors four pairs of solar panels will be connected in parallel. The connection scheme

Figure 3.8: Microcontroller pinout positions.

Figure 3.9: EPS2 and BAT4C 3D models assembled.

of the solar panels is visible in Figure 3.11. The input connectors for the solar panels power are described in subsection 3.9.2.

Figure 3.10: EPS2 battery connectors.

Pin	*Row*
1	+Vbat
2	+Vbat
3	+Vbat
4	+Vbat
5	−Vbat
6	−Vbat
7	−Vbat
8	−Vbat
9	Vbat_Common
10	Vbat_Common

Table 3.5: Battery connector pinout.

3.5 I2C Buffer

The microcontroller I2C interface have a dedicated IC buffer, which improve the signal quality throughout the various connectors and offers reliability enhancements, since it protects the bus in case of failures. This measure was adopted in all the satellite modules due to previous failures in I2C buses. Using this scheme, the modules connected though this protocol might have shared connections without losing performance or reliability.

The buffer selected for this function is the Texas Instruments *TCA4311*. Besides the I2C inputs and outputs, it features control and status signals that are connected to GPIOs in the microcontroller: an enable and an operation ready status. Both inputs and outputs in these I2C lines have external pull-up resistors. Its circuit schematic can be see Figure 3.12.

Figure 3.11: Solar panels connection to EPS2.

Figure 3.12: I2C buffer circuit.

3.6 MPPT Subsystem

On the MPPT subsystem, the main components are the MPPT boost converters, solar panels voltage and current sensors. These measurement circuits are used to generate a voltage proportional to the variable being measured, in a range accepted by the MCU internal ADC.

3.6.1 MPPT Boost Converters

There are three boost converters in the system, one for each couple of solar panels in parallel connection. Each one is a discrete boost with a *MSS1278T-104MLD* inductor, a *SI4166DY* MOSFET as the switch and a *B340LA-13-F* diode. There are six *GRM32ER1E226KE15L* capacitors and two *GRM216R71H103KA01D* capacitors connected in parallel in the boost output. The output filter is the same for all the converters as their outputs are tied together. The control PWM signals are generated by the MCU

at a frequency of 100 kHz. One of the MPPT boosters circuit schematic can be seen in Figure 3.13 and location of all three at the PCB in Figure 3.14.

Figure 3.13: One of the MPPT boosters schematic circuit.

Figure 3.14: MPPT boost converters circuit on the PCB.

3.6.2 Solar Panels Current Measurement

The main component of the solar panels currents measurement circuit is the *MAX9934TAUA+* current sense amplifier. It generates an output current proportional to the differential input

voltage. The gain is 25 μA/mV. It was considered a maximum current load sourced by two solar panels in parallel on the most optimist scenario of 1.5 A . To make the measurements possible, the current goes through 20 mΩ, 0.5 % resistors, connected to the inputs of the amplifier, and the outputs are connected to 3.3 kΩ resistors. The output voltage of the circuit is given by:

$$V_{out} = I_{load} \cdot R_{sense} \cdot G_m \cdot R_{out} \tag{3.1}$$

The voltage drop in R_{sense} generates a proportional I_{out} on the IC of approximately 750 μA. In total there are six of these current measurement circuits for the six sides of the CubeSat. Two of the inputs of the solar panels circuit schematic can be seen in Figure 3.15 and location of all six at the PCB in Figure 3.16.

Figure 3.15: Solar panels −Y and +X input circuit schematic.

3.6.3 Solar Panels Voltage Measurement

The solar panels voltage measurement circuit is composed by a voltage divider and an op-amp in a buffer configuration. The voltage divider is composed of a 93.1 kΩ resistor and an 100 kΩ resistor. The op-amp is a *TLV341AIDBVR* chip. The output voltage is given by:

$$V_{out} = V_{sp} \cdot \frac{R_2}{R_1 + R_2} \tag{3.2}$$

In total there are three of these voltage measurement circuits, the solar panels sides that are measured together are: −Y with +X, −X with +Z and −Z with +Y. One of the voltage measurement circuit schematic can be seen in Figure 3.17.

Figure 3.16: Solar panels current measurement circuits on the PCB.

Figure 3.17: Solar panels −Y and +X voltage measurement circuit schematic.

3.7 Batteries Management Subsystem

On the batteries management subsystem the main components are the battery control circuit, external ADC chip, solar panels and batteries kill-switches, heater drivers and voltage sensors for the boosters output and main power bus.

3.7.1 Boost Converters Output Voltage Measurement

The boost converters output voltage measurement circuit is very similar to the solar panels voltages measurement circuit, with the exception that the voltage divider is composed by a 300 kΩ resistor and an 100 kΩ resistor. The schematic for the voltage measurement circuit of the solar panels can be seen again for reference in Figure 3.17.

3.7.2 Kill-Switches and Remove Before Flight

These switches are used to separate the solar panels and the batteries from the load during pre-flight and launch. Each one is composed of two *SI4403-CDY-T1-GE3* P-channel MOSFETs in parallel, as a redundancy. The Kill-Switches and RBF are interfaced on the EPS board via external PicoBlade cables to its respective external mechanisms.

RBF functions by simply making a short circuit in the pins of the pin header present on an external interface[7], and for the kill-switches it is required to press the spring buttons on the CubeSat structure, this is naturally done when the nanosatellite in integrated into the deployer. On subsection 3.9.3 and subsection 3.9.4 it is showed the pinouts and locations for these connectors.

3.7.3 Battery Monitor Circuit

The batteries are monitored by the *DS2777* chip. It measures several parameters and sends them to the EPS2 MCU via I2C protocol. The factory default value for its slave address is 0x59. Also it automatically protects the batteries against short-circuits, overvoltage and undervoltage situations by switching two *FDS6898AZ* MOSFETs. Its circuit schematic can be seen in Figure 3.18 and location on the PCB in Figure 3.19.

Figure 3.18: Battery monitor circuit schematic.

Figure 3.19: Battery monitor circuit on the PCB.

3.7.4 Main Power Bus Voltage Measurement

The main power bus voltage measurement circuit is identical to the boost converters output voltage measurement circuit. The schematic for the voltage measurement circuit of the solar panels can be seen again for reference in Figure 3.17.

3.7.5 External ADC

The *ADS1248* chip generates a precise reference current to the RTDs, and samples the voltage proportional to the temperature established over the sensors. This voltage is converted to digital data and sent to the MCU via SPI protocol. The location the IC and its subcircuitry can be seen in Figure 3.20.

Figure 3.20: ADS1248 circuit on the PCB.

3.7.6 Heaters Drivers

The drivers are chopper converters managed by the MCU, either switched to on/off or controlled with a PWM signal with a frequency of 50 kHz, depending on the mode of operation (detailed in chapter chapter 4). The switches of the chopper converters are *Si4010DY* MOSFETs. Its circuit schematic can be seen in Figure 3.21 and location on the PCB in Figure 3.22.

Figure 3.21: Heater drivers circuit schematic.

Figure 3.22: Heater drivers circuit on the PCB.

3.8 Power Converters Subsystem

The EPS2 has 6 integrated buck DC-DC regulators, all these are powered from the main power bus. Some regulators are always enabled, others are can be enabled or disabled by the EPS2 or other module.

3.8.1 EPS/TTC Regulator

To supply the TTC MCU (also called "Beacon MCU") and EPS2 MCU and its subcircuits a *TPS5420QDRQ1* regulator is used, with and output voltage of 3.3 V and 2 A current capability. This regulator is always on. The EPS/TTC circuit location on the PCB can be seen in Figure 3.23.

Figure 3.23: EPS/TTC regulator circuit on the PCB.

There is also a current measurement at the output of the EPS/TTC regulator. It also uses a *MAX9934TAUA+* current sense amplifier, but with a shunt resistor of 75 mΩ, 0.5 % and the output connected to a 4.02 kΩ resistor. The circuit schematic is almost the same as Figure 3.15 only changing the resistors and capacitor values, its location on the PCB is the labeled U27 IC and its passive components in Figure 3.16.

3.8.2 OBDH Regulator

The OBDH is powered by a *TPS5410QDRQ1* regulator, with an output voltage of 3.3 V and 1 A current capability. Its location on the PCB can be seen in Figure 3.24.

Figure 3.24: OBDH regulator circuit on the PCB.

3.8.3 Antenna Deployer Regulator

The antenna deployment system has a dedicated regulator *TPS5420QDRQ1*, with 3.3 V output voltage and 2 A current capability. This regulator is always on. Its location on

the PCB can be seen in Figure 3.25.

Figure 3.25: Antenna deployer regulator circuit on the PCB.

3.8.4 Radio 0 Transceiver Regulator

The radio 0 XCVR responsible for the Downlink/Uplink of the CubeSat is powered by a *TPS54540QDDARQ1* regulator, with an output voltage of 5V and 5A capability. The OBDH can enable/disable this regulator. Its location on the PCB can be seen in Figure 3.26.

Figure 3.26: Main radio transceiver regulator circuit on the PCB.

3.8.5 Radio 1 Transceiver Regulator

The radio 1 XCVR responsible for a periodic data sent status of the CubeSat is powered by a *TPS54540QDDARQ1* regulator, with 6V output voltage and 5A capability. The

Beacon MCU can enable/disable this regulator. Its location on the PCB can be seen in Figure 3.27.

Figure 3.27: Beacon radio transceiver regulator circuit on the PCB.

3.8.6 Payloads Regulator

To power the payloads a *TPS5430QDDARQ1* regulator is used. It has an output voltage of 5 V and 3 A current capability. The EPS2 can enable/disable this regulator. Its location on the PCB can be seen in Figure 3.28.

Figure 3.28: Payloads regulator circuit on the PCB.

3.9 External Connectors

The EPS2 module is connected to the other modules using the PC104 bus. The solar panels, the kill-switches, the remove before flight, the RTDs, the heater, the batteries

charger connector and the JTAG pins are connected using Molex PicoBlade connectors. The EPS2 module also has a jumper that connects the MCU VCC to the JTAG VCC and a header to debug the board via UART protocol.

In the following sections each connector is detailed, with a picture showing the location on the EPS2 PCB and a table explaining each pin function.

3.9.1 PC104

The connector referred as PC-104 is a junction of two double row 28H headers (*SSW-126-01-G-D*). These connectors create a solid 104-pin interconnection across the different satellite modules. The Figure 3.29 shows the PC-104 interface from the bottom side of the PCB, which allows visualize the simplified label scheme in the board. Also, the Table 3.6 provides the connector pinout[1] for the pins that are connected to the module.

Figure 3.29: Bottom view of PC-104 and simplified labels.

3.9.2 Solar Panels PicoBlades

There are six PicoBlade connectors that can be connected to solar panels. Each one of them is to be used with its respective positive or negative cartesian axis reference label: X, Y or Z. Note that the total current for each individual PicoBlade pin must not exceed 1000mA, this means that the maximum current per connector is 2000mA. Their pinout is showed in Table 3.8 and position on the PCB in Figure 3.30.

3.9.3 Kill-Switches PicoBlades

There are two PicoBlade connectors to be connected to two separated kill-switch spring button mechanisms, one of the mechanism is illustrated in Figure 3.31. The connection is done by manually soldering and isolating with a heat shrink tube, the other end of the cable goes to PicoBlades of the EPS, their pinout is showed in Table 3.9 and position on the PCB in Figure 3.32.

3.9.4 RBF PicoBlade

The RBF PicoBlade interconnects the separation switches circuit present on the EPS to be accessed in a external interface. Its pinout is showed in Table 3.11 and position on the PCB in Figure 3.33.

[1]This pinout is simplified since additional interfaces were omitted. Refer to *option sheet* in chapter 5.

Pin [A-B]	*H1A*	*H1B*	*H2A*	*H2B*
1-2	-	-	-	-
3-4	-	-	-	-
5-6	-	-	UART_RX	-
7-8	-	-	UART_TX	-
9-10	-	EN_PWR_5	-	-
11-12	-	EN_PWR_6	-	-
13-14	-	-	-	-
15-16	-	-	-	-
17-18	-	-	-	-
19-20	-	-	-	-
21-22	-	-	-	-
23-24	-	-	-	-
25-26	-	-	PWR_4_5V	PWR_4_5V
27-28	-	-	PWR_7_3V3	PWR_7_3V3
29-30	GND	GND	GND	GND
31-32	GND	GND	GND	GND
33-34	-	-	-	-
35-36	-	-	PWR_1_3V3	PWR_1_3V3
37-38	-	-	-	-
39-40	-	-	-	-
41-42	-	-	-	-
43-44	-	-	-	-
45-46	PWR_2_3V3	PWR_2_3V3	PWR_3_BAT	PWR_3_BAT
47-48	PWR_4_5V	PWR_4_5V	-	-
49-50	PWR_5_5V	PWR_5_5V	I2C_SDA	-
51-52	PWR_6_6V	PWR_6_6V	I2C_SCL	-

Table 3.6: PC-104 connector pinout.

3.9.5 RTDs PicoBlade

EPS reads temperature from RTDs present in the BAT4C module with a external PicoBlade cable connected between both boards. The two connectors RTD1 and RTD2 pinouts are showed in Table 3.11 and positions on the PCB in Figure 3.34.

3.9.6 Heater PicoBlade

The PWM signals that control the heaters present on the BAT4C module is also brought by a external PicoBlade cable. The connector pinout is showed in Table 3.12 and positions on the PCB in Figure 3.35.

3.9.7 External Batteries Charger PicoBlade

When the EPS and BAT4C are assembled together the batteries can be charged from a PicoBlade connector, which is accessed in a external interface. The same current restriction as for the solar panels connectors is applied here, the external batteries charger PicoBlade

Signal	Pin(s)	Description
GND	H1-29, H1-30, H1-31, H1-32, H2-29, H2-30, H2-31, H2-32	Ground reference
PWR_1_3V3	H2-35, H2-36	Power bus 1, 3.3 V, 2 A max.
PWR_2_3V3	H1-45, H1-46	Power bus 2, 3.3 V, 1 A max.
PWR_3_BAT	H2-45, H2-46	Power bus 3, battery terminals (+)
PWR_4_5V	H1-47, H1-48, H2-25, H2-26	Power bus 4, 5 V, 3 A max.
PWR_5_5V	H1-49, H1-50	Power bus 5, 5 V, 5 A max.
PWR_6_6V	H1-51, H1-52	Power bus 6, 6 V, 5 A max.
PWR_7_3V3	H2-27, H2-28	Power bus 7, 3.3 V, 2 A max.
I2C_SDA	H2-49	Primary communication bus (data signal)
I2C_SCL	H2-51	Primary communication bus (clock signal)
UART_RX	H2-5	Secondary communication bus (RX)
UART_TX	H2-7	Secondary communication bus (TX)
EN_PWR_5	H1-10	Enable signal of the power bus 5
EN_PWR_6	H1-12	Enable signal of the power bus 6

Table 3.7: PC-104 bus signal description.

Pin	*Row*
1	Panel [side reference] positive input
2	Panel [side reference] positive input
3	PGND
4	PGND

Table 3.8: Solar panels PicoBlades pinout.

Pin	*Row*
1	Common
2	Common
3	NO
4	NO
5	NC
6	NC

Table 3.9: Kill-switches PicoBlades pinout.

must not exceed 2000mA. The connector pinout is showed in Table 3.13 and position on the PCB in Figure 3.36.

3.9.8 JTAG PicoBlade

The EPS module can be programmed and debugged through its JTAG PicoBlade connector as showed in Figure 3.37. The pin header labeled CN1 serves as a jumper for

Figure 3.30: Solar panels power input connectors on the PCB.

Figure 3.31: Kill-switch spring button mechanism.

shorting the power from the flash emulator tool to the 3V3 power used for the EPS MCU and subcircuits. This feature enables powering the board partially using MSP-FET, be advised to leave this connector open if the EPS is powered from other means, for more details see chapter 6. The JTAG connector pinout is showed in Table 3.14.

Figure 3.32: Kill-switches PicoBlade connectors on the PCB.

Pin	*Row*
1	+RBF
2	−RBF
3	+RBF
4	−RBF

Table 3.10: RBF PicoBlade pinout.

Figure 3.33: RBF PicoBlade connector on the PCB.

Figure 3.34: RTDs PicoBlade connectors on the PCB.

Figure 3.35: Heaters PicoBlade connector on the PCB.

3.9.9 Debug UART Pin Header

For debugging via UART using log messages during test phase a pin header can be easily acessed with jumper wires. This connector is not meant to be soldered in the flight model of the EPS. The connector pinout is showed in Table 3.15 and position on the PCB in Figure 3.38.

Pin	*Row*
RTD1 PicoBlade	
1	BAT_GPIO1
2	BAT_GPIO2
3	RTD_Common
4	RTD_RTD3
5	RTD_Common
6	RTD_RTD2
7	RTD_Common
8	RTD_RTD6
RTD2 PicoBlade	
1	RTD_Common
2	RTD_RTD5
3	RTD_Common
4	RTD_RTD4
5	RTD_Common
6	RTD_RTD1
7	RTD_Common
8	RTD_RTD0

Table 3.11: RBF PicoBlade pinout.

Pin	*Row*
1	−Heater1_Voltage
2	−Heater1_Voltage
3	VBUS
4	VBUS
5	−Heater2_Voltage
6	−Heater2_Voltage
7	VBUS
8	VBUS

Table 3.12: Heater PicoBlade pinout.

Pin	*Row*
1	V_Charging_Batteries
2	V_Charging_Batteries
3	PGND
4	PGND

Table 3.13: External batteries charger PicoBlade pinout.

Figure 3.36: External batteries charger PicoBlade connector on the PCB.

Pin	*Row*
1	3V3_MCU
2	TDO
3	TCK
4	UART_Debug_Tx
5	UART_Debug_Rx
6	DGND

Table 3.14: JTAG PicoBlade pinout.

Figure 3.37: JTAG PicoBlade connector on the PCB.

Pin	*Row*
1	UART_Debug_Tx
2	UART_Debug_Rx
3	DGND

Table 3.15: Debug UART pin header pinout.

Figure 3.38: Debug UART pin header connector on the PCB.

CHAPTER 4

Firmware

4.1 Product tree

The product tree of the firmware part of the EPS 2.0 module is available in Figure 4.1.

4.2 Operating System

As operating system the FreeRTOS 10 [8] is being used. FreeRTOS is a market-leading real-time operating system (RTOS) for microcontrollers and small microprocessors. Distributed freely under the MIT open source license, FreeRTOS includes a kernel and a growing set of IoT libraries suitable for use across all industry sectors. FreeRTOS is built with an emphasis on reliability and ease of use.

The main configuration parameters of the operating system in this project are available in Table 4.1.

Parameter	Value	Unit
Version	v10.2.0	-
Tick rate (Hz)	1000	Hz
CPU clock (HZ)	32	MHz
Max. priorities	5	-
Heap size	40690	bytes
Max. length of task name	20	-

Table 4.1: FreeRTOS main configuration parameters.

More details of the used configuration parameters can be seen in the file *firmware/-config/FreeRTOSConfig.h* from [4].

4.3 Hardware Abstraction Layer (HAL)

As the Hardware Abstraction Layer (HAL), the DriverLib [9] from Texas Instruments is begin used. It is the official API to access the registers of the MSP430 microcontrollers.

The DriverLib is meant to provide a "software" layer to the programmer in order to facilitate higher level of programming compared to direct register accesses. By using the high level software APIs provided by DriverLib, users can create powerful and intuitive

Figure 4.1: Product tree of the firmware of the EPS 2.0 module.

code which is highly portable between not only devices within the MSP430 platform, but between different families in the MSP430/MSP432 platforms.

4.4 Tasks

A list of the firmware tasks can be seen in the Table 4.2.

name	priority	initial delay [ms]	period [ms]	stack [bytes]
startup (boot)	highest	0	aperiodic	500
watchdog reset	lowest	0	100	150
heartbeat	lowest	0	500	160
system reset	low	0	36000000	128
heater controller	medium	0	tbd	160
read sensors	low	0	60000	512
mppt	high	0	tbd	160
parameter server	high	0	called by isr	300
time control	high	0	1000	128

Table 4.2: Firmware tasks.

4.4.1 Startup (boot)

This task is responsible for initializing all devices, drivers, libraries and data structures upon system power up or reboot. It's the first task executed and it unblocks the remaining tasks after initialization is completed.

Task configuration parameters are shown in Table 4.2.

4.4.2 Watchdog reset

This task is responsible for resetting the internal watchdog timer every $100ms$. The internal watchdog timer has a $1s$ timeout.

This task has the lowest possible priority in order to prevent the watchdog from not triggering in case an anomaly happens in other tasks (longer than expected execution times for example).

Task configuration parameters are shown in Table 4.2.

4.4.3 Heartbeat

This task is responsible for providing a visual feedback indicating that the system is running properly. This feedback is given through blinking an LED ("SYSTEM_ON" LED on Figure 2.6) at a frequency of $1Hz$ emulating a "heartbeat" for the system.

The usefulness of this task comes during development with engineering model of the module, providing information to the developers. The flight model has no LEDs so this task serves no purpose in that case.

Task configuration parameters are shown in Table 4.2.

4.4.4 System reset

This task is responsible for resetting the microcontroller, via software, every 10 hours. This helps to cleanup possible corrupted values in variables, retry a failed initialization,

cleanup RAM memory, etc.

Task configuration parameters are shown in Table 4.2.

4.4.5 Heater controller

This task is responsible for monitoring and controlling the batteries' heaters.

The heaters and batteries temperature are monitored through a set of RTDs, read through an external ADC (*ADS1248*), and activated through MOSFET drivers controlled by the MCUs GPIOs to prevent the batteries from operating below a critical temperature value.

This task has two modes of operation, automatic mode and manual mode.

In automatic mode, the batteries' temperature readings occur every 2 seconds, and the heaters' status are updated accordingly, based on the new reading and a predefined temperature limit, through the algorithm defined in the heater device. This algorithm switches the heaters on of off based on the set temperature limits.

In manual mode, the heater are controlled manually through telecommands, independent of the temperature readings. In this mode, the heaters are controlled with a PWM signal with a duty cycle defined via telecommand.

Task configuration parameters are shown in Table 4.2.

4.4.6 Read sensors

This task is responsible for reading all sensors available to the EPS 2.0 module, recording their values. The readings occur every 60*s*, and the results are written in the eps buffer data structure.

Task configuration parameters are shown in Table 4.2.

4.4.7 MPPT

This task is responsible for running the MPPT algorithm for the solar panels, in order to operate them in their maximum power point. The algorithm itself is defined in the MPPT device.

The solar panels current and voltage sensors are read every 300*ms* and the results are passed as inputs to the MPPT algorithm, which then controls the MPPT Boost circuit through a set o PWM outputs.

Task configuration parameters are shown in Table 4.2.

4.4.8 Parameter Server

This task is responsible for processing and responding commands received from the OBDH and TTC modules.

The commands are for reading and writing to the EPS buffer data structure, seen in Table 4.3

Task configuration parameters are shown in Table 4.2.

4.4.9 Time Control

This task is responsible for the time management of the system. At every second it increments the system time (epoch). Also, at every minute it saves the current system time in the non-volatile memory.

Task configuration parameters are shown in Table 4.2.

4.5 Variables and Parameters

A list of all the variables of EPS with their identification number (ID) and variable type that can be read from the sensors and peripherals is seen in the Table 4.3.

ID	Name/Description	Type	Access
0	Time counter in millseconds	uint32	R
1	Temperature of the μC in K	uint16	R
2	EPS circuitry current in mA	uint16	R
3	Last reset cause: - 0x00 = No interrupt pending - 0x02 = Brownout (BOR) - 0x04 = RST/NMI (BOR) - 0x06 = PMMSWBOR (BOR) - 0x08 = Wakeup from LPMx.5 (BOR) - 0x0A = Security violation (BOR) - 0x0C = SVSL (POR) - 0x0E = SVSH (POR) - 0x10 = SVML_OVP (POR) - 0x12 = SVMH_OVP (POR) - 0x14 = PMMSWPOR (POR) - 0x16 = WDT time out (PUC) - 0x18 = WDT password violation (PUC) - 0x1A = Flash password violation (PUC) - 0x1C = Reserved - 0x1E = PERF peripheral/configuration area fetch (PUC) - 0x20 = PMM password violation (PUC) - 0x22 to 0x3E = Reserved	uint8	R
4	Reset counter	uint16	R
5	-Y and +X sides solar panel voltage in mV	uint16	R
6	-X and +Z sides solar panel voltage in mV	uint16	R
7	-Z and +Y sides solar panel voltage in mV	uint16	R
8	-Y side solar panel current in mA	uint16	R
9	+Y side solar panel current in mA	uint16	R
10	-X side solar panel current in mA	uint16	R
11	+X side solar panel current in mA	uint16	R
12	-Z side solar panel current in mA	uint16	R
13	+Z side solar panel current in mA	uint16	R
14	MPPT 1 duty cycle in % (writable just in manual mode)	uint8	R/W
15	MPPT 2 duty cycle in % (writable just in manual mode)	uint8	R/W

16	MPPT 3 duty cycle in % (writable just in manual mode)	uint8	R/W
17	Total solar panels output voltage after MPPT in mV	uint16	R
18	Main power bus voltage in mV	uint16	R
19	RTD0 temperature in K	uint16	R
20	RTD1 temperature in K	uint16	R
21	RTD2 temperature in K	uint16	R
22	RTD3 temperature in K	uint16	R
23	RTD4 temperature in K	uint16	R
24	RTD5 temperature in K	uint16	R
25	RTD6 temperature in K	uint16	R
26	Batteries voltage in mV	uint16	R
27	Batteries current in mA	uint16	R
28	Batteries average current in mA	uint16	R
29	Batteries accumulated current in mA	uint16	R
30	Batteries charge in mAh	uint16	R
31	Battery monitor IC temperature in K	uint16	R
32	Battery monitor status register	uint8	R
33	Battery monitor protection register	uint8	R
34	Battery monitor cycle counter	uint8	R
35	Battery monitor Remaining Active-Absolute Capacity (RAAC) in mAh	uint16	R
36	Battery monitor Remaining Standby-Absolute Capacity (RSAC) in mAh	uint16	R
37	Battery monitor Remaining Active-Relative Capacity (RARC) in %	uint8	R
38	Battery monitor Remaining Standby-Relative Capacity (RSRC) in %	uint8	R
39	Battery heater 1 duty cycle in % (writable just in manual mode)	uint8	R/W
40	Battery heater 2 duty cycle in % (writable just in manual mode)	uint8	R/W
41	Hardware version	uint8	R
42	Firmware version (ex.: “v1.2.3”’ = 0x00010203)	uint32	R
43	MPPT 1 mode (0x00 = automatic, 0x01 = manual)	uint8	R/W
44	MPPT 2 mode (0x00 = automatic, 0x01 = manual)	uint8	R/W
45	MPPT 3 mode (0x00 = automatic, 0x01 = manual)	uint8	R/W
46	Battery heater 1 mode (0x00 = automatic, 0x01 = manual)	uint8	R/W
47	Battery heater 2 mode (0x00 = automatic, 0x01 = manual)	uint8	R/W
48	Device ID (0xEEE2)	uint16	R

Table 4.3: Variables and parameters of the EPS 2.0.

CHAPTER 5

Board Assembly

5.1 PCB Fabrication

The board is not designed to be fabricated without a solder mask, but if possible a Class 3 fabrication is recommended. A list with the fabrication specifications can be seen in Table 5.1 and layer stack up can be seen in Table 5.2.

Parameter	Value
Size	86.26 × 92.13 mm
Layers	4
Thickness	1.6 mm
Minimum Hole Size	0.254 mm
Maximum Hole Size	3.2 mm
Silkscreen Color	White
Surface Finish	HASL with lead
Via Process	Tenting vias
Material	FR-4: TG150
Minimum Track/Spacing	6/6 mil (0.1524/0.1524 mm)
Solder Mask Color	Green
Gold Fingers	No
Impedance Control	No
Fiducials	3 on top and bottom layers already placed
Finish Copper	Outer and inner copper 1 oz (35 μm Cu)

Table 5.1: PCB fabrication specifics.

Layer	Material	Thickness mm
Top Layer	Cooper	0.035
Dielectric 1	Prepreg	0.12
Signal Layer 1	Cooper	0.035
Core	FR-4: TG-150	1.2
Signal Layer 2	Cooper	0.035
Dielectric 2	Prepreg	0.12
Bottom Layer	Cooper	0.035

Table 5.2: PCB stack up.

CHAPTER 6

Usage Instructions

6.1 Powering the Board

The EPS 2.0 is the energy provider module within a satellite, its nominal operation is alongside a battery pack from the BAT4C module and solar panels.

During development and testing the board can also be partially powered using the JTAG interface (only the MCU and peripherals connected to the 3V3 EPS regulator), or fully powered directly through the battery connector with an equivalent power supply.

The module's PC104 power pins are available to be accessed externally, but it is advised to be used only for test probes and not powering directly from them.

In the following subsections powering from the JTAG interface and batteries connector are explained in detail. It is advised to to have either the kill-switches on "NO" position and/or the RBF switch active before connecting the probes or cables to power the module.

6.1.1 Powering through JTAG Interface

The JTAG interface is used for programming and debugging the module using a Flash Emulation Tool (refer to subsection 3.9.8 on the hardware chapter). This is done using a MSP-FET with the Spy-Bi-Wire protocol. The tool should provide $3.3V$ and a maximum of 100 mA, due to this current limitation only the EPS MCU can be powered for minimal testing and debugging purposes, for this the CN1 header pins must be shorted. Remember that the CN1 connector should only be used if the EPS board is not being powered by any other means like batteries or external power supply.

For the interfacing the 14 pin cable of the MSP-FET to the EPS it is required an adapted cable or an external interface. The IIP[7] Nº1 and Nº2 boards have a 14 pin header that translates in a picoblade connector for the required connector counterpart on the EPS module, any of the pin header slots from 1 to 4 can be used. When the MSP-FET is correctly connected and the necessary cable connections are done the kill-switches can be put on "NC" position and/or the RBF switch can be released. The 3V3 MCU (refer to subsection 2.5.2 on the system overview chapter) indicates that the 3.3V power is being sourced, the system on led can be checked to see any easily dectable missbehavior right after the programming of the board.

6.1.2 Powering through Power Supply

To power the EPS module through external power supply, two voltage sources are needed. The external power sources can be set up in two ways, either two $3.7V$ sources

in series, or a $3.7V$ source for reference and a $7.4V$ source for power. In both cases the voltage sources should be connected to one of the battery board connectors (P1 or P5).

To use two $3.7V$ sources, the first source should have it's negative and positive terminals connected to Vbat- and Vbat_common pins of the batteries connector respectively, the second source should have it's negative and positive terminals connected to Vbat_common and Vbat+ pins of the batteries connector respectively (refer to Figure 3.10 and Table 3.5 on the hardware chapter).

To use $3.7V$ and $7.4V$ sources, the $3.7V$ source should have it's negative and positive terminals connected to Vbat- and Vbat_common pins of the batteries connector respectively, the $7.4V$ source should have it's negative and positive terminals connected to Vbat- and Vbat+ pins of the batteries connector respectively (refer to Figure 3.10 and Table 3.5 on the hardware chapter).

If the external power supply has configurable current limitation capability, it should be adjusted accordingly to the expected consumption of the load connected to the EPS module (other modules being powered by the EPS, batteries heaters being active, etc.), if powering the EPS in isolation (no load connected) a limit of $80mA$ may be applied.

After correctly connecting the voltage sources, the kill-switches can be put on "NC" position and/or the RBF switch can be released.

6.1.3 Powering through Batteries

To power the EPS module from the batteries the BAT4C module must be connected through the P5 connector (refer to Figure 3.10 on the hardware chapter). The batteries can be charged if needed using the P3 PicoBlade connector (refer to subsection 3.9.7 on the hardware chapter), it will also required an external interface or an adapted cable to be used for interfacing the charger device to the PicoBlade. The batteries also can be charged directly from the BAT4C module, for this refer to its documentation on the usage instruction chapter for more details [2].

6.2 Log Messages

The EPS 2.0 has a UART interface dedicated for debugging, which is described in Table 3.3. It follows a log system structure to improve the information provided in each message. The messages can be acquired by connecting an USB cable to the IIP Nº3 board that has an integrated FTDI FT4232H IC [7].

CHAPTER 7

Test Procedures

This chapter follows a standard workflow created by SpaceLab for testing its CubeSat modules, these procedures are first referenced in FloripaSat-2 documentation chapter 7[1]. The Table 7.1 resumes the workflow and each one of the tests types, subtests and code identification. Some subtests can be considered to not be applicable for the platform, the order in which they are done doesn't need to follow the numeration and other have generic titles that don't require too much explanation to be accomplished. The particularities of the EPS 2.0 for each test type are described in the following sections, and the results of those tests in the v0.1 and v0.2 versions of the EPS 2.0 are presented in the Appendix B and ??.

7.1 Visual Inspection

The first steps when receiving a manufactured (and in some cases assembled) PCB is to inspect visually to see if it is according to its expected appearance. The EPS 2.0 has more than 300 components, so it is advised for this test to not take a long time to accomplish, the major noted problems should be spotted and reported. The Table 7.2 resumes the visual inspection steps.

7.2 Mechanical Inspection

These tests verify if the board has the expected mechanical specs prior to integration. The Table 7.3 resumes the mechanical inspection steps.

7.3 Integration Inspection

These tests verify the integration accordance prior to the module's full assembly on the CubeSat. The Table 7.4 resumes the integration inspection steps.

7.4 Electrical Inspection

These tests verify if the components used in the mounted module are the correct ones and if they are well soldered, as well if the their any other error in the layout. Table 7.5 resumes the electrical inspection steps.

Test type	Subtests	ID
A. Visual Inspection	1. Packaging quality assessment	TA1
	2. Board manufacturing and assembly quality	TA2
	3. 3D model comparison	TA3
	4. Layers marker	TA4
	5. Labels (schematics comparison)	TA5
	6. High resolution photos for documentation	TA6
B. Mechanical Inspection	1. Board dimensions and mounting holes positioning	TB1
	2. Board weight measurement	TB2
C. Integration Inspection	1. Check connectors pinout against the documentation	TC1
	2. Check connectors positioning	TC2
D. Electrical Inspection	1. Solder shorts	TD1
	2. Missing components	TD2
	3. Lifted pins	TD3
	4. Poor soldering	TD4
	5. Swapped components	TD5
	6. Components partnumber	TD6
E. Electrical Testing	1. Continuity test	TE1
	2. Power up procedures	TE2
	3. Average input power consumption measurement	TE3
	4. Average output power source measurement	TE4
	5. Power tracks temperature	TE5
	6. Simple signal integrity	TE6
F. Functional Testing	1. Simple test code run	TF1
	2. System code run	TF2
	3. System hardware self-test flags check	TF3
	4. Monitor LEDs behavior	TF4
	5. Monitor the debug serial port logs	TF5
G. Module Testing	1. Review operation behavior	TG1
	2. Review features and requirements fulfillment	TG2
	3. Review communication buses configuration and protocol	TG3
	4. Review data packages, power buses and control signals	TG4
	5. Review edge cases and evaluate damage	TG5
	6. Run remote automated code tests	TG6
	7. Run system test codes in the board	TG7
	8. Run latest stable code version and review behavior	TG8

Table 7.1: Test workflow table.

7.5 Electrical Testing

Now it is necessary to use equipment such multimeter , oscilloscope, power supplies or anything that can be used to evaluate the interface between components, connectors. Table 7.6 resumes the electrical testing steps.

Table 7.2: Visual Inspection test steps.

Test type	Visual Inspection
subtests code	Description
TA1	Verify the received package, review the packaging protection used and if it maintained the physical and/or aesthetically integrity of the board.
TA2	See if the overall quality specified on the manufacturing process is according with its IPC class, the engineering model should be Class 2 while the flight model Class 3.
TA3	Inspect the top and bottom side of the PCB and verify if all the components are well soldered, if their polarity and labels are correct according to schematics and 3D model. Figure 3.1 and 3.2 for reference. If the board is not yet assembled detail the inconsistencies found during the assembly process.
TA4	Take a high resolution centered photo of the both sides of the board for documentation, avoid reflections if possible.
Success Criteria	The board needs to appear functional regarding its visual electrical specs.

Table 7.3: Mechanical Inspection test steps.

Test type	Mechanical Inspection
subtests code	Description
TB1	Verify board dimensions outline and mounting holes size and positioning with a measurement tool or through the board's draftsman sheet 1:1 scale print.
TB2	Measure board weight with an electronic balance with at least 0.1 grams of precision.
Success Criteria	The fabricated board needs to follow the mechanical specification of the draftsman document.

7.6 Functional Testing

It is necessary to evaluate if the module is able to run any code, and if it is ready to be used during FSW development and debugging. Table 7.7 resumes the functional testing steps.

7.7 Module Testing

The module itself is going to be tested, to evaluate if it can communicate with other modules, if its sensors are measuring correctly all the variables (e.g., voltage, current,

Table 7.4: Integration Inspection test steps.

Test type	Integration Inspection
subtests code	Description
TC1	Check external connectors pinout labels against the documentation present on section 3.9.
TC2	Check connectors positioning and possible integration issues from connected cables with the mounted BAT4C board and solar panels.
Success Criteria	The labels and placement of the external connectors must be according to the documentation and don't present any possible integration issues.

Table 7.5: Electrical Inspection test steps.

Test type	Electrical Inspection
Subtests code	Description
TD1	Check for any visible short circuit in the buses of the module. At this moment, it is not necessary to use any equipment, it is a "visual inspection" only.
TD2	Check if all the components were soldered. If there are missing components, it is necessary to evaluate if it was intended or not.
TD3	Check if the pins of the ICs and other components are damaged.
TD4	Check if there is any poor soldering.
TD5	Check if the components are swapped or even connected "backwards".
TD6	Check the partnumber of the components.
Success Criteria	All the intended components are connected correctly to the board, without any visible errors.

temperature) and if it is operating as expected. Table 7.8 resumes the module testing steps.

Table 7.6: Electrical Testing steps.

Test type	Electrical Testing
Subtests code	Description
TE1	Using a multimeter, check for any short circuit in the buses of the module, primarily the power buses.
TE2	If there the module do not present short circuits between a GND and VCC signals, power up the module with a power supply to see if it turns on.
TE3	Check the power consumption of the module.
TE4	Check if the converters of the module are working as intended.
TE5	Check if the module is not overheating.
TE6	Check if the signals are being transmitted correctly through the buses.
Success Criteria	The modules do not present any short circuit and can be turned on without any critical problem.

Table 7.7: Functional Testing steps.

Test type	Functional Testing
Subtests code	Description
TF1	Check if the module is able to run a code. Preferably something extremely simple, such as a “blinking LED".
TF2	Check if the module is able to run its firmware, even if the firmware is not complete.
TF3	Check if the module is able to make some self-tests. The results of these self-tests are going to be presented in its log.
TF4	Check if the LEDs are working as intended.
TF5	Check the logs for a longer period of time.
Success Criteria	The module has the minimal expected functionalities.

Table 7.8: Module Testing steps.

Test type	Functional Testing
Subtests code	Description
TG1	Run and review the firmware of the module and evaluate it, using any equipment that the SpaceLab offers.
TG2	Check and review the drivers, devices, tasks to see if the features are correctly being implemented.
TG3	Check and review the configuration of the communication protocols used.
TG4	Check and review all the buses to see if there is any anomaly or unexpected behavior.
TG5	Check and review the operation of the module when submitted a edge situations.
TG6	Run the unit tests.
TG7	Run system test codes in the board.
TG8	Run latest stable code version and review behavior.
Success Criteria	The is operating correctly.

Bibliography

[1] Space Technology Research Laboratory (SpaceLab). *FloripaSat-2 Documentation*, 2021. Available at <https://github.com/spacelab-ufsc/floripasat2-doc>.

[2] Space Technology Research Laboratory (SpaceLab). *Battery Module 4C Documentation*, 2021. Available at <https://github.com/spacelab-ufsc/battery-module-4c>.

[3] Space Technology Research Laboratory (SpaceLab). *EPS 1.0 Documentation*, 2019. Available at <https://github.com/spacelab-ufsc/eps>.

[4] Space Technology Research Laboratory (SpaceLab). *EPS 2.0 Documentation*, 2021. Available at <https://github.com/spacelab-ufsc/eps2>.

[5] Space Technology Research Laboratory (SpaceLab). *OBDH 2.0 Documentation*, 2020. Available at <https://github.com/spacelab-ufsc/obdh2>.

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

[7] SpaceLab. Interstage Interface Panels, 2021. Available at <https://github.com/spacelab-ufsc/interface-board>.

[8] Amazon Web Services, Inc. FreeRTOS - Real-time operating system for microcontrollers, 2020. Available at <https://www.freertos.org/>.

[9] Texas Instruments. MSP Driver Library, 2020. Available at <https://www.ti.com/tool/MSPDRIVERLIB>.

APPENDIX A

Test Report of v0.1 Version

This appendix is a test report of the first manufactured and assembled PCB (version v0.1).

- **PCB manufacturer**: PCBWay (China)
- **PCB assembly**: PCBWay (China)
- **PCB arrival date**: 2021/04/14
- **Execution date**: 2021/06/13 to 2021/07/01
- **Tester**: André M. P. Mattos

A.1 Visual Inspection

Figure A.1: Top view of the EPS 2.0 v0.1 board.

Figure A.2: Bottom view of the EPS 2.0 v0.1 board.

- **Test description/Objective**: Inspection of the board, visually and with a multimeter, searching for fabrication and assembly failures.
- **Procedures:** Table 7.2.
- **Material**: None.
- **Results**: The results of this test can be seen in Figures B.1 (top view of the board) and B.2 (bottom view of the board).
- **Conclusion**: No major problems were identified on this test, just some labels with typos (all in the PC-104 region) that should be already be corrected for v0.2 and four 4-pin picoblades that were not soldered by the manufacturer due the lack of clearance (differently from the headers and PC-104 which were intentionally removed from the process). This last item might be ignored since the manufacturer used an automated soldering process and the manual placement of these connectors is totally feasible.

A.2 Mechanical Inspection

- **Test description/Objective**: Evaluate if the board has the nominal mechanical specs prior to integration.
- **Procedures:** Table 7.3.
- **Material**: None.
- **Results**: The inspection was not performed due to the lack of the proper tools.

- **Conclusion**: Even without the test, the board should not present any problem due to the heritage from the previous models and the good manufacturing quality.

A.3 Integration Inspection

- **Test description/Objective**: Analyze the integration accordance prior to the module's full assembly on the CubeSat.
- **Procedures:** Table 7.4.
- **Material**: None.
- **Results**: Schematic files and pinouts identified in the chapter 3.
- **Conclusion**: No problems were identified on this test.

A.4 Electrical Inspection

- **Test description/Objective**: Inspect the visually detectable electrical features.
- **Procedures:** Table 7.5.
- **Material**: None.
- **Results**: The results of this test can be seen in Figures B.1 (top view of the board) and B.2 (bottom view of the board).
- **Conclusion**: No problems were identified on this test, components were correctly selected, placed and soldered (except for those already mentioned in the visual inspection).

A.5 Electrical Testing

- **Test description/Objective**: Perform basic tests to evaluate the board with nominal operating parameters.
- **Procedures:** Table 7.6.
- **Material**:
  - Multimeter Fluke 179
  - Keysight N6705B DC Power Analyzer
- **Results**: Results reported with the following images: A.3, A.4, A.5, A.6, A.7, A.8, A.9 and A.10.
- **Conclusion**: The boards power-up as expected and present stable power consumption. The power outputs (step-down regulators) underperform with nominal or slightly higher load parameters. The issue might be related to poor sizing of passive components required for the regulators. This problem **must** be solved for the next version (it is expected to be performed minor changes in components values).

(a) Test setup for power characterization.

(b) Board connectors and harness used during the tests.

Figure A.3: Electrical test setup of the power-up sequence and output power supply channels.

Figure A.4: Power consumption during standby with any intensive firmware task.

(a) Load: 0mA.

(b) Load: 500mA.

(c) Load: 1000mA.

(d) Load: 1500mA.

Figure A.5: Payload step-down regulator power characterization.

(a) Load: 0mA.

(b) Load: 500mA.

(c) Load: 1000mA.

(d) Load: 1500mA.

Figure A.6: Antenna step-down regulator power characterization.

(a) Load: 0mA.

(b) Load: 500mA.

(c) Load: 1000mA.

(d) Load: 1500mA.

Figure A.7: Radio 0 step-down regulator power characterization.

(a) Load: 0mA.

(b) Load: 500mA.

(c) Load: 1000mA.

(d) Load: 1500mA.

Figure A.8: Radio 1 step-down regulator power characterization.

(a) Load: 0mA.

(b) Load: 500mA.

(c) Load: 1000mA.

(d) Load: 1500mA.

Figure A.9: OBDH step-down regulator power characterization.

(a) Load: 0mA.

(b) Load: 500mA.

(c) Load: 1000mA.

Figure A.10: EPS/TTC step-down regulator power characterization.

A.6 Functional Testing

A.6.1 Firmware Programming

- **Test description/Objective**: Evaluate the board behavior under a firmware programming sequence.
- **Material**:
  - Code Composer Studio v10.0.0
  - MSP-FET Flash Emulation Tool
  - USB-UART converter

  - PuTTy

- **Results**: The results of this are available in Figure B.7, where the log messages of the first boot of the board can be seen.

- **Conclusion**: Major problems were identified on this test, but it was expected since the available firmware version was at early stages of refactoring and development.

(a) Board connections using the MSP-FET and USB-UART converter.

(b) Pin connections using the MSP-FET and USB-UART converter.

Figure A.11: Setup used for the first firmware boot

Figure A.12: Log messages during the first boot.

A.7 Conclusion

The board needs to be tested more, to evaluate its functioning with the firmware that is being developed.

APPENDIX B

Test Report of v0.2 Version

This appendix is a test report of the first manufactured and assembled PCB (version v0.2).

- **PCB manufacturer**: PCBWay (China)
- **PCB assembly**: PCBWay (China)
- **PCB arrival date**: 2022/04/14
- **Execution date**: 2022/04/15 to 2022/05/15
- **Tester**: João Cláudio and Ramon Borba

B.1 Visual Inspection

- **Test description/Objective**: Inspection of the board, visually, searching for fabrication and assembly failures.
- **Procedures:** Table 7.2.
- **Material**: None.
- **Results**: First, the top and bottom side of the EPS 2.0 are shown in Figure B.1 (top view of the board) and Figure B.2 (bottom view of the board). It was possible to see that some components were not soldered. At the bottom side the R89, R90, R91, versioning resistors V1 and V4, R93, R87, R88, R_ESD were not soldered. At the top side only half of the PC-104 and P1 were not connected. Furthermore, the P5 was connected in the wrong side.
- **Conclusion**: No major problems were identified on this test.

Figure B.1: Top view of the EPS 2.0 v0.2 board.

Figure B.2: Bottom view of the EPS 2.0 v0.2 board.

B.2 Mechanical Inspection

- **Test description/Objective**: Evaluate if the board has the right dimensions and the necessary mechanical specs prior to integration.
- **Procedures:** Table 7.3.
- **Material**: Pachymeter, rule and a digital scale.
- **Results**: The inspection was performed with the available tools. The board presented the right dimensions and has 90g.
- **Conclusion**: The board should not present any problem due to the heritage from the previous models and the good manufacturing quality.

B.3 Integration Inspection

- **Test description/Objective**: Analyze the integration accordance prior to the module's full assembly on the CubeSat.
- **Procedures:** Table 7.4.
- **Material**: None.
- **Results**: Schematic files and pinout identified in the chapter 3.
- **Conclusion**: No problems were identified on this test, except from those found during the visual inspection.

B.4 Electrical Inspection

- **Test description/Objective**: Inspect for visually detectable electrical errors.
- **Procedures:** Table 7.5.
- **Material**: None.
- **Results**: The top and bottom side of the EPS 2.0 are shown in Figure B.1 (top view of the board) and Figure B.2 (bottom view of the board).
- **Conclusion**: No problems were identified on this test, components were correctly selected, placed and soldered (except for those already mentioned in the visual inspection).

B.5 Electrical Testing

(a) 5V bus to the payload.

(b) 5V bus to the TTC.

(c) 6V bus to the TTC.

(d) 3,3V bus to multiple modules.

Figure B.3: Measuring the voltage of the power buses without a load.

- **Test description/Objective**: Perform basic tests to evaluate the board with nominal operating parameters.
- **Procedures:** Table 7.6.
- **Material**:
  - Fluke's DMMs (Digital Multimeters) from 170 series
  - N6705B, Keysigh's DC Power Analyser
- **Results**: First, the multimeter was used to search for short circuits between the VCC and GND buses. Then, the EPS 2.0 was turned on with a power supply. Moreover, it was measured the voltage on the power buses without any load, which the results are presented in Figure B.3. Furthermore, a load was connected to these buses to evaluate if it would occur a voltage drop. The setup for 5V and 6V is shown in Figure B.4 and for 3,3V is shown on Figure B.5. For these setups, the load would drain approximately 1A.
- **Conclusion**: The board turned on as expected and present stable power consumption. The voltage at the output of the step-down regulator of the 3,3V bus to power up the Beacon had a significant drop. The issue might be related to poor sizing of passive components required for the regulators. This problem **must** be solved for the next version (it is expected to be performed minor changes in components values).

Figure B.4: Setup to evaluate the 5V and 6V buses with load.

Figure B.5: Setup to evaluate the 3,3V buses with load.

B.6 Functional Testing

B.6.1 Firmware Programming

(a) Board connections using the MSP-FET and USB-UART converter.

(b) Pin connections using the MSP-FET and USB-UART converter.

Figure B.6: Setup used for the first firmware boot

- **Test description/Objective**: Evaluate the board behavior under a firmware programming sequence.

- **Material**:
  - Code Composer Studio v10.0.0
  - MSP-FET Flash Emulation Tool
  - USB-UART converter
  - PuTTy
- **Results**: The results of this are available in Figure B.7, where the log messages of the first boot of the board can be seen.
- **Conclusion**: Major problems were identified on this test, but it was expected since the available firmware version was at early stages of refactoring and development.

Figure B.7: Log messages during the first boot.

B.6.2 Communication Buses

- **Test description/Objective**: Test the communication buses of the board, as listed below:

  - I^2C bus, to communicate with the OBDH
  - UART bus, to communicate with the TTC

- **Material**:
  - Saleae Logic Analyzer (24 MHz, 8 channels)
  - Saleae Logic software (v1.2.18)
  - MSP-FET Flash Emulation Tool

- **Results**: The results of this test can be seen in Figure B.9 and Figure B.8

- **Conclusion:** Minor problems were identified on this test, but were corrected. The figures shows the buses after the corrections, working as expected.

(a) View of the communication.

(b) Zoom in the data sent by EPS.

Figure B.8: Communication between the TTC and EPS

Figure B.9: Communication between the OBDH and EPS.

B.6.3 Sensors

- **Test description/Objective**: Test the EPSs' sensors, as listed below:
  - Voltage sensors
  - Current sensors
  - Temperature sensors
- **Material**:
  - Code Composer Studio v9.3.0
  - MSP-FET Flash Emulation Tool
  - USB-UART converter
  - Screen (Linux software)
- **Results**: The values of current, voltage and temperature are correct. The temperature values are shown in Figure B.10. The values shown in the figure were obtained during the tests of the heater's on-off controller.
- **Conclusion:** In a first moment, the values of voltage and current were not correct, but it was figured out that the problem was in the firmware. The values of temperature were, in a first moment, all 0°C. But, again, the problem was in the configuration of the ADS1248. When these problems were corrected, the values started to be correct.

```
[ 304404 ] Heater Controller: Temp Reading [K]: 305
[ 304465 ] Heater Controller: Temp Reading [K]: 305
[ 306404 ] Heater Controller: Temp Reading [K]: 305
[ 306465 ] Heater Controller: Temp Reading [K]: 305
[ 308404 ] Heater Controller: Temp Reading [K]: 305
[ 308465 ] Heater Controller: Temp Reading [K]: 305
[ 310404 ] Heater Controller: Temp Reading [K]: 305
[ 310465 ] Heater Controller: Temp Reading [K]: 305
[ 312404 ] Heater Controller: Temp Reading [K]: 305
[ 312465 ] Heater Controller: Temp Reading [K]: 305
[ 314404 ] Heater Controller: Temp Reading [K]: 304
[ 314465 ] Heater Controller: Temp Reading [K]: 304
[ 316404 ] Heater Controller: Temp Reading [K]: 304
[ 316465 ] Heater Controller: Temp Reading [K]: 304
[ 318404 ] Heater Controller: Temp Reading [K]: 304
[ 318465 ] Heater Controller: Temp Reading [K]: 304
[ 320404 ] Heater Controller: Temp Reading [K]: 304
[ 320465 ] Heater Controller: Temp Reading [K]: 304
[ 322404 ] Heater Controller: Temp Reading [K]: 304
[ 322465 ] Heater Controller: Temp Reading [K]: 304
[ 324404 ] Heater Controller: Temp Reading [K]: 304
[ 324465 ] Heater Controller: Temp Reading [K]: 304
[ 326404 ] Heater Controller: Temp Reading [K]: 304
[ 326465 ] Heater Controller: Temp Reading [K]: 304
[ 328404 ] Heater Controller: Temp Reading [K]: 303
[ 328465 ] Heater Controller: Temp Reading [K]: 304
[ 330404 ] Heater Controller: Temp Reading [K]: 303
[ 330465 ] Heater Controller: Temp Reading [K]: 304
[ 332404 ] Heater Controller: Temp Reading [K]: 303
[ 332465 ] Heater Controller: Temp Reading [K]: 304
[ 334404 ] Heater Controller: Temp Reading [K]: 303
[ 334465 ] Heater Controller: Temp Reading [K]: 303
[ 336404 ] Heater Controller: Temp Reading [K]: 303
[ 336465 ] Heater Controller: Temp Reading [K]: 303
[ 338404 ] Heater Controller: Temp Reading [K]: 303
[ 338465 ] Heater Controller: Temp Reading [K]: 303
[ 340404 ] Heater Controller: Temp Reading [K]: 303
[ 340465 ] Heater Controller: Temp Reading [K]: 304
[ 342404 ] Heater Controller: Temp Reading [K]: 303
[ 342465 ] Heater Controller: Temp Reading [K]: 304
[ 344404 ] Heater Controller: Temp Reading [K]: 304
[ 344465 ] Heater Controller: Temp Reading [K]: 304
[ 346404 ] Heater Controller: Temp Reading [K]: 304
[ 346465 ] Heater Controller: Temp Reading [K]: 304
[ 348404 ] Heater Controller: Temp Reading [K]: 304
[ 348465 ] Heater Controller: Temp Reading [K]: 304
```

Figure B.10: Values of temperature during the evaluation of heater's on-off controller.

B.6.4 Peripherals

B.6.5 Battery monitor

- **Test description/Objective**: Test the functionality of the DS2777.
- **Material**:
  - Saleae Logic Analyzer (24 MHz, 8 channels)
  - Saleae Logic software (v1.2.18)
  - MSP-FET Flash Emulation Tool
- **Results**: The results of this test can be seen in Figure B.11, Figure B.12 and Figure B.13.
- **Conclusion:** In a first moment, the CI was not working as expected. It was found that the Zener diode, D6, was interfering in the I^2C communication. It was removed and the CI started to work properly.

```
[ 196 ] Startup: FreeRTOS V10.2.0
[ 203 ] Startup: Hardware revision is 1
[ 210 ] Startup: System clocks: MCLK=31981568 Hz, SMCLK=31981568 Hz, ACLK=32768 Hz
[ 225 ] Startup: Last reset cause: 0x00
[ 232 ] LEDs: Initializing system LEDs...
[ 239 ] MPPT: Initializing the MPPT...
[ 247 ] Battery Monitor: Initializing the battery monitor...
[ 256 ] DS277X: Reading data from DS2777G+...
Register: 0x00
Value: 0x4F
[ 266 ] DS277X: Reading data from DS2777G+...
Register: 0x00
Value: 0x4F
[ 277 ] DS277X: Reading data from DS2777G+...
Register: 0x60
Value: 0x0C
[ 288 ] DS277X: Reading data from DS2777G+...
Register: 0x69
Value: 0x64
[ 299 ] DS277X: Reading data from DS2777G+...
Register: 0x64
Value: 0xA4
[ 310 ] DS277X: Reading data from DS2777G+...
Register: 0x65
Value: 0x18
[ 321 ] DS277X: Reading data from DS2777G+...
Register: 0x66
Value: 0x99
[ 333 ] DS277X: Reading data from DS2777G+...
Register: 0x67
Value: 0x12
[ 344 ] Current Sensor: Initializing the current sensor...
```

Figure B.11: First part of the log.

```
[ 459 ] Read Sensors: Reading sensors...
[ 2260 ] DS277X: Reading data from DS2777G+...
Register: 0x0C
Value: 0x5E, 0x80
[ 2272 ] DS277X: Reading data from DS2777G+...
Register: 0x1C
Value: 0x5E, 0x40
[ 2333 ] DS277X: Reading data from DS2777G+...
Register: 0x0A
Value: 0x18, 0x00
[ 2394 ] DS277X: Reading data from DS2777G+...
Register: 0x0E
Value: 0xFE, 0xA6
[ 2456 ] DS277X: Reading data from DS2777G+...
Register: 0x08
Value: 0xFE, 0xA2
[ 2518 ] DS277X: Reading data from DS2777G+...
Register: 0x01
Value: 0x27
[ 2579 ] DS277X: Reading data from DS2777G+...
Register: 0x00
Value: 0x4F
[ 2640 ] DS277X: Reading data from DS2777G+...
Register: 0x02
Value: 0x05, 0xE7
[ 2701 ] DS277X: Reading data from DS2777G+...
Register: 0x04
Value: 0x05, 0xE7
[ 2762 ] DS277X: Reading data from DS2777G+...
Register: 0x06
Value: 0x0C
[ 2823 ] DS277X: Reading data from DS2777G+...
Register: 0x07
Value: 0x0C
```

Figure B.12: Second part of the log.

B.6.6 Heater's on-off controller

- **Test description/Objective**: Test the functionality of the heater's on-off controller.

```
[ 196 ] Startup: FreeRTOS V10.2.0
[ 203 ] Startup: Hardware revision is 1
[ 210 ] Startup: System clocks: MCLK=31981568 Hz, SMCLK=31981568 Hz, ACLK=32768 Hz
[ 225 ] Startup: Last reset cause: 0x00
[ 232 ] LEDs: Initializing system LEDs...
[ 239 ] MPPT: Initializing the MPPT...
[ 247 ] Battery Monitor: Initializing the battery monitor...
[ 259 ] Current Sensor: Initializing the current sensor...
[ 277 ] Voltage Sensor: Initializing the voltage sensor...
[ 286 ] Temperature Sensor: Initializing internal MCU temperature sensor...
[ 297 ] Temperature Sensor: Current MCU temperature: 24 oC
[ 307 ] Temperature Sensor: Initializing ADS1248 device...
[ 339 ] OBDH: Initializing OBDH device...
[ 346 ] TTC: Initializing TTC device...
[ 354 ] Startup: Boot completed with SUCCESS!
[ 361 ] Time Control: The last saved system time is not available!
[ 373 ] Read Sensors: Reading sensors...
[ 2175 ] Read Sensors: Battery voltage [mV]: 7372
[ 2235 ] Read Sensors: Battery temperature [K]: 297
[ 2295 ] Read Sensors: Battery instantaneous current [mA]: -54
[ 2356 ] Read Sensors: Battery average current [mA]: -55
[ 2416 ] Read Sensors: Battery monitor status register: 0x27
[ 2475 ] Read Sensors: Battery monitor protection register: 0x4F
[ 2534 ] Read Sensors: Battery RAAC [mAh]: 2425
[ 2593 ] Read Sensors: Battery RSAC [mAh]: 2425
[ 2652 ] Read Sensors: Battery RARC [percent]: 13
[ 2711 ] Read Sensors: Battery RSRC [percent]: 13
```

Figure B.13: Last part of the log.

- **Material**:
  - Logs
  - MSP-FET Flash Emulation Tool
- **Results**: The results of this test can be seen in Figure B.14.
- **Conclusion:** The heater's on-off controller is working as expected. It was decided to used, during the tests in the workbench, thresholds such 26°C and 32°C.

Figure B.14: Temperature of the batteries being controlled.

B.6.7 MPPT controller

- **Test description/Objective**: Test the functionality of the MPPT controller.
- **Material**:
  - Oscilloscope
  - Unit tests
  - MSP-FET Flash Emulation Tool
- **Results**: The PWM signal generated by the MPPT controller can be seen in Figure B.15 and Figure B.16, as well as the test workflow can be accessed in the GitHub page.
- **Conclusion:** The algorithm was tested using the unit tests, and worked as expected. The PWM signal for 100kHz and 50kHz are being generated correctly.

Figure B.15: PWM signal, at 50kHz, generated by the MPPT controller.

Figure B.16: PWM signal, at 100kHz, generated by the MPPT controller.

B.7 Conclusion

The board can be connected with the rest of the satellite, using a FlatSat platform, to make system level tests.