OBDH Engineering Model #1

AcubeSAT-OBC-EC-013

Theodoros Katzalis
Konstantinos Kanavouras

December 3, 2019
Version: 1.0

Aristotle University of Thessaloniki

Aristotle Space and Aeronautics Team
CubeSat Project

2019

Contents

Changelog

Date	Version	Document Status	Comments
3/12/2019	1.0	INTERNALLY RELEASED	First release

This is the latest version of this document (1.0) as of December 3, 2019. Newer versions might be available at https://helit.org/mm/docList/AcubeSAT-OBC-EC-013.

Acronyms

BGA Ball Grid Array

BOM Bill Of Materials

CAD Computer-aided Design

CAN Controller Area Network

COBS Consistent Overhead Byte Stuffing

DMA Direct Memory Access

EMC Electromagnetic compatibility

EPS Electrical Power Supply

FM Flight Model

FPU Floating Point Unit

GPIO General Purpose IO

HAL Hardware Abstraction Layer

IC Integrated Circuit

IO InputOutput

LL Low Level

LSF Libre Space Foundation

MCU Microcontroller

MRAM Magnetoresistive Random Access Memory

NSS Negative Signal Select Port

NV Non-Volatile

PCB Printed Circuit Board

PTC Positive Temperature Coefficient

PWM Pulse Width Modulation

Q Quarter

QFN Quad Flat No-leads

QFP Quad Flat Package

RAM Random Access Memory

SMD Surface-Mount Device

SWD STMicroelectronics Serial Wire Debug

TC Telecommand

TIF Thessaloniki International Fair

TSOP Thin Small Outline Package

TXRX TransmissionReception

USART Universal Synchronous Asynchronous Receiver Transmitter

via Vertical Interconnection Access

1 Introduction

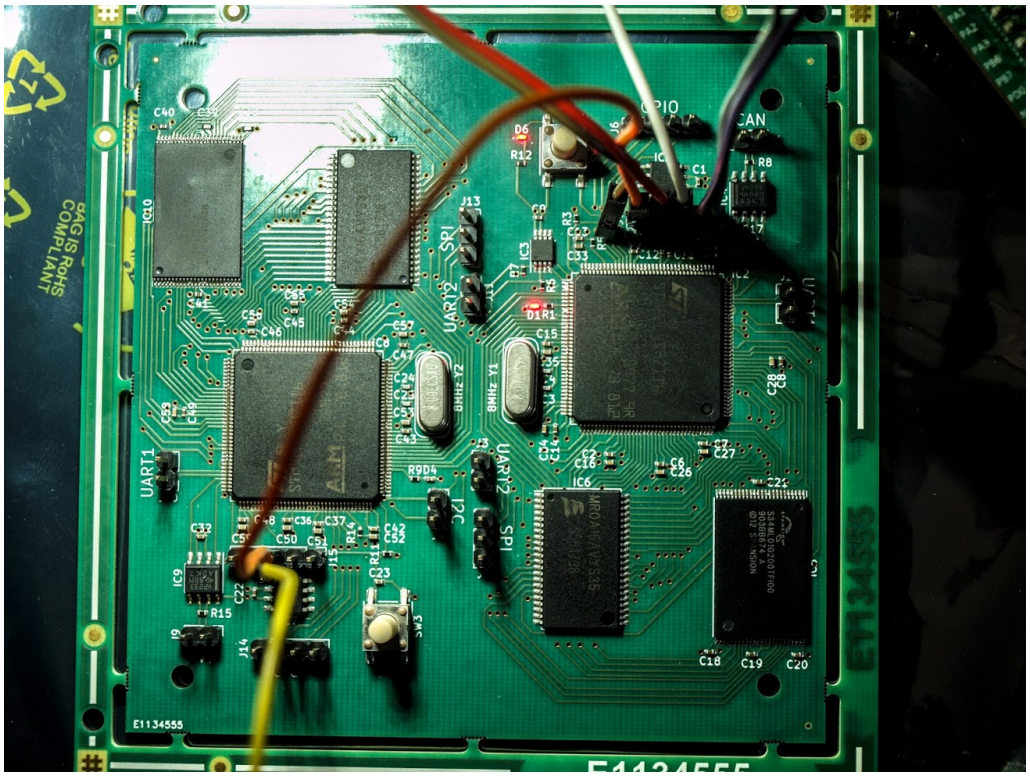


***Figure 1**: Top view of the OBDH EM*

During Q3 2019, an **Engineering Model** (EM) of the On-Board Data Handling or On-Board Computer module was designed by our team, assembled by Prisma Electronics, and used in a public demonstration of the AcubeSAT mockup. This report aims to explain the rationale behind all decisions for the EM, as well as the test results and suggestions for the next EM redesign.

The design is based on an older iteration of the OBDH board, which included a dual-redundant MCU architecture.

The EM proved useful:

1. As a prototype for the satellite's OBDH Flight Model (FM), in order to evaluate the extent of its functionality and handling
2. As a development board for the selected MCU, allowing us to develop satellite firmware much before the integration
3. As a practice target for Printed Circuit Board (PCB) design & development, in order to gain experience with the manufacturing procedure and its specifics
4. As a functional demonstration of the team's designs for public events, such as the Thessaloniki International Fair (TIF)

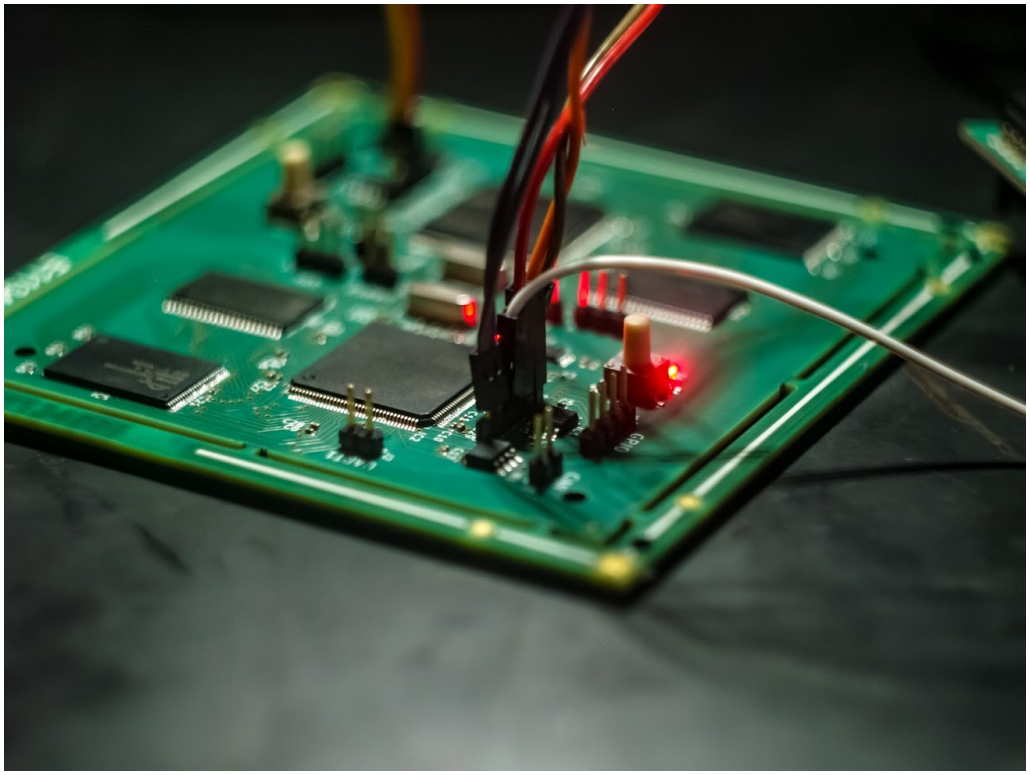


***Figure 2**: The OBDH EM*

2 Selected components

The design of a space-targeted and not batch-produced PCB, manufactured by Prisma Electronics, entailed the following constraints:

- Avoidance of **thru-hole** components, preference of **SMD** components, to ease techniques such as reflow soldering
- Avoidance of **No-leads** packages (e.g. QFN, BGA), preference of QFP ones. Packages with no visible leads cannot be visually inspected or manually repaired. Additionally, BGA packages, while easing routing, may be more prone to physical damage.

For the EM, in order to reduce procurement costs, we considered *low-capacity* and *standard-grade* versions of the components.

The selection of the components has been largely based on previous CubeSat designs, as the one used on UpSat (AcubeSAT-OBC-BC-015). The decision on the components is based on the team's design as of Q3 2019, detailed in AcubeSAT-OBC-MI-001 and AcubeSAT-OBC-MI-002:

- **Microcontroller**: STM32L4S9ZIT6
  STMicroelectronics' **STM32L4** series has been chosen as it offers:
  - A modern ARM Cortex-M4 with a Floating Point Unit (FPU) and a 120 MHz maximum clock, reducing any performance considerations
  - Impressive low-power and sleep mode power consumption figures
  - A large number of peripherals and IO ports, including USART, CAN, SPI,

I^2C, DMA, ADC, dual external memory access, ...
  - Hardware cryptography functions (used to verify signed TC)

  This particular MCU was chosen as the product with the highest **pin count** and **memory** (both volatile and non-volatile) out of the STM32L4 line.
  Do note that this particular product is not space proven; there is no documented reference of STM32L4s being used in space, however there is a large number of planned STM32L4 uses. In addition, STMicroelectronics' MCUs have been used in space extensively for the last years.

- **Non-volatile memory**: SkyHigh S34ML01G200TFI003
  TSOP-48 packages are the typical non-BGA variants of high-density NV memory chips. A flash memory was chosen due to its high density in a low price and supported interface with the MCU.
  Due to the **large amount of data** needed to be stored, **flash Integrated Circuits (ICs)** and **SD cards** were the only considerable options for our design. We have opted for NAND flash chips due to:
  - No mechanical considerations (i.e. no need to glue an SD card on a memory holder)
  - SD cards having a difficult and error-prone protocol to implement (as shown by past experience and Libre Space Foundation (LSF)'s recommendations
  - A fast and low-latency parallel interface between the memory and MCU
  - Direct access and knowledge of the underlying architecture of the memory, larger utilisation and easier debugging

  On the other hand, NAND memories tend to be more expensive ($/kB), do not include wear levelling & error-correcting controllers (which can be remedied by file systems, see AcubeSAT-OBC-BT-027), and require many more PCB traces.

  > Note that, in future designs, the NAND flash will be placed on the SU PCB, while the OBDH board will contain a lower-capacity NAND for packet storage purposes. See AcubeSAT-OBC-MI-005 for the rationale behind this decision.

  See AcubeSAT-OBC-EC-009 for utilization instructions of the NAND flash.

- **Watchdog**: Renesas/Intersil ISL88705IB846Z
  This particular watchdog was selected amongst a list of suitable products, as described in detail in AcubeSAT-OBC-BC-014.

- **"Volatile" RAM**: Everspin MR0A16AVYS35
  See AcubeSAT-OBC-BC-023 for an explanation on the selection of MRAMs as a temporary storage medium.
  The particular product was chosen due to its low latency features, and relatively low capacity, resulting in a lower price as well.

- **Temperature Sensor**: Microchip MCP9808T-E/MS
  A low-profile yet highly accurate temperature sensor, initially chosen by the SU subteam in AcubeSAT-SCI-BC-006.

- **CAN Bus Transceiver**: TI SN65HVD233
  A robust CAN transceiver frequently used with various applications requiring reliability. Radiation-tolerant version exists, but unknown availability.

2.1 Dual-redundant architecture

A **dual-redundant** architecture was initially chosen for the OBDH EM, the idea being that one MCU can take over if the other one fails.

In essence, the board was comprised of two separate OBDH platforms, **sharing the following parts**:

- The VCC, GND power supply
- Two "**heartbeat**" GPIO pins, connected between the two Microcontrollers (MCUs), to let each microcontroller monitor for the other's failure
- An I^2C line connected to a single MCP9808 temperature sensor

> Due to the lack of available space, increased complexity & desire to focus on other issues, and a low risk impact of a dual-redundant architecture, all subsequent OBDH boards will not implement the dual-redundant architecture.

Dual-redundant architectures were researched in AcubeSAT-OBC-BH-029.

2.2 Interfaces

In this EM, no effort was made to emulate the actual communication bus of the satellite.

Due to its debugging and development purpose, the following **interfaces** were added to the board (per MCU), in the form of 2.54mm pitch **male pin headers**:

- **Power input**: VCC & GND to power the MCUs.
- **SWD**: The most significant interface. Necessary to program and debug MCUs.
- 2x **USART**: The ubiquitous "serial" ports are used to **interface with computers via USB**[1].
- **SPI**: Interface used for medium-speed peripherals (e.g. transceivers). Included the hardware NSS port.
- **I^2C**:[2] Interface used for low-speed peripherals (e.g. sensors).
- **CAN**: The CANH and CANL lines for inter-MCU communication
- 4x **GPIO pins**: General Purpose IO (GPIO) pins serve as "*wildcard*" pins that can be used for any purpose (e.g. sensor interrupts, LEDs, SPI NSS, ...). If possible, assign *timer* or *interrupt* MCU pins here, to allow for extended flexibility.
- **Reset button**: Linked to the MCU's or the watchdog's *reset* port, this button is the first station to visit when something goes wrong.

It is crucial to include an ample amount of interfaces in order to serve **any interfacing needs that might come up** during debugging.

[1] 2 ports were added in order for one to be the computer interface, and the other to communicate with an external MCU or transceiver to send/receive packets. We considered the possibility of a wireless connection not being available, in which case a wired USART port could be used instead

[2] Only a single shared I^2C line was routed between the MCUs

3 PCB design & production

The PCB design is comprised of low-power digital signals and clock frequency up to 120MHz.

In order to decide if the circuit should be approached as lumped or distributed, the faster rising time of the signals and the maximum frequency of interest should be evaluated. According to the datasheet of the MCUs (the source of the digital signals), the faster rise time for an IO port is 3 ns. So, the bandwidth of the signal is calculated approximately 120 MHz using the following equation:

$$f = 0.35/t$$

where t is the rise time. As a rule of thumb, when the interconnection length is greater than the $\lambda/7$ of the maximum frequency of interest, then these interconnections should be treated as transmission lines. Using this rule of thumb in conjunction with the velocity of propagation in FR4 dielectric, lines that exceed roughly 180 mm should be handled by the distributed-element model. Finally, the team decided to approximate the circuit as lumped and not distributed as long as the aforementioned length limit of tracks remained unviolated. On the other hand, in the real world rising times might be much faster than the specifications and in general it would be wise to include transmission line techniques when designing the PCB.

The guidelines that the team used to achieve signal integrity and EMC are the following:

1. Traces are kept as **short** as possible (minimize loop areas) to mitigate transmission line phenomena. Thus, matching impedance and termination assumed redundant techniques.
2. **Crosstalk**
   The clearance of adjacent traces increased as much as possible to oppose unwanted coupling between signals.
3. **Circuit segregation**
   Partitioning the circuit requires to identify sources of noise (high speed signals like clock and bus lines) as well as sensitive components and the coupling mechanism (paths). In our case, analog and sensitive parts weren't included in the design, so our primary focus was to isolate the clock circuitry from the rest of the tracks and to keep it close to the MCU's external clock interface. Bus lines and any high speed signals should be treated in the same way with the clock traces and discontinuities ,using vias, along these lines should be avoided to ensure deterministic return paths to the adjacent ground layer.
   Another important factor to be considered for the segregation is the interface between the PCB and the "outside" world. **Connectors** should be kept in a specific area near the edges and away from high speed circuitry to avoid unwanted voltage differences between the connectors.
4. **PCB stack-up**
   As mentioned before, high speed signals should be routed adjacent to a ground layer. To accomplish that, a 4 layer stack-up had been used. The stack-up configuration was 1. signal layer, 2. ground plane, 3. power plane, 4. signal layer. This configuration provides low impedance path for the return signals contributing to EMC and signal integrity. It is worth mentioning that the planes should be solid

(not gapped). Otherwise slots interrupt the path of the return signal creating a loop area that increase the self-inductance and radiated emissions.

5. **Power Decoupling**
   Decoupling capacitors had been located in each power supply pair as near as possible. The values of the capacitors are recommended from the datasheets of the electrical components. The power and ground planes due to the selected stack-up contribute to the decoupling as well.
6. **Surface-mounted devices**
   The main benefits of SMD over through hole components are the reduced inductance and the smaller available form factors that result for the SMDs to be placed closer to other components.

More information about the design can be found on the **PCB handbook** AcubeSAT-OBC-G-009.

3.1 Workflow

The CAD tool used to design the PCB was KiCad. It is a reliable and free software suite with sufficient features for our board designs. KiCad will be preferred for the next iterations too.

After the selection of the components, the first thing that should be done to start designing a PCB, is the configuration of the libraries (schematic symbols and footprints). The SamacSys component search engine was used for this reason.

Having complete PCB libraries, in order to start with the schematic design, the datasheets of the components and application notes regarding the hardware development, had been examined carefully. Recommendations from these documents to ensure proper functionality (e.g. values for decoupling capacitors, terminating resistors for the CAN bus) had been taken into account. Finally, the layout of the PCB had been designed according to the guidelines for routing and placement referred to section 3.

Git had been used for version control during the implementation. Additionally, the PCB designed with parallel collaboration of two team members. One member had been selected to operate and the other one to review. This type of approach helped to solve problems and concerns very efficiently due to the merging of opinions and knowledge.

The entire design of the PCB is available at https://gitlab.com/acubesat/obc/obc-pcb/tree/1.1 (including post-production fixes).

3.2 Technical specifications

The PCB manufactured by Eurocircuits. It is a 4 layer board (the decision explained in item 4) and has the following characteristics:

- The board thickness is approximately 1.6mm.
- For inner layers the standard copper foil thickness is 35µm/1oz.
- For outer layers the standard copper foil thickness is 18µm/½oz. Over this 20 – 25µm (+½oz) of copper is plated so the final thickness is equivalent to (just over) 35µm/1oz copper.

- Base material FR4 (approx 0.71 mm).
- The Imagecure XV501T-4 LDI soldermask (approx. 25-45 μm).
- Electroless nickel immersion gold as surface finish (approx. 7 μm).

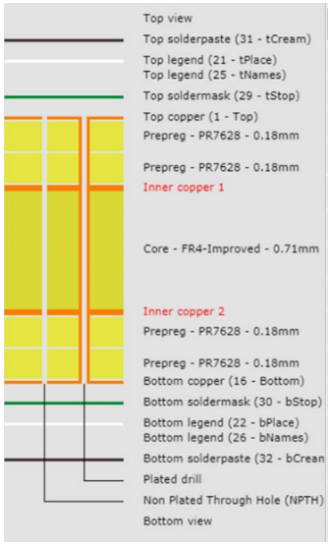


***Figure 3**: Board layers*

4 Testing

To evaluate the functionality of the received EM, the following tests were performed before programming:

1. **Visual inspection**
   We performed visual tests for dust and any unwanted solder joints[3]. Furthermore, the **orientation of the components** was thoroughly checked, according to the design files.
2. **Resistance measurements**
   To ensure the lack of any trivial **shorts**, the resistance between the VCC and GND was measured, and found sufficiently high (> 1 kΩ) to proceed with testing.
3. **Application of voltage**
   3.3V were applied from a current-limited supply to ensure LED operation and lack of any other spurious shorts. A current consumption measurement is also needed to ensure lack of low-resistance paths.
   At this timepoint, the temperature of the components can be tested by thermal camera, non-contact thermometer, or (in cases of low enough current) by touch, to ensure no **overheating** of critical components.
4. **SWD connection and device identification**
   The board is now ready to be programmed. Tools such as STM32CubeProg or OpenOCD can identify the board. We used the following command to read the MCU's identifying information:

   ```
   STM32CubeProg/bin/STM32_Programmer_CLI -c port=SWD
   ```

[3]Due to a design mishap (see section 6), visual inspection could not be performed as efficiently

Success in this step means that the microcontroller is **booting successfully**, receiving a stable power supply, and not being reset. Any piece of software can be loaded via SWD.

5. **Hello world**
   The first LED blinking and USART communication programs were uploaded to the MCU.
6. **Component testing**
   Communication with each of the on-board components (sensors, memories, transceivers) can now start. In most cases, a complex program will need to be written to interface with each component. An oscilloscope will be handy to debug connection issues.

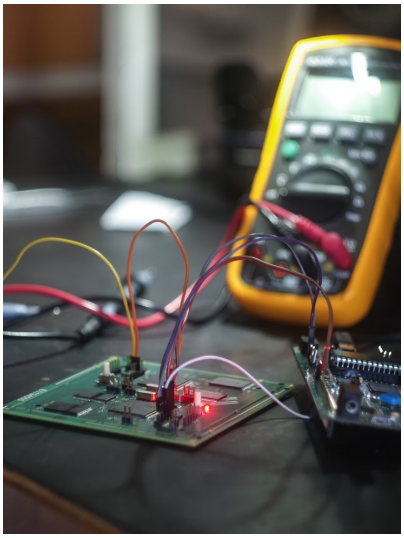


***Figure 4**: EM testing*

After validating that all components of the board can successfully work *in a concurrent and parallel fashion*, the main functionality of the board can be considered validated.

4.1 Preliminary testing outcomes

No irreparable faults were discovered during the above preliminary testing. In conjuction with the ones mentioned in section 6, the foremost issues discovered were:

1. A **misconnection of the watchdogs that permanently reset microcontrollers**, rendering them unable to be programmed.
   This was fixed by *physically disconnecting* the watchdogs' links to the MCUs.
2. A **high power consumption** (and thermal dissipation) by the **MRAM** chips.
   Estimated to occur due to a lack of pull-up resistors in the Chip Select pins of the memories, resulting in spurious operations due to their inputs being in a Hi-Z state.
3. A few misnamed or swapped pins on the schematic

4.2 Tests to be performed

- Operation tests of the **NAND Flash**
- Tests for concurrent operation of all components
- **Crosstalk** test with oscilloscope between neighbouring lines. Rise time and various other timing tests.
- Mechanical stress tests

***Figure 5**: EM programming*

5 Usage in 4th mockup

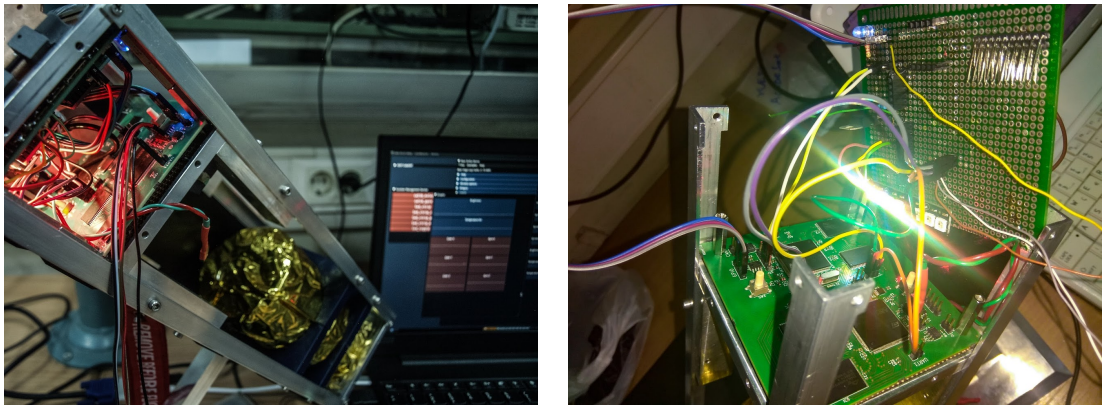


***Figure 6**: The EM inside the 4th mockup*

The OBDH EM was the "heart" of the 4th AcubeSAT 3U mockup, presented over 9 days in TIF 2019 to a wide audience.

Source code for the above mockup can be found at https://gitlab.com/acubesat/obc/mockup-4.

The mockup was programmed on one of the STM32L4S9 MCUs available on the board in FreeRTOS/C++, and included the following functionality:

- AcubeSAT **ECSS Services** implementation (ST[08] function management and ST[20] parameter management)
- Procurement of **sensor data** (gyroscope, temperature, brightness)
- Wireless transmission and reception
- Lighting of an LED strip

While the above are not a responsibility of the OBDH subsystem, they were directly interfaced with it in the name of simplicity.

5.1 Daughter board

The EM was connected (via male header) to a hand-soldered **perfboard** including all the components required to implement the above functions. The schematic of the former can be found at `https://gitlab.com/acubesat/obc/mockup-4/tree/master/Schematics/Satellite`.

The included components were:

- An additional MCP9808 temperature sensor
- A 2.4 GHz AT86RF233 transceiver
- A BH1750 brightness sensor
- An MPU9250 gyroscope-accelerometer-magnetometer set
- An WS2812B LED strip
- An AX5043 transceiver (*not used*)

Note that the above components were not soldered directly into the board. Instead, **sockets** were used to make the soldering process easier & not error-prone, and to enhance component reusability.

5.2 Power board

Due to the lack of an EPS board, we soldered a perfboard solution containing a battery, an Aliexpress DC-DC converter to provide the necessary 3.3V for the MCU, a voltage indicator, and an on/off pushbutton.

Previous experience has shown the need for a hard on-off switch with **clearly distinguished** states, to let operators know in an indisputable manner whether power has been turned on or off.

5.3 Software

The OBDH MCU was programmed in FreeRTOS using C++ and served to wirelessly transmit and receive some data, in order to demonstrate some of the basic functions of a satellite.

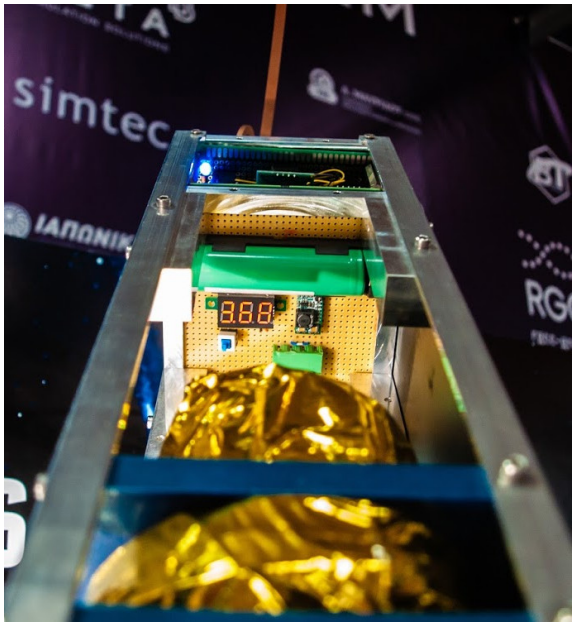


***Figure 7**: View of the power board*

The following FreeRTOS **tasks** were created:

1. **AT86RF233**: Wireless TXRX and ECSS Message parsing.
   Only the following services' messages were sent and parsed:
   (a) ST[08] function management sevice
   (b) ST[20] parameter management service
2. **Sensor tasks**: Reception of data from sensors through serial interfaces
   (a) BH1750 (Brightness)
   (b) MCP9808 (Temperature)
   (c) MPU9250 (Gyroscope)
3. **ECSS periodic**: Periodic ECSS service operations, to be called automatically at specific intervals
   Only the following service contained periodic functions:
   (a) ST[03] housekeeping service
4. **LED strip**: PWM output to control the lighting through the connected LED strip
5. **USART task**: Transmission of USART messages, being either ECSS packets or log messages. All data is handled through DMA and encoded using Consistent Overhead Byte Stuffing (COBS).
6. **Task list**: A periodic task that updates the task statuses and utilisation statistics every 500 ms

The following **libraries** were used for the above:

1. Madgwick AHRS for conversion of gyroscope data to Euler angles
2. In-house AT86RF233 drivers based on a public implementation
3. cobs-c COBS implementation
4. In-house ECSS Servies implementation
5. In-house MCP9808 implementation
6. In-house AX5043 implementation

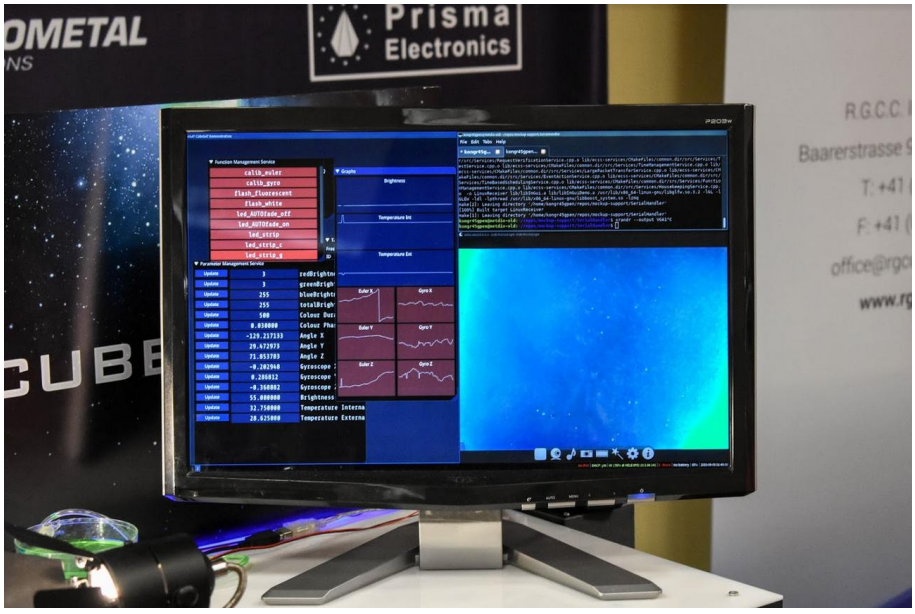


***Figure 8**: Display of the received data using the SerialHandler*

7. In-house BH1750 implementation
8. In-house MPU9250 implementation
9. STM32 HAL and LL libraries

5.4 Ground Segment

The "ground" receiver was based on the **NUCLEO-L432KC** and only included the AT86RF233 receiver and a couple of LEDs. The contraption was placed in a 3D-printed enclosure. The firmware of the receiving MCU can be found at https://gitlab.com/acubesat/obc/mockup-support/tree/master/GroundStation.

The display of data on a computer screen was performed using a C++ application based on ImGui. The application's source code can be found at https://gitlab.com/acubesat/obc/mockup-support/tree/master/SerialHandler.

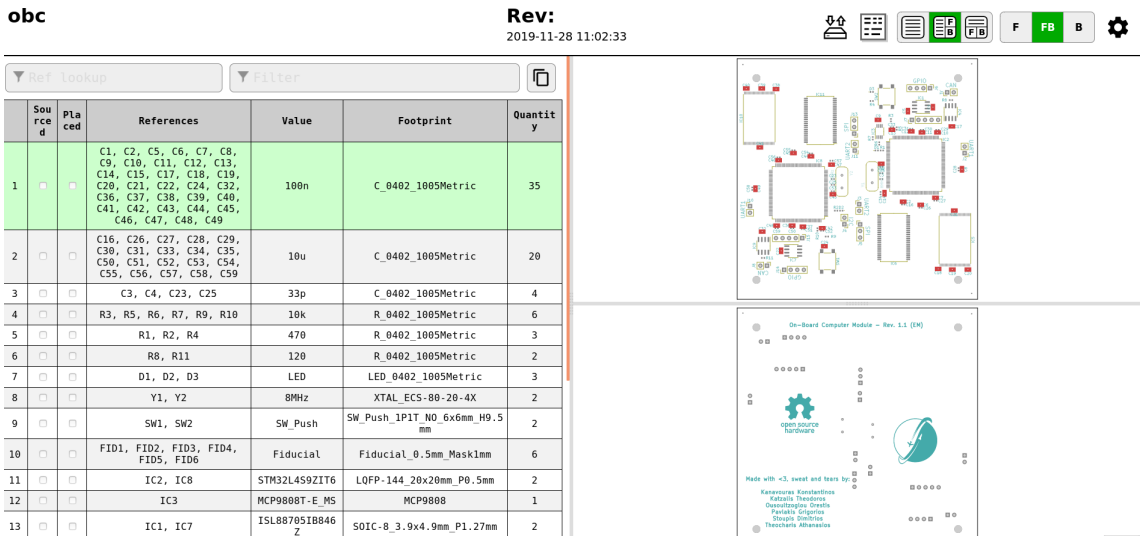
6 Lessons Learned

1. Most problems in PCBs originate from wrong **libraries** (schematic symbols and footprints) and the EM of the OBDH unfortunately wasn't an exception. Specifically, the schematic symbol of the watchdog (ISL88705) had been designed by our team and wrong pin numbers had been assigned to it that led to wrong PCB connections. The 6^{th} and the 7^{th} pin numbers had been associated as Active-Low Reset Output ($\overline{RST}$) and Watchdog Input (WDI) instead of WDI and $\overline{RST}$ respectively. To prevent this in future iterations, in case we won't find reliable databases for libraries and we need to design a schematic symbol or footprint by ourselves, then we should check thoroughly the datasheets of the components and a library review should be made at least by another person. The same applies to

the footprint creation and we could be compatible also with IPC standards. Additional testing will take place to verify the functionality of the selected watchdog.

2. **3D CAD models** are essential for ensuring the available area to place components and that everything fits in the PCB, so the assembler won't have any issues in the process. In our design we had some problems with the size of components due to the lack of these models. Precisely, we didn't take into account the size of the male pins that we had ordered, so they would come in touch with the nearby components. The problem solved using pins with the same pitch but with different size in the plastic body around them.

3. During the ordering of the components, a mistake was made in the quantity due to the error prone distinction of SMD form factors in metric and imperial code. Therefore the BOM wasn't applicable for our PCB design stopping the assembly process. The order included 100nF capacitors with form factor 1005 in imperial code instead of 1005 in metric code. To make this procedure easier and less error prone, a useful Kicad plugin will be used for the next iterations.



	Sourced	Placed	References	Value	Footprint	Quantity
1	☐	☐	C1, C2, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14, C15, C17, C18, C19, C20, C21, C22, C24, C32, C36, C37, C38, C39, C40, C41, C42, C43, C44, C45, C46, C47, C48, C49	100n	C_0402_1005Metric	35
2	☐	☐	C16, C26, C27, C28, C29, C30, C31, C33, C34, C35, C50, C51, C52, C53, C54, C55, C56, C57, C58, C59	10u	C_0402_1005Metric	20
3	☐	☐	C3, C4, C23, C25	33p	C_0402_1005Metric	4
4	☐	☐	R3, R5, R6, R7, R9, R10	10k	R_0402_1005Metric	6
5	☐	☐	R1, R2, R4	470	R_0402_1005Metric	3
6	☐	☐	R8, R11	120	R_0402_1005Metric	2
7	☐	☐	D1, D2, D3	LED	LED_0402_1005Metric	3
8	☐	☐	Y1, Y2	8MHz	XTAL_ECS-80-20-4X	2
9	☐	☐	SW1, SW2	SW_Push	SW_Push_1P1T_NO_6x6mm_H9.5mm	2
10	☐	☐	FID1, FID2, FID3, FID4, FID5, FID6	Fiducial	Fiducial_0.5mm_Mask1mm	6
11	☐	☐	IC2, IC8	STM32L4S9ZIT6	LQFP-144_20x20mm_P0.5mm	2
12	☐	☐	IC3	MCP9808T-E_MS	MCP9808	1
13	☐	☐	IC1, IC7	ISL88705IB846Z	SOIC-8_3.9x4.9mm_P1.27mm	2

***Figure 9**: Output of the interactive BOM*

4. In PCB assembly, **orientation** of components and **reference points** are necessary inputs for the automated machines and for the soldering in general. Thus, a gerber file that will indicate the orientation should be considered. About the reference points, fiducial marks should be located on the corners of the board. An auxiliary axis origin for plot file formats could be also helpful for the fabrication house.

5. When the team started to use the board, we discovered the need of **labeling** the function of each individual pin of the connectors. However, the **silkscreen** hadn't been configured with that in mind and only the reference designators had been printed, forcing the users of the board to check the designs all over again in identification-attempts.

6. **Mounting holes** should be configured thoroughly. It is a good practise to apply a large clearance for plated and particularly for non-plated mounting holes. This

helps to avoid shortening unwanted internal copper areas due to potential damage of the internal surface caused by the sharp screw thread. Plated through holes for screws can be used also for the unibody structure to function as the chassis ground, but attention should be given for the ground of the sensitive circuits. Additionally, plated holes can contain a grounded pad so that the interface to the screw/rail/spacer is conductive, and the board has access to the structure's ground.

7. The pins of integrated circuits that are very close to each other make difficult for the manufacturer the application of soldermask between them. Therefore, if **adjacent pins** has been connected as shown in Figure 10, then the trace is exposed along with the pads making the assembly and the optical inspection confusing and error prone. If adjacent pins should be connected then the solder mask clearance should be considered and the track should follow another route.

***Figure 10**: Improper example of shorted adjacent pins*

8. For the EMs, the ability to test multiple **features** of the design in conjunction with the need of **interfacing** with other boards, should be included to the guidelines. **More GPIO pins** that correspond to PWM or to other useful characteristics along with the accessibility of headers from different sides of the PCB (preferably near the outline) would be convenient.

9. In order to monitor the signal integrity of the PCB and to be able to understand and identify problems, **test points** are necessary tools. Circuit analysis isn't always capable to completely point out the potential problems that may occur in the actual implementation and the test points are practical ways to make this possible.

10. A pull up/down resistor should be connected to the **chip select** of the memory modules to ensure a default and deterministic mode in the hardware layer.

11. Where there are a number of closely-spaced vias or through-holes there is danger that their clearance holes through a plane will merge together, creating a huge gap. This gap creates a loop area increasing the inductance. So the clearance should be configured accordingly to prevent these gaps letting copper flow between them. Buried vias can also mitigate the gap phenomenon.

12. The connectors weren't properly located in terms of radiated emissions requirements.

13. Care for the location of connectors should be taken. Frequently used connectors (e.g. power, SWD, USART) *should* be placed **close to the edges** so that they are easily accessible when the model is inside the mockup.

***Figure 11**: Gap in the plane*

14. The analog power supply (VDDA/VSSA) was not connected. Even though this seemed not necessary, as the MCU did not interface with analog components, it turned out that it was impossible to use the internal temperature sensor and reference voltage without enabling the analog domain.
15. USART connectors can come along with a GND port, to ease interfacing with isolated voltage levels
16. Decoupling capacitors should be located in each power supply pair and not only if the datasheets prompts to place them.
17. In next iterations we should take into account aging and temperature coefficients in the design guidelines.
18. Care needs to be taken when handling **batteries inside conductive enclosures**. Spurious shorts may occur!
19. A **reset pin** could be added to the SWD interface, to allow for a hardware reset via the ST-Link
20. For frequently used Engineering Models (EMs), it is a good idea to use a polarity, overcurrent & overvoltage **protection circuit**. There exist chips such as the LTC4361 that can take care of all these. Additionally, PTC thermistors or resettable fuses could be used for short-circuit protection.

All in all, despite the above shortcomings, the EM was a joy to work with, and proved equally (if not more) useful as a **development board** customised to our team's needs. The manufactured PCB turned out to be a valuable addition to the team's display on TIF, as well as on future demonstrations.

References

[1] L. B. Gravelle and P. F. Wilson, "EMI/EMC in printed circuit boards-a literature review," in IEEE Transactions on Electromagnetic Compatibility, vol. 34, no. 2, pp. 109-116, May 1992.

[2] M.K. Armstrong, "PCB design techniques for lowest-cost EMC compliance. Part 1" in Electronics & Communication Engineering Journal, Volume 11, Issue 4, August 1999

[3] M.K. Armstrong, "PCB design techniques for lowest-cost EMC compliance. Part 2" in Electronics & Communication Engineering Journal, Volume 11, Issue 5, August 1999

[4] R. Goyal, "Managing signal integrity [PCB design]," in IEEE Spectrum, vol. 31, no. 3, pp. 54-58, March 1994.