



CZECH TECHNICAL
UNIVERSITY
IN PRAGUE

F3

**Faculty of Electrical Engineering
Department of Measurement**

Bachelor's Thesis

On-board computer for PC104 format CubeSats

Filip Geib
Cybernetics and Robotics

February 2021
<https://github.com/visionspacetec/VST104>
Supervisor: Ing. Vojtěch Petrucha, Ph.D.

Draft: 8. 4. 2021

Acknowledgement / Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended. In accordance with Article 46 (6) of the Act, I hereby grant a nonexclusive authorization (license) to utilize this thesis, including any and all computer programs incorporated therein or attached thereto and all corresponding documentation (hereinafter collectively referred to as the "Work"), to any and all persons that wish to utilize the Work. Such persons are entitled to use the Work for non-profit purposes only, in any way that does not detract from its value. This authorization is not limited in terms of time, location and quantity.

In Prague on February 25, 1999

.....

Abstrakt / Abstract

Klúčové slová: CubeSat; PC104; OBC; hardvér; PCB dizajn; schémy.

Preklad titulu: Palubný počítač pre CubeSaty formátu PC104

Keywords: CubeSat; PC104; OBC; hardware; PCB design; schematics.

/ Contents

1 Board Sierra - description	1
1.1 CubeSat concept	1
1.2 OBC requirements	1
1.2.1 Radiation and redundancy . .	1
1.2.2 Capabilities and features . .	1
1.2.3 Components certification . .	1
1.2.4 Components selection	2
1.3 PCI104 standard	2
1.3.1 Mechanical specification . . .	2
1.3.2 Power consumption	2
1.3.3 Main header pinout	2
1.4 PCB design and assembly	2
1.4.1 PCB specifications	2
1.4.2 Routing and fanout	2
2 Board Sierra - submodules	3
2.1 Microcontroller	3
2.1.1 Schematic design	3
2.1.2 PCB design	4
2.2 External clock sources	4
2.2.1 Schematic design	5
2.2.2 PCB design	5
2.3 Power management	5
2.3.1 Schematic design	7
2.3.2 PCB design	7
2.4 Peripheral isolators	7
2.4.1 Schematic design	9
2.4.2 PCB design	10
2.5 External memory	10
2.5.1 Schematic design	10
2.5.2 PCB design	10
2.6 CAN bus drivers	10
2.6.1 Schematic design	11
2.6.2 PCB design	12
2.7 Temperature monitoring	12
2.7.1 Schematic design	13
2.7.2 PCB design	14
2.8 Programming port	15
2.8.1 Schematic design	15
2.8.2 PCB design	15
References	18

Tables / Figures

2.1 Microcontroller characteristics ... 3	2.1 Microcontroller circuitry 4
2.2 Power management rating 7	2.2 Clock sources schematic 5
2.3 Temp. sensors description 14	2.3 Clock sources circuitry 6
	2.4 Power management diagram..... 6
	2.5 Power management schematic ... 8
	2.6 Power management circuitry 8
	2.7 Peripheral isolators diagram9
	2.8 Peripheral isolators schematic ... 9
	2.9 Peripheral isolators circuitry .. 10
	2.10 External memory schematic ... 11
	2.11 External memory circuitry..... 11
	2.12 CAN drivers schematic 12
	2.13 CAN drivers circuitry 13
	2.14 Temp. monitoring schematic .. 14
	2.15 Thermal bridge illustration 15
	2.17 Programming port schematic.. 15
	2.16 Temp. monitoring circuitry.... 16
	2.18 Programming port circuitry ... 16
	2.19 Programming port pinout 17

Chapter 1

Board Sierra - description

1.1 CubeSat concept

1.2 OBC requirements

1.2.1 Radiation and redundancy

Occasionally traveling through weaker parts of the Earth's magnetic field and not shielded by the Earth's atmosphere, the CubeSats have to operate in an environment full of radiation. A direct hit of a high-energy particle might have serious consequences for OBC functionality. These include transistor gate ruptures, memory bit flips, software upsets, or latch-ups. A proper strategy must be taken to increase the OBC durability and ability to handle such an error, resulting in maximizing a possible mission life.

One approach is to use special radiation-hardened components. This strategy is typical for professional and more expensive satellites than the CubeSats. These components are usually bigger, more costly, and have less functionality than ordinary COTS.

Considering the size and budget requirements of our OBC, we chose to implement another option. Instead of the previously mentioned physical hardening technique, a logical one was realized. The OBC hosts many schematic design-related features ensuring the proper handling of any radiation-related event. These include: i) over-current sensing power management, ii) separate peripheral isolators, iii) full high-impedance mode requested by the higher logic, iv) triple-redundant memories, v) multiple temperature sensors. A fully double-redundant OBC is presented in chapter ??.

1.2.2 Capabilities and features

[1–9]

Having in mind the expectations of the VST supervisors, features common for OBCs by different professional manufacturers, and design requirements implied by the radiation, we created and implemented the following list of desired OBC's features:

- **Microcontroller:**
- **External clock sources:**
- **Robust power management:**
- **Isolation of the peripherals:**
- **Redundant external memory:**
- **CAN bus peripherals:**
- **Temperature monitoring:**
- **Maximal payload sector:**

1.2.3 Components certification

[chap:componentsSelection]

■ **1.2.4 Components selection**

After listing all of the technical requirements for a specific electronic part, we had to chose a particular component. As there are usually multiple similar components from various manufacturers, we had to use additional criteriums for the selection process.

As this OBC module is not primarily designed for an actual space flight (explained in chapter ??), an individual component's price is not negligible. With a lower overall cost of the OBC, a broader project expansion in the LibreCube community can be achieved. Thus components with sufficient attributes but lower price were favored.

Another criterium in the component selection was the available PCB surface. As a result of maximizing the payload sector of the PC104 module, the actual OBC subsystem area was significantly decreased. Therefore the components of smaller dimensions available in fine-pitch packages (e.g. SSOP or QFN) were preferred. After researching the technical capabilities and related costs of PCB manufacturers, we decided not to use the BGA package components. Their significantly small footprints would require more precise fanout, resulting in increased manufacturing difficulty and price.

Different components are available from various distributors, which can prolong the assembling process. We chose to use Mouser Electronics for purchasing the components. Therefore the availability of a specific component in this store was also a selection factor. This decision was influenced by the VST supervisors and previous experiences.

■ **1.3 PCI104 standard**

■ **1.3.1 Mechanical specification**

■ **1.3.2 Power consumption**

■ **1.3.3 Main header pinout**

■ **1.4 PCB design and assembly**

■ **1.4.1 PCB specifications**

■ **1.4.2 Routing and fanout**

Chapter 2

Board Sierra - submodules

2.1 Microcontroller

The processing unit is the most important part of the OBC. Professional designs use a wide variety of different instruction set architectures [1, 3, 5–6, 9]. However, the ARM architecture seems to be more popular [2, 7–8, 4]. This architecture is known for its good multiprocessing support, low power consumption, affordable pricing, and broad spectrum of existing applications. Considering these benefits and influenced by the LibreCube and TUDSaT, our VST supervisors decided to pick an STM32 MCU.

Our task was to choose a particular model of this 32-bit, Arm and Cortex-M based MCU. Aiming rather for a low-power than high-performance characteristics, we decided to select an L series. Particularly the L4 series as it combines the largest flash memory size with the highest number of general-purpose input/output pins (GPIOs) [10]. Although, only one device from the L4 series is equipped with two CAN bus channels. As the presence of a second bus is crucial for our double-redundant approach, the STM32L496xx option was selected. This family comes in six different packages. Avoiding all of the BGA-like ones (as described in chapter 1.2.4) narrows the selection to Zx, Vx, and Rx variants. The Zx is the most capable one in terms of flash size and GPIO count. Therefore the STM32L496ZG (where G stands for extended operating temperature range) was our final choice [11]. From now on, we will refer to this particular device as the MCU. Some of its key characteristics are listed in table 2.1.

Max. frequency	80 [MHz]	SPI	3
Flash memory	1 [MB]	I ² C	4
Static RAM	320 [kB]	UART	5
Comparators	2	CAN	2
Op. amplifiers	2	GPIO	115
Temperature	–40 to 125 [°C]	DAC	2

[tab:microcontroller]

Table 2.1. Highlighted characteristics of the STM32L496ZG microcontroller [11].

2.1.1 Schematic design

The MCU pin assignment was continuously changing during the entire process of OBC schematic and PCB design. Its final state is presented in figure ???. We attempted to maximize the number of user-free GPIOs with added functionalities such as ADC or PWM while keeping the fanout manageable. A significant help during this process was the CubeMX tool of the STM32CubeIDE, visualizing all of the pinout combinations with a specific functionality. Each of the 3.3[V] tolerant pins was used only for the internal circuitry, resulting in a fully 5[V] tolerant main PC104 header connection.

A significant number of filtering and reservoir capacitors is needed to ensure the proper MCU functionality. The assignment of correct capacitors to required MCU pins

was pretty straightforward, following the device datasheet [11] and STM32L4 hardware development application note [12]. Furthermore, a $10[\mu\text{H}]$ choke and $120[\Omega]$ at $100[\text{MHz}]$ ferrite bead were placed in series with the analog power input. This LC filter supported by the bead should effectively eliminate both low and high-frequency interference.

The clock and data lines of both I²C busses are equipped with standard $4.7[\text{k}\Omega]$ resistors. Multiple $22[\Omega]$ resistors were placed in series with high-speed clock signals of the SPI interface. Two $0[\Omega]$ resistors are present on *FAULT* and *MODE* lines to facilitate an optional hardware isolation. A $10[\text{k}\Omega]$ resistor at *PH3* is suggested by [12].

2.1.2 PCB design

Location and fanout of the MCU are shown in figure 2.1. The MCU is placed on the PCB top side, covering most of the non-payload area. This big footprint of a LQFP144 package (approximately $2 \times 2[\text{cm}]$) is a consequence of avoiding the BGA packages while still trying to keep the pin count high. All of the capacitors were placed as close to their assigned pins as possible. In some cases, it was necessary to use the bottom side of the PCB. The same approach was applied to all of the resistors.

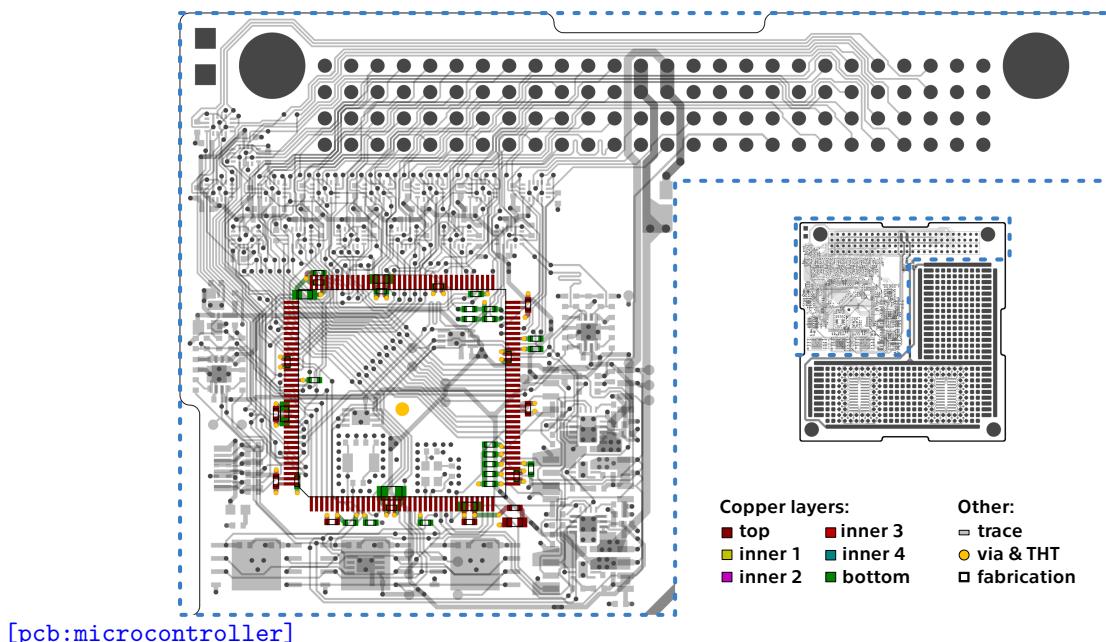


Figure 2.1. Highlighted location of microcontroller circuitry.

2.2 External clock sources

Proper timing and synchronization are the key features while dealing with high-speed data busses, ADCs, or other precise applications. The MCU is equipped with two internal RC oscillators that can be used to drive a master system clock and other auxiliary clocks [11]. These internal oscillators generally have a significantly lower frequency stability, a higher temperature dependency, and smaller overall accuracy than their external equivalents [13]. Therefore, to increase the clock precision and reliability in the harsh space environment, we had to implement external clock sources.

A $4 - 48[\text{MHz}]$ high speed external oscillator (HSE) can drive the system clock. Supported types are crystal, ceramic resonator, or silicone oscillator [11]. The last option seems to be the best as it is insensitive to electromagnetic interference (EMI) and vibration. The only downside is its slightly lower temperature rejection [14]. We chose

the SiT8924B, a 26[MHz] silicon microelectromechanical system (MEMS) oscillator [15]. Accordingly to the clock configuration tool of the stm32cube software, we can reach various system clock and auxiliary clocks frequencies up to 78[MHz] (maximum is 80[MHz] [11]). This is done using the phase-locked loop (PLL) clock generation.

A 32.768[kHz] low speed external oscillator (LSE) can drive the real time clock (RTC), hardware auto calibration, or other timing functions [11]. Table 7 in [16] recommends individual crystal resonators for combination with STM32 MCUs. After a consideration of these options, we decided to pick the ABS07AIG ceramic base crystal [17].

2.2.1 Schematic design

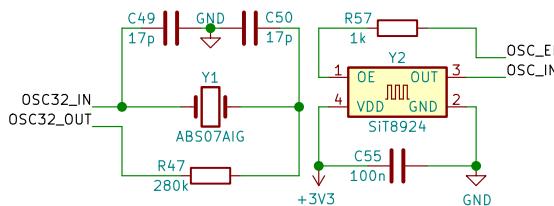
The HSE circuitry follows the oscillator datasheet [15] and is shown on the right side of figure 2.2. Only a decoupling capacitor and a terminator resistor are required. The clock output *OSE_IN* can be enabled or disabled by the binary *OSC_EN* signal.

The LSE circuitry is based on a reference design in the oscillator design guide [16]. To achieve a stable frequency of this Pierce oscillator, it is required to find the values of load capacitors C_{L1} , C_{L2} and external resistor R_E . This can be done using equations

$$C_L = \frac{C_{L1}C_{L2}}{C_{L1} + C_{L2}} + C_S \quad \wedge \quad C_{L1} = C_{L2}, \quad (1)$$

$$R_E = \frac{1}{2\pi f C_{L2}}, \quad (2)$$

where C_L is the crystal load capacitance, f is the crystal oscillation frequency and C_S is the stray capacitance [16]. Values of C_L and f are listed in the crystal datasheet [17]. We can assume as a rule of thumb, that $C_S = 4[\text{pF}]$. The final LSE circuitry with computed values of the components is shown on the left side of figure 2.2.



[sch:clockSources]

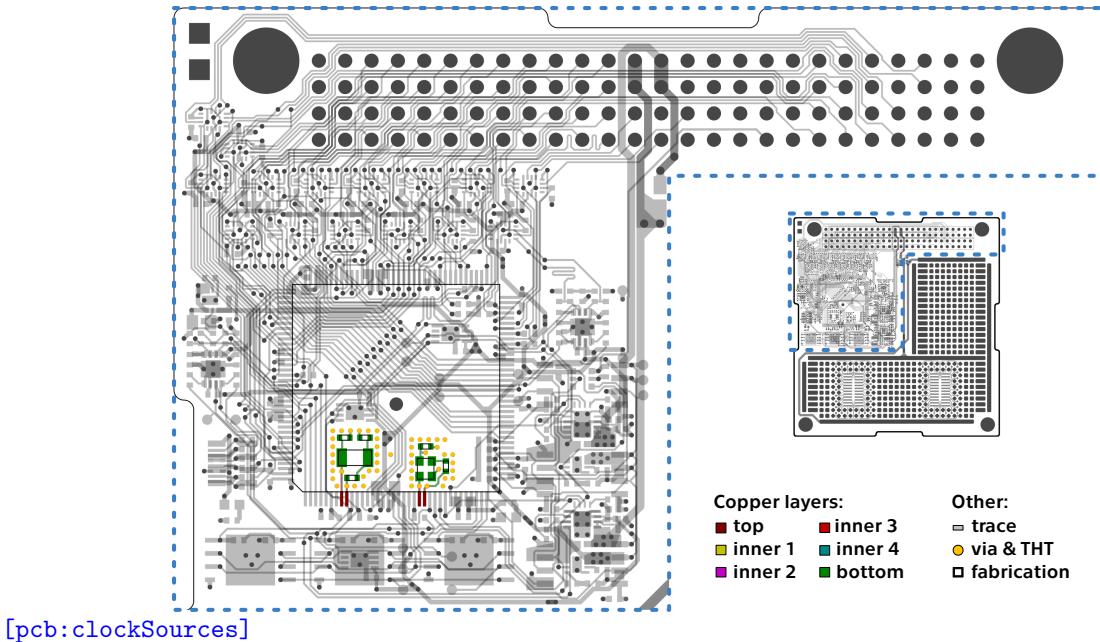
Figure 2.2. Schematic diagram of external clock sources.

2.2.2 PCB design

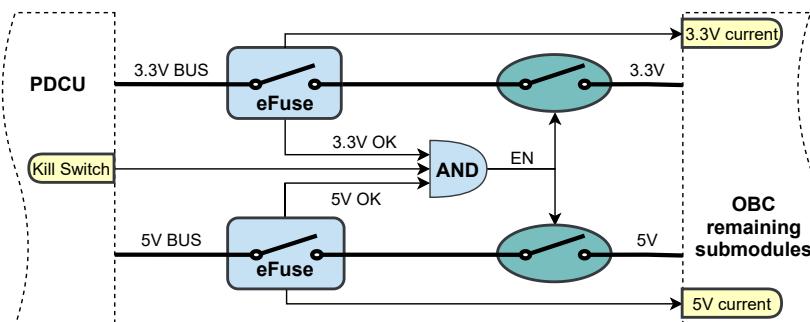
Location and fanout of the external clock circuitry are shown in figure 2.3. All of the components are placed on the bottom side of the PCB. The circuitry layout follows multiple tips presented in the oscillator design guide [16]. Separate GND planes are assigned to both the HSE and LSE circuitry. These planes are bounded by guard rings, formed by series of vias. Each of the planes is connected to a common GND only at one point. This approach provides proper EMI shielding while reducing a ground loop effect. We also minimized the distance between the MCU pins and both oscillators. All these measures combined should improve the clock generation stability and robustness.

2.3 Power management

The electric power subsystem (EPS) is known to be the most vital subsystem of a spacecraft. Its reliability and error handling should be ensured by the power control and

**Figure 2.3.** Highlighted location of external clock sources circuitry.

distribution unit (PDCU). However, it is a good practice by professional manufacturers to include an additional power monitoring and control to their OBC module designs [1–9]. We have to implement circuitry that can sense a power bus malfunction and, as a response, power down the OBC. This feature is also important for some of the OBC on-earth user cases. During a hardware development or a system presentation, the user might misconnect the power line or use an unsupported power source by a mistake.

**Figure 2.4.** Functional diagram of power management circuitry.

A functional diagram of the implemented power management is shown in figure 2.4. To increase the overall efficiency and decrease complexity, we decided to avoid any voltage conversion independent from the PDCU. Therefore, our OBC requires two separate inputs from the main power buss (3.3[V] and 5[V]). The OBC is connected to each of these inputs through an electronic fuse (eFuse). This device continuously monitors the bus for events of under-voltage, over-voltage, and over-current¹. As a response to such an event, the eFuse will switch into high impedance and pull down specific input of an AND logic gate. This gate simultaneously controls two load switches, one for each power line. This approach ensures that a fault on one power bus will result in a high impedance of both OBC power inputs. It also eliminates the risk of a death

¹ This is a crucial feature in handling and resolving a latch-up event.

loop, in which a reset of eFuses is not possible as they are switching each other off. Added benefits of this design are an inbuild current measurement and a Kill Switch integration into the logic gate. A summary of the final power management rating is listed in table 2.2. These values were chosen considering the power requirements of the remaining OBC components and are a subject of change by a future user.

Power input	Parameter	Min	Typ	Max	Unit
3V3 BUS	voltage	2.9	3.3	3.5	V
	current	0.0	-	1.2	A
5V BUS	voltage	4.6	5.0	5.4	V
	current	0.0	-	1.2	A

[tab:powerManagement]

Table 2.2. OBC power rating. Value out of range will cause a protective shutdown.

[secc:powerManagementSch]

2.3.1 Schematic design

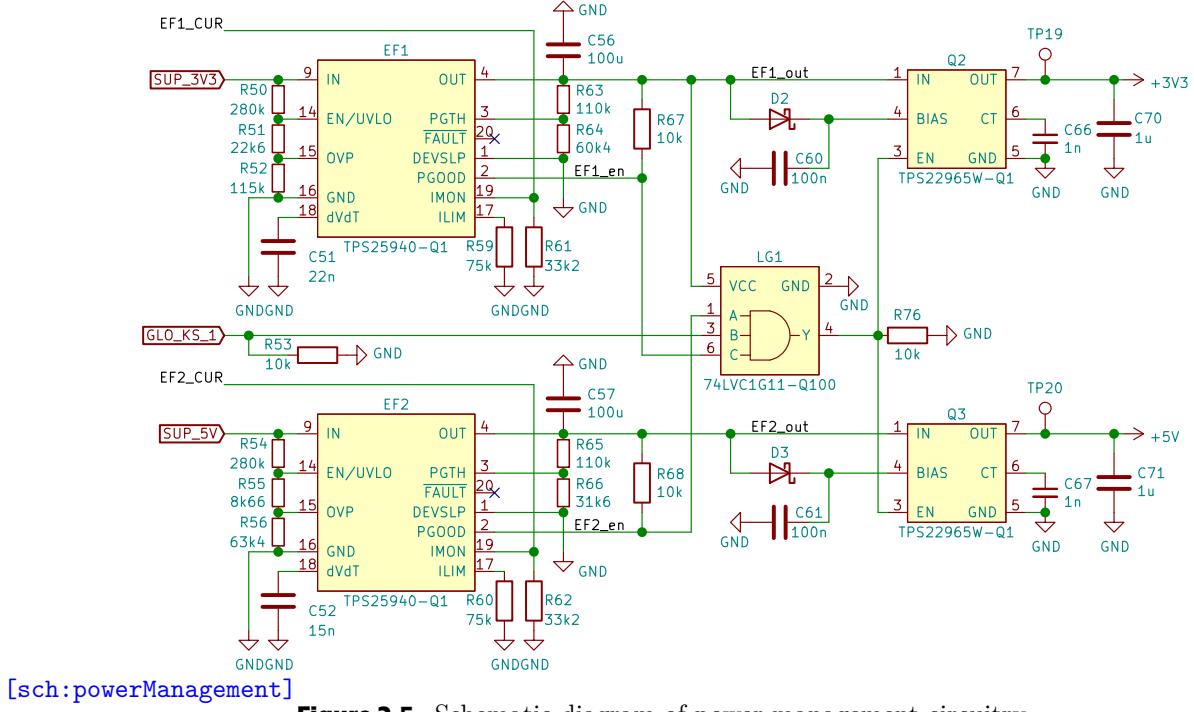
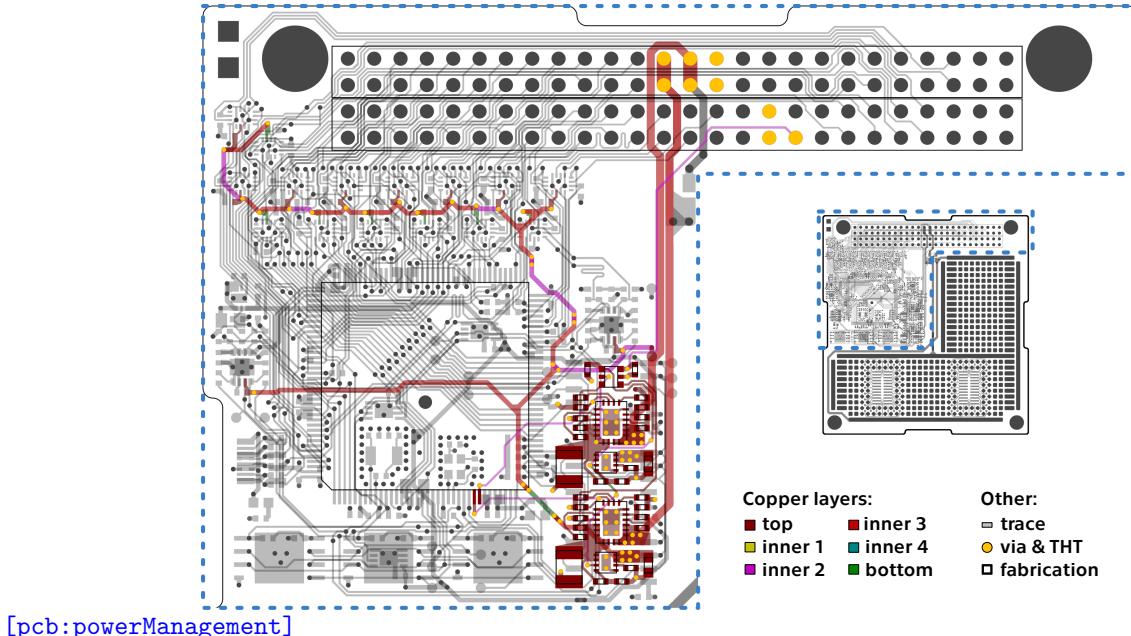
An actual schematic diagram of power management circuitry is shown in figure 2.5. The most important part of this design is the eFuse, as it covers all of the power control features. We decided to use the TPS25940-Q1 device [18]. Custom threshold values can be set following the typical application schematic in the datasheet [18]. This is done by connecting specific resistors, with values calculated using the TPS2594x design calculation tool [19]. As the logic AND gate, we chose the 74LVC1G11-Q100 [20]. This device is designed to operate in a mixed logic level environment, what corresponds with our application. The last important component is the load switch. In our case, the TPS22965W-Q1 with an inbuilt output discharge function [21]. For a correct operation of the switch, the *VBIAS* pin should stay saturated for a while after disconnecting the *VIN* voltage. We achieved this behavior by charging a capacitor connected to the *VBIAS* from the *VIN* through a Schottky diode. Four reservoir capacitors are placed on both sides of the load switches, following the suggestions in both datasheets [18, 21]. Nominal logic values of all switching signals are set by pull-up or pull-down resistors.

2.3.2 PCB design

Location and fanout of the power management circuitry are shown in figure 2.6. All of the power management components are placed on the top side of the PCB. Similarly, all of the power tracks are on the top copper layer. Only a few signal traces are running within the second or the bottom layer. Both the 3.3[V] and 5[V] control circuitry share the same layout and tracing regarding recommendations presented in the eFuse and load switch datasheets [18, 21]. The 3.3[V] part is situated closer to the main PC104 header, while the 5[V] part is right below it. Both eFuses are connected to the main PC104 header power pins through strengthened 0.77[mm] traces in the third copper layer. Considering a standard copper thickness of 35[μm], each of the input traces is rated to deliver up to 2[A] of current. The logic gate with associated pull-down resistors is located above all of the remaining circuitry, closest to the main PC104 header.

2.4 Peripheral isolators

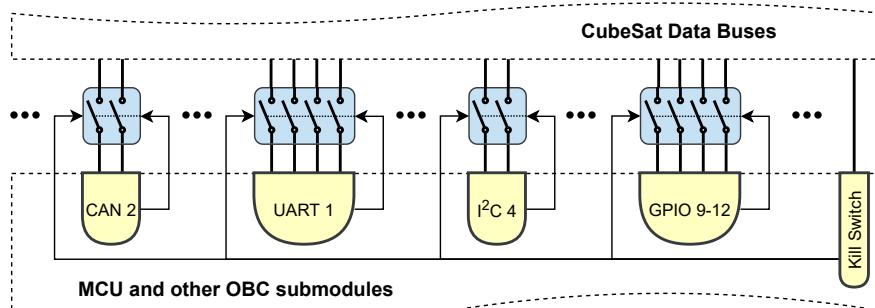
The spacecraft OBC is connected to many data buses shared among all other subsystems. In some scenarios, the OBC must be able to isolate itself from a specific or

**Figure 2.5.** Schematic diagram of power management circuitry.**Figure 2.6.** Highlighted location of power management circuitry.

multiple data buses. For example, to switch between OBCs in a redundant configuration, to handle a failure on a data bus, or to prevent unintentional interference. Standard approaches to address this feature are based on using analog switches [22], optocouplers [23], or FPGAs [1]. Furthermore, these isolators should also guarantee that all data lines are in a high impedance state when the OBC is powered off.

After a brief survey, we decided to implement the design using robust analog switches. This approach is more straightforward, less expensive, and requires a smaller PCB area than the optocoupler or the FPGA-based ones. A functional diagram of the implemented circuitry is shown in figure 2.7. All of the OBC data lines are connected

to the rest of the spacecraft through a series of analog switches. These data lines are grouped by particular data buses and are assigned to a separate switch. The OBC can enable or disable a specific switch and therefore isolate a particular data bus from the remaining spacecraft subsystems. Pulling low the Kill Switch will result in high impedance of all switches and completely isolating the OBC data lines.

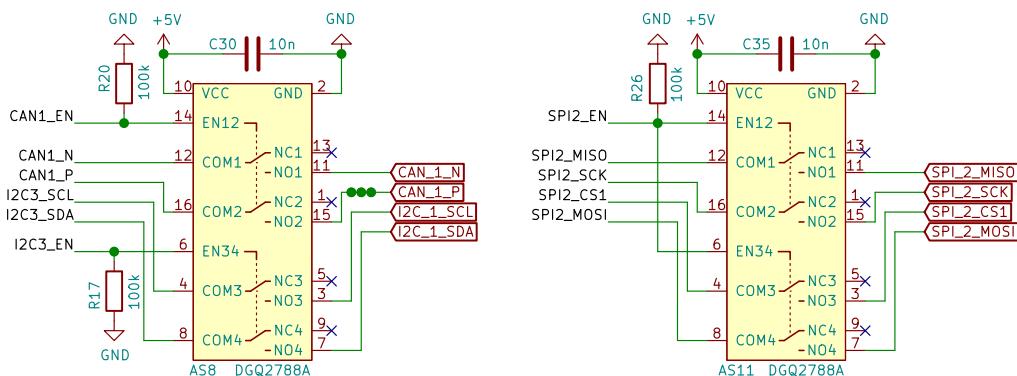


[dia:peripheralIsolators]

Figure 2.7. Functional diagram of peripheral isolators circuitry.

2.4.1 Schematic design

Schematic diagrams of two isolators are shown in figure ???. We decided to use the DGQ2788A device [24]. To cover all of the data buses, the OBC hosts fifteen of these analog switches in a dual double pole double throw (DPDT) configuration. The OBC data lines are connected to common terminals (*COM*). Normally closed terminals (*NC*) are left floating, whereas normally open terminals (*NO*) are connected to the spacecraft data lines. The important Kill Switch functionality is implemented using the device's power down protection. If the switch loses power, it will enter the normal state. This approach simplifies the circuitry a lot as it substitutes an otherwise necessary system of multiple logic gates. Other beneficial features of this analog switch are a high latch-up current of 300[mA] and inbuild signal clamping. The device will clamp all of the signals exceeding its supply voltage by internal diodes. As the MCU pins connected to these analog switches are 5[V] tolerant, we chose to power the switches from the 5[V] power bus. A potential problem could be caused by the switch's enable terminals (*EN*), as they do not include internal pull-down or pull-up resistors. We decided to use hardware pull-down resistors to avoid a floating state of these control signals. The placement decoupling capacitors follows the device datasheet [24].



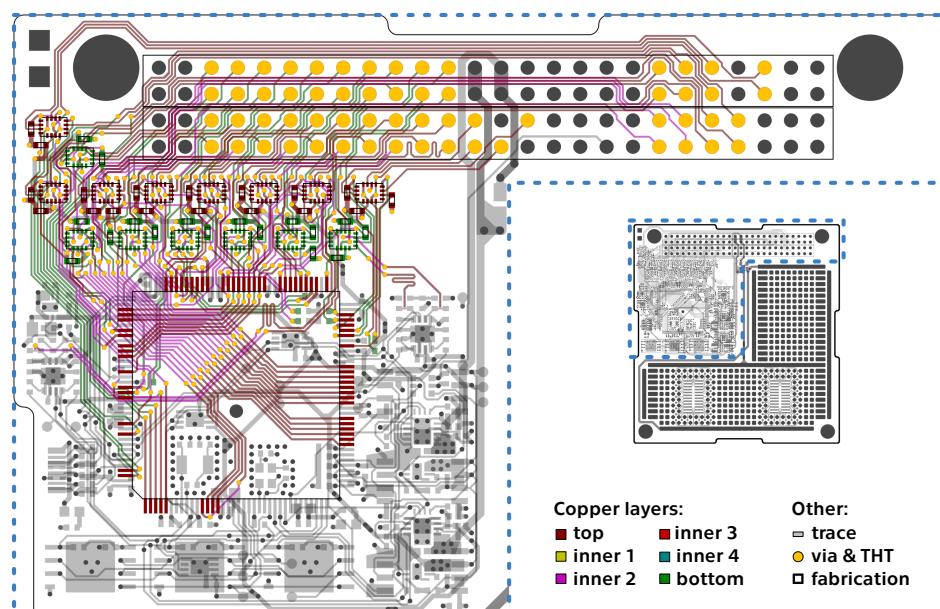
[sch:peripheralIsolatorsS]

Figure 2.8. Schematic diagram of two peripheral isolators.

[peripheralIsolatorsPCBDesign]

2.4.2 PCB design

Location and fanout of the isolators circuitry are shown in figure 2.9. Sixty separate signals are running from the MCU pins through analog switches up to assigned pins in the PC104 header. Hence this part of the PCB was the most challenging to design. The switches are placed in two main rows, each on one side of the PCB (only two switches are not aligned). The position of every switch was determined by its assigned MCU and PC104 header pins. It took several iterations to find out the current layout. The routing network is quite dense, using all three copper signal layers. To accommodate all of the signal traces, the standard signal trace width was decreased from 200[μm] to 173[μm]. This new value was acquired as a maximal width possible to squeeze three traces between two pins of the PC104 header. To ensure CAN buses signal integrity, we addressed a length matching of its differential pairs. As a finishing step of the routing, lengths of separate CAN traces were measured and tuned with serpentine patterns.



[pcb:peripheralIsolators]

Figure 2.9. Highlighted location of peripheral isolators circuitry.

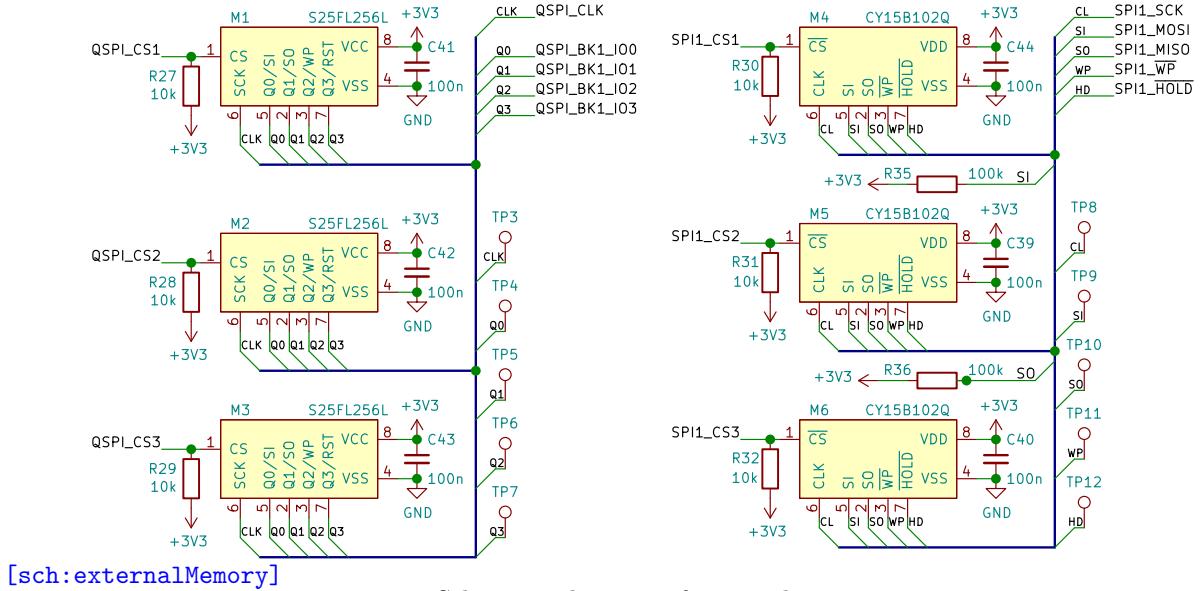
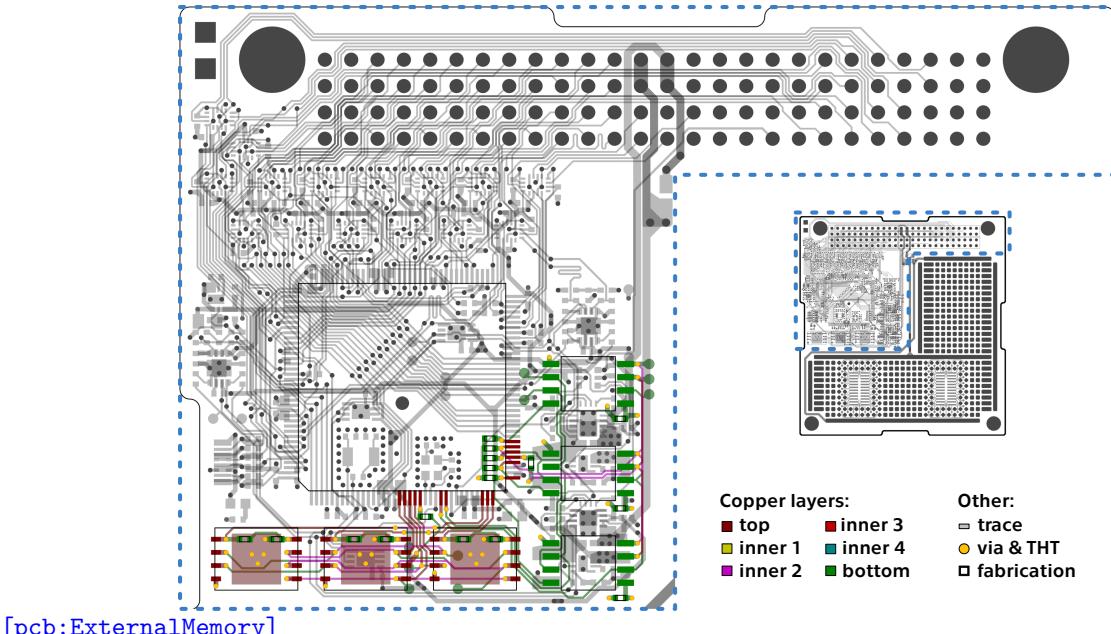
2.5 External memory

2.5.1 Schematic design

2.5.2 PCB design

2.6 CAN bus drivers

As described in chapter ???, the SpaceCAN is considered a primary control and monitoring bus of a LibreCube spacecraft. Therefore it was required to ensure its full support by the OBC. An external CAN transceiver is usually added to a microcontroller, as its internal physical layer has only limited properties or does not even exist. The separate transceiver provides a stable and reliable physical environment. In our case,

**Figure 2.10.** Schematic diagram of external memory circuitry.**Figure 2.11.** Highlighted location of external memory circuitry.

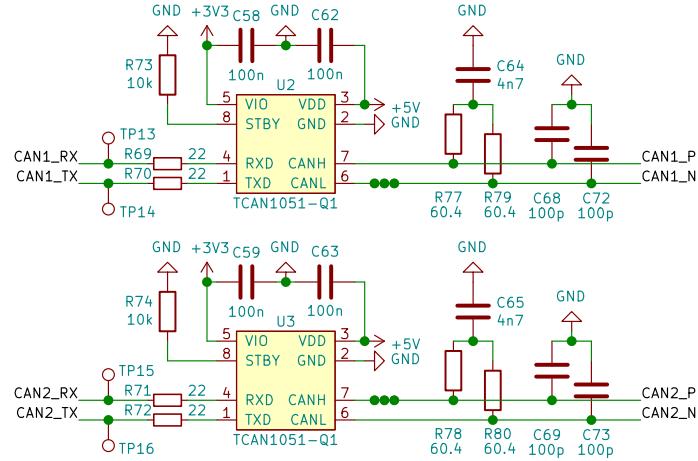
the MCU's BxCAN is compatible with both 2.0A and 2.0B CAN specifications with a bit rate up to $1[\text{MB s}^{-1}]$ [11]. As the MCU's CAN drivers are equipped with the 2nd network layer only, an external transceiver implementation to our design was required.

2.6.1 Schematic design

A schematic diagram of implemented CAN driving circuitry is shown in figure 2.12. As the MCU supports two independent CAN busses, we had to accommodate each of them. For the CAN transceiver, we decided to use a TCAN1051V-Q1 device [25]. The Rx/Tx lines of the MCU are connected to the device with a $22[\Omega]$ terminating resistors. A test point is also present on each of these lines to assist potential debugging. This version of the device comes with a level shifting feature. Different voltage levels on CAN and Rx/Tx sides are supported. Since the SpaceCAN is a $5[\text{V}]$ bus and the MCU

operates at 3.3[V], we set the device's power levels accordingly. As recommended in the device datasheet, decoupling capacitors were added to power pins. Considering the CAN bus's importance, we expect it to stay active straight from powering on the OBC. To save some complexity and MCU pins, we decided to ignore the option of controlling the device standby mode. A pull-down resistor on the *STBY* pin forces an active mode.

A well-designed CAN network usually contains a terminating resistor, filtering, and a transient & ESD protection. To correctly implement these optional features, we followed application reports [26–27]. Instead of a simple 120[Ω] terminating resistor, we chose a more advanced terminating node. A difference is in added filtering as the node consists of two 60[Ω] series resistors connected to a GND through a 4.7[nF] capacitor. Furthermore, 100[pF] filtering capacitors were added to signal lines. Recognizing the suggestions in the reports, we did not include any common-mode chokes or ESD protection in our design. Since our OBC is not intended to operate near heavy machinery, no extra improvement of susceptibility to electromagnetic disturbance or EMC is required.



[sch:canBusDrivers]

Figure 2.12. Schematic diagram of CAN bus driving circuitry.

2.6.2 PCB design

Location and fanout of the CAN bus driving circuitry are shown in figure 2.13. The transceivers are placed on the PCB top side in two different locations. Prioritizing the PCB surface's optimal usage, we were unable to keep the devices in a mutual area. The transceiver circuitry layout follows suggestions in the datasheet [25] and the application report [26]. This layout is the same for both devices. As mentioned in section 2.4.2, serpentine patterns were used to match trace lengths of particular differential pairs.

2.7 Temperature monitoring

Temperature is a critical parameter in the space environment, and all electronic components are sensitive to its variation. Any increase in temperature may reduce their lifespan and even result in irreversible damage. Such a temperature increase can be evoked by an ambient temperature or by the component's activity. Most electronic components are designed to dissipate heat into the ambient air. This is problematic in the space due to its lack of an atmosphere. Instead, a component transfers heat into the PCB's thermal capacity.² This conduction can produce temperature gradients

² And then dissipated into the spacecraft environment by the thermal radiation.

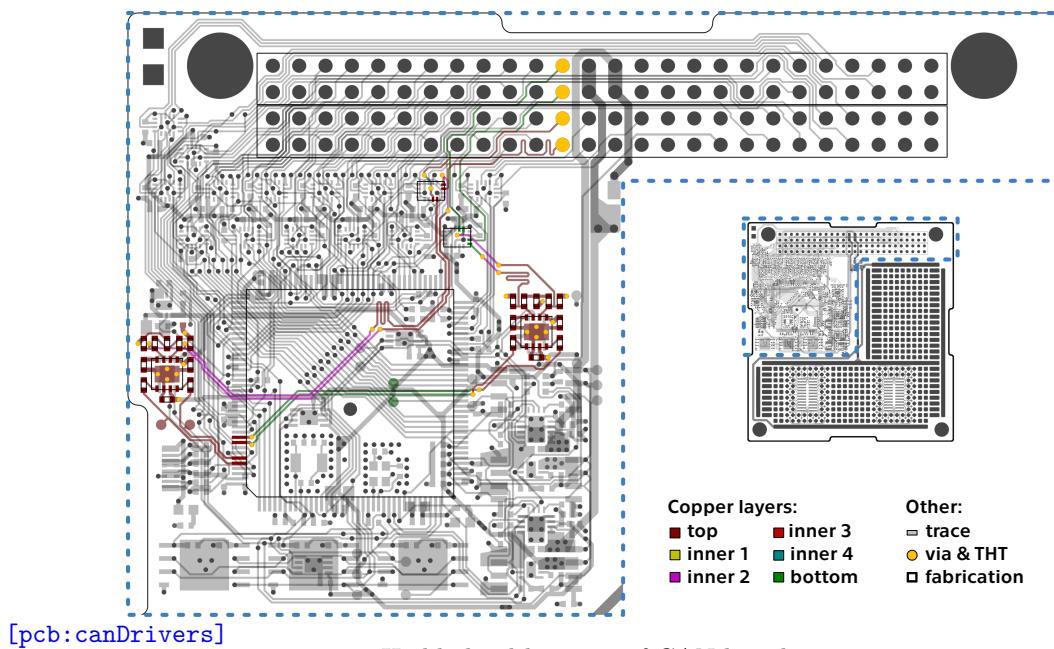


Figure 2.13. Highlighted location of CAN bus drivers circuitry.

throughout the PCB and influence other components. Furthermore, a change in a component's temperature can indicate its malfunction. Sudden temperature increase is a common sign of a latch-up event [28]. Accordingly, the OBC temperature is an essential part of a spacecraft telemetry and worthy of continuous monitoring. Professional OBC manufacturers have also included multiple temperature sensors in their designs [1–3, 6].

Standard temperature sensor technologies include integrated circuit (IC) sensors, thermistors, resistance temperature detectors (RTDs), and thermocouples. Their key features are compared in the guide to temperature sensing [29]. The IC sensors appeared to be the best choice for our design challenge. These sensors are typical for their good accuracy, small footprint, easy complexity, and excellent linearity. A processing unit can usually communicate with the IC sensors using one shared data bus and receive ready-to-use temperature data. In contrast, the implementation of the other sensor technologies requires extra analog components and circuitry. For example, amplifiers and ADCs for thermistors and thermocouples or precise current sources and ADCs for RTDs. These non-IC technologies would also use multiple MCU pins and require additional calibration and shielding. As our goal was to design a compact PCB and a simple system, we decided to implement the IC sensors approach.

2.7.1 Schematic design

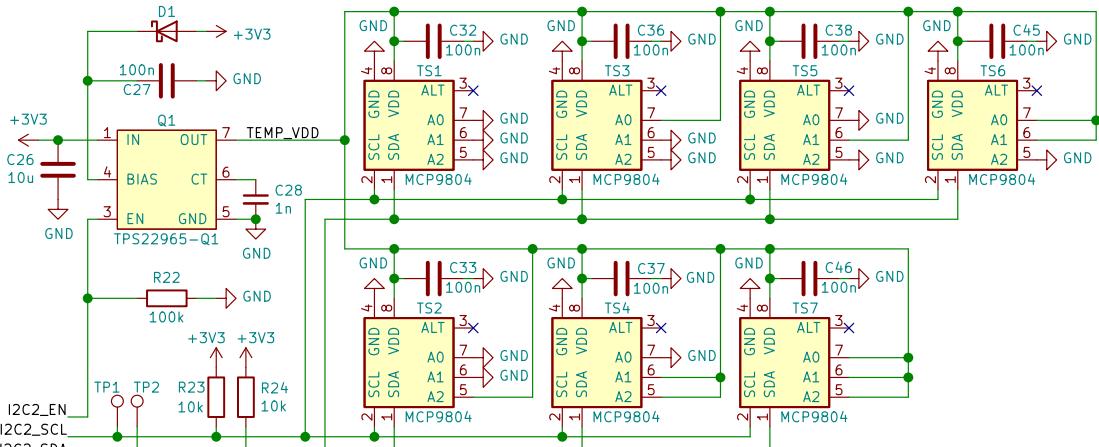
A schematic diagram of a temperature sensor network is shown in figure 2.14. After a brief survey, we selected a MCP9804 device [30]. This temperature sensor has an accuracy of $\pm 0.25[^\circ\text{C}]$ and communicates through an I²C interface. The device offers eight different I²C addresses, selected by different logic levels on three slave address pins (*A*₀-*A*₂). Thanks to this feature, we were able to use only one I²C bus for all of the devices. The final address assignment to particular sensors is listed in table 2.3. As the temperature monitoring is continuous, we decided not to use the device's inbuild alter function. Placement of a decoupling capacitor and 10[kΩ] pull-up resistors on I²C lines were suggested by the device datasheet. It is worth mentioning that this sensor supports a low-power standby mode, accessible through a special register.

Designator	Targeted component		Slave addr.	I ² C addr.
TS1	M2	- Flash memory	000	0x18
TS2	U1	- MCU, central west	100	0x1C
TS3	EF2	- 5[V] eFuse	001	0x19
TS4	U1	- MCU, north east	110	0x1E
TS5	U3	- CAN2 driver	010	0x1A
TS6	EF1	- 3.3[V] eFuse	011	0x1B
TS7	U2	- CAN1 driver	111	0x1F

[tab:temperatureMonitoring]

Table 2.3. List of temperature sensors location and addresses.

Accordingly to the survey on CubeSat electrical bus reliability [31], the I²C interface is the most likely to fail. Over half of investigated space crafts experienced at least one I²C lockup³. Hence it was essential to apply measures assuring the proper functionality of our I²C based temperature monitoring network. We had to implement a mechanism that can either prevent the lockup from occurring or is capable of resolving it. We chose the second option, as we consider it as a more robust and reliable. A simple but efficient approach to resolve an I²C lockup is to reset the power of all its slave devices. For this purpose, we implemented the TPS22965W-Q1 load switch in the very same configuration as we have used in the power management circuitry (subsection 2.3.1).



[sch:temperatureMonitoring]

Figure 2.14. Schematic diagram of temperature monitoring circuitry.

2.7.2 PCB design

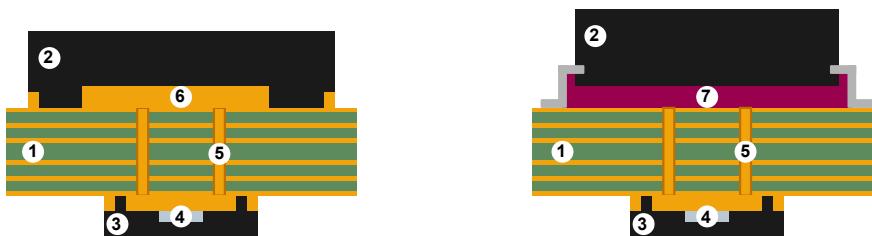
The layout of the temperature monitoring circuitry is dependent on the other subsystems. Therefore, it had to be implemented as the very last one. Each of the IC sensors was assigned to monitor a temperature of a particular device from another subsystem. These monitored devices were carefully chosen to cover all of the OBC's most critical components. These are various ICs used in the other subsystems as they execute multiple tasks, produce most of the heat, and are vulnerable to latch-up events. A listing of the monitored devices and their assigned temperature sensors is stated in table 2.3.

As we cannot exceed a total number of eight IC temperature sensors (MCP9804 devices) at one I²C bus, we decided to monitor only one of the three FLASH devices.

³ A continuous busy state of the I²C bus, where is the master prevented from starting a new transaction.

We selected the one in the middle as a good approximation of the two remaining FLASH devices. None of the three F-ram devices is monitored directly as the opposite side of the PCB in their location contains a dense layout of the power management circuitry. However, the F-ram devices are neighbored by three temperature sensors: TS5, TS6, and TS3. Measurements obtained from these sensors should be sufficient to monitor the F-ram devices. The MCU contains an inbuilt temperature sensor suitable only for applications that detect temperature changes only [11]. We decided to monitor the MCU with a pair of IC sensors as we prefer to measure the absolute temperature.

To ensure correct temperature measurement, we have to create a sufficient thermal bridge between the temperature sensor and its targeted device. A common approach is to place the sensor on the other side of the PCB, right opposite the device. A thermal bridge is then created using a set of PCB vias. This method is recommended and described in the temperature sensors guideline for SMDs [32]. Its illustration for our use case is shown in figure 2.15. It is worth mentioning that the added vias help dissipate the heat into the PCB and are usually required by the device's datasheet.



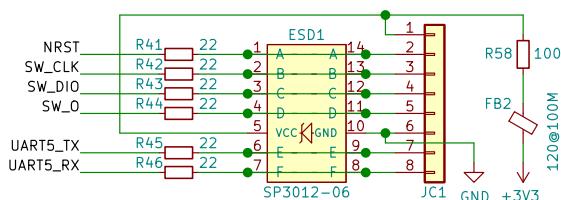
[temperatureMonitoring]

Figure 2.15. Illustration of a thermal bridge between the temperature sensor and WSON package (left) or LQFP package (right). Legend: (1) PCB, (2) targeted device, (3) IC temperature sensor, (4) measurement die, (5) PCB via, (6) thermal pad, (7) epoxy resin.

Location and fanout of the temperature monitoring circuitry are shown in figure 2.16. All IC sensors are placed at the bottom side of the PCB, directly under their targeted devices. The load switch is located at the PCB's west-central part. The switch itself and most of its auxiliary components are placed on the top side of the PCB. The I²C bus signal traces and separate power bus runs between all of the sensors.

2.8 Programming port

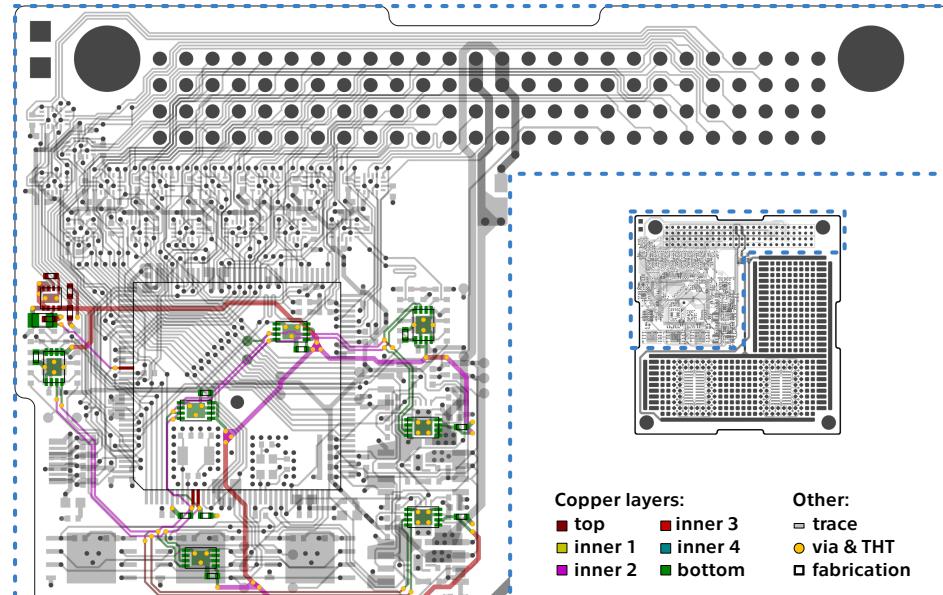
2.8.1 Schematic design



[sch:programPort]

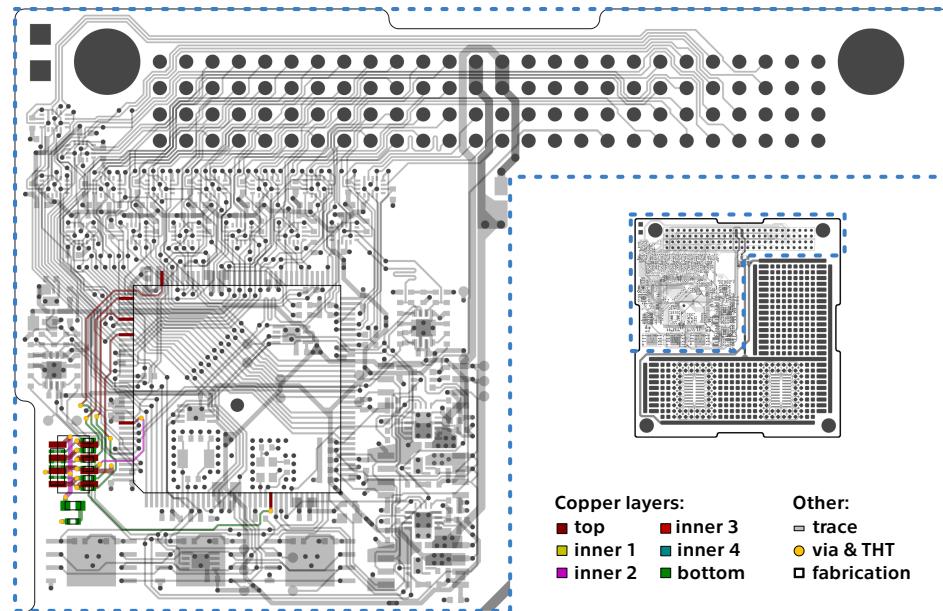
Figure 2.17. Schematic diagram of programming port circuitry.

2.8.2 PCB design



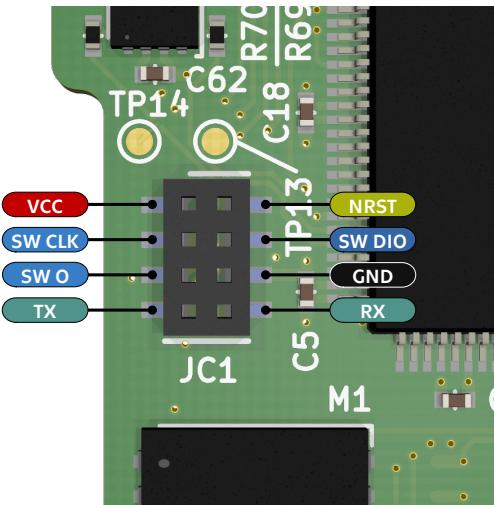
[pcb:temperatureMonitoring]

Figure 2.16. Highlighted location of temperature monitoring circuitry.



[pcb:programPort]

Figure 2.18. Highlighted location of programming port circuitry.



[mod:programPort]

Figure 2.19. Programming port pinout.

References

- [[\[obc:dataPatterns\]](#)] [1] *DP-OBC-0402-QAC-DS-1V00-349.* DP-OBC-0402 On Board Computer. Bengaluru: Data Patterns Pvt. Ltd..
- [[\[obc:isis\]](#)] [2] *iOBC.* ISIS On board computer. Delft: ISIS - Innovative Solutions In Space B.V..
- [[\[obc:imt\]](#)] [3] *MIT OBC.* Cubesat On-Board Computer. Rome: IMT s.r.l., c2016.
- [[\[obc:gos\]](#)] [4] *GOS CUBESAT OBC.* ON BOARD COMPUTER. Berlin: German Orbital Systems GmbH.
- [[\[obc:pumpkin\]](#)] [5] *CubeSat Kit™ Motherboard (MB).* Single Board Computer Motherboard for Harsh Environments. San Francisco: Pumpkin Inc., 2012. Rev E.
- [[\[obc:gauss\]](#)] [6] *ABACUS_201702.* GAUSS OBC ABACUS 2017. Rome: G.A.U.S.S. Srl, 2017.
- [[\[obc:aac\]](#)] [7] *KRYTEN-M3.* Command & Data Handling. Uppsala: AAC Clyde Space, 2020.
- [[\[obc:antelope\]](#)] [8] *ANTELOPE OBC.* On-board computer designed to keep your mission safe. Gliwice: KP Labs Sp. z o.o..
- [[\[obc:nanosatpro\]](#)] [9] *NANOSATPRO.* Space Qualified Processor Unit. Ankara: STM Savunma Teknolojileri Mühendislik ve Ticaret A.Ş..
- [[\[dat:mcuSeries\]](#)] [10] *STM32L series.* Ultra-low-power 32-bit MCUs Releasing your creativity. Geneva: STMicroelectronics, c2019.
- [[\[dat:mcu\]](#)] [11] *DS11585.* Ultra-low-power Arm® Cortex®-M4 32-bit MCU+FPU, 100 DMIPS, up to 1 MB Flash, 320 KB SRAM, USB OTG FS, audio, external SMPS. Geneva: STMicroelectronics, c2021. Rev 13.
- [[\[app:getStart\]](#)] [12] *AN4555.* Getting started with STM32L4 Series and STM32L4+ Series hardware development. Geneva: STMicroelectronics, c2019. Rev 8.
- [[\[app:oscInt\]](#)] [13] *AN5067.* How to optimize STM32 MCUs internal RC oscillator accuracy. Geneva: STMicroelectronics, c2018. Rev 1.
- [[\[app:oscComp\]](#)] [14] *APP2154.* Microcontroller Clock—Crystal, Resonator, RC Oscillator, or Silicon Oscillator? San Jose: Maxim Integrated, 2003.
- [[\[dat:hse\]](#)] [15] *SiT8924B.* Automotive AEC-Q100 Oscillator. Santa Clara: SiTime Corporation, 2019. Rev 1.7.
- [[\[app:oscDes\]](#)] [16] *AN2867.* Oscillator design guide for STM8AF/AL/S, STM32 MCUs and MPUs. Geneva: STMicroelectronics, c2020. Rev 13.
- [[\[dat:lse\]](#)] [17] *ABS07AIG.* Automotive & Industrial Grade 32.768kHz Ceramic Base SMD Crystal. Hidden Creek: Abracon LLC, 2020. Rev 12-11-20.
- [[\[dat:efuse\]](#)] [18] *TPS25940-Q1.* TPS25940xx-Q1 2.7-V to 18-V eFuse with Integrated Short-to-Battery Protection. Dallas: Texas Instruments Inc., 2021. Rev January 2021.
- [[\[app:tps2594Calc\]](#)] [19] *SLVC571.* TPS2594x Design Calculation Tool.
- [[\[dat:gate\]](#)] [20] *74LVC1G11-Q100.* Single 3-input AND gate. Nijmegen: Nexperia B.V., 2016. Rev 2.

- [dat:switch]
- [21] *TPS22965-Q1*. TPS22965x-Q1 5.5-V, 4-A, 16-m Ω On-Resistance Load Switch. Dallas: Texas Instruments Inc., 2019. Rev December 2019.
- [the:wurzburg]
- [22] BUSCH, Stephan. *Robust, Flexible and Efficient Design for Miniature Satellite Systems*. Würzburg: University of Würzburg, Faculty of Mathematics and Computer Science, 2016. Doctoral thesis.
- [pap:satelliteLeo]
- [23] FAJARDO, Isai, Aleksander A. LIDTKE, Sidi Ahmed BENDOUKHA et al. Design, Implementation, and Operation of a Small Satellite Mission to Explore the Space Weather Effects in LEO. 2019, Vol. 6, No. 10. ISSN 2226-4310.
- [dat:isolator]
- [24] *DGQ2788A*. Automotive 125 °C Analog Switch Dual DPDT / Quad SPDT, 0.37 , 338 MHz Bandwidth. Malvern: Vishay Intertechnology, INC., 2021. Rev 01-Jan-2021.
- [dat:canDriver]
- [25] *SLLSET0C*. TCAN1051-Q1 Automotive Fault Protected CAN Transceiver with CAN FD. Dallas: Texas Instruments Inc., 2017. Rev May 2017.
- [app:canDesign]
- [26] KISLING, Daniel, Abhi AAREY, and Bhavin KUMAR. *SLLA419*. How to Design Isolated CAN Systems With Correct Bus Protection.
- [app:canChokes]
- [27]
- [app:badBoys]
- [28] HASELOFF, Eilhard. *SLYA014A*. Latch-Up, ESD, and Other Phenomena.
- [boo:temperature]
- [29] *The Engineer's Guide to Temperature Sensing*.
- [dat:temperature]
- [30] *DS22203CMCP9804*. MCP9804: ±0.25°C Typical Accuracy Digital Temperature Sensor. Chandler: Microchip Technology Inc., 2012. Rev May 2017.
- [pap:reliability]
- [31] BOUWMEESTER, Jasper, Martin LANGER, and Eberhard GILL. Survey on the implementation and reliability of CubeSat electrical bus interfaces. *CEAS Space Journal*. Springer, Available from DOI 10.1007/s12567-016-0138-0.
- [app:temperature]
- [32] KASEMSADEH, Ben, Aaron HENG, and Amit ASHAR. *SNOA967A*. Temperature sensors: PCB guidelines for surface mount devices.

Requests for correction