



AGH University of Science and Technology

Faculty of Mechanical Engineering
and Robotics
Cracow, Poland



The European Organization
for Nuclear Research

Technology Department
Vacuum, Surfaces & Coating Group

Meyrin, Geneva, Switzerland

Master Thesis

Karol Adrianek

Name and surname of author

Automation Control and Robotics

Field of Study

**The design and the prototype of the Vacuum Gauges Controller
for high and ultra-high vacuum measurements for the Large Hadron
Collider at CERN**

Master Thesis Theme

PhD. Eng. Piotr Micek (AGH)

.....

MSc. BSc. Gregory Pigny (CERN)

.....

Master Thesis Promoters

Grades

ACKNOWLEDGEMENTS

I would like to thank my supervisors: *PhD. Eng. Piotr Micek* and *MSc. BSc. Gregory Pigny* for their guidance and support through the creation of this master thesis and project and the project itself.

Furthermore, I would like to thank *MSc. BSc. Tomasz Włostowski* for practical guides, hints and the possibility to use his laboratory. His experience and support were crucial during this project.

I would like to express my special appreciation to *MSc. BSc. Paweł Krakowski* for brilliant comments, suggestions, ideas, feedbacks, consultations and insights.

I am also grateful to all the people listed below for precious help and support.

Petter Nittermark Wien, Rafał Celejewski, Bartosz Golec, Jose Nuno Gracio Rosa, Miguel Valente Santos and Sandra Nizialek.

Table of Contents

1	Introduction.....	10
1.1	Motivation.....	10
1.2	About CERN.....	11
1.2.1	CERN	11
1.2.2	The Large Hadron Collider	11
1.2.3	Vacuum, Surfaces & Coating Group.....	13
1.3	Master thesis goals.....	13
1.4	Project guidelines.....	14
2	Vacuum at CERN	16
2.1	Vacuum control system at CERN.....	16
2.2	Vacuum instrumentation.....	17
2.3	Technical overview of current Vacuum Gauge Controllers used at CERN	20
2.3.1	TPG300 by Pfeiffer	20
2.3.2	VGC1000 by Volotek.....	21
3	New Vacuum Gauge Controller architecture.....	22
3.1	VGC system block diagram.....	22
3.2	Motherboard.....	25
3.3	Front panel	27
3.4	External cards	29
3.5	Power supply system	31
3.6	Controller configuration	32
3.6.1	Helmer gauge configuration	32
3.6.2	General configuration.....	33
3.7	Mechanical design	34
4	Hardware.....	36
4.1	STM32 microcontroller	36
4.2	TFT LCD display.....	40

4.3	External memories	42
4.3.1	FLASH	42
4.3.2	SDRAM.....	44
4.3.3	EEPROM.....	46
4.4	Real Time Clock	46
4.5	Power supply system	47
4.5.1	STM32.....	47
4.5.2	USB	48
4.5.3	Power system input.....	49
4.5.4	Main power converter.....	51
4.5.5	RTC battery and power Backup	52
4.5.6	TFT LCD Backlight	53
4.6	USB OTG	54
4.7	STM32 programmer	57
4.8	IO port buffers	58
4.9	External clock sources	60
4.10	User buttons	61
5	Firmware	63
5.1	Firmware requirements and guidelines	63
5.1.1	Software requirements:.....	63
5.1.2	Software guidelines:	64
5.2	Hardware drivers layer.....	65
5.2.1	General block diagram.	65
5.2.2	Clocks system configuration	66
5.2.3	DMA & DMA2D	69
5.2.4	STM32 Inputs-Outputs Ports Configurations.....	72
5.2.5	RTC - Real Time Clock.....	73
5.2.6	Interrupt configurations	75

5.2.7	FMC – Flexible Memories Controller	78
5.2.8	TFT LCD display	80
5.2.9	Communication interfaces.....	85
5.2.10	USB OTG controller.....	88
5.2.11	External bootloader	90
5.2.12	RNG, Watchdog and other drivers	91
6	Software & Firmware application layer.....	92
6.1	General main algorithm block diagram.	92
6.2	Communication protocol	94
6.3	Memory architecture	96
6.4	GUI – Graphical User Interface	98
6.5	GUI Templates.....	99
6.6	PC C++ console application	102
6.7	Software development tools.....	104
7	VGC controller project; summary and tests	105
7.1	Methodology of project development.....	105
7.2	Electronics design and the prototype	106
7.3	Tests system architecture	107
7.4	Hardware tests.....	108
7.5	Hardware drivers layer tests	110
7.6	Software test	114
7.6.1	External card protocol	114
7.6.2	PC C++ console application	116
7.6.3	GUI.....	117
7.7	Final tests	117
8	Summary	118
9	Attachments	129
9.1	Motherboard schematics	129

9.2	Backplane board schematics	142
9.3	PCB designs	146
9.4	Prototype photos	152

Tables and Figures

Table 1. The DMA1 requests mapping of STM32 [42].....	70
Table 2. The DMA2 requests mapping of STM32 [42].....	70
Table 3. The hardware tests [100].....	108
Table 4. The hardware drivers tests [100].....	110

Figure 1. The Large Hadron Collider (LHC) overview schematic [7].....	12
Figure 2. The architecture of the vacuum control system [19].	16
Figure 3. Vacuum measurement techniques classification [20].....	17
Figure 4. Measurement ranges of vacuum gauges [21].	18
Figure 5. The TPG300 controller [22].	20
Figure 6. The VGC1000 controller [25].....	21
Figure 7. The block diagram of the VGC system architecture [100].....	22
Figure 8. The motherboard block diagram [100].	25
Figure 9. The Front Panel of the Vacuum Controller [100].....	27
Figure 10. The block diagram of external card architecture [100].....	29
Figure 11. The power supply architecture [100].	31
Figure 12. External cards configuration for Helmer Gauge only [100].	32
Figure 13. General configuration for external cards [100].....	33
Figure 14. The 42HP 3U 19-inch Europe standard Plug-in-Unit [28].	34
Figure 15. The mechanical design (cross-section view) [29] [30] [31] [32] [100].....	35
Figure 16. STM32 microcontroller in the TQFP208 package [34].....	36
Figure 17. The STM32F429xx microcontroller architecture [42].	37
Figure 18. The STM32F429 Discovery evaluation board [44].	39
Figure 19. The 5-inch TFT LCD display [45].....	40
Figure 20. The circuit of the TFT LCD display [100].	41
Figure 21. The 64Mbit NOR Flash circuit [100].	42
Figure 22. The graphical comparison of NOR and NAND Flashes [49].	43
Figure 23. The 256Mbit SDRAM circuit [100].	44
Figure 24. The block diagram of the chosen SDRAM [51].	45
Figure 25. The 512kb Serial 24FC512 EEPROM circuit [100].	46
Figure 26. The power supply system of the STM32 [14].	47

Figure 27. The DC/DC boost converter circuit of the USB OTG [100].	48
Figure 28. Circuit of the TPS77601 LDO voltage regulator [100].	49
Figure 29. The power supply input circuit [100].	50
Figure 30. The main step-down DC/DC converter circuit [100].	51
Figure 31. The RTC power supply circuit [100].	52
Figure 32. The boost DC/DC converter circuit of the TFT LCD backlight [100].	53
Figure 33. The ESD and EMI circuit of the USB OTG [100].	54
Figure 34. Circuit description of the STF202-22 chip [67].	55
Figure 35. Circuit description of the NZF220TT1 [68].	55
Figure 36. The SWD programmer circuit of the STM32F429 Discovery board [44].	57
Figure 37. The SN74LVC2G17 (left) and SN74LVC2T45 (right) buffers [72] [73].	58
Figure 38. External crystal oscillator circuit [100].	60
Figure 39. The Circuit of the user button debouncer [100].	61
Figure 40. Switch bounce effect of buttons [81].	61
Figure 41. The hardware drivers' architecture [100].	65
Figure 42. The STM32F429xx clock system [83].	68
Figure 43. The block diagram of the STM32 DMA controllers [42].	69
Figure 44. The STM32 DMA2D block diagram [42].	71
Figure 45. The block diagram of the STM32 RTC [42].	74
Figure 46. The list of selected interrupts vectors of the STM32.	75
Figure 47. The FMC block diagram [42].	78
Figure 48. The FMC memory map [42].	79
Figure 49. The graphic hardware configuration driver with external SDRAM [46].	80
Figure 50. The LTDC controller block diagram [46].	81
Figure 51. The blending layer order [42].	81
Figure 52. The LTDC display data frame [46].	82
Figure 53. The LTDC timings and resolutions [46].	83
Figure 54. The colours formats versus pixel data mapping [46].	84
Figure 55. The pixel formal conversion from the RGB565 to the ARGB888 [46].	84
Figure 56. The SPI block diagram of the STM32F4xx microcontrollers [42].	85
Figure 57. The USART(RS232) block diagram [88].	86
Figure 58. The I ² C block diagram [42].	87
Figure 59. The USB application diagram, based on the ST libraries [89].	88
Figure 60. The bootloader-enabling procedure block diagram [90] [100].	90
Figure 61. The firmware flow chart [100].	93

Figure 62. The SPIs protocol flow chart [100].....	95
Figure 63. The memory map of the VGC Controller [100].	96
Figure 64. The GUI flow chart [100].	98
Figure 65. The GUI Templates – first part [100].	99
Figure 66. The GUI Templates – second part [100].	100
Figure 67. The GUI Templates – third part [100].	101
Figure 68. The C++ console application [100].	102
Figure 69. The console application flow chart [100].	103
Figure 70. The logos of used software development tools [85] [94] [95] [96].	104
Figure 71. The project managing block diagram [100].....	105
Figure 72. The test architecture block diagram [100].	107
Figure 73. The external card protocol test block diagram [100].....	114
Figure 74. SPIs' protocol testbench setup (during testing) [100].	115
Figure 75. The PC console application testbench setup [100].	116
Figure 76. The Vacuum Gauges Controller – final version [100].	118
Figure 77. The motherboard design in Altium Designer (top view) [100].	146
Figure 78. The motherboard design in Altium Designer (bottom view) [100].	147
Figure 79. The 3D motherboard design in Altium designer (top view) [100].	148
Figure 80. The 3D motherboard design in Altium designer (bottom view) [100].	149
Figure 81. The Backplane board design in Altium Designer (top view) [100].....	150
Figure 82. The 3D backplane board design view in Altium designer [100].	151
Figure 83. The motherboard PCB top view (after soldering) [100].....	152
Figure 84. The motherboard PCB bottom view (after soldering) [100].....	153
Figure 85. The Motherboard fixed to the mechanical chassis front panel [100].	154
Figure 86. The backplane PCB top view (after soldering) [100].....	155
Figure 87. The backplane PCB bottom view (after soldering) [100].....	155
Figure 88. The VGC motherboard fixed to the chassis front panel (perspective) [100].	156
Figure 89. The VGC motherboard fixed to the backplane board [100].	157
Figure 90. The VGC controller with mounted the backplane board (side view) [100]. .	158
Figure 91. The GUI screens (selected) [100].	159
Figure 92. The VGC controller prototype in the chassis (rear perspective) [100].....	160
Figure 93. The VGC controller in the 42HP 3U housing (perspective) [100].	160
Figure 94. The VGC controller in the mechanical chassis (side and rear view) [100]. ..	161
Figure 95. The final Vacuum Gauges Controller [100].	162

1 Introduction

1.1 Motivation

A high and ultra-high vacuum is mandatory part of performing physics experiments related to probing fundamental structure of the universe. Electron, neutron and proton particles require the highest possible the vacuum, in order to achieve a suitable environment for the particle collisions. An environment with a gas or air molecules, inside of the LHC (Large Hadron Collider) facilities, provides impossibility to achieve particle collisions with a stable concentrated beam (the concentration is used in order to increase the probability and total amount of collisions). The vacuum is a basis of physical experiments and the crucial parts of the CERN experimental installations [1] [2].

The previous versions of the Vacuum Gauges Controllers have been designed in the 80's and they are still used in the LHC installations at CERN (2018). The old controllers are currently obsoleted and they will not be produced any longer. Old vacuum controllers were designed to drive only 2-3 type of vacuum sensors, and their user interfaces are not optimised and user-friendly.

The technology development, increased requirements and new communication interfaces contributed to the design of a new controller that drives all the vacuum gauges used by CERN. Simultaneously, the new controller must be designed accordingly to industrial standards. For listed reasons, it was necessary to design the new universal controller for the LHC vacuum system, described in this master thesis.

1.2 About CERN

1.2.1 CERN

The European Organisation for Nuclear Research, called the CERN (“the name CERN is derived from the acronym for the French “Conseil Européen pour la Recherche Nucléaire” or European Council for Nuclear Research” [1]), is the largest scientific organisation for the particle physics in the world [1] [3]. The main mission of CERN physicist and engineers is “the probing the fundamental structure of the universe to find out what the elementary particles are and how they interact” [4].

The CERN experiments are performed with the use of the particle accelerators and detectors. The main goal of acceleration is that the “accelerators boost beams of particles to high energies before the beams are made to collide with each other or with stationary targets” [1]. In order to record the results of the particle collisions, the special detectors are used.

What it is worth to highlight, the CERN associates the engineers and physicists “from over 70 countries and with 105 different nationalities – half of the world’s particle physicists” [5]. Moreover, the CERN complex consumes 1.3 terawatt hours of electricity annually, which is sufficient to supply 300,000 standard homes for one year [6].

1.2.2 The Large Hadron Collider

The Large Hadron Collider (LHC) was started up in 2008 and currently, “the LHC is the world’s largest and most powerful particle accelerator” [3]. Furthermore, “the LHC consists of a 27-kilometre ring of superconducting magnets with a number of accelerating structures to boost the energy of the particles along the way” [3]. The accelerator chain overview of the LHC is shown in Figure 1 [1].

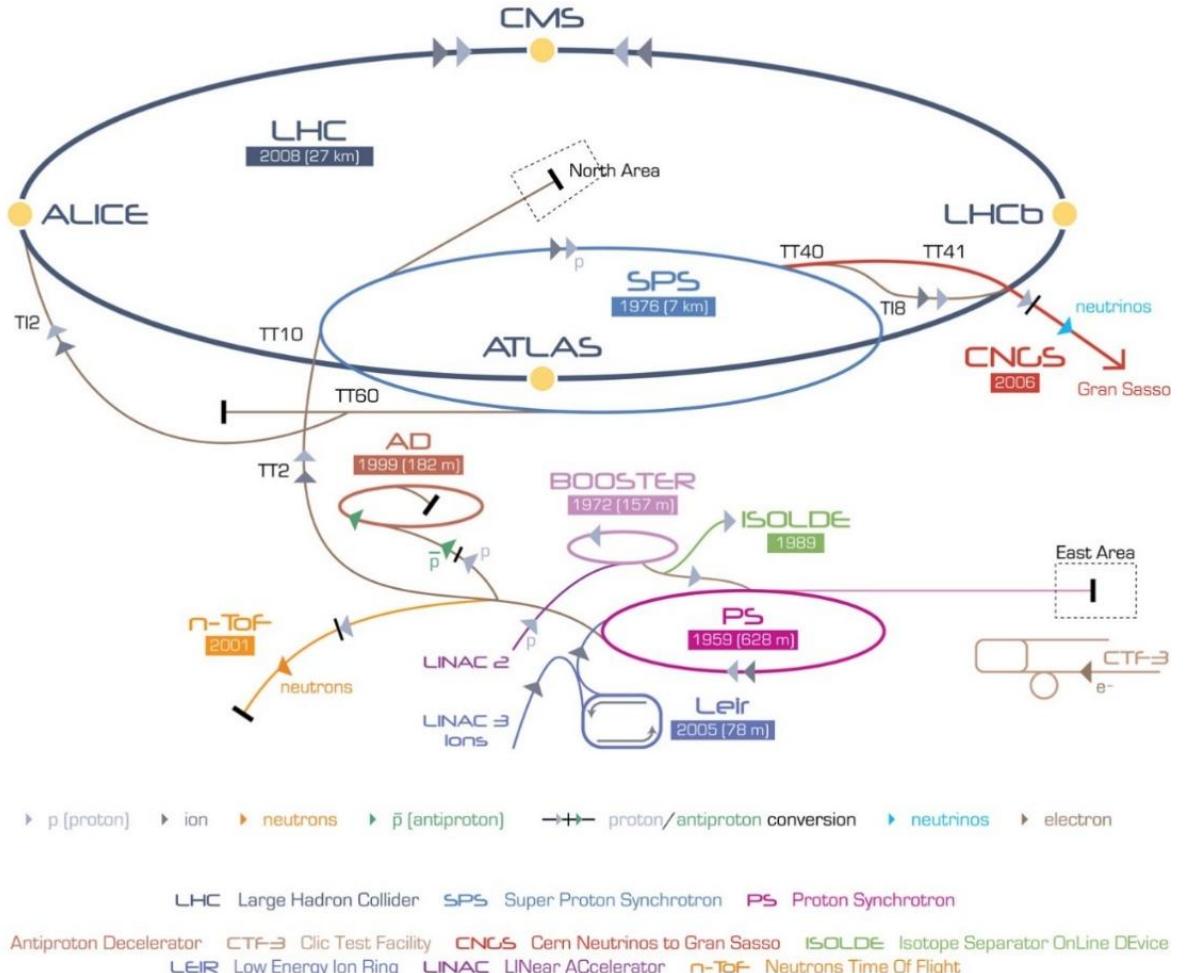


Figure 1. The Large Hadron Collider (LHC) overview schematic [7].

Inside the LHC, “two high-energy particle beams travel at close to the speed of light before they are made to collide. The beams travel in opposite directions in separate beam pipes – two tubes kept at ultrahigh vacuum” [3]. The trajectory of particle beams is maintained with the use of ultra-high power electromagnets, operated in the superconductivity state [3].

The LHC complex is equipped with approx. 104km of vacuum pipes (the largest vacuum system in the world) [8]. Moreover, the complex is “unusual in that it has three separate vacuum systems: one for the beam pipes, one for insulating the cryogenically cooled magnets and one for insulating the helium distribution line” [8]. The vacuum is crucial part used “to avoid colliding with gas molecules inside the accelerator, the beams of particles in the LHC must travel in a vacuum as empty as interstellar space” and to be thermal insulator for the magnet parts [8].

Moreover, this accelerator is “the largest cryogenic system in the world and one of the coldest places on Earth” [9]. The cryogenic system is used to enter a superconductive magnets to the superconductivity state, where they operate at the 1.9K (-271.3°C) temperature [9].

1.2.3 Vacuum, Surfaces & Coating Group

The Technology Department (TE) at CERN is responsible for performing engineering activities for particles accelerators and detectors of the LHC. The TE department is divided into seven groups including Vacuum, Surfaces & Coatings (VSC), which consists of 5 sections: the Interlock, Controls & Monitoring (ICM), Beam Vacuum Operation (BVO), Design, and Logistic & Methods (DLM), Surfaces, Chemistry & Coatings (SCC), Vacuum Studies and Measurements (VSM). Master project of the new Vacuum Gauge Controller has been done for the ICM, section of the VSC group [10] [11].

The ICM is in charge of the design, prototyping, implementation, maintenance and services of vacuum controls of all CERN accelerators. Vacuum interlocks (safety feature of vacuum systems) and monitoring tools are an integral part of the ICM [10].

1.3 Master thesis goals

The main goal of the following master thesis was to design and prototype the new Gauges Controller.

The main goals of the masters project can be divided into following parts:

- **Versatility** – the controller has to drive all vacuum sensors in use. It should allow easy integration and compatibility with other industrial and commercial devices and systems.
- **Modularity** – the controller has to be divided into several small blocks make the system easily to maintenance and to add new functionalities.
- **Upgradability** – hardware and software architecture should allow easy controllers functionalities upgrade for future needs.
- **Industrial standards** – the design must follow industrial standards of mechanical, electronics and software parts such as the ISO¹ [12] or IPC² [13] standards.
- **Easy maintenance** – remote failures handlers should be implemented.
- **Intuitive user panel** – user friendly and comprehensible for end-users with different backgrounds. Data display, options descriptions, layout of control buttons, windows and menu should be clear and understandable, without the need of reading technical documentation.

¹ ISO – International Organization for Standardization.

² IPC – Association Connecting Electronics Industries.

- **Universal communications interfaces** – controller should be able to operate with a majority of the most popular communication interfaces used in embedded or industrial systems.
- **External bootloaders³** – equipped with options to change remotely and effortlessly firmware of the controller. External bootloader should use one of the universal communications standards.
- **Reasonable components single-piece price** – the price cannot be too high and simultaneously, the chosen components properties must be sufficient for this project.
- **Solid mechanical construction** – design of the mechanical chassis have to be resistant as possible to mechanical damage and hazards of industrial environments.

1.4 Project guidelines

- **Main Microcontroller** – main microcontroller is an STM32⁴, based on ARM⁵ core [14].
- **User Display Panel** – is a 5-inch 800 x 480 pixels TFT⁶ LCD Display with 24-bit depth of colour [15].
- **User Interface** – based on the standard push-pull buttons. Eight main buttons were assumed, additionally one button is only dedicated for reset.
- **The USB (Universal Standard Bus)** – communication standard used in almost all electronic devices. The USB is dedicated to external bootloader and communication with a PC applications.
- **External System Cards** – various external, interchangeable measurement, communication and power supply cards are assumed.
- **Main Communication Interface** – is the SPI (Serial Peripheral Interface).
- **Industrial Housing** – controller is designed for 42HP⁷ 3U⁸ industrial Plug-in Unit (also called “Cassette”) compliant with 19-inch rack standard [16] (42HP is the width and 3U height).

³ Bootloader – for microcontrollers, part of a chip used to load program file into memory of the chip [82].

⁴ STM32 – name of high performance microcontrollers produced by the STMicroelectronics.

⁵ ARM – Advanced RISC (Reduced Instruction Set Computer) Machine.

⁶ TFT LCD – Thin-Film-Transistor Liquid-Crystal Display.

⁷ HP – Horizontal Pitch, 1HP = 5.08mm.

⁸ 3U – Rack Unit of height, 1U = 44.50mm.

- **Main Operating Controller Configurations** – two main configuration are set. The first of them is dedicated for only one kind of vacuum sensors, whereas the second is used to drive all others vacuum sensors.
- **Printed Circuit Board (PCB) guidelines** – the guidelines complies with the EUROCIR-CUITS® rules ([17]).
- **SMT⁹ technology** – if it is possible, all components should be of SMT package.
- **Dimensions of electronics components** – all components should be fitted for manual soldering. The RLC¹⁰ components are in 0603¹¹ SMD¹² package. The minimal assumed raster¹³ is not smaller than 1.27mm (50mils) for chips and 0.5mm for sockets.
- **Dimensions of PCB laminate** – 201mm width and 95mm height. Thickness of the laminate is 1.55mm (motherboard) and 3mm (backplane board). The Dimensions are set according to the dimensions of the 42HP 3U industrial housing for three versions of industrial Plug-in Unit used at CERN by the Vacuum Group (2018).
- **Industrial standard of the Power Supply** – +24VDC input power supply (industry standard voltage). Input voltage of the main power supply is 230VAC 50Hz (standard voltage network in the Europe).

⁹ SMT – Surface Mount Technology.

¹⁰ RLC – Resistors, Inductors, Capacitors.

¹¹ 0603 package – 60 mils (length) x 30 mils (width). One mils = 0.001 inch.

¹² SMD – Surface Mount Device.

¹³ Raster – it is a dimension between ports of chips, sockets or others electronics components.

2 Vacuum at CERN

Following chapter presents a short overview of the vacuum controls and sensors used by CERN facilities.

2.1 Vacuum control system at CERN

Figure 2 shows vacuum controls architecture at CERN. The system consists of three main levels: tunnels, underground service and controls & monitoring areas. The PLC (Programming Logic Controller) and SCADA (Supervisory Control and Data Acquisition) are of the backbone of the system. The PLCs communicate with underground devices via the PROFIBUS (Process Field Bus) interface. All new device have to be driven through this interface [18].

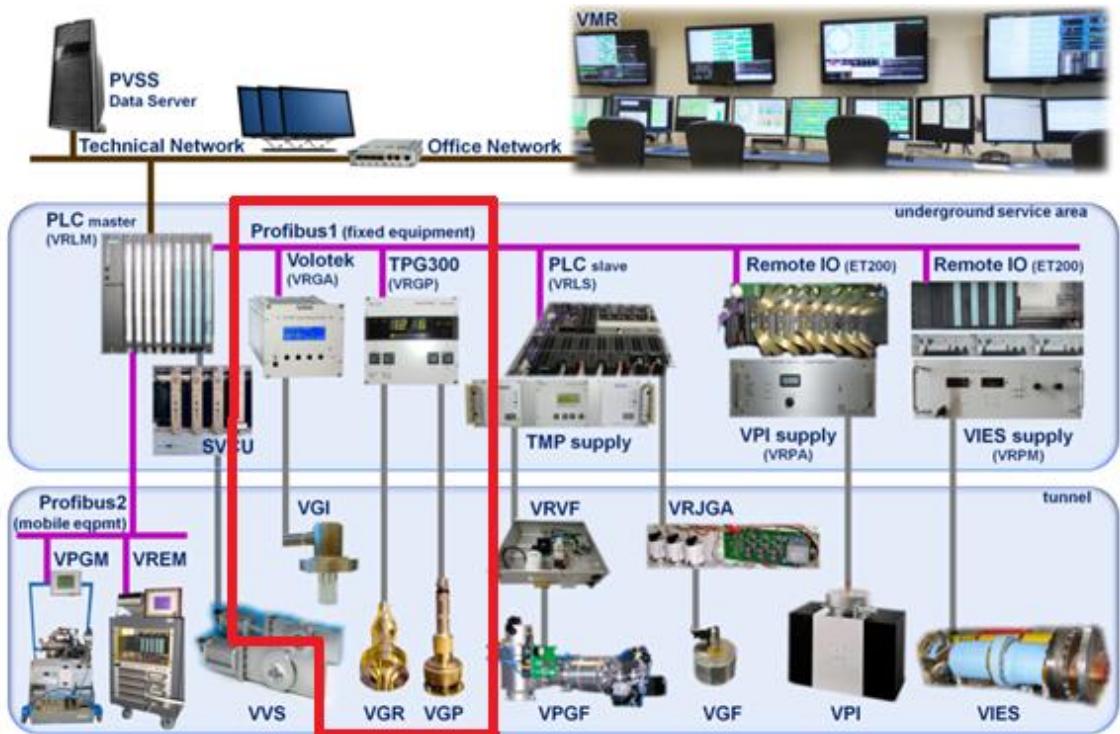


Figure 2. The architecture of the vacuum control system [19].

The **VGI** (hot cathode ionisation gauges), **VGR** (thermal conductivity gauges) and **VGP** (cold cathode ionisation gauges) gauges are driven by an old vacuum controllers (the TPG300 and the VGC1000). These controllers (red frame on the Figure 2) are connected with the PLC master, trough the PROFIBUS1 network [18] [19]. These controllers will be replaced with the new Vacuum Gauges Controller that will be responsible for driving the listed gauges.

The tunnel and underground service areas are separated by concrete walls to shield against radiation. All connections between them are realised by long cables [19] [18].

All vacuum instrumentation present in the underground service and tunnel areas are remotely driven from central controls server based on the SCADA with built-in WinCCoA data server. The access to all controls is possible from central control room only. The user friendly human-machine interface is performed by the SCADA, responsible for displaying all control processes [18].

2.2 Vacuum instrumentation

Vacuum instrumentation can be divided into two main groups: direct and indirect sensors. The main principle of direct sensors is that the measured pressure is proportional to mechanical deformation, caused by mechanical tensions gases or constant materials imposed on a measurable element. This effect can be seen in piezoelectric materials, where voltage or resistance gradient is related to applied external force. Direct sensors do not reach high accuracies of measurements and they are mainly used for lower vacuum or high and ultra-high pressure. [20] [21].

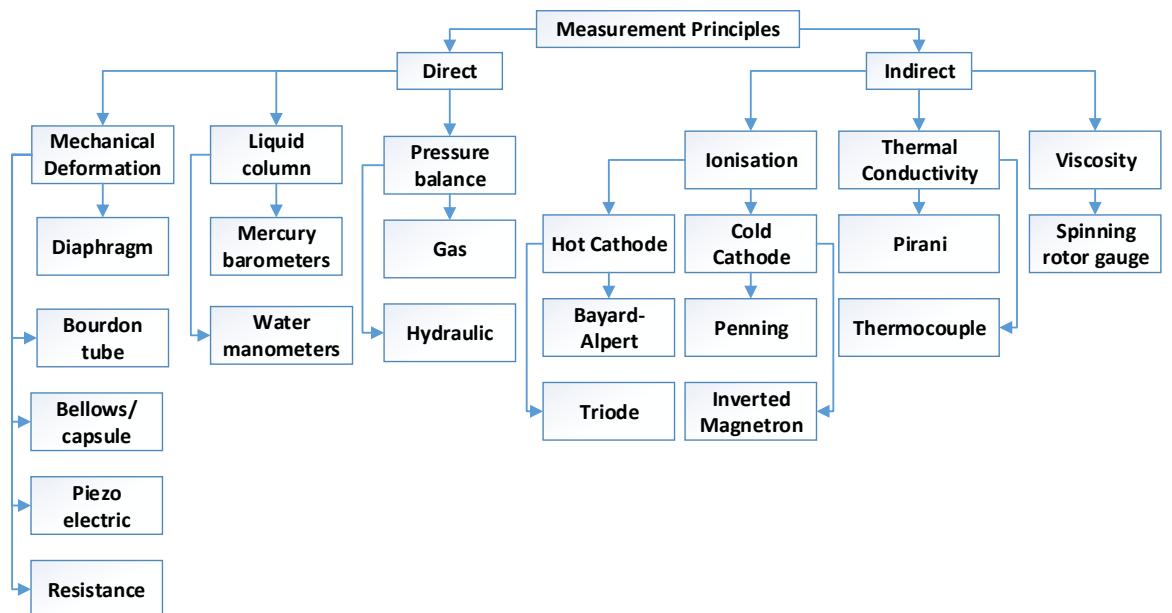


Figure 3. Vacuum measurement techniques classification [20].

The second group of vacuum sensors consist of indirect sensors pressure indication is based on indirect measurement methods through the measurement of the sensors parameters such as resistance, capacitance, magnetic field density, frequency, power, voltage, thermal power, temperature or current. Listed parameters indicate a pressure in indirect way and they

enable to achieve higher accuracies of measurements than direct sensors. Figure 3 presents three main subgroups of indirect sensors: ionisation, thermal conductivity and viscosity gauges [20] [21].

The thermal conductivity gauges (Pirani) principle relies on measurements of thermal power loss or temperature. The main operating principle is kept constant surrounding temperatures of measure element inside vacuum chamber. In higher vacuum, when number of molecules inside vacuum chamber decreases, the amount of power dissipation used to keep constant temperature is also decreased. The other type of thermal conductive gauges are thermocouples sensors, which main operating principle based on measurements of differing voltage between sensor electrodes, dependents on temperature (being depended on pressure) [20] [21].

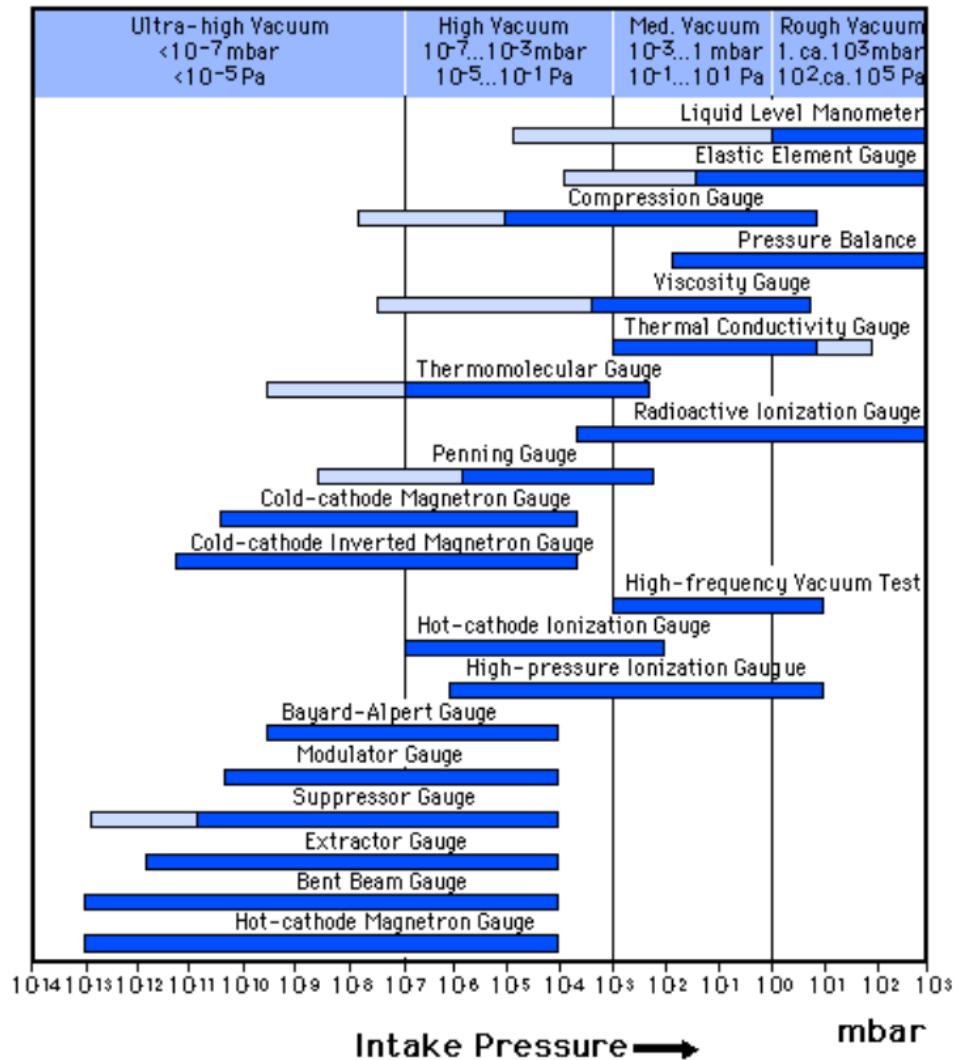


Figure 4. Measurement ranges of vacuum gauges [21].

The measurement ranges of different vacuum sensors are presented on the Figure 4. All direct type vacuum gauges have smaller measurement range, and they cannot be used in high and ultra-high vacuum. The CERN's facilities operate with the presence of UHV¹⁴ and the indirect vacuum gauges have to be used to measure pressure below 10^{-5} mbar.

The new Vacuum Gauges Controller is dedicated to drive the following instruments:

- **Helmer** – upgraded hot ionisation gauge, with a measurement range from 10^{-4} up to even 10^{-13} mbar (the most accurate sensor in use) [20] [21].
- **Bayard-Alpert** – high precision hot cathode ionisation gauge with the measurement range from 10^{-4} up to even 10^{-12} mbar [20] [21].
- **Penning** – cold cathode ionisation gauge with a measurement range from 10^{-2} up to 10^{-11} mbar [20] [21].
- **Pirani** – thermal conductivity gauge with measure range from 10^{-2} up to 10^{-4} mbar (newer versions of Pirani). Usually coupled with Penning gauges [20] [21].
- **Piezo-Resistive** – vacuum sensor with a measurement range only up to 10^{-3} mbar [20] [21].

¹⁴ Ultra High Vacuum

2.3 Technical overview of current Vacuum Gauge Controllers used at CERN

Following chapter briefly describes the VGC1000 (by Volotek) and the TPG300 (by Pfeiffer) Vacuum Gauge Controllers currently widely used at CERN's installations.

2.3.1 TPG300 by Pfeiffer

The TPG300 is only dedicated for Pirani's and Penning's sensors, and it cannot drive other type of vacuum sensors. The front panel of the TPG300 is presented on Figure 5, where four main user buttons and LED displays are visible. The TPG300 was designed accordingly to the 28HP 3U ($\frac{1}{3}$ 19-inch) industrial chassis and it is made in compliance with 19-inch EURO rack standard. What is more, the controller can be equipped with four interchangeable external cards. Two of them are always measurement cards (maximum four sensors at a time: two Piranis and two Pennings), main power supply and communication cards. The main communication interface card is dedicated to the PROFIBUS [22].

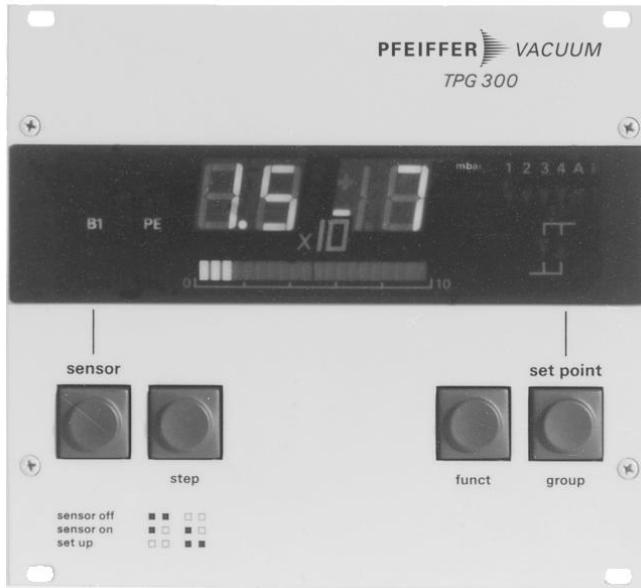


Figure 5. The TPG300 controller [22].

The main power supply provides several different levels of voltages in parallel. All references and power voltage lines are carried through an auxiliary backplane board to the measurement sensors. This configuration of power system can be problematic and result in a decrease in the precision of the measurements (this architecture has not been implemented in the project) [23].

2.3.2 VGC1000 by Volotek

This controller is only dedicated for driving the Bayard-Alpert gauge. In contrary to the TPG300, the VGC1000 can only serve one vacuum sensor at time [24].



Figure 6. The VGC1000 controller [25].

The Volotek controller is equipped with four interchangeable external cards, similar to the TPG300. Those cards are respectively: main and grid power supplies, measurement and PROFIBUS communication cards. The controller power supply is only dedicated to Bayard-Alpert grid, because of its high overhead for electrical power and non-standard voltage levels. The user front panel (Figure 6) of the controller consists of four main user push-pull buttons, reset button, serial interface programming and LCD display. It must be mentioned that this controller is out-of-date (as for 2018). The VGC1000 is equipped with alpha-graphical LCD Display with the internal HD44780 standard driver of this LCD displays. Similarly, to the TPG300 it is housed in 28HP 3U standard industrial chassis [24].

3 New Vacuum Gauge Controller architecture

Following chapter presents the hardware architecture of the New Vacuum Gauges Controller (VGC) for the $\frac{1}{2}$ 19-inch standard industrial chassis. This chapter contains brief architecture descriptions of the controller system, motherboard, external cards, front panel and user interface. Moreover, this chapter describes the system configurations for vacuum gauges with internal arrangements of the external cards.

3.1 VGC system block diagram

Figure 7 presents the block diagram of the VGC hardware architecture. This diagram consists of the Motherboard, GUI (Graphical User Interface), Main Power Supply, auxiliary Backplane board, and different communication, measurement and adjustable power supply cards. The controller has built-in seven external sockets. One of them is only dedicated to the main power supply, the second serves the communication with a PC via USB (Universal Serial Bus) and the others are general purpose ones (driving the communication, measurements and power supplies cards).

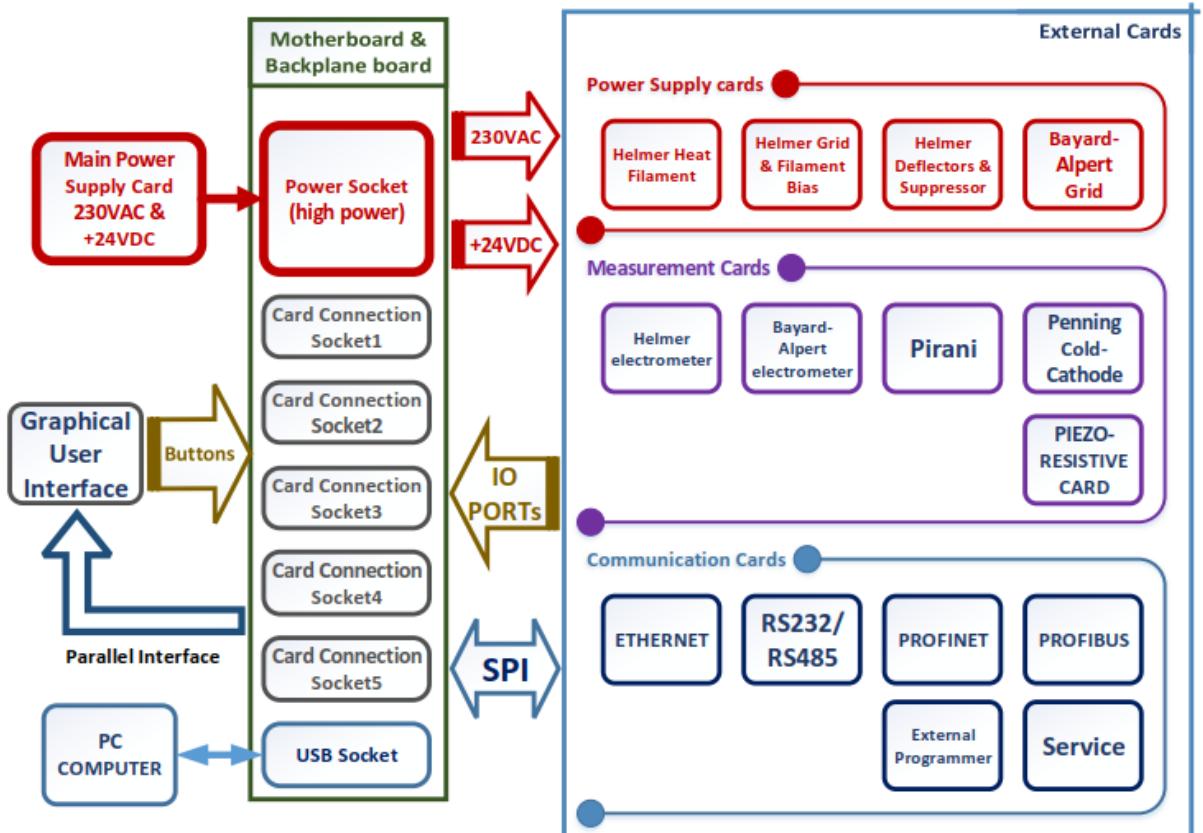


Figure 7. The block diagram of the VGC system architecture [100].

The auxiliary backplane board is placed between the motherboard and external cards. The Motherboard operates as the master to all measurement and power supply cards. The communication cards can operate as a master or a slave (it is possible to switch the operating mode between devices) in the point of view of the motherboard. In default mode, the motherboard operates as the slave for a communication card and a PC.

External cards are organised in three main groups: power supplies, measurements and communication. The project assumes that the maximum of sixteen external cards (measurements, communication, power supplies) will be available to use, but only six of them (taking into account the main power supply that always must be connected) can be connected to the controller at time.

Power Supply cards:

- **Main Power Supply** – providing +24VDC and 230VAC to the controller, with maximum power consumption up to couple hundred watts provided to others power supply cards.
- **Helmer Gauge Heat Filament** – dedicated for non-standard reference voltages generation, which require relatively high power.
- **Helmer Gauge Grid and Filament Bias** – does not require high-power source. The card is dedicated for high voltages (even up to couple of kV) to the bias and filament the gauge.
- **Helmer Gauge Deflectors and Suppressors** – card generates non-standard reference voltages.
- **Bayard-Alpert Gauge Grid, Filament heating and Bias** – high power card (even up to 500W), dedicated to supply the grid, filament and bias through non-standard high voltages.

Measurement cards:

- **Helmer gauge electrometer** – dedicated to ultra-low current measurements, operates down to 1fA (1 femto = 10^{-15}).
- **Bayard-Alpert electrometer** – dedicated to current measurements down to 100fA. The card must be shielded by the faraday cage against electromagnetic field interferences (EMI).
- **Penning Cold-Cathode** – with a measurement range down to pA. This card contains high-voltage circuits (providing a couple of kV), integrated with an analogue measurement circuits. Penning gauges do not require high power source.

- **Pirani** – it provides a pressure measurement down to 10^{-4} mbar, when the Penning gauge can be safely switched on. It is possible to place the Pirani circuit into the Penning card (solution implemented into the TGP300).
- **Piezo-Resistive** – it provides a pressure measurement from 10^{-3} up to 2000 mbar.

Communication cards serve a communication interfaces such as:

- **PROFIBUS** – industrial communication protocol for differential RS485 interface. It is widely used by majority devices of the CERN facilities and industry [26].
- **PROFINET** – industrial communication protocol using Ethernet interface [26].
- **Ethernet** – standard computer network interface used in the LAN (Local Area Network) [27].
- **RS232/RS485** – an old serial (RS232) and differential (RS485) interfaces. The card is dedicated to achieve backwards compatibility with majority of old industrial devices.

It must be mentioned, that the controller will be equipped with a test card, dedicated for simulation of the measurement cards, flashing the main microcontroller and performing controller tests. The user interface of the controller displays all available information about sensors states, settings, or external cards details. The motherboard drives the GUI through parallel buses.

The USB is a default interface for an external bootloader. Its main role is to provide the ability to update the firmware of the controller. What is more, the USB can be used to download the measurements, configurations, or other system data to a PC. Alternatively, the data transfer between the controller and a PC/Laptop can be also carried out through emulated RS-232 serial port set by the motherboard.

3.2 Motherboard

The motherboard is responsible for driving all system parts. The main chip is the STM32 microcontroller equipped with embedded controllers for several interfaces and devices (Figure 8).

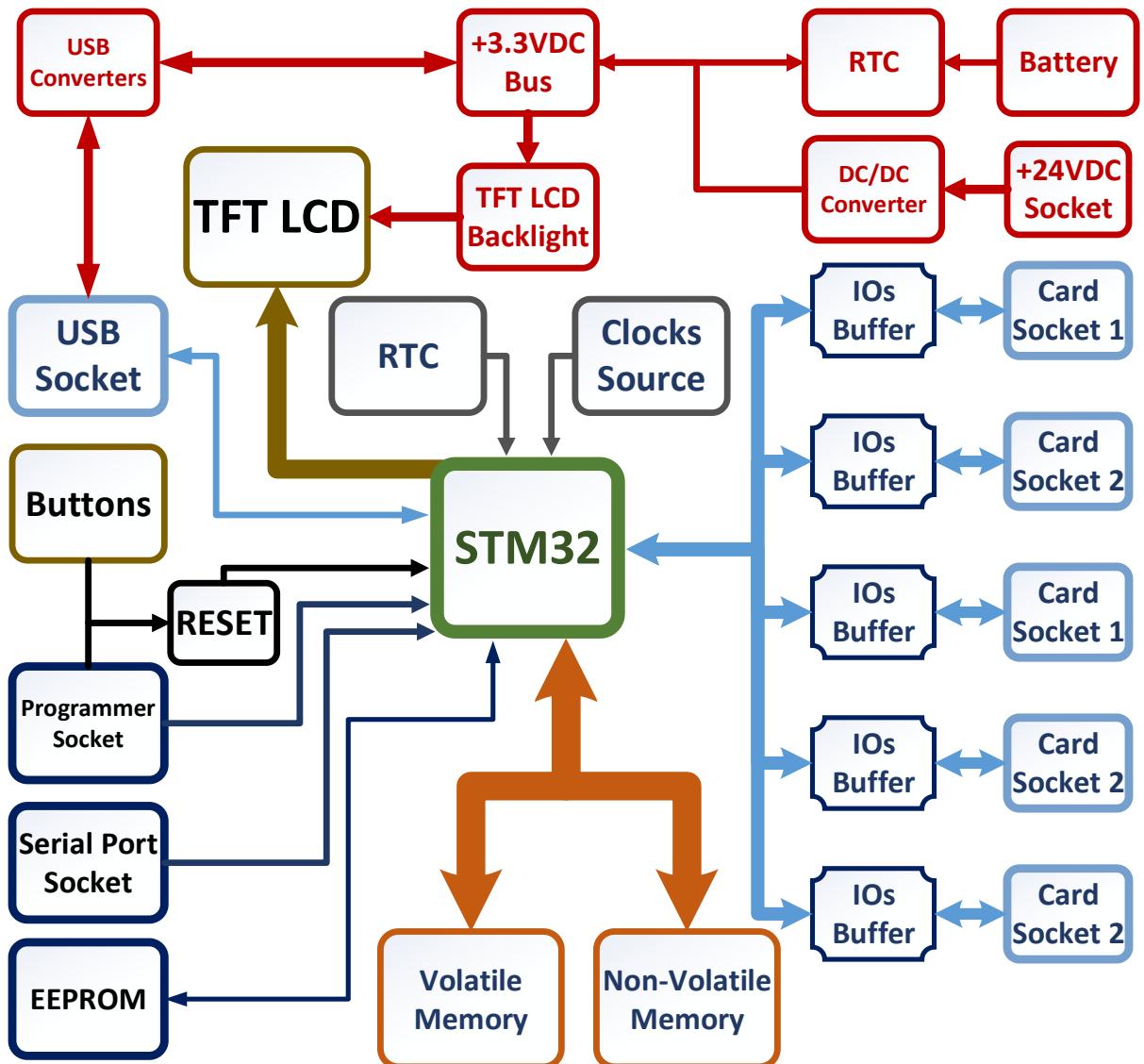


Figure 8. The motherboard block diagram [100].

Each card socket (except power socket) has an independent SPI controller with a buffer containing an auxiliary IO port. This port is used to recognise an external card when connected and generate interrupts of the STM32. What is more, the communication card SPI port has two-directional buffer for switching the operation mode between cards and the motherboard (details in the chapter 6.2).

Non-volatile and volatile memories are data buffers of the TFT LCD. The volatile memory is a buffer of the standard operation mode. The data stored in a volatile memory will be lost, when the main power supply is disabled. A non-volatile memory is used to keep the data independently of the power supply state. Both memories share the same communication bus. The EEPROM (Electrically Erasable Programmable Read-Only Memory) is dedicated to hold events time-stamps like, pressure or power supply cards changes, SPIs communication errors or other critical events.

For the purpose of this project, an internal Real Time Clock (RTC) of the STM32 is used. What is crucial, the RTC will hold data, only when additional back-up power source is on (details in the chapter 4.4). Critical system event time-stamps (such as pressure drops) are stored in the non-volatile memory.

The overall performance of data transfer depends on a single SPI throughput. The maximum sampling frequency depends on how many measurement cards are connected to the motherboard at the time (maximum four measurements cards can be connected and the SPI throughput has to be divided by four). Presented architecture is not recommended for systems requiring ultra-high sampling frequencies in parallel (for this project, it is sufficient).

The user interface contains standard push-pull buttons. Despite out-of-date construction, they are widely used in industry due to its very low price, high reliability of operations for long time span and high resistance against mechanical tensions or vibrations. Instead of standard buttons, majority of electronics devices use touch TFT LCDs. This solution could be better for this project, but touch LCDs are not recommended for industrial environments (usually, they have low mechanical durability and they are very sensible for noises). Taking into account pros and cons, the standard push-pull buttons have been chosen.

The power supply system consists of the main power converter, TFT LCD backlight converter; USB and RTC back-up power circuits. The main power circuit converts the +24VDC input to +3.3VDC, provided to crucial chips. Besides, the TFT LCD has additional backlight LEDs, which require higher voltage than the +3.3VDC. The USB and TFT LCD Backlight must have additional DC/DCs to convert the +3.3VDC to required higher voltages. The USB power circuit must be flexible in operating. The motherboard must provide opportunity to supply external devices on the USB bus and simultaneously, these devices can provide a power to the motherboard.

An auxiliary serial port is added for the purpose of prototyping and testing. When the controller will operate conventionally, this port will not be used. This port is accessible only when the controllers housing is disassembled. The STM32 can be flashed through the USB or a standard STM32 programmer. The programmer socket is available only during prototyping and testing, when the housing is disassembled. Moreover, this module and an additional push pull button can generate the reset state of the STM32.

3.3 Front panel

Figure 9 presents the main view of the controller front panel. It consists of the TFT LCD, eight user and reset button, and an USB socket. The TFT LCD is 5-inch diagonal. Due to the housing dimensions constrains it was not possible to fit larger display into the front panel and finally the widest possible 5-inch TFT LCD has been chosen.

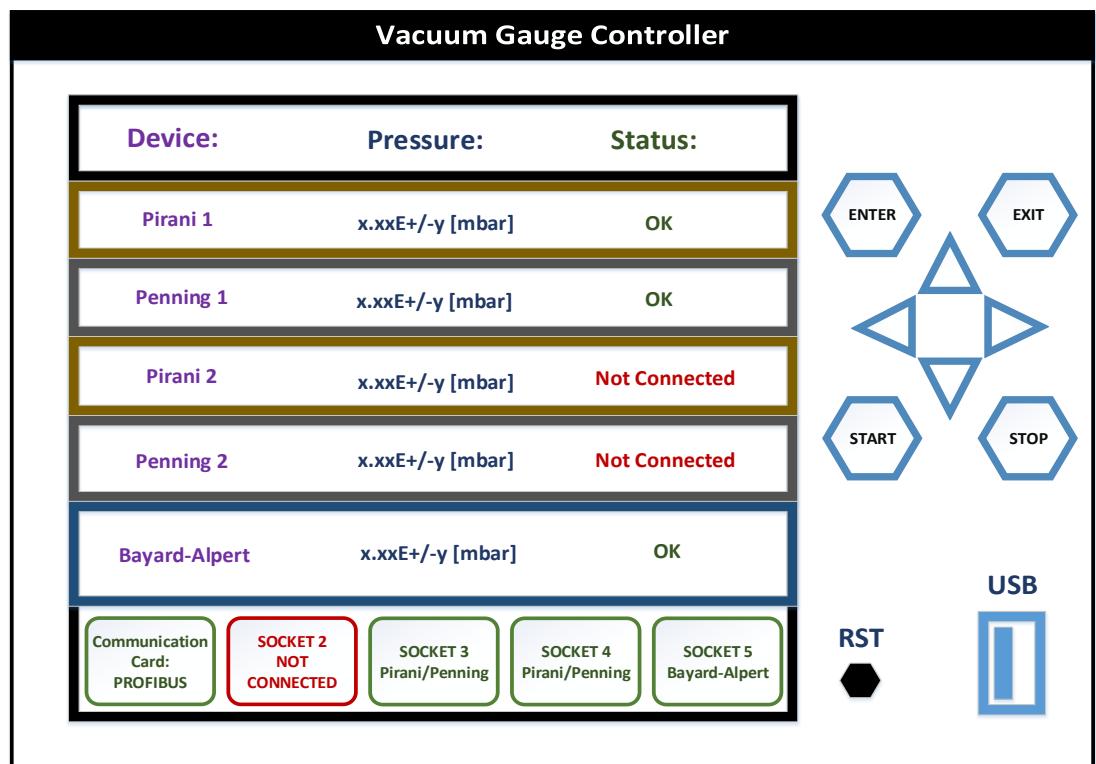


Figure 9. The Front Panel of the Vacuum Controller [100].

This TFT LCD gives the ability for an advanced graphical visualisation implementation of the control system.

The project goal assumes that a couple dozen GUI templates will be available, such as:

- “welcome” screen with status bar indicating a progress of initialisation,
- main operational view presenting general information about the gauges (Figure 9),
- measurements display window (as a graph or zoom of displaying data),
- control template for easily setting desired values of pressure,
- controller configuration windows with options to set main operational parameters,
- window with details of connected cards,
- errors and events histogram window.

The controller uses different templates. A single template requires approx. 1Mbits of memory capacity. Internal STM32 capacity is not sufficient to hold all data of the TFT LCD (templates, fonts and logos). For these reasons, an external non-volatile and volatile memories are used as the buffers for the TFT LCD data. In order to facilitate the manual update of the controller firmware, the USB socket has been placed in the front panel.

3.4 External cards

External cards architecture is the same for all types of cards, except the main power supply. Each card has common digital part and minimum one DC/DC converter. Figure 10 presents an example architecture of external cards, with a division into three main parts such as: digital, analogue and power circuits.

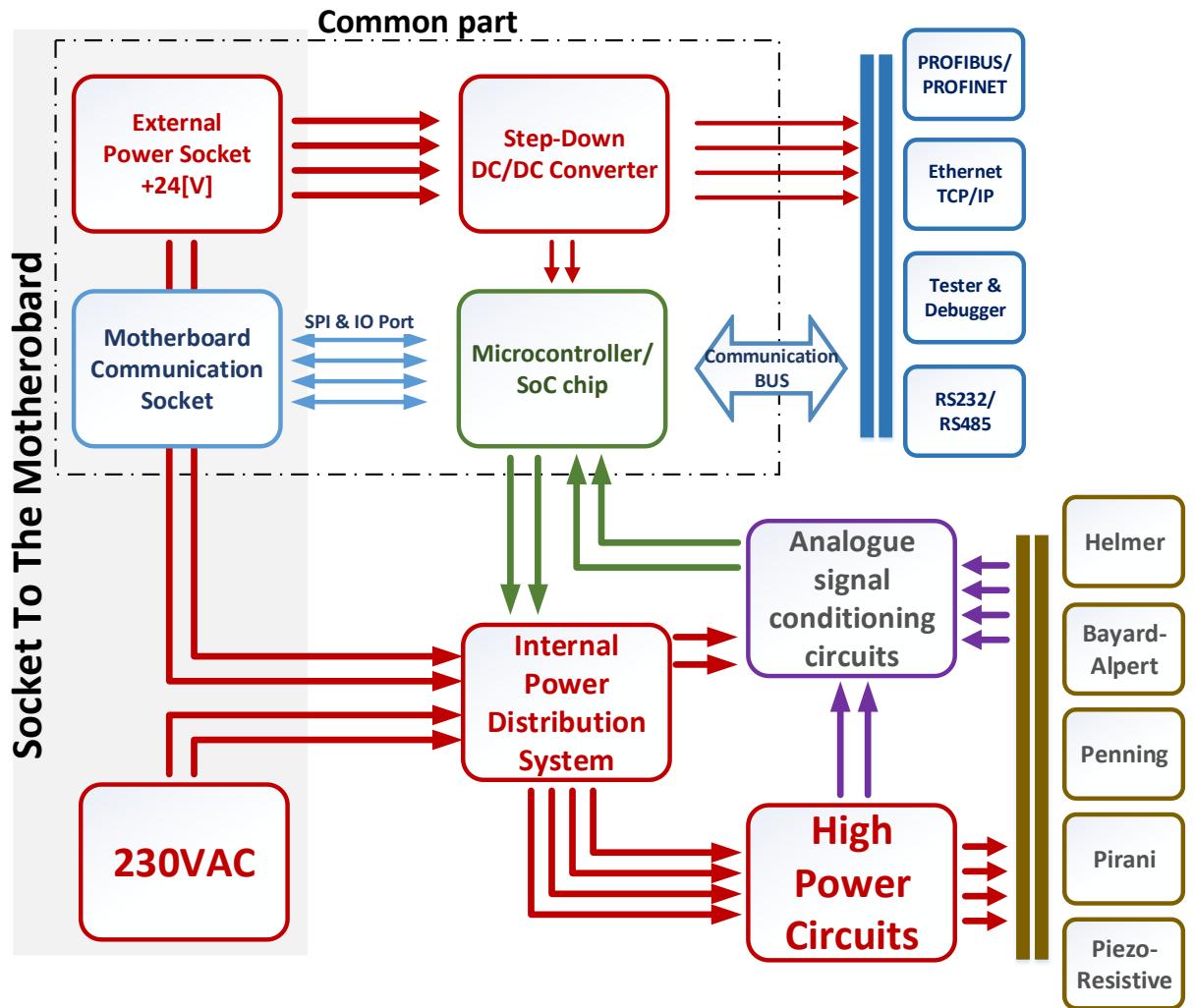


Figure 10. The block diagram of external card architecture [100].

The common part is based on a microcontroller or SoC (System-on-Chip) chip. Depending on a card type, this chip is responsible for collecting measurements, driving SPI communication, controlling adjustable power supplies and being a hub for external communication networks (e.g. PROFIBUS).

The measurement cards consist of analogue signals conditioning circuits, equipped with additional ADC (Analogue-to-Digital Converter). Usually, measurement cards are equipped with dedicated low-power chips together with internal compensation modules (e.g. temperatures, humidity), to provide reference voltages for ADCs. Moreover, some measurement cards require additional power circuits to generate ultra-high or high voltages.

The power supply cards chips software is responsible for driving the control regulators used to keep output voltages stable. These cards contain a DAC (Digital-to-Analogue Converter), for driving high and ultra-high voltages. The Control regulators calculate the output values of DACs, based on the output values of power supplies. Usually, internal DACs of standard microcontrollers are a sufficient solution for low accuracy sensors. The power supplies cannot directly communicate with measurement cards. For this reason, the motherboard must be used for controlling voltages accordingly to the output pressure of sensors. The exception is the main power supply, which is a standard high-power not-adjustable supply, free of any digital circuits.

3.5 Power supply system

Figure 11 shows the power supply architecture, used to provide +24VDC and 230VAC to the external cards' sockets, and its implemented on the auxiliary backplane board.

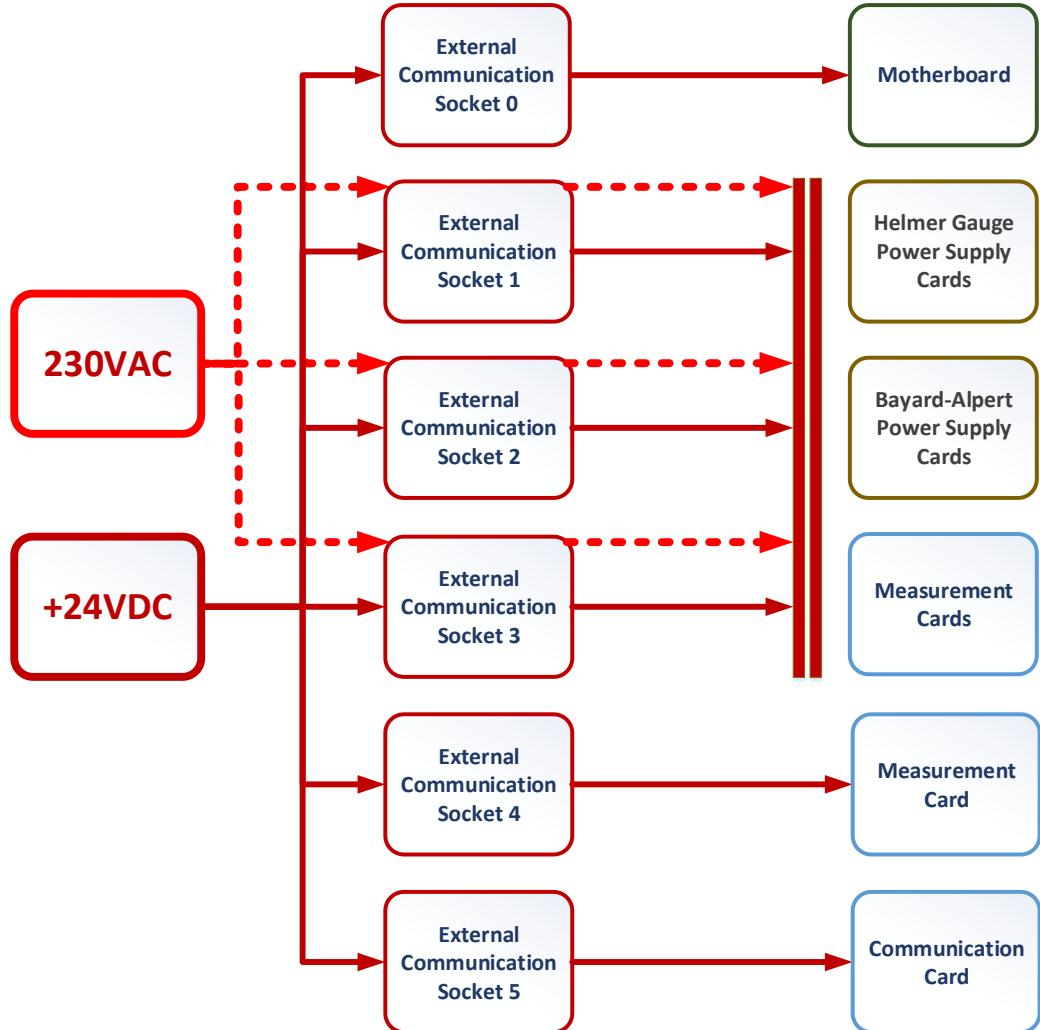


Figure 11. The power supply architecture [100].

The 230VAC is dedicated to feed only the power supply cards, and it is used as an auxiliary high-power AC source for the heat filament or grid elements of Helmer and Bayard-Alpert gauges. The other measurement and communication cards are only using +24VDC.

Measurement and power supply cards share the same three sockets (socket 1-3). Only two of them (socket 0 and socket 5) are used by digital cards (the motherboard and a communication card). A socket number 4 is dedicated only to a measurement card.

What is worth to highlight that a ground topology of the boards has to be divided into the digital, power and analogue ground planes, connected in only one point (star topology) in order to achieve higher measurements precision.

3.6 Controller configuration

Following chapter presents all configurations of external plug-in-units in industrial chassis, dedicated to the 42HP 3U industrial cassette, with maximum of six cards.

3.6.1 Helmer gauge configuration

The Figure 12 and Figure 13 show external cards widths and their placement pattern in the housing. For the Helmer Gauge configuration, four cards operate as power supplies and two respectively as a communication and measurement cards. The widths and positions of the cards are fixed for all configurations and the order of cards for the Helmer gauge cannot be changed.

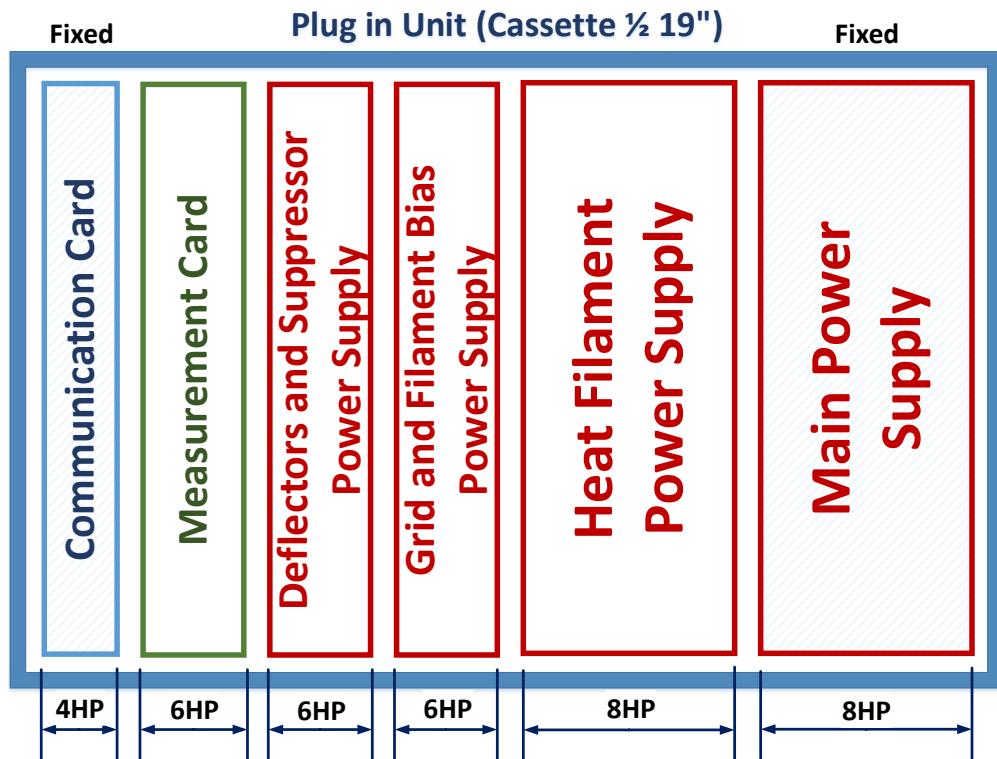


Figure 12. External cards configuration for Helmer Gauge only [100].

The width of the main power and heat filament power supplies cards are 8HP. A 6HP width is the minimum width, highly recommended for power supply cards due to the bigger mechanical dimensions of the components, wider traces of conductors, bigger elements for dissipating thermal power and bigger clearances between circuits.

The thinnest card is a communication card. The card has the 4HP width, which is sufficient to fit the communication interfaces sockets (such as the RJ45 of Ethernet or D-SUB9 of Profibus) into the card. The communication card would occupy less space, if a newer and smaller

versions of the sockets were used (it is not possible, because backward compatibility with older industrial controllers is required).

3.6.2 General configuration

The general operational configuration of the controller, without Helmer gauge, is presented on the Figure 13. The external cards widths of this configuration are identically to the Helmer Gauge configuration.

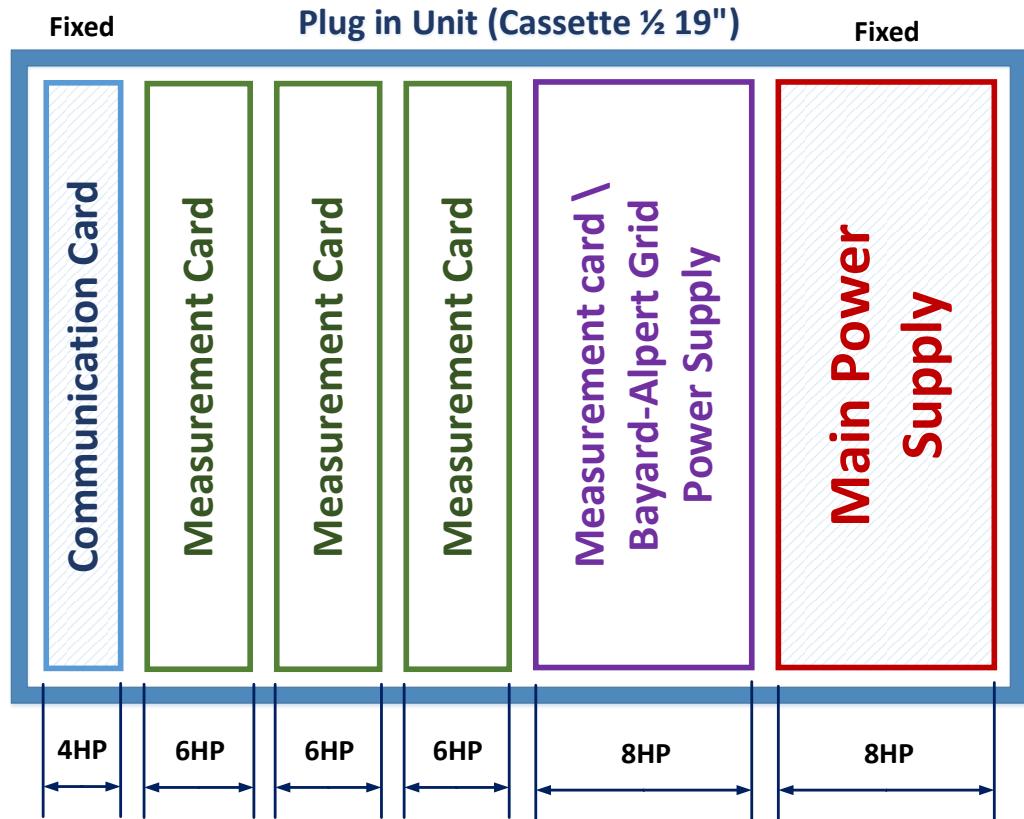


Figure 13. General configuration for external cards [100].

The Bayard-Alpert gauge requires additional high power supply and the total width of this card must be also 8HP. This card position is the same as the power supply of the heating filament. Moreover, when this gauge is not used, a measurement card can be inserted into this position.

For the configuration, it is possible to drive only one Bayard-Alpert gauge at the time, but simultaneously other measurement cards can be inserted to remaining three free sockets. Positions of a communication card and the main power supply are again fixed and cannot be changed.

3.7 Mechanical design

The controller is designed for the 42HP 3U industrial Plug-in-Unit (Figure 14). Figure 15 shows the mechanical design of the controller inside the mechanical chassis (shown on the Figure 14). The setup shows mutual placements of the components such as the TFT LCD, the Motherboard, buttons, USB, sockets, and the backplane board.

The depth of the Plug-in-Unit is 227mm, and the maximum length of external cards is 167mm [16]. Between the front panel and the end of external cards is a 50mm gap therefore the backplane board must be used (Figure 15).

Moreover, special board-to-board connectors are used to connect boards due to the 50mm gap. Besides, these connectors should not carry forward any mechanical tensions. The backplane board contains additional special industrial connectors, used for external cards.

The Motherboard and the backplane boards are mounted to the front panel of the Plug-in-Unit with the use of auxiliary standoffs and screws. The force of mechanical tensions directly affects the front panel.



Figure 14. The 42HP 3U 19-inch Europe standard Plug-in-Unit [28].

A PCB laminate is not a flexible material and it has to be secured from any mechanical hazards. The backplane board is based on the boosted 3mm thin laminate, which high stiffness contributes to more equalled distribution of mechanical stress on all of the standoffs and screws.

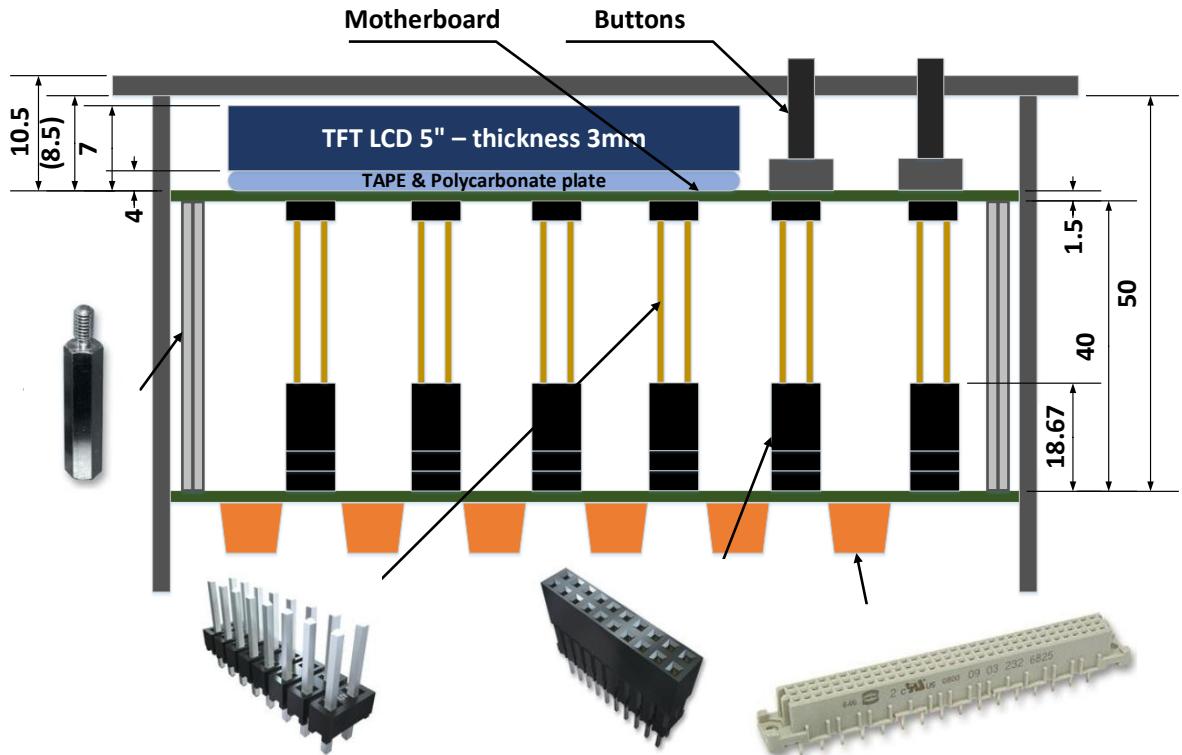


Figure 15. The mechanical design (cross-section view) [29] [30] [31] [32] [100].

The TFT LCD is fixed on the bottom side of the motherboard with the use of the 3mm thick double side tape. Adhesive properties of this tape are sufficient to hold the TFT LCD on the PCB (even during high vibrations). Problems can occur, when an ambient temperature significantly increases (adhesive properties of tape glue will decrease). E.g. for the standard rubber adhesive tape, the ambient temperature cannot exceed approx. 70-Celsius degrees [33].

4 Hardware

Following chapter presents hardware solutions used for the Vacuum Gauges Controller. This chapter contains brief overviews of hardware components, circuit descriptions with partial schematics, surveys and reviews of used solutions, advantages and disadvantages of the design, main functionalities of chips and proposals for alternative solutions.

4.1 STM32 microcontroller

The STM32F429BIT6 microcontroller, shown in Figure 16, from F4 series TQFP208 package, is the main chip of the project. It offers high performance, real-time capabilities and support of the digital signal processing (DSP) instructions. The STM32 is based on the high-performance 32-bit RISC ARM Cortex-M4 core, operated up to 180MHz. The chip is incorporated with high-speed embedded memories such as the 2Mbyte NAND Flash memory, 256Kbytes SRAM (Static Random Access Memory) and 4Kbytes Back-up SRAM. The chip contains the FPU (Floating Point Unit), able to perform floating-point operations as atomic [14].



Figure 16. STM32 microcontroller in the TQFP208 package [34].

The STM32 contains several core peripherals. The ST vendor provides multiple software development tools that decrease total hardware and software development time. The ST also provides free hardware driver libraries with multiple examples for multiple devices and free Integrated Development Environments (IDE) for programmers. Moreover, the STM32 fully supports the DSP instructions with the released DSP libraries, including examples [35] [36]. Based on the resources of the AMD corporate and the Linley Group, in 2012, the ARM core products were the most significant part of the global microprocessor market (75% of all used microcontrollers had an ARM core) [37]. In 2018, the trend of the ARM core shares of the microcontroller market is increasing [38]. ARM cores are widely used in multiple

electronics devices such as smartphones (e.g. the Snapdragon 625 microprocessor of the Xiaomi Redmi smartphones, based on the ARM Cortex A53) [39] [40]. The ARM RISC architecture makes up the small number of assembly instructions, and the Cortex-M4 cores are targeted to industrial or automotive applications [41].

Taking into account all pros, the STM32 with the ARM Cortex-M4 core has been chosen as the main chip of this project. Moreover, the single piece price of the STM32 to offered performance is reasonable.

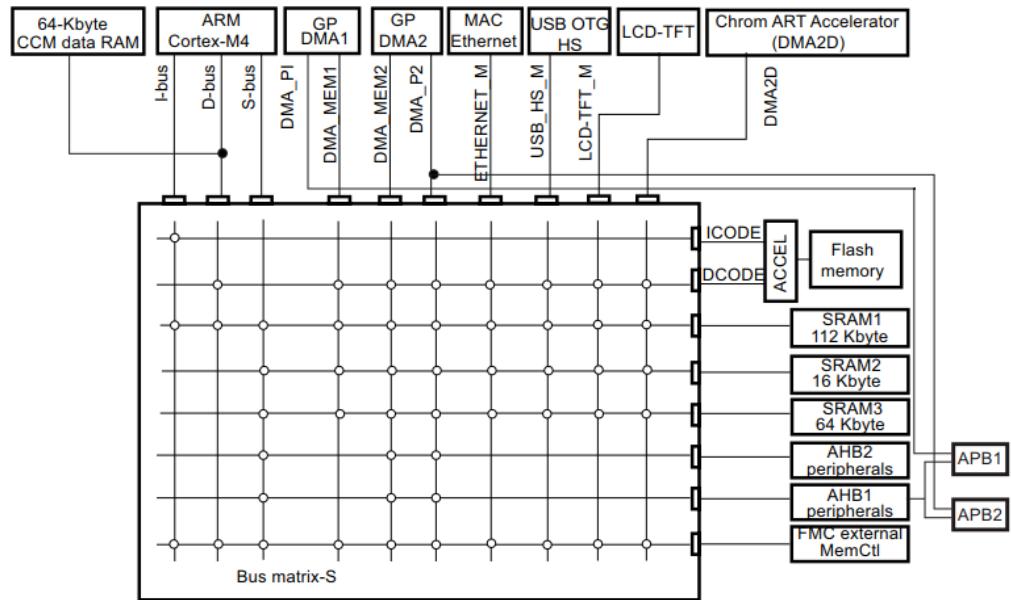


Figure 17. The STM32F429xx microcontroller architecture [42].

On the Figure 17 there is presented the system architecture of the STM32F429xx microcontrollers. The figure shows mutual connections between the ARM core and its peripherals on the main 32-bits multilayer bus matrix-S. It must be mentioned that the system exchanges data through crucial data and address buses such as [41] [42]:

- internal ARM Cortex-M4 I-Bus, D-bus, S-Bus, ICode and DCode buses,
- AHB1 and AHB2 (Advanced High-performance Bus-Lite),
- APB1 and APB2 (Advanced Peripheral Bus),
- FMC (Flexible Memory Controller),
- DMA2D (Advanced Chrom-Art Accelerator) and DMAs (Direct Memory Access).

The main chip has advanced embedded communication interface and device controllers. For this project, the following controllers are used([42] [14]):

- **2 x USB OTG FS/HS** – the USB 2.0 On-the-Go Fast-Speed/High-Speed controller with built-in PHY (physical layer) driver, fully supported on the hardware.
- **6 x SPI** – drive up to 45Mbits with full STM32 hardware support.
- **6 x USART (RS232)** – Universal Asynchronous Synchronous Receiver Transmitter controllers enable to drive devices through the RS232 interface.
- **FMC and FSMC** – Flexible and Static Memory Controllers for driving external volatile and non-volatile memory.
- **LTDC for TFT LCD Displays** – Liquid Transistor Display Controller.
- **DMA** – Direct Memory Access controller enables to transfer data from memories to memories, peripherals to memories and from the memories to peripherals, without using core computer power.
- **DMA2D** – enhanced graphic content creation controller, dedicated for supporting LTDC for immediately transferring graphic data to a TFT LCD.
- **RTC** – Real Time Clock unit being an independent 15-bit stages BCD counter contained time and data values, with sub-second accuracy and included hardware calendar.
- **Bootloaders** – enable to exchange firmware of a microcontroller with the use of standard interfaces such as SPI or USB.
- **Hardware general-purpose timers system** – with 16/32-bit resolution.
- **Window and Independent Watchdogs** – controllers based on the electronic timer used to reset a microcontroller, when it will enter a suspend mode.
- **CRC** – Cyclic Redundancy Check unit, used to calculate a special code from 32-bit words.
- **EXTI** – External Interrupt/Event Controller used to drive up to 16 independent interrupts generated by external devices.

Figure 18 presents the STM32F429-Discovery evaluation board that allows user-friendly application development for the STM32 microcontroller, without designing a PCB. This board contains several microcontroller's peripheral devices, such as the 64 Mbits SRAM, gyroscope, accelerator, 2.4" TFT LCD, DAC, USB OTG Micro-AB socket, LEDs, push-pull buttons and programmer with debugger [43].

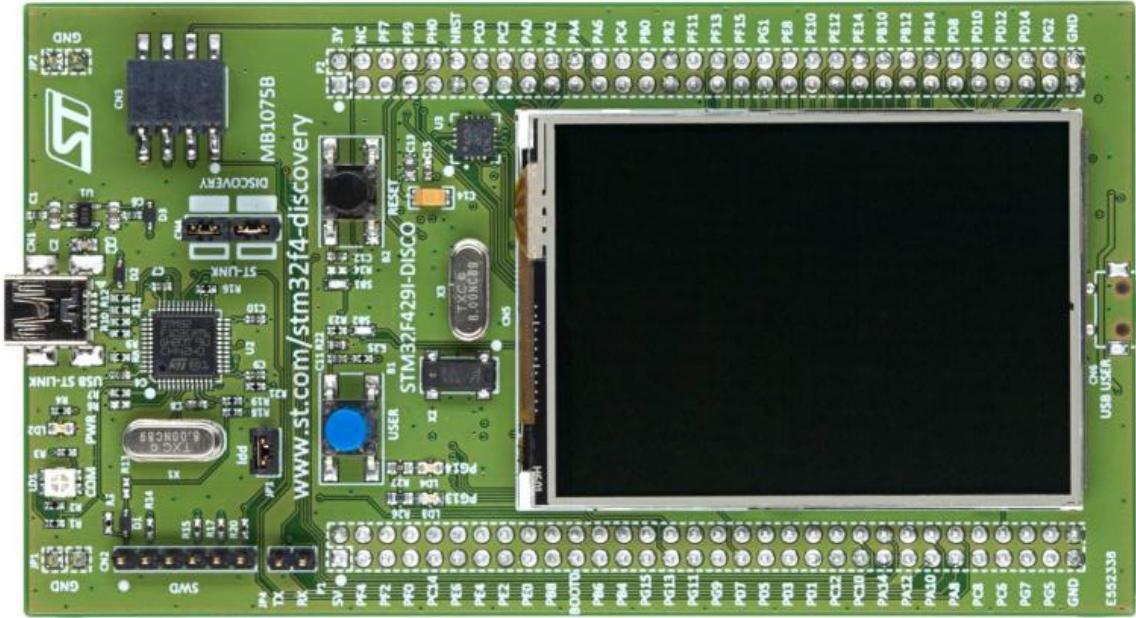


Figure 18. The STM32F429 Discovery evaluation board [44].

This board is providing the opportunity to check and test parameters of the microcontroller, before designing and making a prototype (significant advantage). For this evaluation board, the ST provides multiple free software libraries with multiple examples and schematics of the connections between the STM32 and its peripheral devices [44]. The price of the STM32F429 Discovery is approx. 25\$ for a single board. The chosen evaluation board operates as the test board for the motherboard.

4.2 TFT LCD display

The 5-inch TFT LCD is shown in the Figure 19. The dimensions of the TFT LCD module are 120 (Width) x 75.8 (Height) x 2.8 (Depth) mm respectively, together with the flexible flat conduct tape (the yellowish element on the Figure 19), is approx. 95mm. This height is the maximum matching height of a PCB and components, dedicated to fit to the 42HP 3U housing [15] [16] [28].



Figure 19. The 5-inch TFT LCD display [45].

The matrix resolution of the TFT LCD is 800 (width) x 480 (height) pixels. The active area of the TFT LCD is 108 (width) x 64.8 (height) mm. The dimensions of a single pixel are 0.135 mm (respectively for the width and height) [15]. During displaying, the TFT LCD backlight has to be enabled and its backlight colour is white. The operational voltages of this display are +3.3VDC (for logic part) and +19VDC (for the backlight - details in the chapter 4.5.6) [15].

In order to achieve a connection between the main microcontroller and the TFT LCD, the FFC (Flexible Flat Cable) socket with 0.5mm raster has to be used [15].

The display is fully compatible with the RGB888 (Red, Green, Blue with eight signals for each colour) synchronous interface, frequently used to drive small TFT LCDs. The RGB888 circuit is shown in the Figure 20. This interface provides opportunity to drive TFT LCDs with the 24 bits depth of colours. For each colour 8 signals lines ($R[0:7]$, $G[0:7]$ and $B[0:7]$ signals on the Figure 20) are dedicated. Each of them define the intensity of colours from 0 to 255 values. Three values of colours are used to construct the RGB colour pallet by mixing these colours to each other [15] [14].

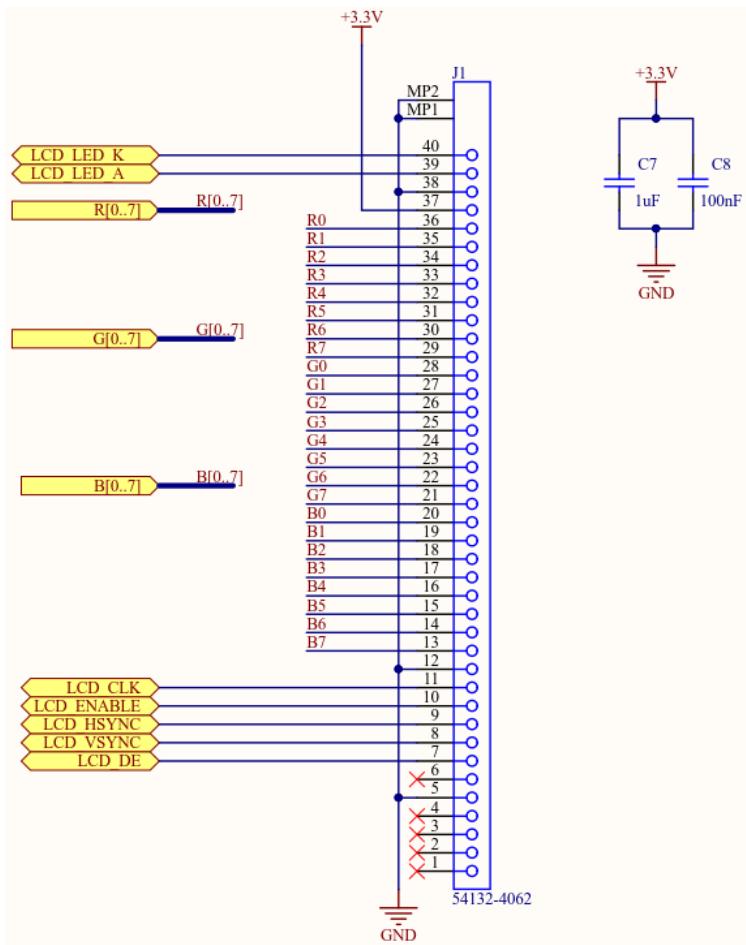


Figure 20. The circuit of the TFT LCD display [100].

The **HSYNC** (Horizontal synchronisation), **VSYNC** (Vertical Synchronisation), **DE** (Data Enable) and **CLK** (clock signal) control lines have to be used to drive all the pixels. These signals are responsible for selecting rows and columns of the LCD matrix, where the data for the single pixel (in the RGB888 format) will be loaded. The data transmission is constantly performed and fully supported by the internal controller of the STM32 (without loss of the core computing power) [42] [46].

4.3 External memories

The following chapter presents external memories. They act like buffers for the TFT LCD template data, that cannot be loaded into the internal memories of the STM32. This chapter contains a short descriptions of memory hardware connections with the STM32, advantages and disadvantages of this solution and main operational principles.

4.3.1 FLASH

The main goal of the external NOR Flash is to hold the TFT LCD templates data independently from the power supply presence. Figure 21 shows the circuit of the NOR Flash. The chip is the 64 Mbits NOR memory, grouped into four 16Mbits banks. It is compliant with the JEDEC single-power-supply Flash standard [47]. The Flash is driven through asynchronous parallel communication interface, with some control lines and two main data (the DQ0 - DQ15 lines on the Figure 21) and address (the A0 – A22 lines) buses. The data word width is 16-bits [47].

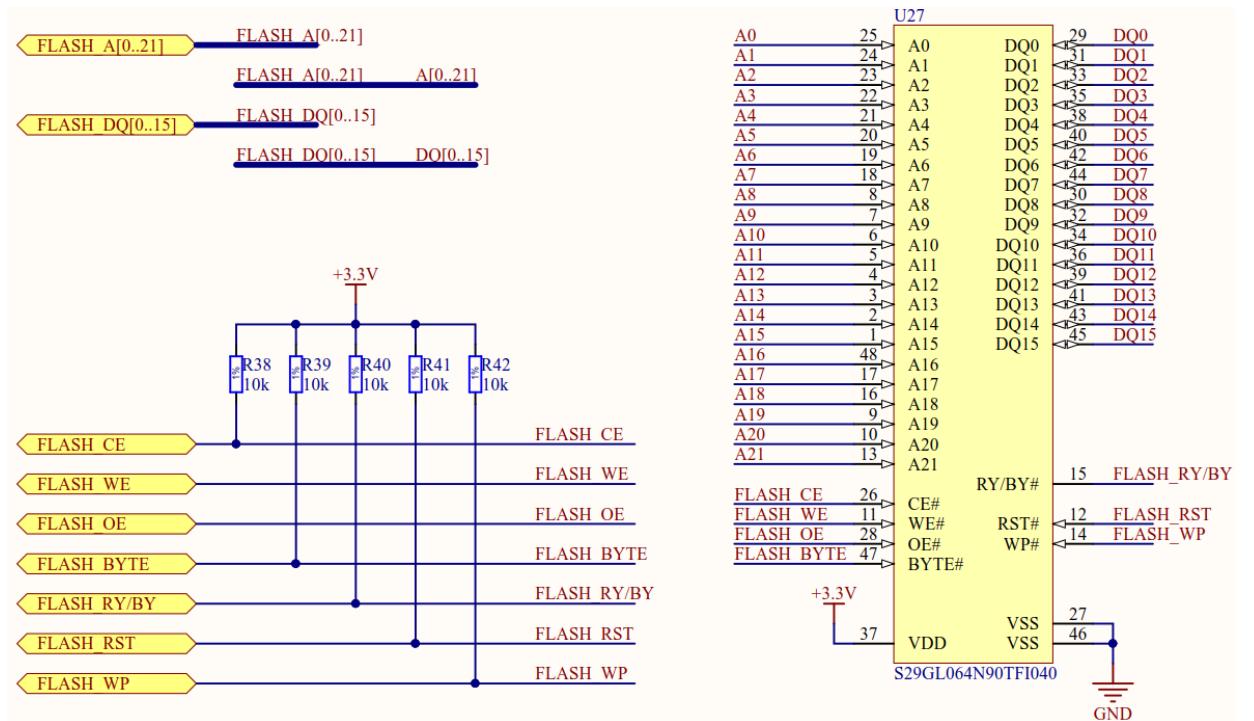


Figure 21. The 64Mbit NOR Flash circuit [100].

The Flash operates at high-speed transfer data rate. For this reason, the chip has to be terminated with resistors connected in series to signal lines between the STM32 and the chip. The resistors termination are used to keep the time integrity of signals [47] [48].

The read time of the Flash is shorter than erasing or writing times. A single page reading time (64kBytes) is 25ns, for comparison, a writing time is 90ns. Despite high performance, the chip cannot be used as an active data buffer of the TFT LCD, due to long read time. Moreover, Flashes cannot be used as a writable data buffer, because of the limited amount of writing/erasing cycles. The chip's manufacturer guarantees the correctness of the operations only up to 100000 writing or erasing cycles [47].

Modern NOR and NAND Flash chips shows similar performances. The NANDs are more frequently used than NORs, due to denser layout and greater storage capacity per chip. Moreover, the price difference between single pieces of NAND and NOR is negligible. Figure 22 presents the comparison between them two.

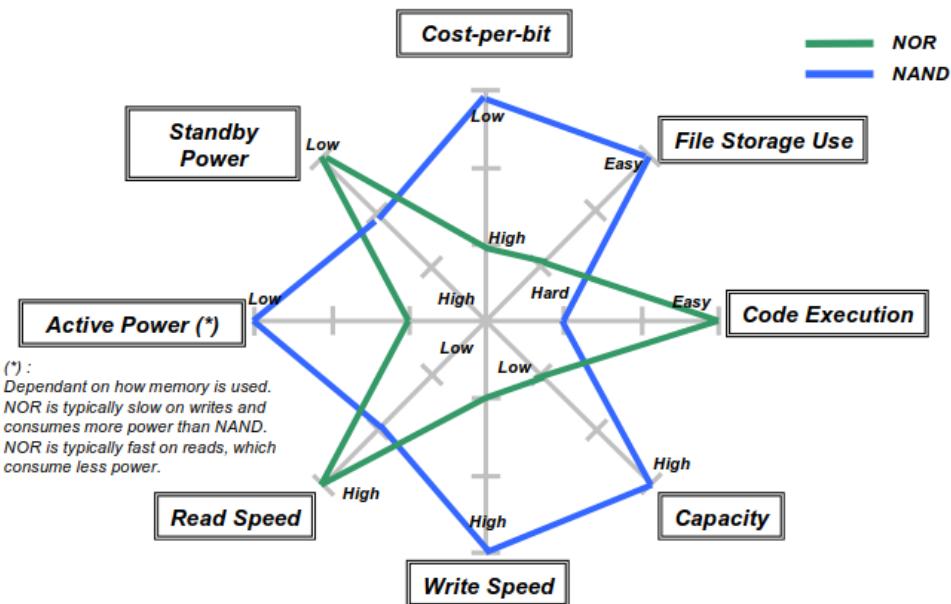


Figure 22. The graphical comparison of NOR and NAND Flashes [49].

The NAND is recommended for memory devices, that will be frequently modified (significantly shorter writing and erasing times than NOR Flashes). For mobile devices, supplied from low-power sources (e.g. batteries), a NAND will be a better choice due to the lower operating voltages than NORs [47] [49].

As it is shown in Figure 22, NORs are highly recommended for being used as a programmable memory of microcontrollers' cores. Moreover, a NOR can easier access to a single memory cell (every cell can be handled separately) [47] [49].

Despite multiple advantages of a NAND, a NOR Flash has been chosen, due to the shorter reading time. This parameter is particularly important for the controller's initialisation, when the TFT LCD template data has to be transferred as fast as possible to a volatile memory.

4.3.2 SDRAM

The SDRAM (Synchronous Dynamic Random Access Memory) and DDR RAM (Double Data Rate Random Access Memory) are frequently used memory chips in a variety of embedded systems or other electronics devices (such as PC computers, laptops, GPS modules or smartphones). The SDRAM is a volatile type of memory, that cannot hold data after disabling the power supply [50].

The standard clock frequencies of SDRAMs are 100MHz or 163MHz, but they can operate with significantly lower frequencies. Theoretically, the SDRAM does not have a hard limitation about minimum frequency, and they can still operate even in the kHz range. In contrary to a SDRAM, a DDRRAM has a predefined minimum clock frequency, and usually it can operate from minimum 100MHz - 200MHz up to even a couple of GHz. The maximum output clock of the STM32 FMC controller is 90MHz, being the main operational frequency of this chip [51] [50].

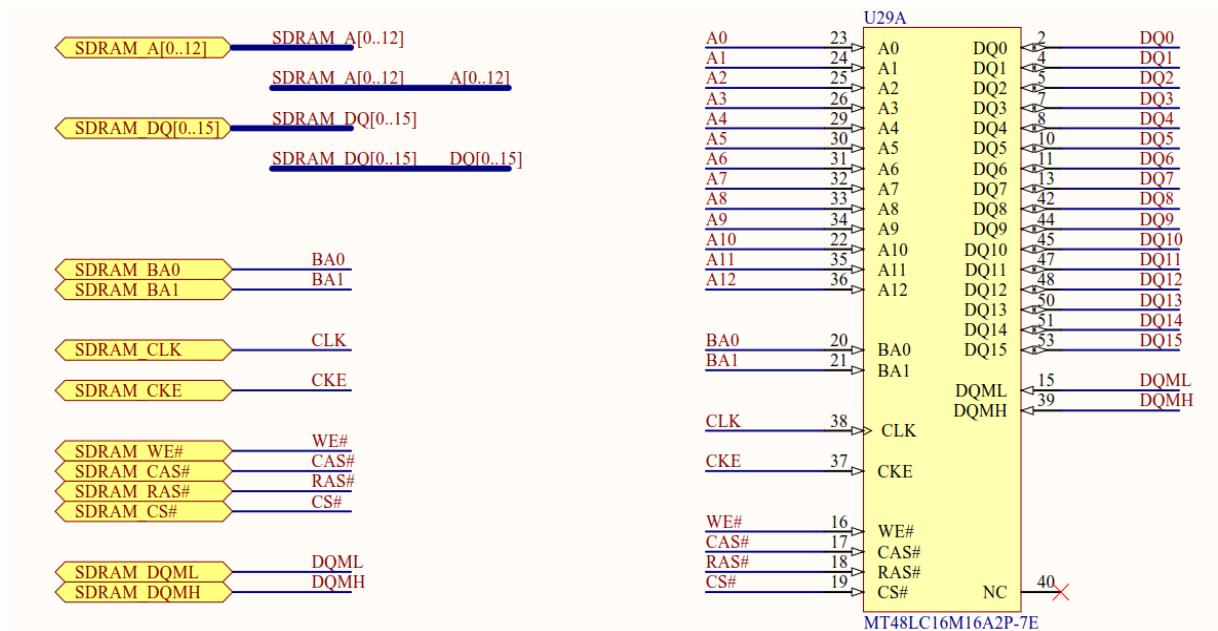


Figure 23. The 256Mbit SDRAM circuit [100].

Figure 23 presents the circuit of 256Mbits high speed SDRAM, connected to the STM32. The internal structure of SDRAM memory is shown in the Figure 24. The SDRAM is internally divided into four banks, and it is driven through a synchronous interface. Each memory bank of the SDRAM is organised by 8192 rows and 512 columns of memory cells, with 16-bit data width. Moreover, operations or commands are latched on the positive edge of the clock signal, which must be provided to the chip [51].

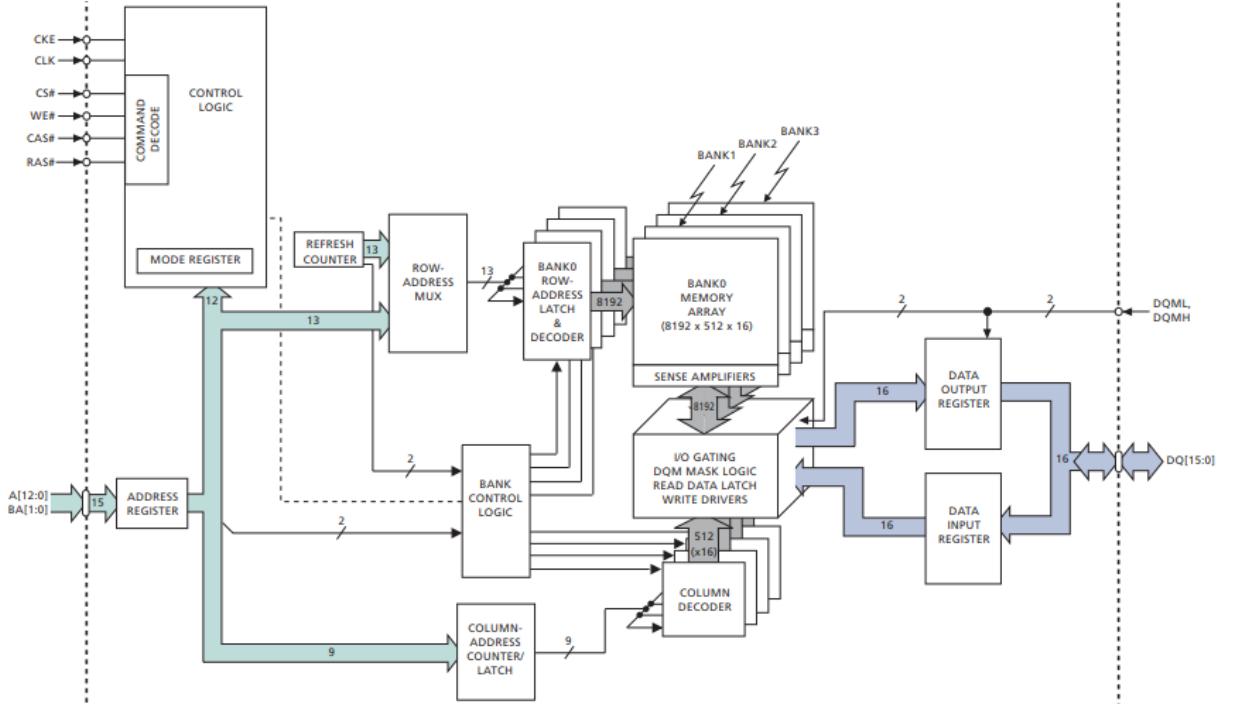


Figure 24. The block diagram of the chosen SDRAM [51].

The SDRAM and NOR share the same data and address buses. Only first sixteen bits of the address bus are dedicated for addressing memory cells of the SDRAM. During standard operations, the FMC data and address buses will be only used by the SDRAM (the NOR Flash will be in reset mode). It must be mentioned that parallel interfaces of the SDRAM and NOR Flash are fully supported by STM32.

What is crucial, the time integrity of control, data and address signals must be kept. These signals have to be terminated with resistors, placed in series with signals between the main microcontroller and the SDRAM [51] [48] [42].

The SDRAM is dedicated to be an active data buffer for the TFT LCD. What is worth to highlight, that the SDRAM/RAM memories are significantly faster at reading and writing than any kind of Flash memories. Moreover, the SDRAM is free of any limitations, connected with maximum number of writing/erasing operations. Taking into account the pros and cons, the SDRAM has been chosen as the main operational memory of this project.

4.3.3 EEPROM

The 24FC512 EEPROM chip circuit is shown in the Figure 25. The chip is driven through the I²C (Inter-Integrated Circuit) communication interface. The EEPROM has 512kB memory space and the maximum operational frequency of the I²C clock is 400kHz [52]. Maximum transfer data speed is up to 50 Kbits and the maximum reading time of a single page is no longer than 5ms [52].

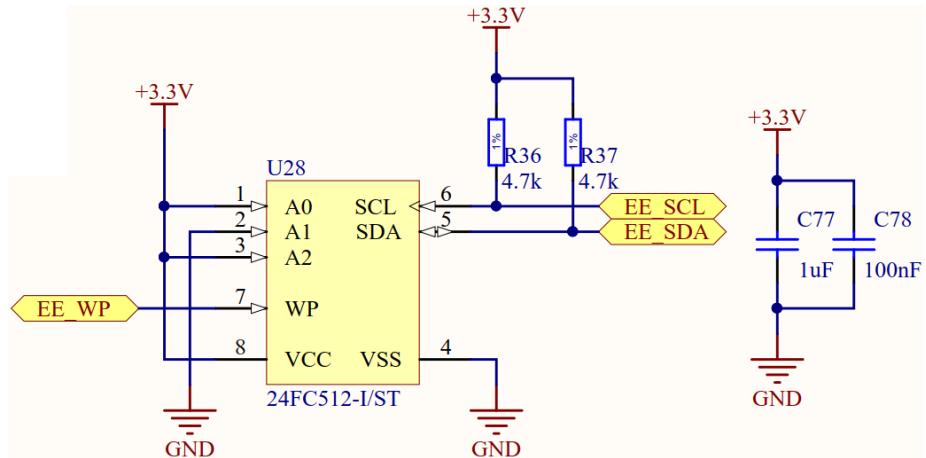


Figure 25. The 512kb Serial 24FC512 EEPROM circuit [100].

It must be mentioned, that the I²C data and clock lines must be pulled-up to positive voltage through resistors. Moreover, the capacitive loads of these signal traces cannot be higher than 400pF [53]. In another case, the EEPROM chip will not be operational. This chip can modify a single byte of non-volatile memory at any time.

4.4 Real Time Clock

The STM32 microcontrollers, from F4 lines, have an independent built-in RTC module [42]. Whenever a high accuracy of time counting (up to nanosecond) is required, it is highly recommended to use an external RTC chip instead of built-in modules. For the VGC project, the accuracy of the RTC used is fully satisfactory [42] [54].

In order to use the STM32 RTC, additional external crystal oscillator must be connected to the two dedicated STM32 IO ports (“OSC32” labels). This oscillator unit must be equipped with two loaded capacitors in parallel (more details in the chapter 4.9). The resonant frequency of the RTC clock source must be the 32.768 kHz (chapter 4.9). Otherwise, the RTC will not be operated correctly [42] [54]. Theoretically, it is possible to use other clock sources (like standard passive RC circuits), but they are not recommended. Moreover, the RTC can be calibrated using the 512Hz signal [42] [54].

4.5 Power supply system

Following chapter presents circuits and components of the power supply architecture. Moreover, the chapter contains short descriptions of power line filters, electrical protection modules, circuits of standard power converters, regulators, and information about power components.

4.5.1 STM32

The internal power supply system of the STM32, shown in the Figure 26, consists of multiple IO power and control ports (approx. 40 IO ports). The STM32 requires +3.3VDC input with a maximum 100mA current consumption [14].

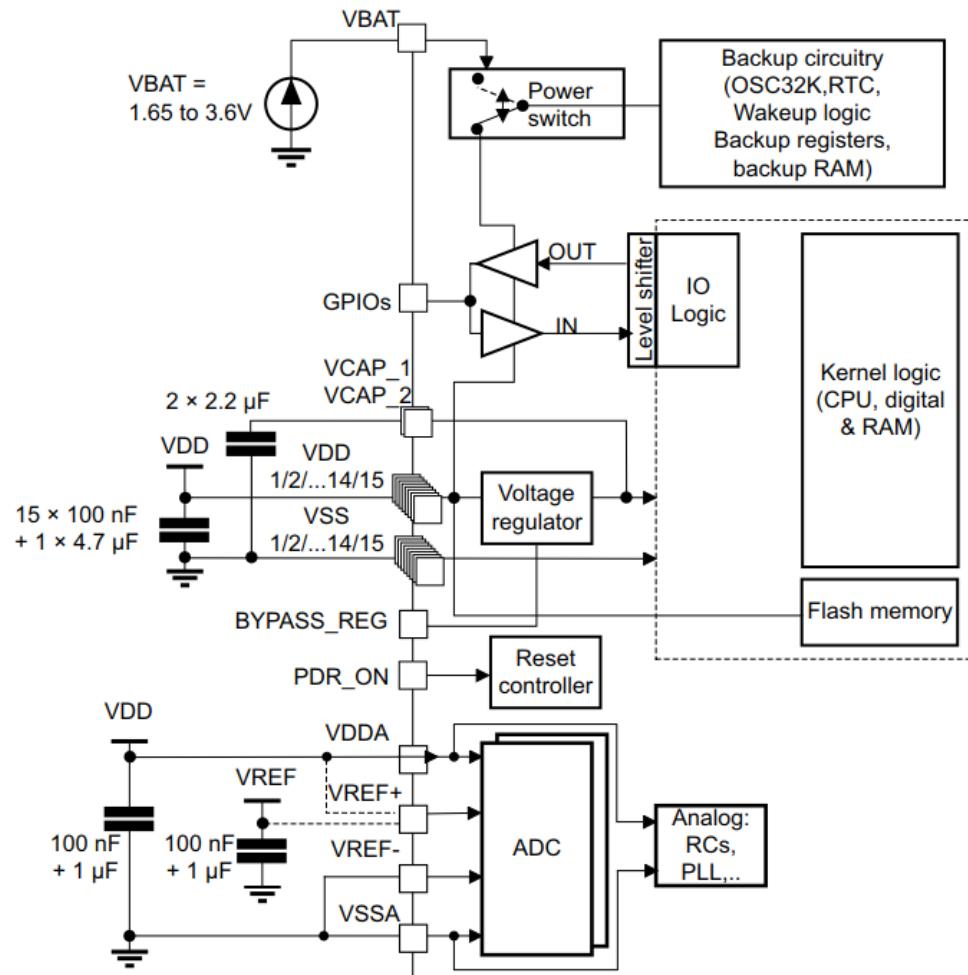


Figure 26. The power supply system of the STM32 [14].

The power supply lines consist of few subsets, and their IO ports names define main functions in the power system. The STM32 contains power lines like the **Vdd** and **Vss** (digital power ports), **Vdda** and **Vssa** (analogue power ports), **Vfer+** and **Vref-** (reference voltages of internal

ADCs/DACs), **Vcap1** and **Vcap2** (bypass ports of the internal regulator), **Vbat** (RTC and back-up registers) and **PDR_ON** (input port of the internal power down reset module) [14].

All power lines must be connected to suitable voltages. The motherboard design has only digital circuits and it has been designed accordingly to guidelines, shown in the technical documentation [14].

4.5.2 USB

The USB OTG 2.0 standard requires +5VDC on the VBUS line, with a maximum 500mA current [55]. The input power of the motherboard is the +3.3VDC and the step-up (boost) DC/DC converter is used to achieve full compatibility with the USB standard. The DC/DC converter circuit is shown in the Figure 27 [55] [56].

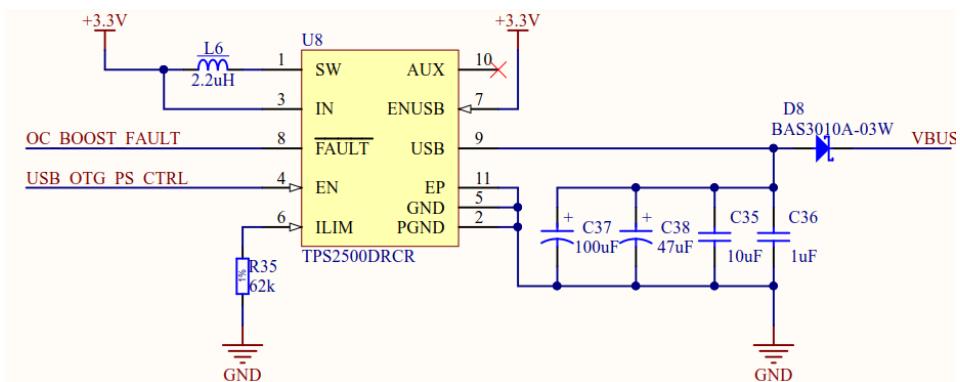


Figure 27. The DC/DC boost converter circuit of the USB OTG [100].

The main goal of the DC/DC converter is converting lower voltage to required higher voltage, with the highest possible performance. The main chip of the converter is the TPS2500DRCR with integrated USB power switch, dedicated for the USB host applications without native +5VDC bus [56].

The chip has internal adjustable current-limit power switch. A current limitation can be regulated from 130mA up to 1400mA. The maximum set output current of the USB switch is approx. 500mA. Moreover, the power switch regulator contains a thermal shutdown protection module [56].

Presented USB switch has high performance level up to 93%. Besides, the chip requires only two external components, the L6 inductor and D8 Schottky diode [56].

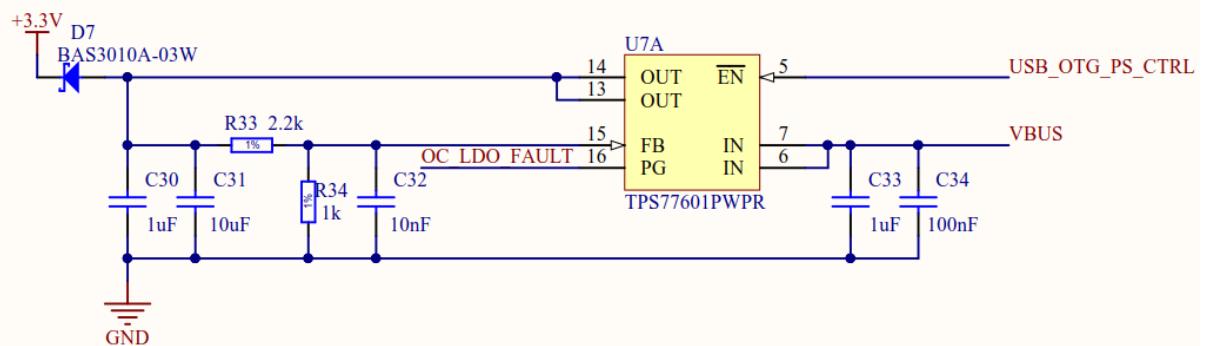


Figure 28. Circuit of the TPS77601 LDO voltage regulator [100].

Figure 28 depicts the circuit of the TPS77601PWPR chip as a linear adjustable LDO (Low Drop Voltage) regulator. This chip is used to convert +5VDC to +3.3VDC with a maximum 500mA output current. The voltage drop of the chip should not be higher than 200mV [57] [58].

The linear voltage regulator is an integrated analogue circuit with low-level noise voltage. The stability of an output signal is significantly higher than DC/DC converters and regulators are widely used as a source of reference voltages, dedicated for analogue signal conditioning circuits [57] [58]. The LDO regulator is supplies the motherboard in the bootloader mode, when the main power supply is absent.

4.5.3 Power system input

Figure 29 shows the motherboard input power supply circuit. It contains standard protection modules with a maximum of 0.5A input current. The ESD protection is based on the two-directional TVS (Transient-Voltage-Suppression) diode (D1), inside of which resistance reaches a value close to zero, when input voltage will exceed a certain voltage. This diode makes a short circuit, which causes a fuse blow, when ESDs occurs. Moreover, the diode response time is ultra-short, in contrary to the other solutions, such as varistors (an electronics component that operational principles are similar to TVSSs) [59].

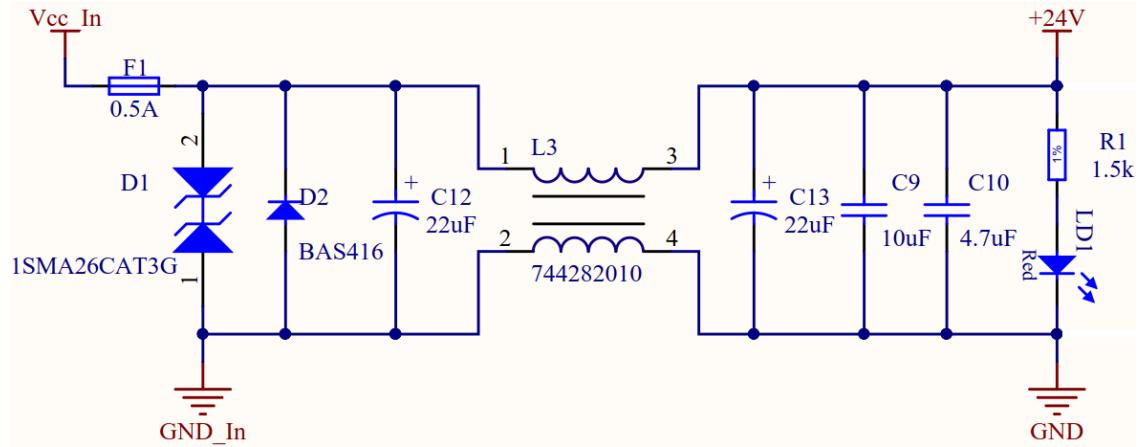


Figure 29. The power supply input circuit [100].

The BAS416 diode (“D2” on the Figure 29), placed in parallel and inversely to the input voltage polarisation, operates as the inversed polarisation protection. When input voltage is connected inversely, the D2 makes a short circuit that blows the fuse. Frequently, a Graetz bridge (four semiconductor diodes setup that always guarantees correct polarisation) or a P-type MOSFET (Metal Oxygen Semiconductor Filed Effect Transistor) setups could be used, instead of the presented solution [59]. For this project, the diode is a sufficient solution, because the controller’s chassis construction guarantees correct power supply connection (this protection is only necessary during prototyping and testing).

The input power circuit has a standard reactive filter with a common choke (L3) and capacitors (C12, C13). The common choke is a component made of two built-in inductors, wounded around a single magnetic core. These built-in inductors are inversely decoupled from each other. A common choke with two capacitors makes a LC filter which suppressing properties are significantly better than the standard RC filters or single capacitors [60].

The overcurrent protection consists of two main components: input 0.5A fuse and current limitation switch of step-down switching regulator of the main DC/DC converter (Figure 30). A standard fuse is an overcurrent protection module, which response time depends on the type of the fuse and gradient of overload current. In a case of the main step-down switching regulator, which has a built-in 2A current limiter, this module usually has shorter reaction time than standard fuses [61] [62]. The 2A current limiter and the 0.5A fuse are dedicated respectively for the +3.3VDC and the +24VDC sides.

4.5.4 Main power converter

The main step-down DC/DC converter circuit is presented on the Figure 30. Filtered +24VDC input is converted to +3.3VDC. The main chip, the L5972D is a step-down switching regulator, with an internal 2A current limit switch and an additional overcurrent and thermal shutdown protection [62].

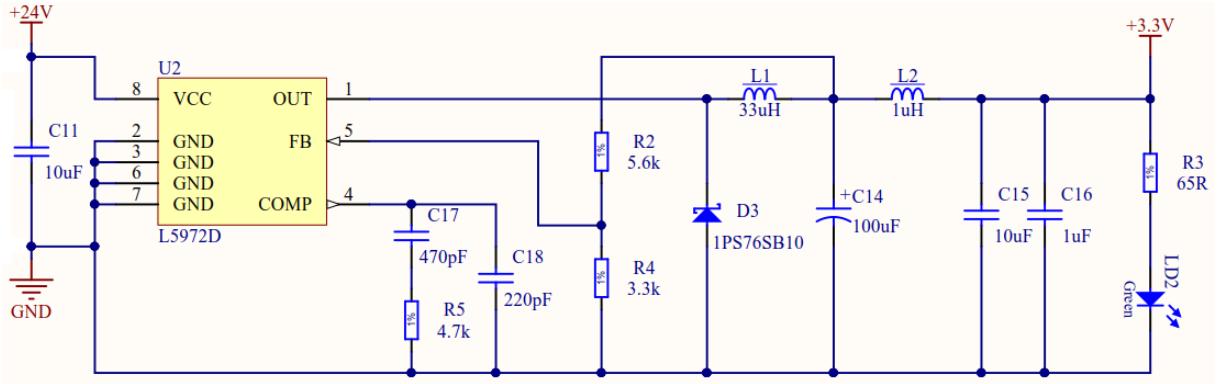


Figure 30. The main step-down DC/DC converter circuit [100].

The R2 and R4 resistors are set for +3.3VDC output and their tolerance is 1% (generally, these resistor tolerances should be as high as possible) [62].

The main role of a step-down DC/DC converter is to decrease the input voltage to required level, along with the lowest power losses. The DC/DC converter performance depends on total consumption current, switching frequency, tolerance of components and multiple other parameters. Usually, standard DC/DC achieve from 80 up to 90% of the performance (significantly more than the linear regulators) [59] [63].

4.5.5 RTC battery and power Backup

The auxiliary RTC power source is a standard +3VDC battery acting like a power back up, when the main power supply is disabled. The STM32 RTC shows approx. $35\mu\text{A}$ of a current consumption during standard operation. On the other hand, the power consumption is approx. $2\mu\text{A}$ in the battery back-up mode. For +3VDC coming from the battery, the RTC data will be held for approx. 12 months, when the main power supply is absent [54] [42].

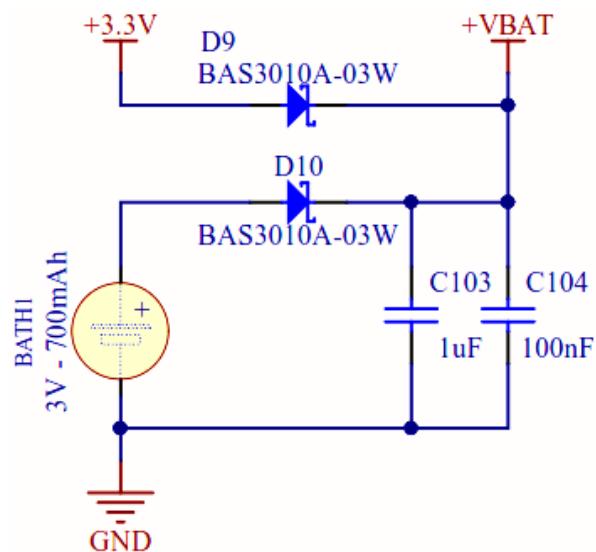


Figure 31. The RTC power supply circuit [100].

Figure 31 shows the RTC power supply circuit. Two Schottky diodes have been used, due to lower drop voltage than present in standard diodes. Moreover, these diodes are used to separate two power sources of the RTC. When the main power supply is present, the RTC is supplied from the main DC/DC converter. In case of the main power absence, only the battery will provide power to the module. Parallel connection of Schottky diodes prevents a battery from immediate discharge. Additionally, these diodes ensure that the battery supplies only and exclusively the RTC.

4.5.6 TFT LCD Backlight

The backlight of the TFT LCD consists of two chains of six diodes connected in series. Diodes emit white light that increases data visibility of the LCD matrix. The backlight diodes require +18VDC up to +20VDC input power supply. Moreover, the current of these diodes cannot be higher than 20mA [15].

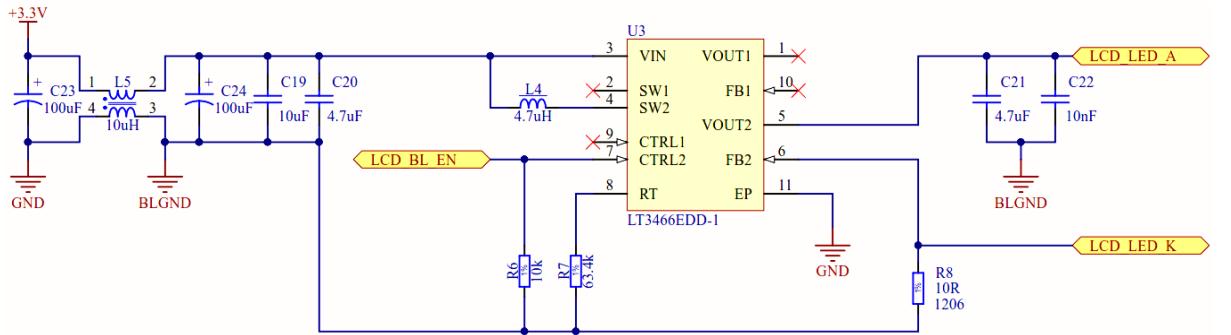


Figure 32. The boost DC/DC converter circuit of the TFT LCD backlight [100].

The boost DC/DC converter circuit of the TFT LCD backlight is presented on the Figure 32. The main chip of the circuit is the LT3466EDD being the step-up switching power regulator with built-in Schottky diode, dedicated for driving several LEDs. The LED dimming is adjustable with the use of the internal adjustable current regulator. The LT3466EDD can drive two LED strings independently. A significant advantage of the chip is that, the Schottky diode is placed inside the chip. Only one external inductor and two resistors are required to run the converter [64].

The R8 resistor sets the maximum output current of 10mA for the 10-Ohm resistance. The maximum input current is approx. 150mA for +3.3VDC side (for assumed +19VDC and 10mA on the chip's output). The R8 power has to be at least 0.5W and its tolerance should be relatively high. The R7 (63.4kOhm) is used to set the switching frequency of step-up regulator chip, which is approx. 500 kHz (it guarantees high performance) [64].

The TFT backlight power circuit contains a standard LC input filter (chapter 4.5.3). Implicitly, the DC/DC converter is disabled and it is controlled directly from the STM32 (“LCD_BL_EN” label on the Figure 32).

4.6 USB OTG

The USB is a differential communication interface dedicated for PC peripheral devices connection (such as keyboards, computer mice, cameras, printers or generally portable devices), that share a common data bus. The newer USB versions feature back compatibility with the older versions [65] [66].

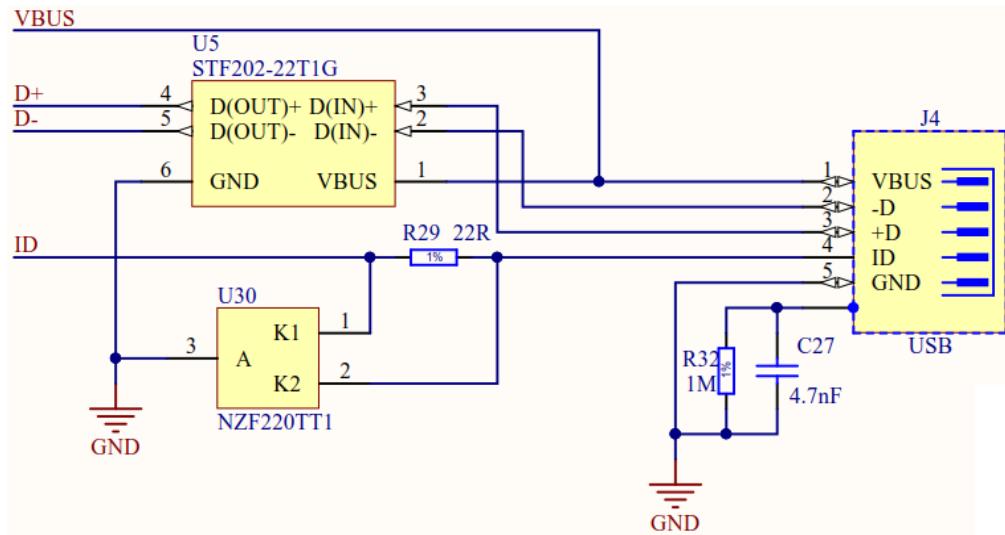


Figure 33. The ESD and EMI circuit of the USB OTG [100].

The STM32 USB OTG controller can operate in the USB OTG FS mode (Figure 33). It can operate as a host (master) or a device (slave) on the same bus. This module contains a self-hardware PHY (Physical Layer) driver [42] [14].

The USB OTG uses three control signal lines, such as the D+ (Data+), D- (Data-), ID, and two power lines like the VBUS and GND. The D+ and D- signals are used to transfer data bi-directionally. Each data is coded as a difference of the voltages between data signals. Moreover, the third logic line (ID) is used as a distinguisher of operational modes. When the ID signal is connected to the ground, the device will operate as a host (master), whereas when floating, a device is defined as a slave [55] [42].

The most meaningful advantage of the differential interface is high resistance against noises and low level of crosstalk (permeability of signals). The USB can correctly operate in high-level noise environment, because inducted noise voltages are the same on the both data

lines (the difference of voltage is constant). Nevertheless, the differential interface requires additional ESD and EMI (Electromagnetic Interference) protection modules for data and power lines [66].

On the USB socket shield, noises or ESDs can be inducted, therefor the USB protection circuit contains an additional RC filter, dedicated exclusively for suppressing noises or ESDs generated on the shield (R32 and C27 on the Figure 33) [67] [68].

The STF202-22 and the NZF220TT1 chips provide the ESD and EMI protections within a line terminations for the USB signals [67] [68].

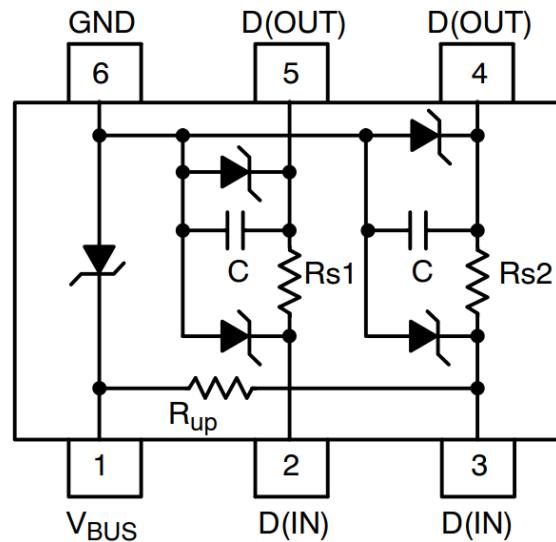


Figure 34. Circuit description of the STF202-22 chip [67].

The STF202-22 contains RC filters with line termination resistors and few TVS diodes (Figure 34). This chip does not provide the protection for the identification port (ID) [67]. For this reason, the NZF220TT1 has been added, which internal structure is shown in the Figure 35. The operational principles and structure of this chip is similar to the STF202-22 [68].

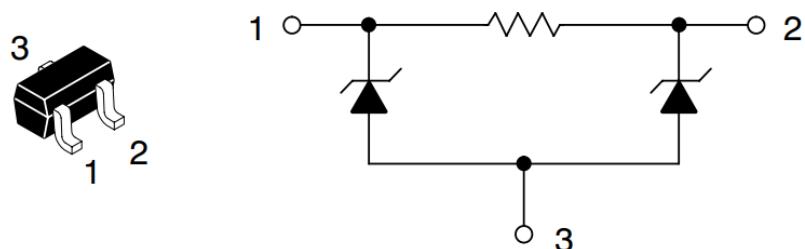


Figure 35. Circuit description of the NZF220TT1 [68].

The STF202-22 ties the D+ to the VBUS through a pull-up resistor on the STM32 USB controller side. Based on the datasheet [67], when the D+ line of the STM32 side is pulled-up to the positive voltage, the USB controller will operate as a host in fast-speed mode. If the USB filter chip is connected inversely to the circuit, the STM32 USB controller will operate as device in low-speed mode (data transfer speed is only up to 1.5Mbits) [66] [55] [67].

In order to detect if passive device (without independent, self-internal power supply) is on the bus, the capacitance difference into the bus is measured. After connecting passive device, the electrical parameters (capacitance) of the USB bus are changed, and the USB controller recognises it as a connected external device. The protection chips are equipped with RC filters, which increase the total capacitance of the bus, that is being used to detect the device by the external host (PCs) [66] [55] [67].

4.7 STM32 programmer

The SWD (Serial Wire Debug) is programmer interface of the STM32 microcontrollers [42]. The scheme of the SWD connections is shown in the Figure 36. The scheme is the part of the STM32F429 Discovery evaluation board (presented on the Figure 17) [44].

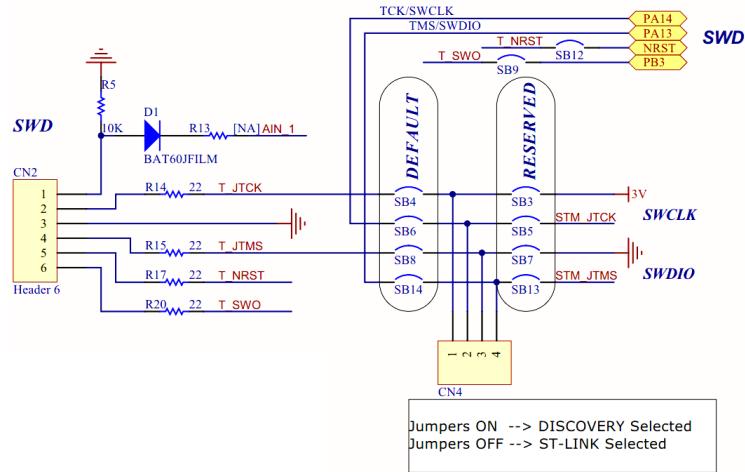


Figure 36. The SWD programmer circuit of the STM32F429 Discovery board [44].

The scheme of the programmer circuit is equipped with the CN4 jumper, which provides opportunity to select target microcontroller flash (an internal microcontroller of an evaluation board or an external ST chip) [44]. The ST evaluation boards allow flashing of all microcontrollers from ST lines, without using any special and expensive programmers such as the JTAG (name of the Joint Test Action, the group that developed this standard of testing integrated devices) [69] [70]. Furthermore, the majority of microcontrollers with the ARM core can be flashed through the SWD standard, designed by the ARM holding [71].

The SWD interface consists of signals such as:

- ***SWDIO*** – Serial Wire Debug Input Output,
 - ***SWCLK*** – Serial Wire Clock,
 - ***SWO*** – Serial Wire Output,
 - ***RST*** – Reset,
 - ***GND, Vcc*** – voltage reference lines.

The SWD interface allows performing full debugging process. The most significant advantage of the SWD, in contrary to the JTAG, is that it uses only four signals (the JTAG interface uses several different signals). In order to use all options of the debugger, additional signal line (SWO) has to be used [70] [71].

4.8 IO port buffers

The Figure 37 presents the SPI buffers, used to achieve a logical separation of the SPIs lines between the STM32 and external cards. Moreover, identification ports of external cards are also protected. These buffers provide protection of a microcontroller IO ports, from general damages. On top of that, these chips contain the ESD and the EMI protections and their operational frequency is up to 150MHz [72] [73].

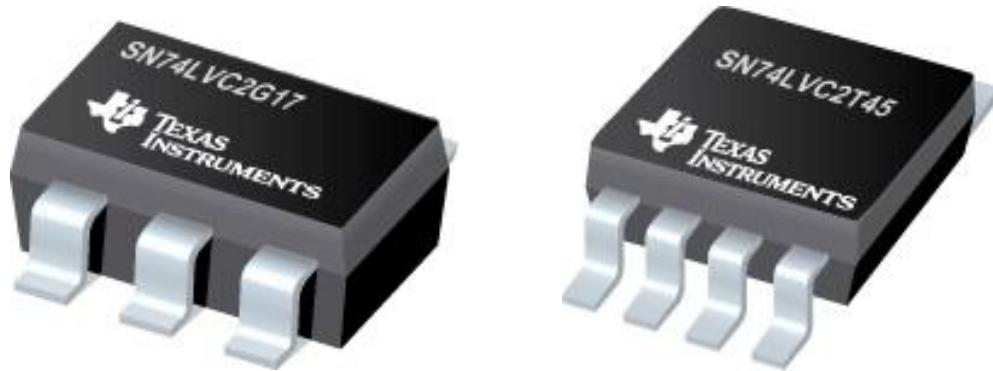


Figure 37. The SN74LVC2G17 (left) and SN74LVC2T45 (right) buffers [72] [73].

The SN74LVC2T45DCUT (on the right of the Figure 37, called transmitter) is the two-directional buffer, used to protect the SPI controller of a communication card only. The direction of transmitting data can be changed through the suitable polarisation of the “DIR” port. The SN74LVC2G17DBVR (on the left of the Figure 37) is one-directional buffer, dedicated for four SPIs controllers [72] [73].

High load or ESD effects can cause physical damages to the main microcontroller, which is being the most expensive chip on the motherboard. In order to protect advanced and expensive chips, it is feasible to sacrifice cheaper buffers, which can be easily exchanged.

The main purposes of using buffers are:

- ***ESD and EMI protection*** – safety features to protect IO ports against electrostatic discharges and interferences from electromagnetic fields [74].
- ***Sacrifice*** – buffers can protect advanced chips through self-destroying (their cost is negligible), which contributes achieving physical isolation [74].
- ***Driving low impedance circuits*** – buffers are highly recommended for high impedance signals (like IO ports of the STM32). They are used to drive low impedance signals (like DC motors, LEDs) [74].
- ***Logical isolation*** – majority of buffers have a tristate logic gate with standard boolean and additional high impedance states (it can be considered as “disconnected devices”, despite physical connection). Tristate logic buffers are used for providing logic insulation of multiple devices signals, shared the same buses or signals [74].
- ***Physical isolation*** – buffers can eliminate interferences between devices, decrease noise level for sensitive devices and improve their communication parameters [74].
- ***Translation of voltage levels*** – buffers allow to connect devices that are supplied from different voltages with each other [74].
- ***Digitalisation and clean-up of transient signals*** – buffers can contain a hysteresis unit provided to a “clean” signal with sharp edges [74].

4.9 External clock sources

The STM32 clock sources are 8 MHz (used to generate multiple high-speed signals of core) and 32.768 kHz (only for the RTC) crystal oscillators (Figure 38).

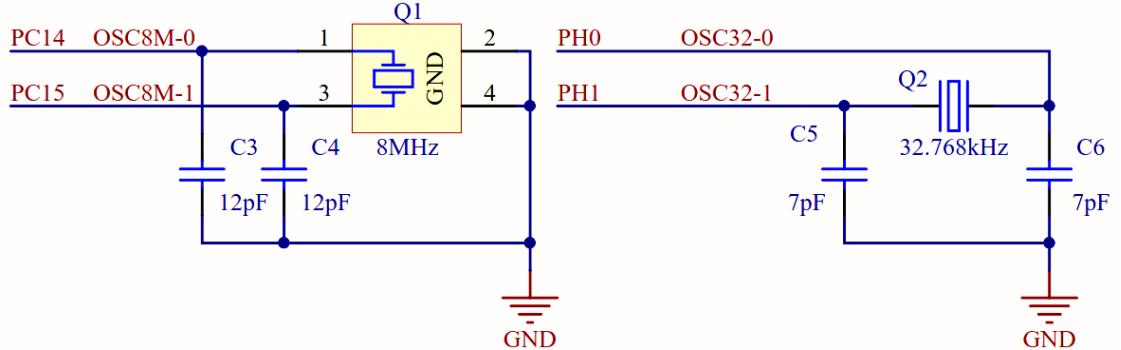


Figure 38. External crystal oscillator circuit [100].

Crystal oscillators are based on the mechanical resonance of a vibrating piezoelectric material. Each crystal oscillator is connected to two load capacitors in parallel. The frequency of crystal oscillators depends on the values of the load capacitors and they are used as resonant frequency trimmers [75] [54] [76].

The load capacitors of used crystals are 12pF and 7pF, respectively for the 8MHz and 32.768 kHz crystals (recommended by manufacturers [77] [78]). Moreover, the placement of these capacitors has to be as close as possible to the crystal package.

The quality of clock signals highly depends on an ambient temperature (crystal cannot be placed close to heat elements). Moreover, the tolerance of crystals is defined by the ppm unit (parts per million) that indicates how much a crystal frequency can be deviated from the nominal value (chosen crystals tolerances are equal to the 30ppm) [77] [78] [76].

4.10 User buttons

Figure 39 presents the circuit of the user button with the MAX6816. The chip is operated as the debouncer, dedicated to handle push-pull buttons [79]. Standard buttons cannot generate “clean” digital output signal. This it is caused by mechanical oscillations occurring on the button junction (during pushing). Described effect must be eliminated as it can translate to the errors in logic states. Figure 40 displays the output signal of the example button during switching on [80] [81].

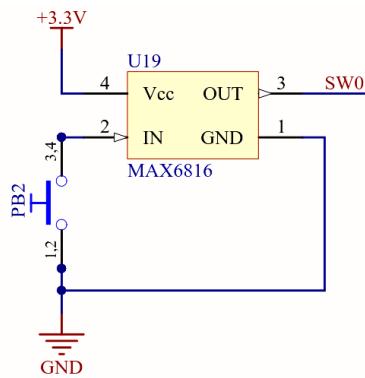


Figure 39. The Circuit of the user button debouncer [100].

The bouncing effect can generate software faults due to incorrect detection of logic states. Moreover, high voltage peaks can contribute to physical damages of IO ports. In order to digitise and clean-up a button signal, the MAX6816 chip has been used. Majority of debouncers contains additional ESD and EMI protection, used to eliminate high voltage peaks on the button junction [80] [81].

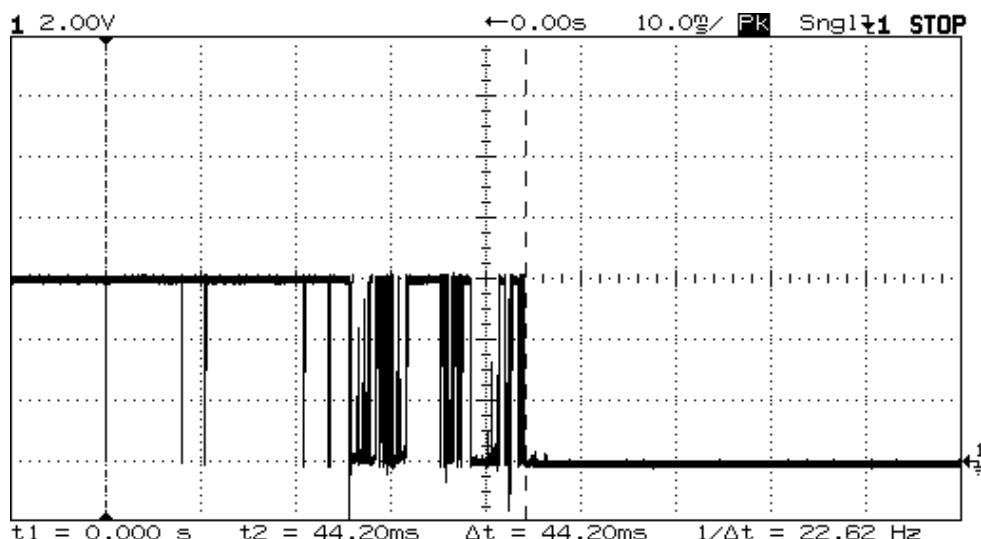


Figure 40. Switch bounce effect of buttons [81].

The main goal of the debouncer chip is a full digitalisation of the button output signal. This signal must be free of oscillations or noises. Dedicated chips for reducing bouncing effects are an expensive solution, but they guarantee that the output signal will be cleaned and digitalised. Usually, debouncer chips are based on a counter and simple logic gates, and internal structure of the chosen debouncer chip is presented in the technical documentation [79].

Debouncing effect can be solved with the use of other methods:

- **RC input filter** – standard low band pass filter set for relatively low cut-off frequency. Its suppressing properties are not sufficient to extract clean digital signal (noises or variations signals are only suppressed, not eliminated) [80].
- **RC filter with logic gates** – improved a RC filter uses logic gates and an oscillation unit. Suppressing properties are better than standard RC circuit, but still are not sufficient [80].
- **Software debouncers** – the cheapest solution, without the use of any hardware components. Debouncing is performed with the use of software implementation with the MAX6816 principles [80].

All listed RC filter setups cannot eliminate bouncing effect itself. In order to achieve fully digitalised output signal, additional software implementation must be designed. For this project, the MAX6816 has been used for reducing the software implementation for the user buttons.

5 Firmware

The following chapter presents descriptions of the software implementation for the VGC controller. The chapter contains brief descriptions of used algorithms with graphs, hardware drivers, surveys and reviews of solutions, software development tools, memories management and protocols. Moreover, the chapter presents a short overview of application's software. The VGC software has been designed and implemented by the author of this master thesis. For the STM32 firmware, the source code contains approx. 12 thousand lines and 1.5 thousands of a PC C++ application (taking into account all the comments, without the ST peripheral drivers' libraries).

5.1 Firmware requirements and guidelines

5.1.1 Software requirements:

- **Portability** – the software implementation must have a feature of easy migration of the program code from the STM32 platform to other microcontrollers' platforms.
- **Maintainability** – the architecture of a software should provide options to easily upgrade a program with new features. Moreover, the program code should be readable for other programmers. On the top of that, any changes in a code should be effortless.
- **Readability** – code must be written accordingly to the clean-code rules [82].
- **Open source IDEs and development tools** – have to be free of charges.
- **Performance** – programming language must provide as high performance as possible to program's size. It is crucial for highly limited embedded platform resources.
- **Development time** – programming languages and tools should provide opportunity to create/built the project with as short time as possible.
- **Safety features** – code must have safety mechanisms implemented provided to prevent the system from unauthorized access to a device.

5.1.2 Software guidelines:

- **The C programming language** – dedicated to design the firmware implementation of the main microcontroller. **The C++ programming languages** – dedicated for PC applications.
- **Two main software layers** – it is assumed that the firmware has to be divided into two parts: the hardware drivers layer and the application layer (in order to increase the portability of the firmware).
- **DMA and DMA2D controller usage** – mandatory part of this project, provided to increase the firmware implementation performance.
- **Open source and free libraries** – the ST manufacturer provides multiple free and open source libraries of hardware drivers that have to be included into the programs' code.

5.2 Hardware drivers layer

Following chapter presents a short description of the hardware drivers layer of the STM32 controllers and their peripheral devices, with suitable graphical block diagrams.

5.2.1 General block diagram.

The architecture of the hardware drivers' layer initialisation is shown in Figure 41. What is crucial to highlight, the STM32 controllers must be configured accordingly to the plan, with the hierarchy and priorities (they cannot be randomly configured). Suitable configuration of the hardware drivers contributes in achieving the reliable system.

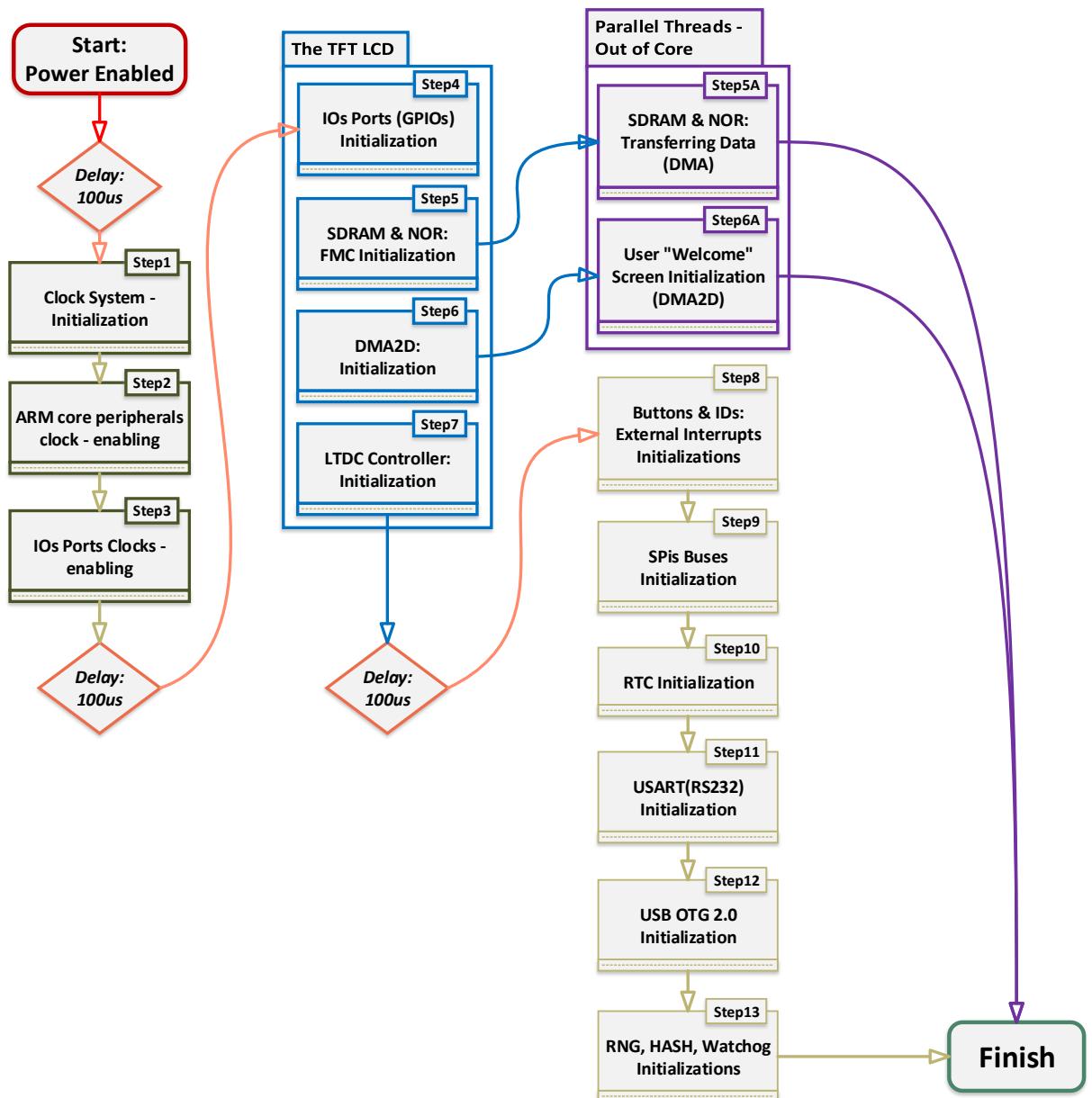


Figure 41. The hardware drivers' architecture [100].

When the main power supply is turned on, the STM32 awaits approx. 100us to achieve a stable level of voltages and to coupling capacitors charging. The input delay highly reduces probability of the STM32 going into the reset state.

In the first step, the clocks system of the STM32 must be initialised, before running any internal controllers. In any other case, the controllers will not operate correctly. The configuration of the clocks system usually takes some time, than any other tasks should not be performed. A configured clocks system enables configuration of the TFT LCD modules and other STM32 controllers. Afterwards, the DMA2D and DMA channels are started to transfer data, in parallel other controllers are configured (“Parallel Threads” in the Figure 41). When the controller is initialising, the slider status of the “welcome” screen is loaded (proportionally to how many controllers are already configured).

5.2.2 Clocks system configuration

The clocks system of the STM32F429xx microcontrollers (Figure 42) with the maximum frequency of the ARM (180MHz). The clock system is split into:

- **Main PLL** (Phase Loop Latency),
- **The RTC** clock multiplier,
- **PLLSAI and PLLI2S** (audio signal conditioning loops - not using in this project) [75].

The STM32 clocks system can start the operation with an external source (“HSE” in the Figure 42) or the internal 16MHz oscillator (HSI) (it guarantees that the system will always run) [42] [75].

The PLL frequency multiplier is used to increase the source frequency, even a couple hundred times. The output frequency depends on the PLL parameter values (divider and multiplier factors such as “Q”, “P”, “M” and “N”) configured accordingly to the PLL equations (values cannot be at will) [42] [75]. Moreover, the value of an external clocks source cannot be optional and its value must be in a range of 4MHz up to 26MHz. In another case, the PLL will run from the HSI. The PLL equation coefficients and configurations procedure are described in the technical documentation [42] [75].

Generated output frequency of the PLL loop are provided to all peripherals of the ARM core through clock buses such as:

- **SYSCLK** (output PLL clock for ARM Cortex core, core system timers and Ethernet),
- **AHB1** and **AHB2** (buses for DMAs, DMA2D and FMC),
- **APB1** and **APB2** (core peripherals buses),
- **48MHz** clock for the USB OTG 2.0 controller [42].

The majority of clock busses contain internal prescalers, used to decrease the PLL frequency. For this project, the set output frequency of the STM32 is the 168MHz. The maximum clock is 180MHz.

The USB OTG 2.0 controller requires precisely 48MHz. This is not achievable from the 180MHz output (limitations of the PLL equation factors). Respective prescalers busses are set to the minimal values generating the highest possible frequencies [42] [75].

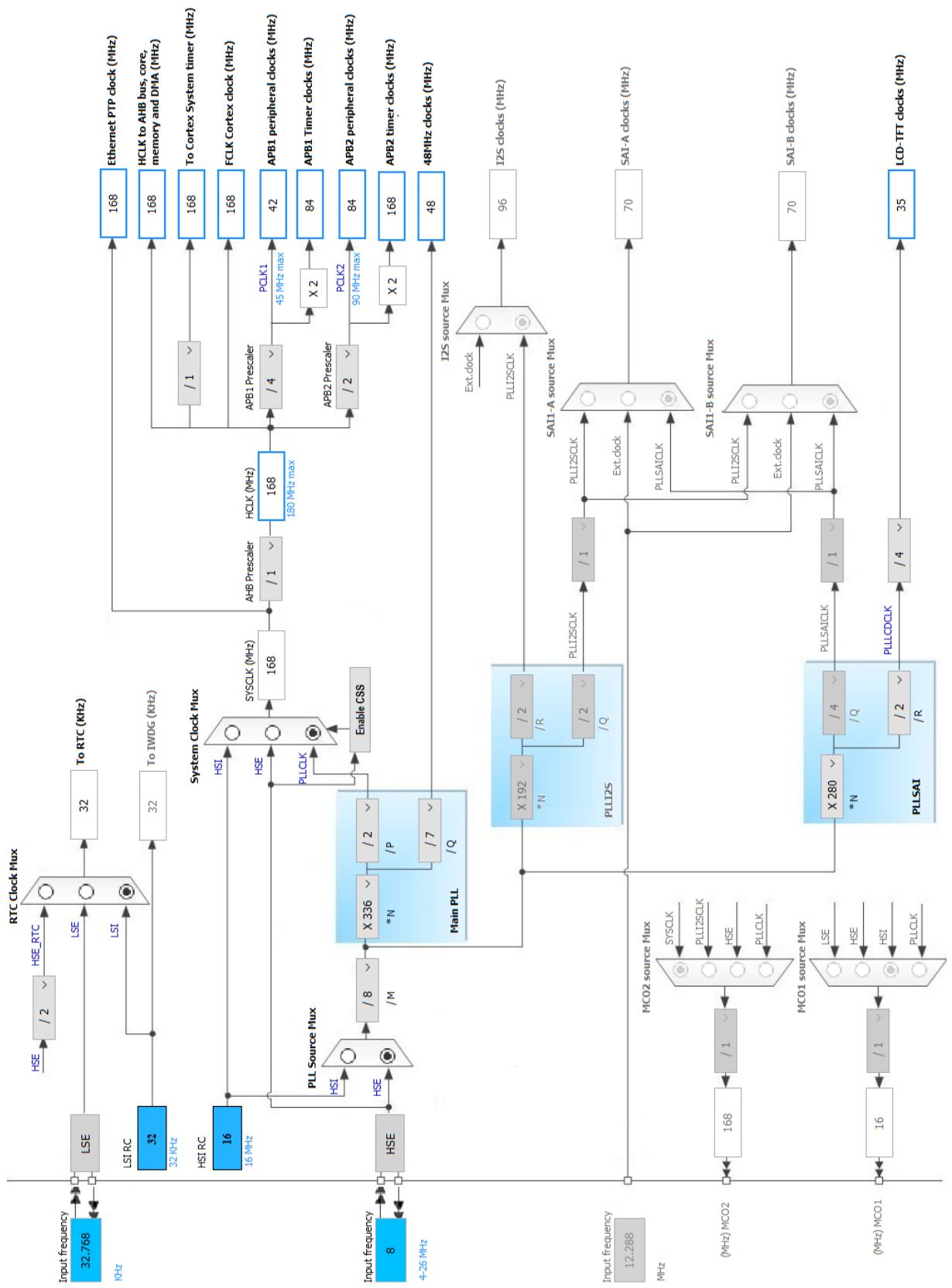


Figure 42. The STM32F429xx clock system [83].

5.2.3 DMA & DMA2D

The STM32F429xx microcontrollers contain two DMA controllers. Their block diagram and their dependencies with the STM32 internal modules are presented in the Figure 43.

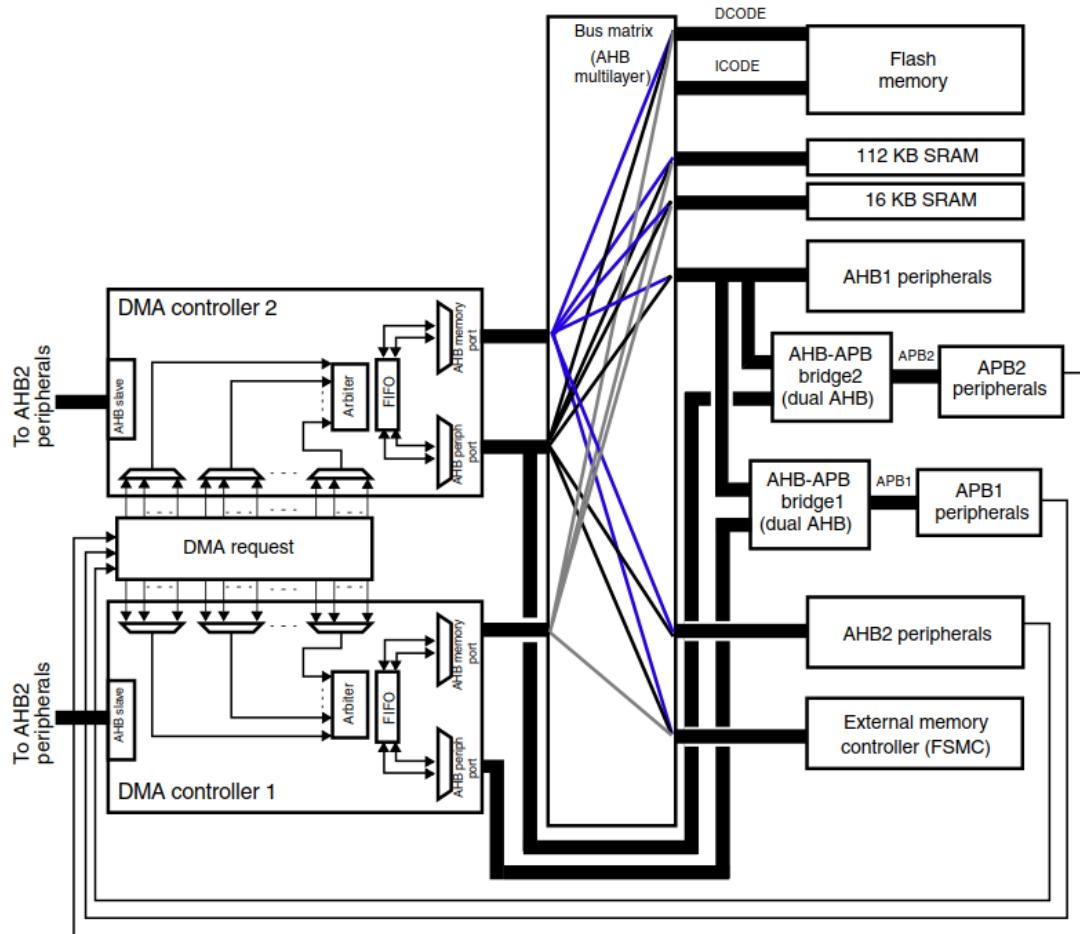


Figure 43. The block diagram of the STM32 DMA controllers [42].

The DMA controllers have the access to almost every internal module of the STM32 through the main communication buses such as the AHB1, AHB2, APB1, APB2 and core buses. The biggest advantage of DMAs is, that they can independently operate, out of core and without the use of the core computing power. Moreover, the DMA controllers enable transferring data according to the circular buffer order, supported by the hardware (useful for managing TFT LCD and SPIs data) [42].

The DMAs can operate in three main modes of transferring data:

- peripheral device to memory,
- memory to peripheral device,
- memory to memory [42].

The DMAs are built of eight independent operational streams. Every of them have the multiplexer with eight channels. Each stream is associated with the only one request from a memory or peripherals that is selected by the suitable multiplexer channel. It is not possible to connect peripherals or memories to an optional DMA stream. Each DMA contains an arbiter module that manages streams compliance with their priorities (for the STM32, it is possible to set four levels of priorities). Moreover, each stream has an independent FIFO (First Input First Output) buffer that can be used, when the DMA operates in non-atomic mode [42].

Table 1. The DMA1 requests mapping of STM32 [42].

Peripheral requests	Stream 0	Stream 1	Stream 2	Stream 3	Stream 4	Stream 5	Stream 6	Stream 7
Channel 0	SPI3_RX		SPI3_RX	SPI2_RX	SPI2_TX	SPI3_TX		SPI3_TX
Channel 1	I2C1_RX		TIM7_UP		TIM7_UP	I2C1_RX	I2C1_TX	I2C1_TX
Channel 2	TIM4_CH1		I2S3_EXT_RX	TIM4_CH2	I2S2_EXT_TX	I2S3_EXT_TX	TIM4_UP	TIM4_CH3
Channel 3	I2S3_EXT_RX	TIM2_UP TIM2_CH3	I2C3_RX	I2S2_EXT_RX	I2C3_TX	TIM2_CH1	TIM2_CH2 TIM2_CH4	TIM2_UP TIM2_CH4
Channel 4	UART5_RX	USART3_RX	UART4_RX	USART3_TX	UART4_TX	USART2_RX	USART2_TX	UART5_TX
Channel 5	UART8_TX ⁽¹⁾	UART7_TX ⁽¹⁾	TIM3_CH4 TIM3_UP	UART7_RX ⁽¹⁾	TIM3_CH1 TIM3_TRIG	TIM3_CH2	UART8_RX ⁽¹⁾	TIM3_CH3
Channel 6	TIM5_CH3 TIM5_UP	TIM5_CH4 TIM5_TRIG	TIM5_CH1	TIM5_CH4 TIM5_TRIG	TIM5_CH2		TIM5_UP	
Channel 7		TIM6_UP	I2C2_RX	I2C2_RX	USART3_TX	DAC1	DAC2	I2C2_TX

Table 2. The DMA2 requests mapping of STM32 [42].

Peripheral requests	Stream 0	Stream 1	Stream 2	Stream 3	Stream 4	Stream 5	Stream 6	Stream 7
Channel 0	ADC1	SAI1_A ⁽¹⁾	TIM8_CH1 TIM8_CH2 TIM8_CH3	SAI1_A ⁽¹⁾	ADC1	SAI1_B ⁽¹⁾	TIM1_CH1 TIM1_CH2 TIM1_CH3	
Channel 1		DCMI	ADC2	ADC2	SAI1_B ⁽¹⁾	SPI6_TX ⁽¹⁾	SPI6_RX ⁽¹⁾	DCMI
Channel 2	ADC3	ADC3		SPI5_RX ⁽¹⁾	SPI5_TX ⁽¹⁾	CRYP_OUT	CRYP_IN	HASH_IN
Channel 3	SPI1_RX		SPI1_RX	SPI1_TX		SPI1_TX		
Channel 4	SPI4_RX ⁽¹⁾	SPI4_TX ⁽¹⁾	USART1_RX	SDIO		USART1_RX	SDIO	USART1_TX
Channel 5		USART6_RX	USART6_RX	SPI4_RX ⁽¹⁾	SPI4_TX ⁽¹⁾		USART6_TX	USART6_TX
Channel 6	TIM1_TRIG	TIM1_CH1	TIM1_CH2	TIM1_CH1	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3	
Channel 7		TIM8_UP	TIM8_CH1	TIM8_CH2	TIM8_CH3	SPI5_RX ⁽¹⁾	SPI5_TX ⁽¹⁾	TIM8_CH4 TIM8_TRIG TIM8_COM

The majority of the core peripherals can be connected to a certain DMA stream. Table 1 and the Table 2 present all available DMA stream configurations versus multiplexer channels. The red frames (into the Table 1 and Table 2) indicate the chosen DMAs streams

of the SPIs and the USART6 (auxiliary serial port) controllers. The SPI1 does not have a dedicated DMA stream for the transmitter, because this stream is used by the SPI5 receiver. This solution is not preferred, but it does not significantly decrease the performance [42].

Figure 44 displays the block diagram of the DMA2D Chrom-Art Accelerator of the STM32. It is a specialized DMA controller, used to generate advanced graphical effects. Similarly to DMAs, the DMA2D is the independently operating module that does not need the core computing power [42].

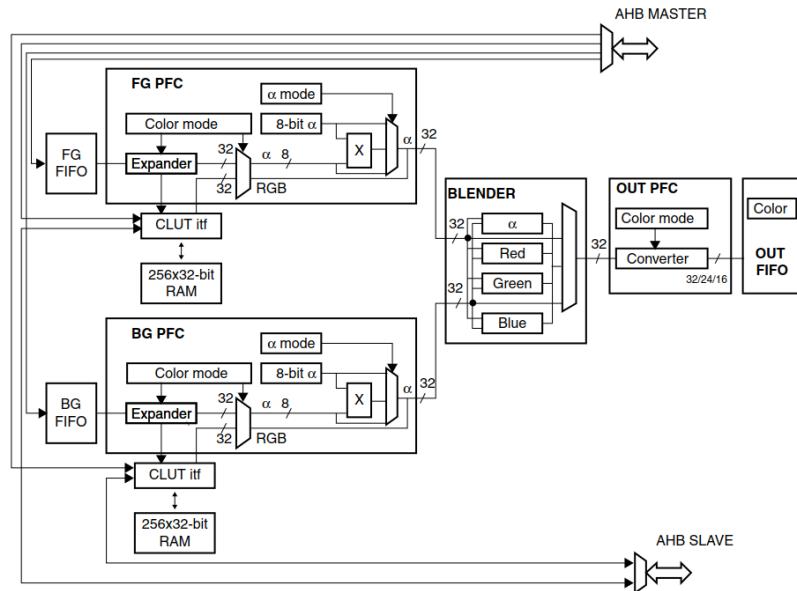


Figure 44. The STM32 DMA2D block diagram [42].

This module can easily fill destination LCD banks with colours. Moreover, it can copy and transfer graphical data with pixel conversion and blending/dithering different templates with each other and with different colour formats and blending/dithering factors. The DMA2D contains the CLUT (Colour Loop-Up Table) unit, which could be considered as an “internal GRAM” (Graphic RAM) [42].

The DMA2D can transfer data in the four main modes such as:

- from registers to memories,
- from memories to memories,
- from memories to memories with PFC (Pixel Format Conversion),
- from memories to memories with PFC and blending [42].

The DMA2D contains the FG (Foreground) FIFO and BG (Background) FIFO buffers (shown in the Figure 44). FIFOs fetch the pixels accordingly to the formats of colours, defined by the PFC unit. All pixels are converted in FG and BG units of the PFC accordingly to graphical coefficients (such as the Alfa). The converted pixel FIFOs of the FG and BG layers are transferred to the blender unit. Later, the pixels are mixed to each other (accordingly to set blender coefficients). Lastly, all blended pixels are converted to output pixel format by the output PFC unit. All operations are executed by the AHB buses [42].

5.2.4 STM32 Inputs-Outputs Ports Configurations

The GPIO (General Purpose Inputs Outputs) controller is responsible for setting operational properties of each IO port [42].

Each IO port can be configured as follows:

- input floating and pull-up/down,
- analogue input,
- output open-drain with pull-up/down capability,
- output push-pull with pull-up/down capability,
- alternate function open-drain or push-pull with pull-up/down capability [42].

Each IO port is +5VDC tolerant and compatible with the TTL¹⁵ standard (used by older chips). Moreover, each IO port can be connected to the core peripherals through the multiplexer, which allows to configure only one alternate function of peripherals to only one IO port. The multiplexer has sixteen channels, mapped onto alternate functions of peripherals (signed as the AF0 – AF15 in the documentation) [42].

The GPIO configures IO ports, based on their or peripherals operational frequencies. It is possible to set four working speeds such as: low (up to 2MHz), medium (up to 25MHz), high (up to 50MHz) and ultra-high (up to 100MHz). Depending on the working speed, the rising and falling times of signals will be shorter or longer. Besides, each IO port can be configured as an input for an external interrupt (e.g. generated by an external card), connected to one channel of the EXTI (used to serve user buttons signals) [42].

¹⁵ TTL – Transistor Transistor Logic.

For this project, the IOs are usually configured to operate with high and ultra-high speeds, almost for all available configurations (without analogue mode).

5.2.5 RTC - Real Time Clock

The STM32 RTC is used to hold and count actual reference time and date. It must be synchronised with the internal reference time system of the CERN facilities. The RTC contains internal correction modules for the leap years and 30th or 31th days of months. Moreover, its registers are coded in the BCD (Binary Coded Decimal) [42] [54].

The VGC requires the time counting accuracy at least up to one millisecond. The total accuracy of the RTC is usually five ppm. What is important, the RTC has an independent power source that enables to hold values of the RTC registers, during the main power supply absence [42] [54]. The time of the RTC can be used as a time-stamp for recording precise interrupts and critical events. Beside, this stored time is also used to synchronise communications between devices. The RTC (Figure 45) is built of the time and date counter registers, A and B type alarms, wake-up interrupt generating unit, input clock calibration unit, and time stamp and tamper detection modules [42] [54].

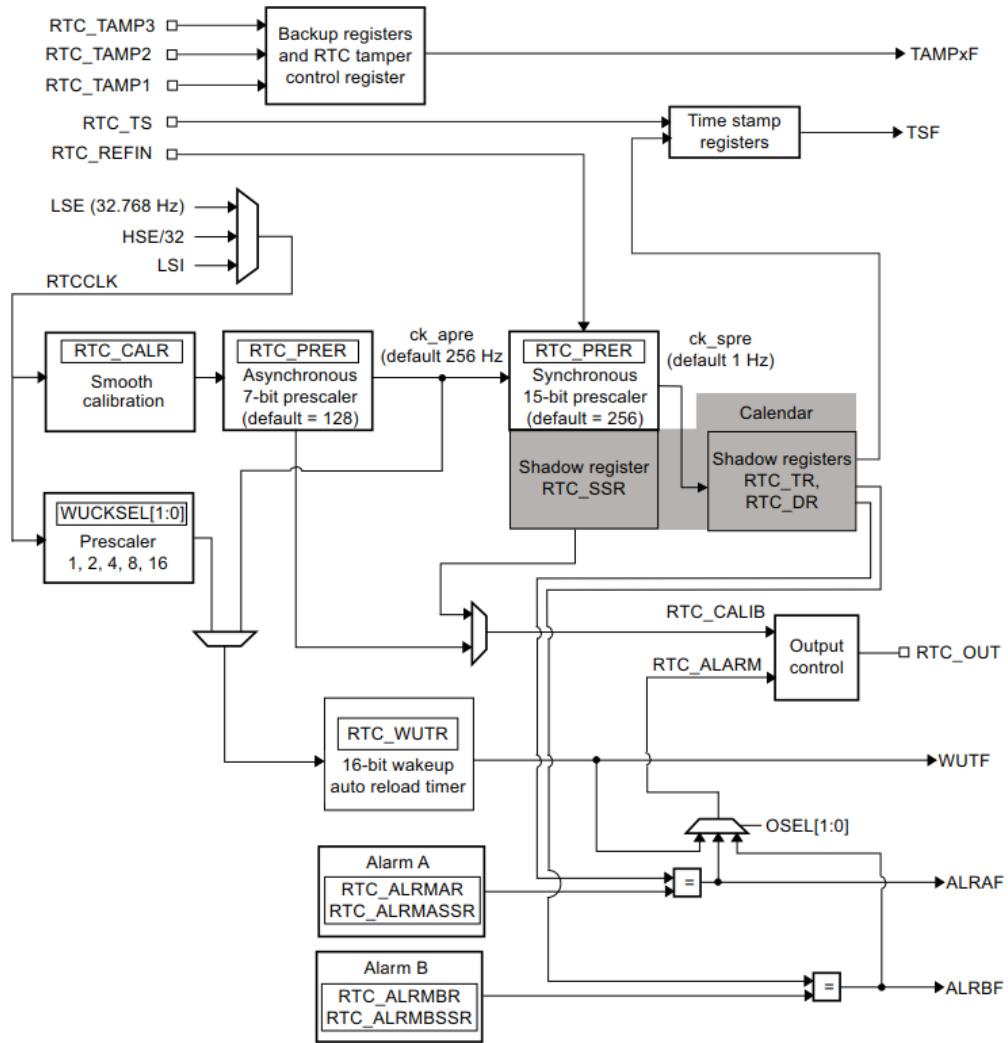


Figure 45. The block diagram of the STM32 RTC [42].

The LSI or LSE input clock is provided to the fifteenth stages clock prescaler. This can be used to generate accurate output signals (“RTC_OUT” in the Figure 45). Frequencies of these signals can be 1Hz or 512Hz. Furthermore, the main power supply (50Hz or 60Hz) as an external reference clock can be connected (the “RTC_REFIN” port in the Figure 45). The build-in programmable alarms (A and B types) provide opportunity to generate accurate events/interrupts at time [42] [54].

The module is built of the shadow registers (Figure 45), used to ensure the consistency between read-out values of registers. During the reading the data and the register values could be changed. For this situation, the shadow registers mechanism is used to keep the coherence of reading data [42] [54].

5.2.6 Interrupt configurations

In embedded systems, the interrupt is a signal generated by hardware or software layers. This signal is an indicator of some occurred events, which should be executed as soon as possible. The interrupts are usually called asynchronously (it can occur at will time) [84].

```

g_pfnVectors:
.word _estack
.word Reset_Handler
.word NMI_Handler
.word HardFault_Handler
.word MemManage_Handler
.word BusFault_Handler
.word UsageFault_Handler
.word 0
.word 0
.word 0
.word 0
.word SVC_Handler
.word DebugMon_Handler
.word 0
.word PendSV_Handler
.word SysTick_Handler

/* External Interrupts */
.word WWDG_IRQHandler /* Window WatchDog */
.word PVD_IRQHandler /* PVD through EXTI Line detection */
.word TAMP_STAMP_IRQHandler /* Tamper and TimeStamps through the EXTI line */
.word RTC_WKUP_IRQHandler /* RTC Wakeup through the EXTI line */
.word FLASH_IRQHandler /* FLASH */
.word RCC_IRQHandler /* RCC */
.word EXTI0_IRQHandler /* EXTI Line0 */
.word EXTI1_IRQHandler /* EXTI Line1 */
.word EXTI2_IRQHandler /* EXTI Line2 */
.word EXTI3_IRQHandler /* EXTI Line3 */
.word EXTI4_IRQHandler /* EXTI Line4 */
.word DMA1_Stream0_IRQHandler /* DMA1 Stream 0 */
.word DMA1_Stream1_IRQHandler /* DMA1 Stream 1 */
.word DMA1_Stream2_IRQHandler /* DMA1 Stream 2 */
.word DMA1_Stream3_IRQHandler /* DMA1 Stream 3 */
.word DMA1_Stream4_IRQHandler /* DMA1 Stream 4 */
.word DMA1_Stream5_IRQHandler /* DMA1 Stream 5 */
.word DMA1_Stream6_IRQHandler /* DMA1 Stream 6 */
.word ADC_IRQHandler /* ADC1, ADC2 and ADC3s */
.word CAN1_TX_IRQHandler /* CAN1 TX */
.word CAN1_RX0_IRQHandler /* CAN1 RX0 */
.word CAN1_RX1_IRQHandler /* CAN1 RX1 */
.word CAN1_SCE_IRQHandler /* CAN1 SCE */
.word EXTI9_5_IRQHandler /* External Line[9:5]s */
.word TIM1_BRK_TIM9_IRQHandler /* TIM1 Break and TIM9 */
.word TIM1_UP_TIM10_IRQHandler /* TIM1 Update and TIM10 */
.word TIM1_TRG_COM_TIM11_IRQHandler /* TIM1 Trigger and Commutation and TIM11 */
.word TIM1_CC_IRQHandler /* TIM1 Capture Compare */
.word TIM2_IRQHandler /* TIM2 */
.word TIM3_IRQHandler /* TIM3 */
.word TIM4_IRQHandler /* TIM4 */
.word I2C1_EV_IRQHandler /* I2C1 Event */
.word I2C1_ER_IRQHandler /* I2C1 Error */
.word I2C2_EV_IRQHandler /* I2C2 Event */
.word I2C2_ER_IRQHandler /* I2C2 Error */
.word SPI1_IRQHandler /* SPI1 */
.word SPI2_IRQHandler /* SPI2 */
.word USART1_IRQHandler /* USART1 */
.word USART2_IRQHandler /* USART2 */
.word USART3_IRQHandler /* USART3 */
.word EXTI15_10_IRQHandler /* External Line[15:10]s */
.word RTC_Alarm_IRQHandler /* RTC Alarm (A and B) through EXTI Line */

```

Figure 46. The list of selected interrupts vectors of the STM32.

Figure 46 presents a list of selected interrupt vectors of the STM32 (in Assembly language). Listed vector are the aliases to program functions, defined by the programmer. These aliases can be considered as a pointer to the function.

When an interrupt is signalled, every non-atomic operation (currently in executing) can be stopped. Majority of the STM32 microcontrollers IDEs (Integrated Development Environments) have interrupt vectors (shown in Figure 46), placed in the “`startup_stm32f4xxxx.s`” file (Atolic TrueStudio program at 2018) [85] [35].

The VGC project uses following interrupts:

- SPIs, USART and I²C buses with DMAs,
- DMAs controllers (in memory-to-memory mode),
- external interrupts of user buttons,
- RTC alarms,
- DMA2D,
- USB OTG 2.0,
- RNG, HASH modules,
- ARM Cortex-M4 and timers system interrupts.

For the STM32 microcontrollers, the interrupt handlers neither take or return parameters (in C, they are a “void” type and take a “void” input argument). An interrupt handler (for e.g. the DMA2D) can be written in C language as:

```
void DMA2D_IRQHandler( void ) {
    if ( DMA2D_GetITStatus( HW_LCD_DMA2D IRQn_TYPE ) == SET ) {
        LCD_SetBusyFlag( RESET );
        DMA2D_ClearFlag( HW_LCD_DMA2D_IT_FLAG );
    }
}
```

The implementation of interrupt handlers can be different depending on microcontroller platforms, but the main idea of interrupt handling is still the same. Interrupt handlers have to be executed in the shortest possible time.

Function cannot be used in the interrupt handlers when it contains software delays, awaits for some events or system flags. This has to be executed outside of interrupt handlers. In other case, a failure of microcontroller system can be occurred (access to the resources of microcontroller, shared by multiple tasks and interrupt handlers, is highly limited) [86].

The STM32 contains the NVIC (Nested Vectors Interrupts Controller), responsible for interrupt management. The NVIC is highly coupled with the ARM core. During the execution, it contributes to lower latencies of interrupt handler. Moreover, interrupts can be handled accordingly to the priorities set in the NVIC registers. The controller provides opportunity to set sixteen priority levels. For the STM32F429xx family, the NVIC and the ARM Cortex-M4 core prioritise and handle the 92 maskable interrupt channels [86] [42] [14].

The STM32 owns the EXTI that can be considered as the sub-module of the NVIC. The EXTI prioritises and handles interrupts generated by external devices on the microcontroller IO ports. Each signal edge of an interrupt signal can trigger the EXTI. This feature is used for handling the user buttons [86] [42] [14].

The interrupts, with higher priorities, are always performed as first. If some lower priority interrupt is in execution, it will be stopped in favour of higher prioritised interrupt (this situation is called as nested interrupts). When the executing process (with higher prioritised interrupt) is finished, the NVIC will come back to previously performed interrupt (with suitable restored content of the handler). For the STM32 microcontrollers, the core interrupts always have the highest priorities [42] [86].

5.2.7 FMC – Flexible Memories Controller

The FMC module is responsible for driving external synchronous, asynchronous, volatile and non-volatile memories such as NOR/NAND Flashes, SDRAM, SRAM, PSRAM and PC Cards. The FMC provides opportunity to drive a couple different memories by only one microcontroller, but only one memory is accessible at the time [42]. The FMC block diagram is shown in the Figure 47.

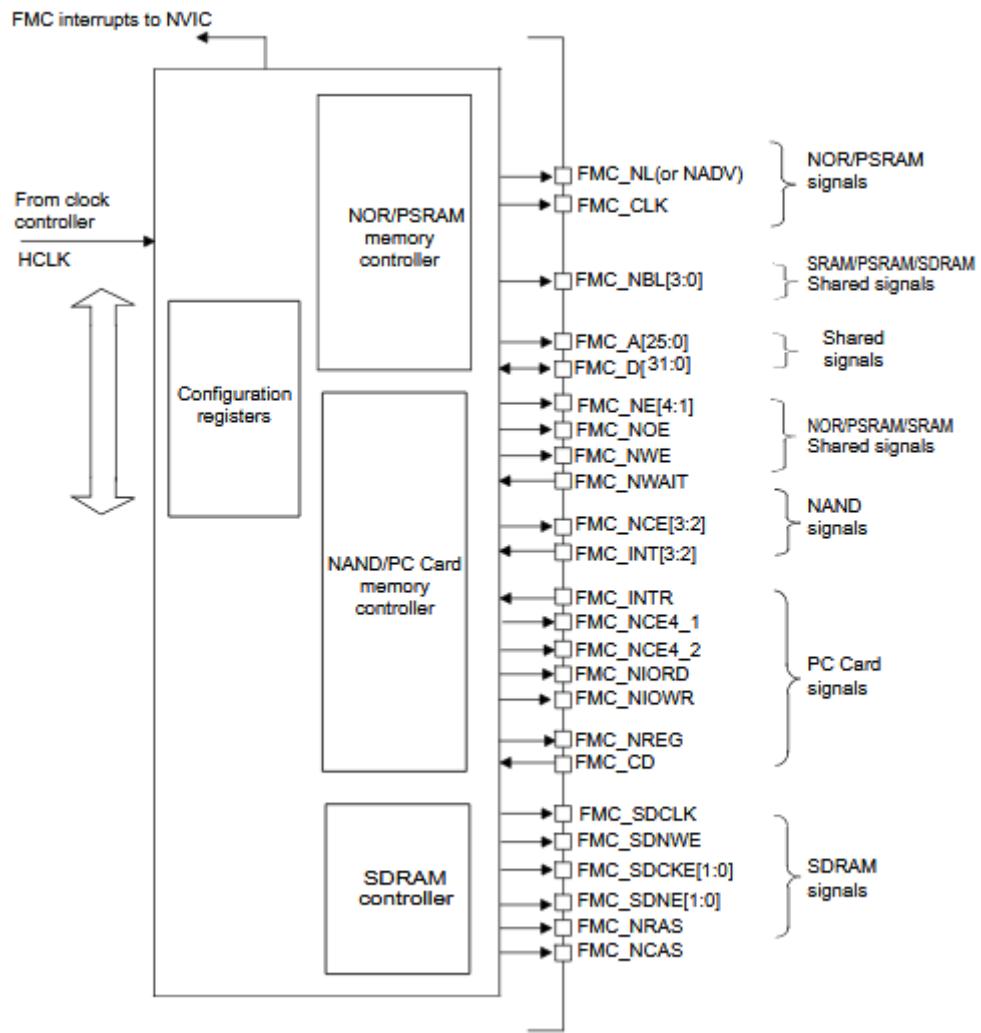


Figure 47. The FMC block diagram [42].

The FMC consists of five main blocks, equipped with several different signals. These signals are used to handle communication between the ARM core and external memory chips. All external memories share data and address busses. External chips can be enabled by using suitable chip-select IO ports. Moreover, memories from the same family (like SRAM/PSRAM or SDRAM) share some common control lines, but only one memory of the family can be driven at the time [42]. This controller is directly connected to the HCLK

clock bus, operated with the maximum available frequency of the microcontroller. Besides, access to external memories is performed through the AHB slave bus. That bus allows the CPU or other core peripherals to manage external memory cells [42].

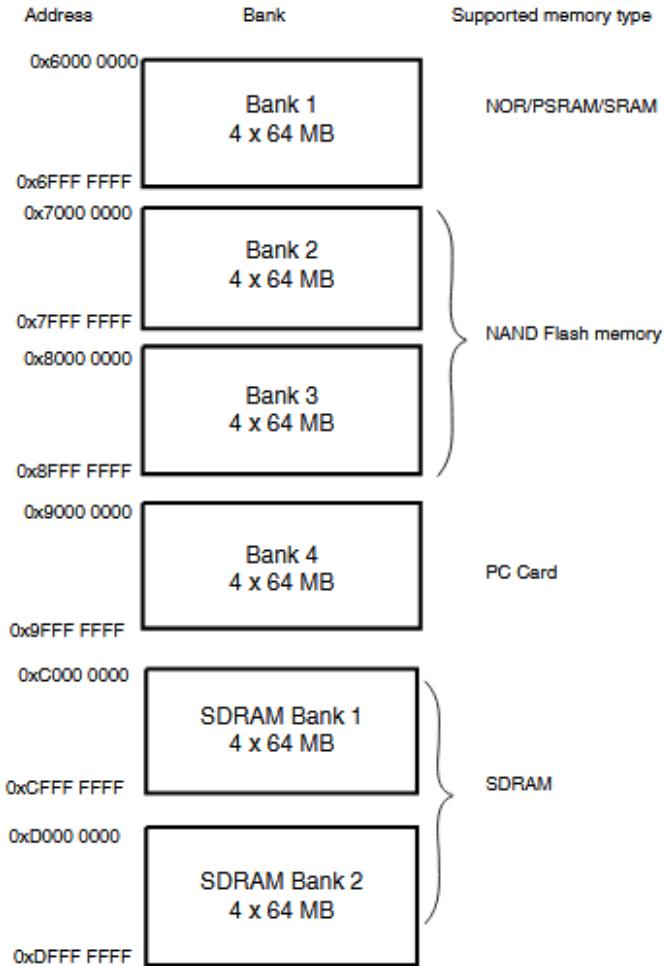


Figure 48. The FMC memory map [42].

The FMC enables data transaction with the 32/16/8 bits width. Maximum width of the address bus is 23 bits. In system's point of view, the external memories are accessible from 0x60000000 up to 0xE0000000 addresses (in hexadecimal code).

The FMC memory map is shown in Figure 48. The ranges of the addresses are fixed and restricted. The FMC is split into six 256Mbyte fixed-size banks.

The Bank1 allows the connection of four small NOR Flashes or PSRAM. In addition, this bank consists of four chips-select signals and four smaller sub-banks (but still only 256Mbyte is available for all four memories). Other banks can operate with only one external memory [42]. In a firmware's implementation point of view, the driving of the external memories is performed on the hardware layers, out of the core [42]. In this project, the Bank1 and the SDRAM Bank1 are used respectively for the NOR Flash and the SDRAM.

5.2.8 TFT LCD display

The DMA2D, LTDC, FMC with an external SDRAM, the Flash and a couple buses of the ARM core are used to drive the TFT LCD display. Mutual dependencies between modules, with data flows, are presented in the Figure 49 [46].

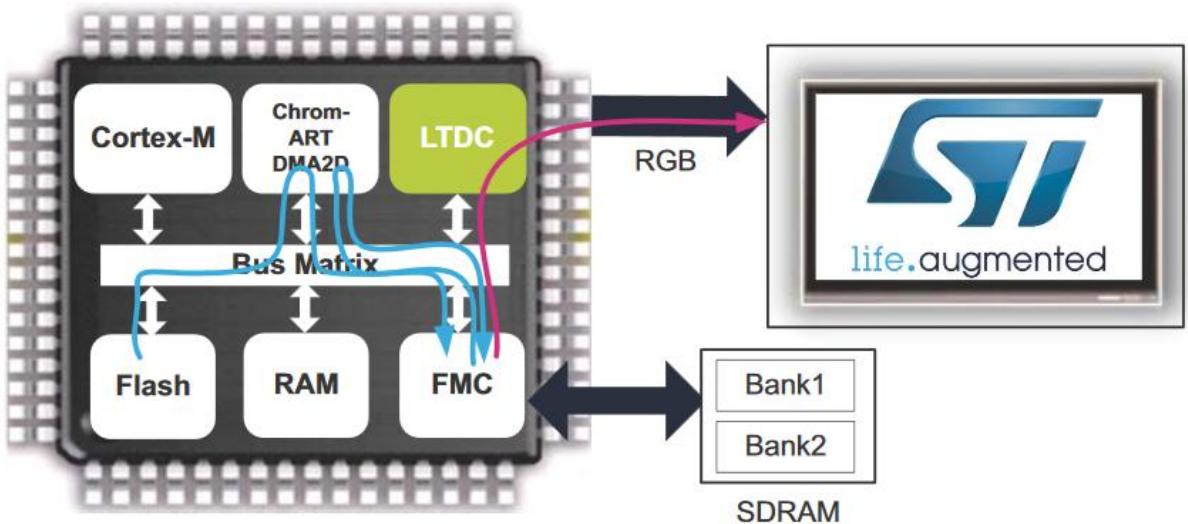


Figure 49. The graphic hardware configuration driver with external SDRAM [46].

The DMA2D exchanges data with external memories using the FMC. Those memories are the data buffers of the TFT LCD display. The DMA2D and the LTDC controllers share the access to the FMC graphical data into the external memories. This data is pushed out to the TFT LCD through RGB interface. Moreover, the operational parameters of the DMA2D and the LTDC (e.g. blending and dithering factors, format of pixels) can be changed from the ARM core, at any time [46].

The DMA2D requires additional auxiliary data buffers (one for each operate layer). Auxiliary buffers of the DMA2D are used to convert and prepare graphical data, before it will be sent to the LCD. Either, single buffer size equals the sizes of the single LCD template [46].

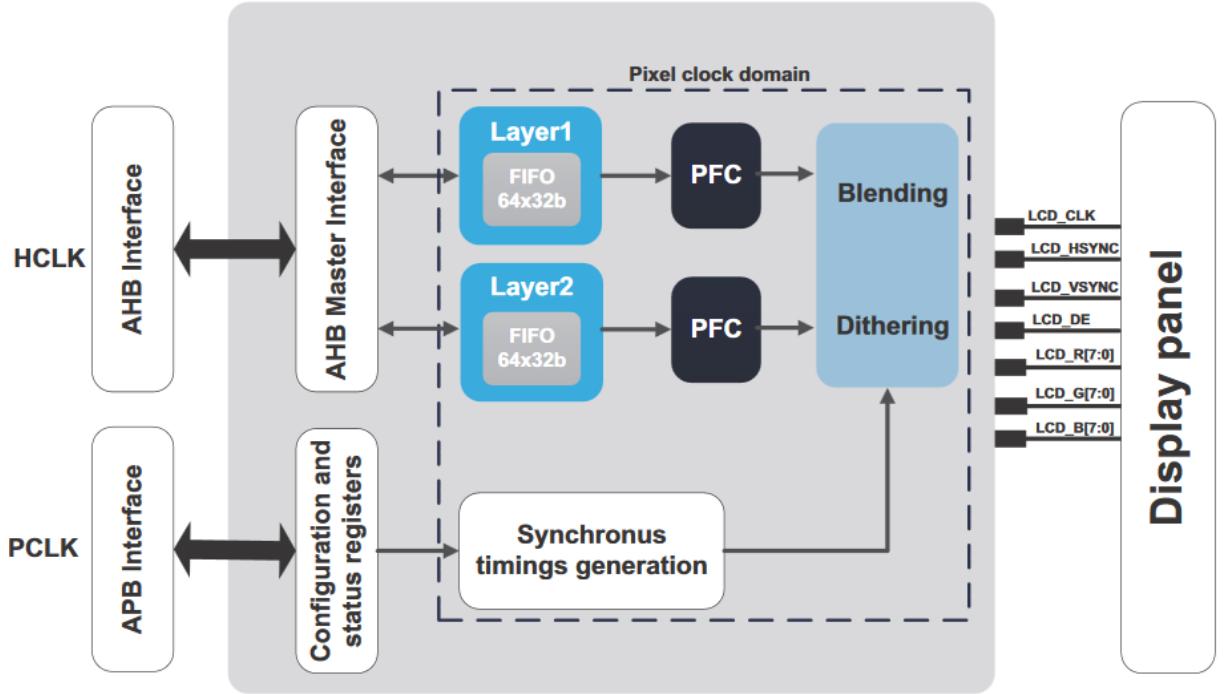


Figure 50. The LTDC controller block diagram [46].

Figure 50 presents the internal structure of the LTDC. The data and setting transfers are performed with the use of the APBs and AHBS buses. The LTDC takes data of two layers, which are provided to the PFC unit. Converted pixels are sent to Blending and Dithering units, which are used to generate graphical effects, such as the backlight of selected fields or flexible screen changes. These effects are achieved by mixing layers with each other accordingly to the parameters of Blending and Dithering units. These parameters can be changed through the APBs buses, without reconfiguration of the LTDC. The order and process of layers blending can be seen on Figure 51 [46].

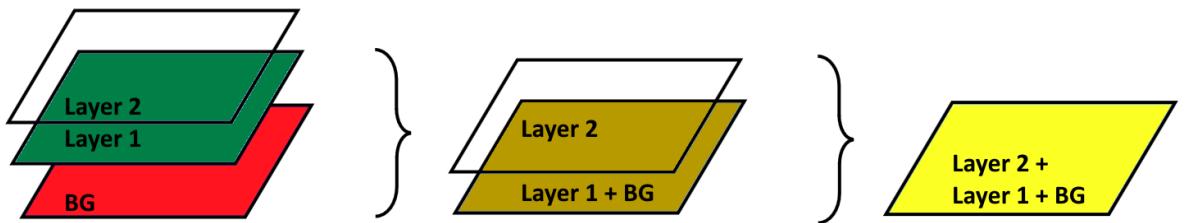


Figure 51. The blending layer order [42].

Default colour of the background is adjustable for each layer. On the first step, the first layer is mixed with a background. Afterwards, mixed first layer is blended with the second layer. The result of the last operation will be displayed on the LCD matrix. Besides, the colours blending parameters can be easily changed by the Alfa factor, independently for each layer and background. The factor can be changed at any time. The blending order is always fixed

[46]. Finally, all blended layers are placed in the internal FIFO buffer of the LTDC. The data is pushed out through the parallel interface. The timings of the parallel interface are presented on the Figure 52. The parallel interface consists of signals, such as the **VSYNC** (Vertical Synchronisation), **HSYNC** (Horizontal Synchronisation), **DE** (Data Enable), **CLK** (clock) and eight signals for each colour (red, green, blue). The graphics data are sent, accordingly to order: pixel by pixel, column by column, line by line [46].

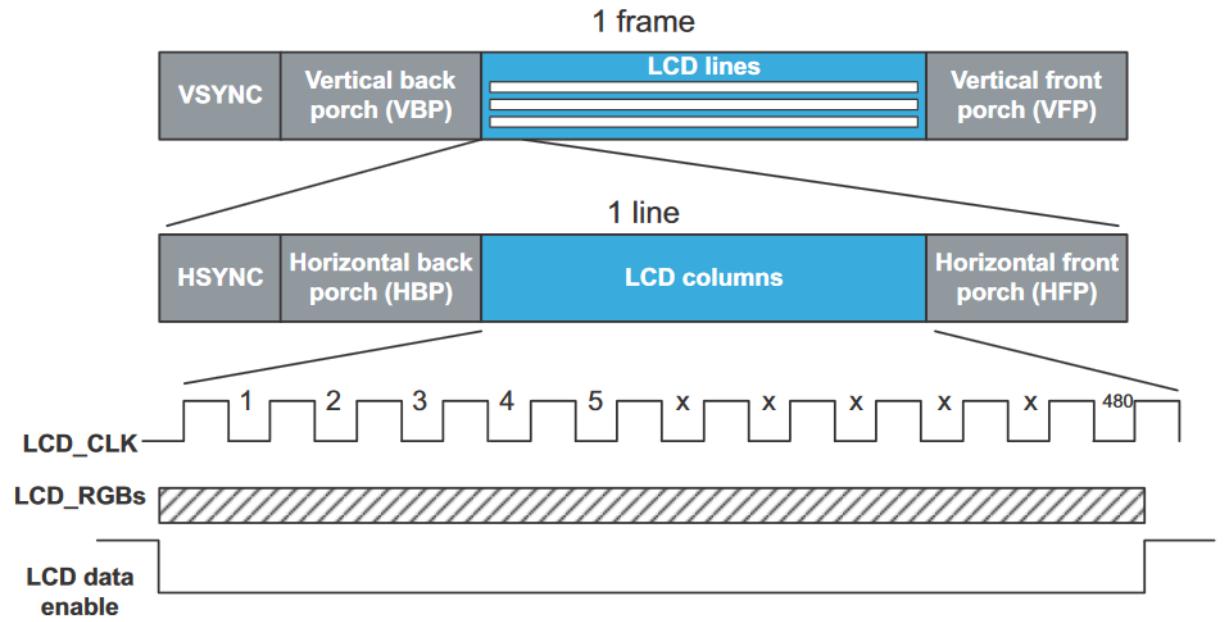


Figure 52. The LTDC display data frame [46].

In order to achieve synchronous timings, the **VBP**, **VFP** (Vertical Back/Front Porch), **HBP**, **HFB** (Horizontal Back/Front Porch) parameters must be configured correctly. Mentioned parameters are used to synchronise columns and lines with each other. Withal, they can be considered as a time “gap” between next line/column (used to prepare a TFT LCD for next part of data). Furthermore, the VSYNC and HSYNC widths should be also suitable configured according to the clock frequency (parameters configured as a number of the TFT LCD clock cycles at time) [46].

Figure 53 shows the graphical representation of the LTDC timings, set out on Figure 52. Based on this figure and the TFT LCD resolution [15], the VSYNC and HSYNC signals will be generated respectively 480 and 800x480 times [46].

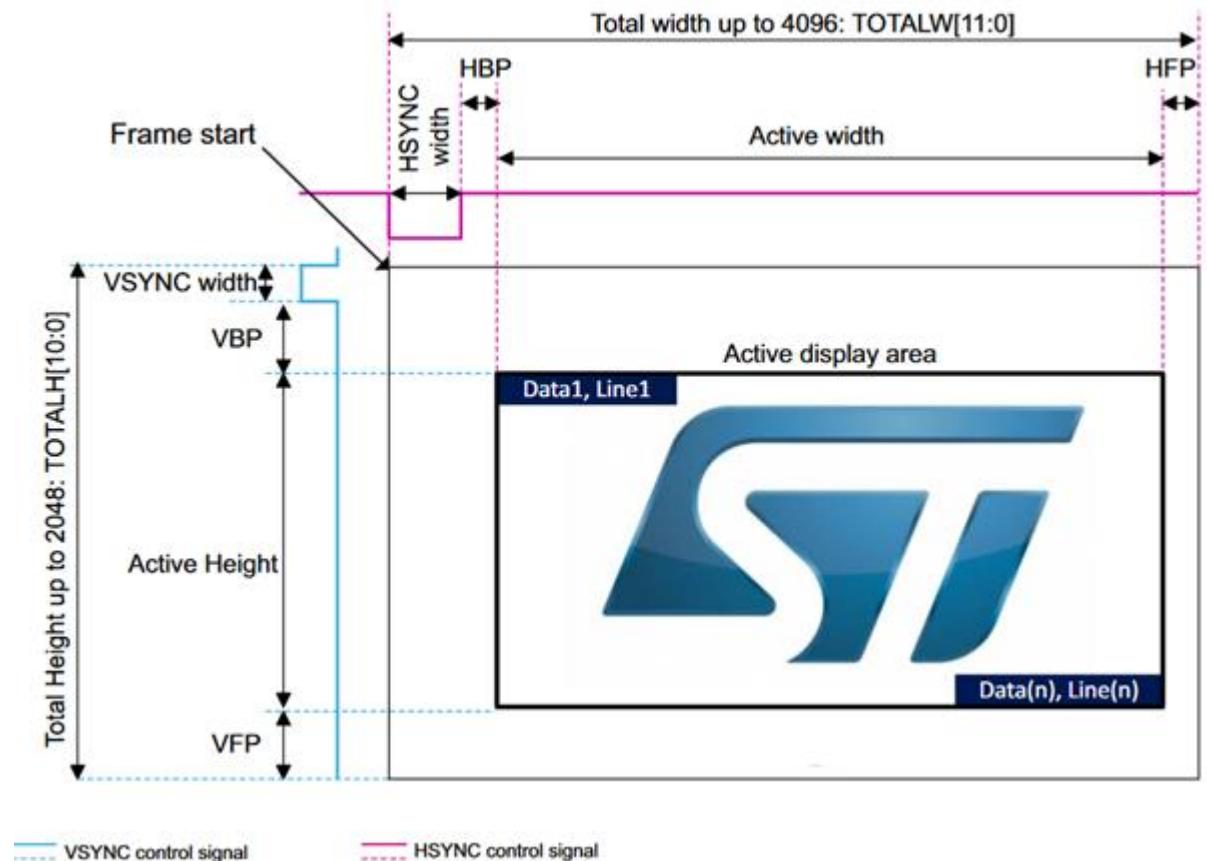


Figure 53. The LTDC timings and resolutions [46].

The maximum resolutions of the VGA (Video Graphics Array) standard are the XGA 1024 x 768 (Extended Graphical Array) and HD 1280 x 720 (High Definition). The TFT LCD resolution is the WVGA 800 x 480 (Wide VGA). It must be mentioned that the LTDC of the STM32F429xx family can drive TFT LCDs, which resolution is up to 4096x2048 pixels [46].

The LTDC can operate with eight different formats of pixels, from which seven of them are shown in Figure 54. For each layer, a pixel format can be configured separately. The colours can be coded in indirect (L8, AL88, AL44) or direct (RGB888, RGB565, ARGB888, ARGB144 and ARGB4444) formats [46].

	@ + 4	@ + 3	@ + 2	@							
Direct color	ARGB8888	Ax[7:0]	Rx[7:0]	Gx[7:0]	Bx[7:0]						
	RGB888	Bx+1[7:0]	Rx[7:0]	Gx[7:0]	Bx[7:0]						
	ARGB1555	Ax+1[0]	Rx+1[4:0]	Gx+1[4:3]	Gx+1[2:0]	Bx+1[4:0]	Ax[0]	Rx[4:0]	Gx[4:3]	Gx[2:0]	Bx[4:0]
	ARGB4444	Ax+1[3:0]	Rx+1[3:0]	Gx+1[3:0]	Bx+1[3:0]	Ax[3:0]	Rx[3:0]	Gx[3:0]	Bx[3:0]		
Indirect color	L8	Lx+3[7:0]	Lx+2[7:0]	Lx+1[7:0]	Lx[7:0]						
	AL88	Ax+1[7:0]	Lx+1[7:0]	Ax[7:0]	Lx[7:0]						
	AL44	Ax+3[3:0] Lx+3[3:0]	Ax+2[3:0] Lx+2[3:0]	Ax+1[3:0] Lx+1[3:0]	Ax[3:0] Lx[3:0]						

Figure 54. The colours formats versus pixel data mapping [46].

For indirect formats, the values of every pixel are pointers to the internal array of the LTDC. The array holds values of colours or the Alfa factor (array is the CLUT). When a pixel direct format is chosen, the pixels' values are equalled the values of the colours or the Alfa factor (the positions of bits and byte widths can be different, depending on format) [46]. Figure 54 displays the positions of bytes/bits and their range depending on the colour and format. The maximum size of single pixel data is four bytes (24 bits), which is standard size of pixels for majority images files (such as .PNG, .JPG that are widely used on the PC) [46].

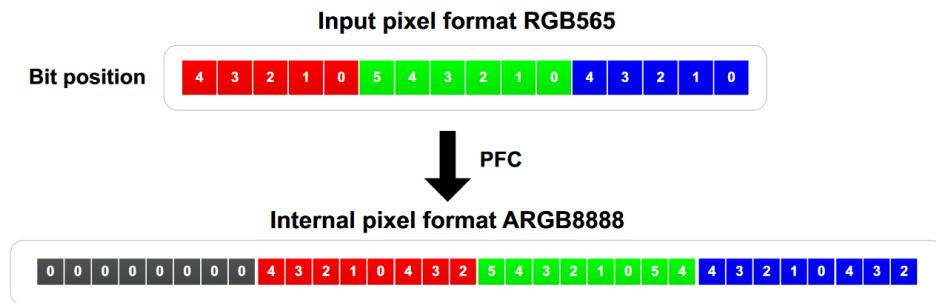


Figure 55. The pixel formal conversion from the RGB565 to the ARGB888 [46].

For this project, the default pixel format is the RGB565, where five, six and five bits are reserved respectively for the red, green and blue colours. Every pixel format is always converted to internal the ARGB888 format [46]. The conversion with bit positions are presented in the Figure 55.

The most significant bits of the RGB565 are copied to the least significant bits of the ARGB888 format. In addition, the Alfa factor byte is added (fulfilled with zeros). What is worth to highlight, the conversion between formats is always disruptive and different colour (than assumed) can be displayed on the TFT LCD matrix [46]. For this project, the disruptive conversion between formats is acceptable, in the point of view of user interface.

5.2.9 Communication interfaces

The main communication interface is the SPI, that is a bidirectional serial interface (Figure 56). The SPI has three main signal lines and usually one additional chip-select pin. The signals are respectively: the **SCK/CLK** (clock signal), **MOSI** (Master Output Slave Input), **MISO** (Master Input Slave Output) and **CS** (Chip Select) [42] [87].

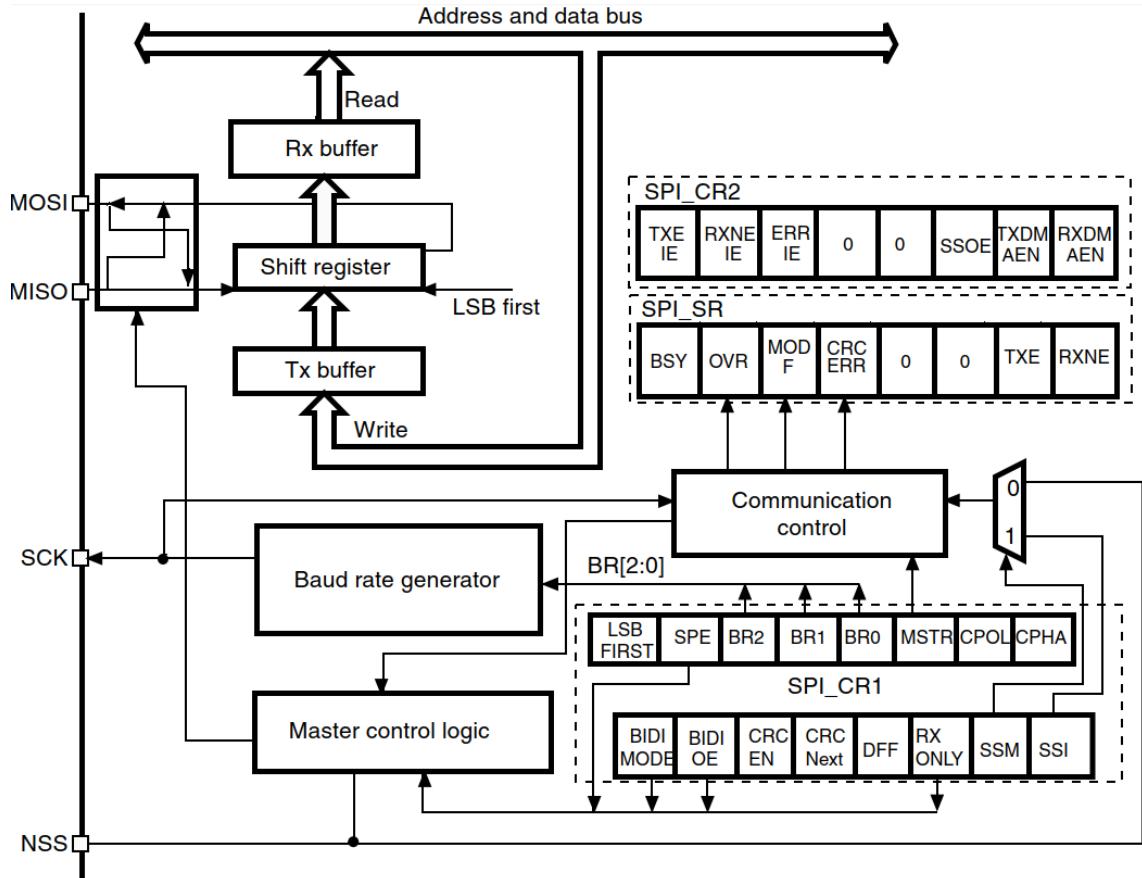


Figure 56. The SPI block diagram of the STM32F4xx microcontrollers [42].

The STM32 SPIs are connected to two dedicated streams of the DMA1 or DMA2 controllers, configured to operate in the full-duplex mode with 16-bits data width. When the full-duplex mode is selected, the receiving and transmitting data will be performed at the same time. For the SPI in full-duplex mode, the MISO and MOSI are synchronised with the use of the clock signal [42] [87].

The SPIs controller provides opportunity to set phase polarity of signals (clock edge type, for which signals will be latched). The SPI controller can operate as a master (when signal clock is provided to external devices) or a slave (when the controller is driven by externally generated clock) [42] [87]. The maximum generated frequency of the clock signal is up to 45MHz, which is available for only four SPIs of STM32F429xx (SPI1, SPI2, SPI5, and SPI6). For other

two SPIs (SPI3 and SPI4) is only up to 25MHz. The maximum baud rates, respectively to the frequencies, are approx. 3Mbps and 1.5Mbps. The SPI controller has a built-in CRC (Cyclic redundancy Check) module used to verify correctness of received and transmitted data [42] [14] [87].

The USART is auxiliary interface for the communication with PC applications (more details in chapters 6.6 and 7.6.2) and its controller block diagram shown in Figure 57. The USART operates as standard synchronous/asynchronous RS232 serial port, with the auxiliary hardware flow control. The controller consists of four main signals such as: **Tx** (Transmit), **Rx** (Receive), **CLK** (Clock), **RTS** (Request to Send), **CTS** (Clear to Send). The RTS and CTS are only used by the hardware flow control (not used in this project). Furthermore, it can operate in half-duplex mode, when one line operates as a clock and second as a bidirectional data line [42] [88].

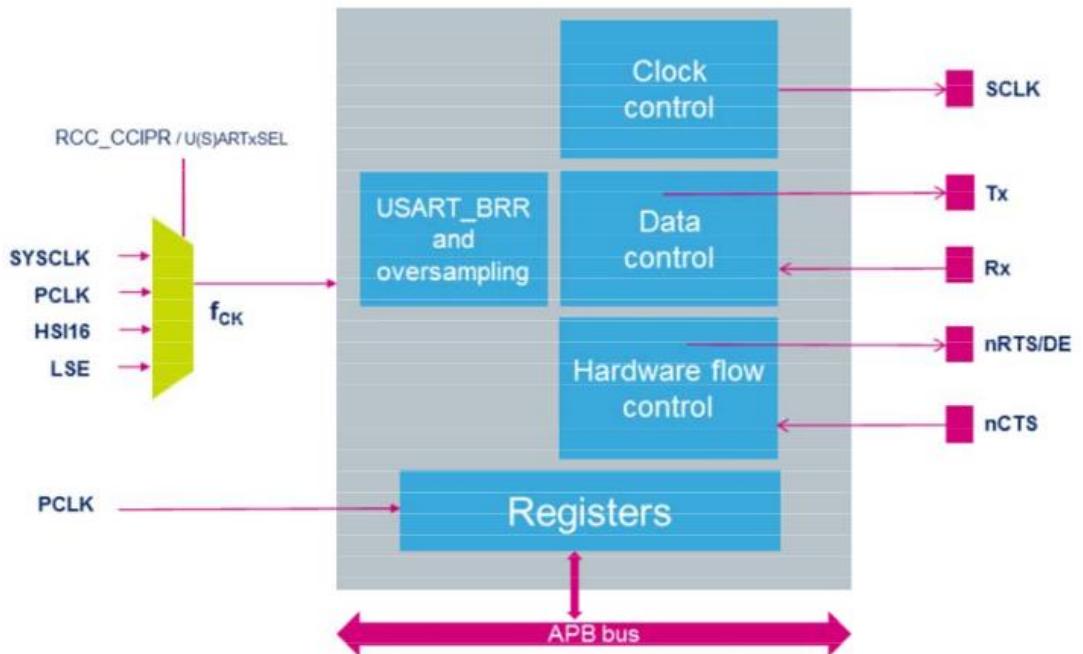


Figure 57. The USART(RS232) block diagram [88].

The default mode of the USART is the asynchronous mode, without hardware flow control. The Tx and Rx signals are not synchronised with each other and therefore the exchange of data can be more flexible. The controller is configured for sending or receiving 8-bit word length with the one parity and one-stop bits. Furthermore, the receiver and transmitter are connected to suitable DMA streams. Moreover, the default baudrate is equal to 256KBps and it can be increased up to 10.5 Mbps [42] [88]. Moreover, the USART uses the oversampling unit. This unit contributes to achieve higher reliability of transmission due to higher accuracy of signal sampling [42] [88].

It must be mentioned that the USART is not a differential interface. The reliability of transmission highly depends on speed of transmission, cable type and environment noises. It is highly recommended to use additional signal filters with EMI and ESD protection modules.

The EEPROM is driven through the I²C being two-directional synchronous interface, operated in the half-duplex mode. The standard I²C bus consists of two main signals: bidirectional data and clock that maximum frequency is up to 100kHz (standard mode) or up to 400kHz (fast mode) [42] [52].

The block diagram of the STM32 I²C controller is shown in Figure 58. It contains additional the SMBus (System Management Bus – not used in this project). The I²C does not have implemented default protocol (I²C is just specification how single data can be received or sent) in contrary to the SMBus, which has up to the nine implemented protocols. Either, the STM fully supports the I²C on the hardware level [42].

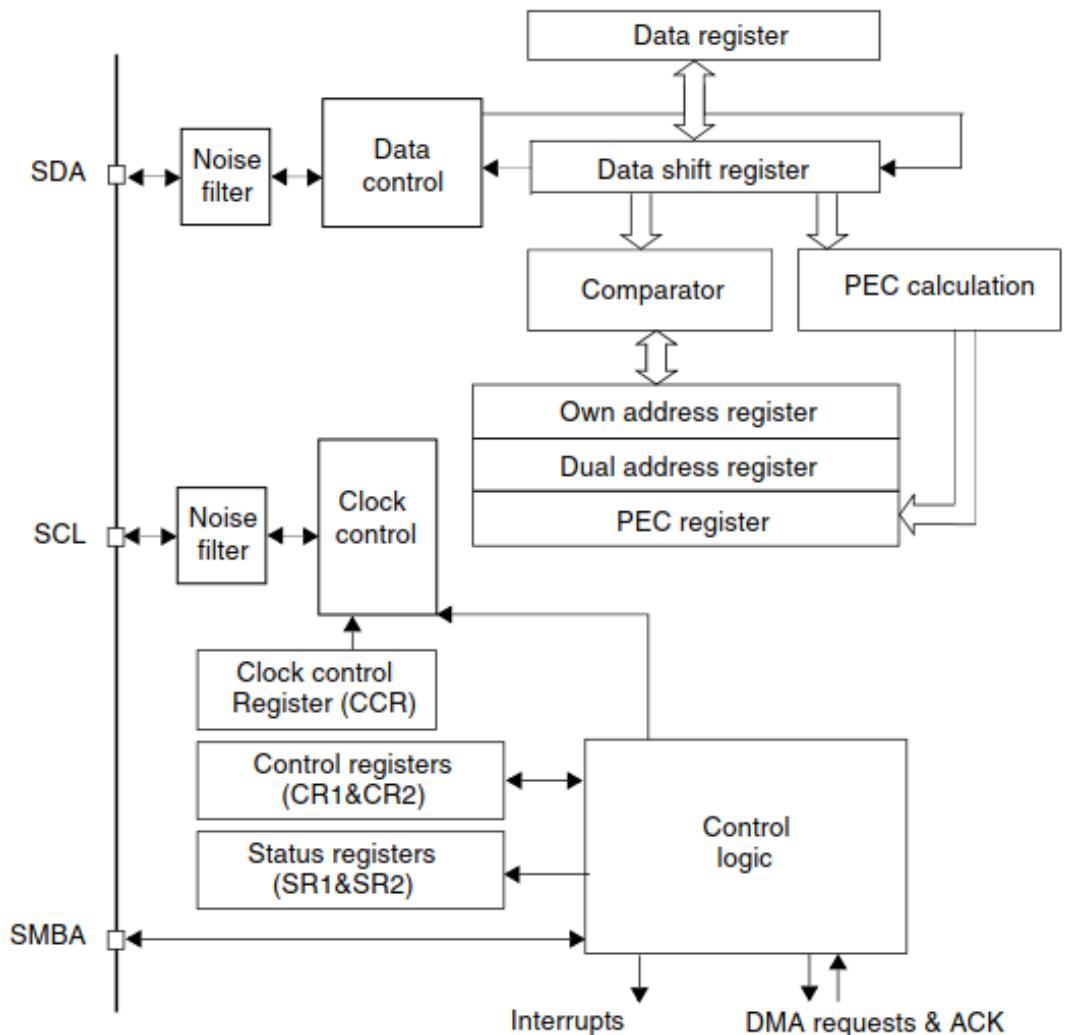


Figure 58. The I²C block diagram [42].

The I²C controller has a noise filter for signal lines (Figure 58) and it can operate as master or slave. The I²C interface's advantage is that multiple different devices can be connected to the same bus at the same time [42].

The identification of a targeted device is performed by sending a start frame equipped with a unique device address. All connected devices receive the start frame, but only targeted device will be activated (other devices will go to Sleep mode). Usually every chip with built-in I²C has adjustable address, configured by suitable polarisation of additional chip's address IO ports [52]. Multiple devices can be connected to the same I²C bus, but every device must have an unique address. In other case, the communication will be not possible.

The EEPROM chip has address 0x55. Further, the address width of I²C chips can be 7 or 10 bits. The internal controller of STM32 is configured to operate in the standard mode as the master. The widths of address and data words are equalled 7bits [42] [52].

5.2.10 USB OTG controller

The USB OTG interface and its protocol are fully supported by the STM32 on the hardware and software layers. Figure 59 indicates the architecture of the USB application dedicated for the STM32F429xx microcontrollers. The controller application is based on the open-source libraries released by the STMicroelectronics [89] [36].

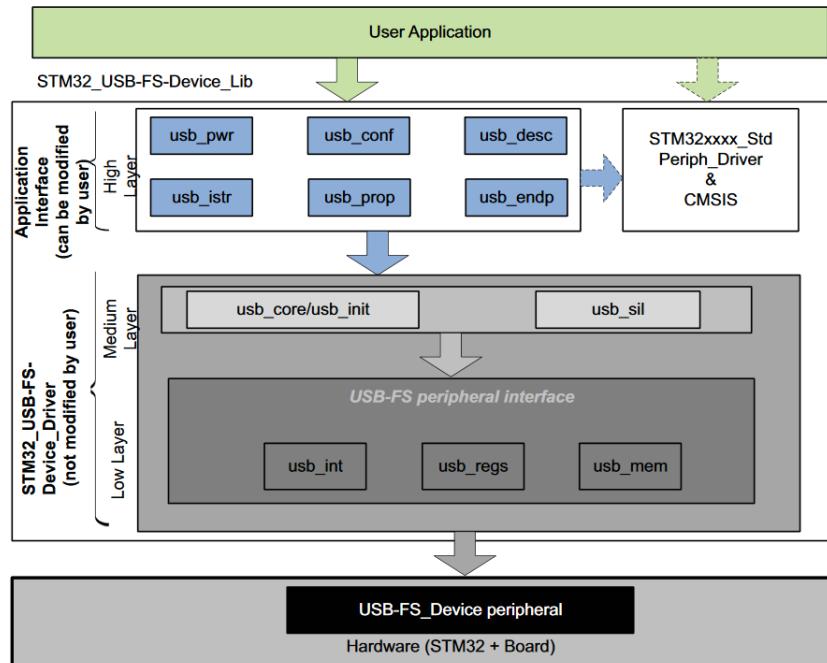


Figure 59. The USB application diagram, based on the ST libraries [89].

The USB software of the ST drivers are incorporated with two main layers:

- ***Device driver*** – used to drive communication with an external device on the hardware and to configure the internal controller of STM32 (low level software layer) [89].
- ***Application Interface*** – used to manage the protocol on the abstraction layer [89].

The interface allows devices for switching operational modes to a host (master) or a device (slave) on the fly. Moreover, the ST libraries contain the USB OTG implementation parts such as the HNP (Host Negotiation Protocol used to automatically detect a host or a device on the bus), SRP (it allows sending and detecting a request of turning on a power bus) and ADP (Attach Detection Protocol used to detect a passive device). All connected devices always assume self and other devices operational roles, based on suitable configurations of hardware signals. By default, the VGC controller is device for PCs (host). ST libraries contain full implementation of the USB OTG and they are highly recommended solution. [42] [65] [55] [66].

5.2.11 External bootloader

The STM32 microcontroller can be flashed through a standard embedded communication interfaces such as the USART, SPI, I²C or USB. The bootloaders are stored into internal boot ROM (Read-Only Memory) and they can be activated by applying a suitable boot pattern, described in the technical documentation [42] [90]. For the STM32, the fifth pattern must be applied. The bootloader-enabling procedure must be performed accordingly to the block diagram, shown in the Figure 60 [42] [90].

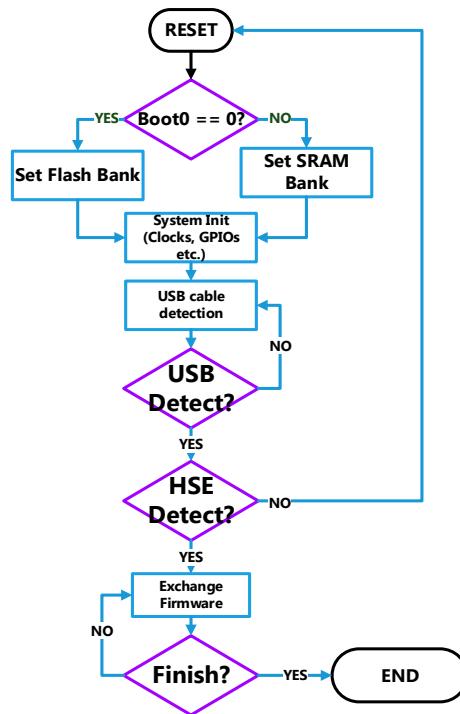


Figure 60. The bootloader-enabling procedure block diagram [90] [100].

In order to enter to the bootloader mode, a reset handler must be activated. Before entering, all peripheral clocks and interrupts must be disabled. What is crucial, the STM32 microcontrollers have the “BOOT0” and “BOOT1” IO ports. These IOs must be suitable polarized, in order to select the targeted bootloader (Flash or SRAM) memory [42] [90]. For used STM32, the BOOT0 port must be connected to ground. Furthermore, when the USB bootloader is chosen, the HSE clock must be in active mode. In another case, an exchange of firmware will be not possible. Figure 60 presents the procedure for the USB bootloader in the DFU (Device Firmware Upgrade) mode [42] [90] [91]. External bootloader adds a crucial functionality of exchanging the firmware in a situation, when access to the programmer socket is strongly inhibited.

5.2.12 RNG, Watchdog and other drivers

The STM32F4xx microcontrollers contain two types of Watchdogs: independent and window. Watchdog is an independent timer unit used to prevent the microcontroller from staying too long in suspend state. In standard operational mode, the watchdog counter register is reset every given period. If a microcontroller is in suspend state, watchdog counter will not be reset and the module will generate a reset signal [42]. The main principle of watchdogs is very simple and it is sufficient tool to prevent majority electronics devices from staying too long in the suspend state.

What is more, the main microcontroller contains multiple advanced 16-bit auto reload timers, which generate multiple interrupts at the time. Timer units are widely used to switch system tasks and to measure time duration or frequency of signals/events/interrupts/tasks. The STM32 timers are independently operated and they can be synchronised with each other or with other core peripherals [42].

The STMF429xx microprocessors contain several internal controllers such as the RNG (Random Number Generator used to generate random numbers with 32-bits width in the test of driving external cards, described in the chapter 7.6.1), HASH and CRYP processors (used to secure hash algorithm), PWR (Power Controller), ADCs, DACs and others [42].

6 Software & Firmware application layer

Following chapter presents a short overview of the VGC software. Brief descriptions contain information about used algorithms, implemented protocols, memory map, GUI, PC applications and software development tools. Moreover, the chapter contains several illustrating the software mechanisms.

6.1 General main algorithm block diagram.

Figure 61 presents the block diagram of the main algorithm. The software implementation is dedicated to perform several tasks approximately at parallel, and it is called the RTOS (Real Time Operating System). The tasks are considered as parallel and independent threads of the program. The STM32 has only one single core and each task has a set timeout for self-execution. When the task is not finished and timeout occurs, it will be suspended with saved context (in RTOS, all data of executing program/function, used to restore last state in the next iteration, e.g. loop iterator) and next task will be executed. Described mechanism (called the context switch) is widely used in RTOSs for devices equipped with a single core [92].

The controller program is divided into five subroutines (Figure 51). All listed subroutines are performed in the main loop and interrupt handlers. Each task, similar to the interrupts, has a self-priority. It must be mentioned, that this asynchronously executed application is free of any software delays.

With limited memory space it is not possible to collect measurement data of all gauges, with high sampling frequency for a long time span. If an external master (which could receive collected data from controller) is not present, the measurement data of gauges, placed at dedicated buffers, will be overwritten. Furthermore, the sampling frequency cannot be too high. Moreover, additional software power off switch is added, which should be used before disabling the main power supply (in order to prevent from unexpected behaviours of handled sensors). Additionally, the application records multiple data, which can be used to generate a histogram of critical events and to detect correctness of performing operations.

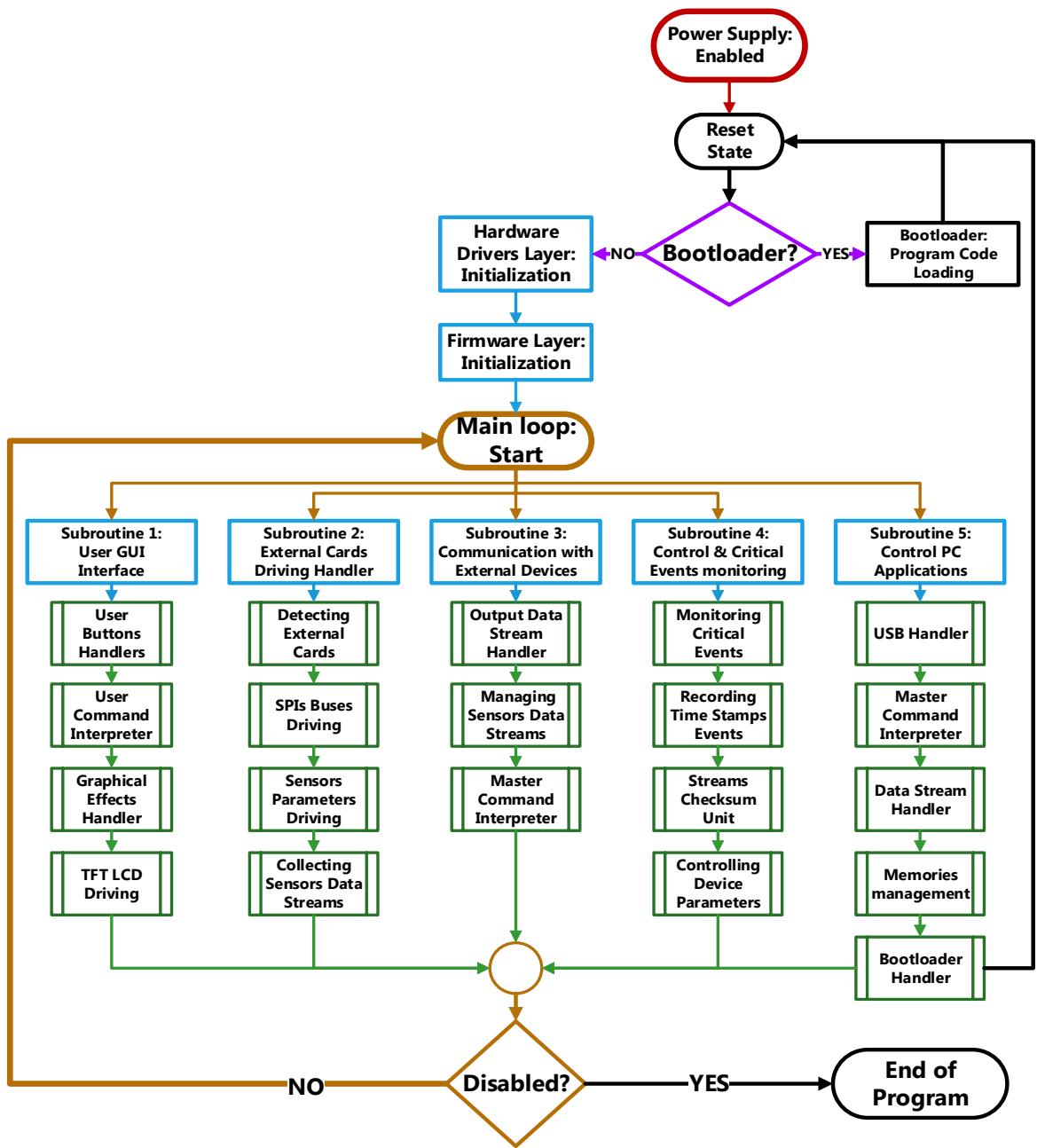


Figure 61. The firmware flow chart [100].

6.2 Communication protocol

Figure 62 presents the protocol flow chart for all external cards, driven through SPIs. The flow chart is presented from the motherboard side (external cards are slaves). When the motherboard is configured as a slave for a communication card, the protocol sequences are similar. The communication is always started by the motherboard. If an external master will be connected, during re-initialisation state, the motherboard will change the operational mode accordingly to sequences shown in the Figure 62.

The ID port of SPIs is used as an indicator of a card status. When a card will be connected to the backplane board, the ID port triggers an STM32 interrupt (“IRQn()” on the Figure 62). Afterwards, the communication is started up. The initialisation (“Init loop”) and configuration (“CMD loop”) of cards can occupy relatively long time. During these states, a SPI bus is disabled and the ID port is not active. When processes are finished, a card will trigger an STM32 interrupt with the used of the ID port. An interrupt handler is used to verify and check the status and parameters of a card. When a device will not be responding long period of the time, a timeout will occur. Moreover, when an executed command or re-initialisation is not successfully finished, after achieving the maximum number of tries the communication will be broken. During a standard mode (“Data stream loop”), the ID port is active and data streams are continuously performed.

Each data or command is placed in the dedicated data frame. This has fields such as delimiters and CRC bytes, used to check if received/sent frames are correct. When the system detects that a received frame is incorrect, a SPI master will go to re-initialisation loop (“Init loop”) and the start-up procedure will be repeated.

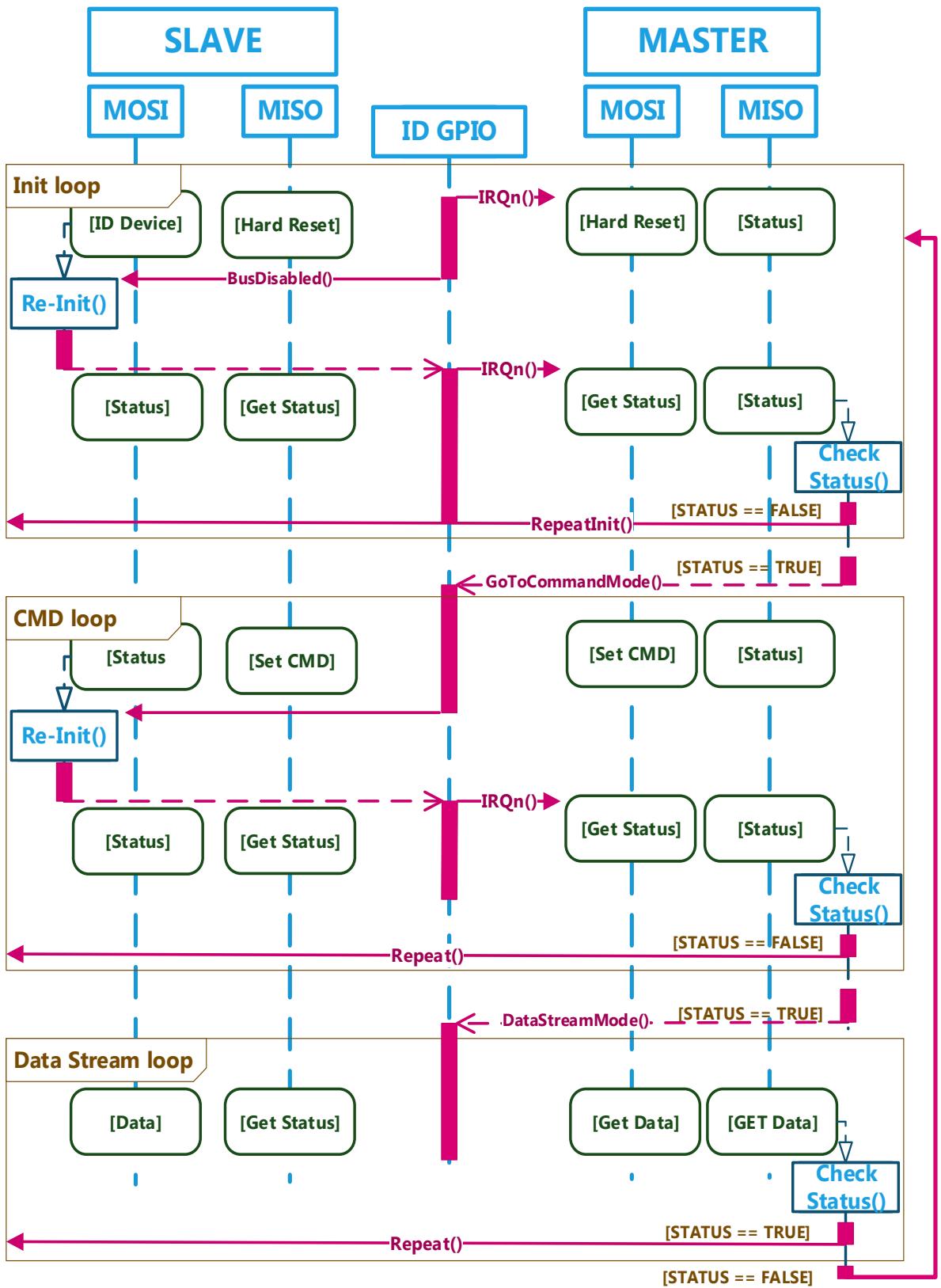


Figure 62. The SPIs protocol flow chart [100].

6.3 Memory architecture

The memory map of the VGC is shown in the Figure 63. It presents a detailed map of the main microcontroller memories, with the addresses and contained data. The memory system of the STM32F4 microcontroller is divided into seven main memory areas, which are fully occupied (4GB of space). The addresses of memories are from 0x0000 0000 up to 0xFFFF FFFF in the hexadecimal code [42] [86].

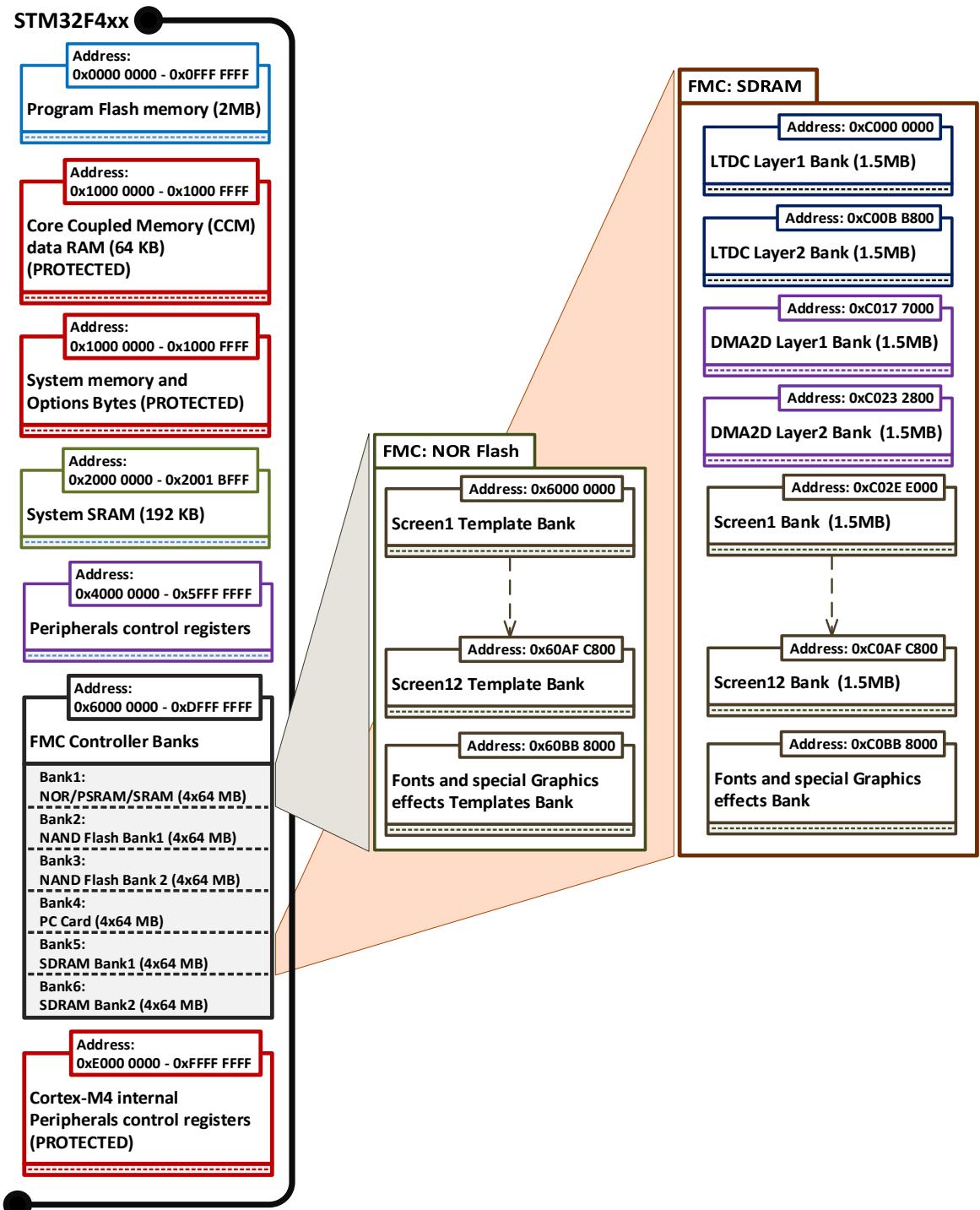


Figure 63. The memory map of the VGC Controller [100].

The main microcontroller has the 2MB program and 256kB SRAMs. Usually, for majority of the microcontrollers, the Flash memory is located into the first sector of the memory map. Moreover, the internal Flash of the STM32 microcontrollers is compliant with the NAND topology and it is internally divided into sectors [42].

The Core Coupled Memory data RAM (CCM), Core System Memory with Options Bytes and Cortex-M4 internal Peripherals Control Registers (red frames on the Figure 63) are special memories, which contain registers of the core peripherals such as the NVIC, system timer (SysTick) or system block control. Listed memories are highly secured from non-authorized access, and these memories are accessible from internal buses of the ARM core [42] [86].

The peripheral control registers (violet frame on the Figure 63) are only accessible for the core and its peripherals with the use of the Bit-Banding mechanism. This feature gives indirect access to the peripheral registers through a memory region that contains aliases to these registers. The Bit-Banding enables to perform operations on the single bits, faster than direct access to register (the access operation is atomic). Moreover, the Bit-Band mechanism is built-in feature of all microcontrollers with Cortex-M4 or Cortex-M3 cores [41] [42] [86].

The FMC region is split into six small banks, respectively for a NOR Flash, two NAND Flashes, PC card (standard SD card) and two SDRAMs. When the FMC will want to get access to addresses from not using banks, the memory management fault occur (only two of six banks are used in this project) [42].

Access to external memory cells, in point of the view of the system, it is the same as to the internal SRAM or Flash. Furthermore, the reading operation is free of any charges and easy to perform. The writing operations to the external NOR Flash requires adding dedicated procedures with relatively long duration [42] [86] [47].

Figure 63 depicts the addresses of the TFT LCD templates and its operational sub-banks, placed into the external memories. The SDRAM has four sub-banks more than the NOR Flash. These banks are operational banks of the LTDC and DMA2D. The addresses of the external EEPROM are not presented on the Figure 63 due to independently addressed memory cells. The chip is not directly connected with the STM32 memory map.

6.4 GUI – Graphical User Interface

Figure 64 presents the block diagram of the GUI architecture. The architecture consists of four main menus, dedicated for all external cards, configurations or information about the system, PC applications and errors or warnings handlers. Moreover, each menu can be selected with the use of buttons, which operational functions description is available in the system information menu or on the front panel of the controller. When a critical event occurs, the error or warning menu will be automatically selected and suitable message displayed. When only a warning occurs, the warning message will be automatically disabled after some timeout. In case of an error, the menu will not be disabled until the error has been resolved.

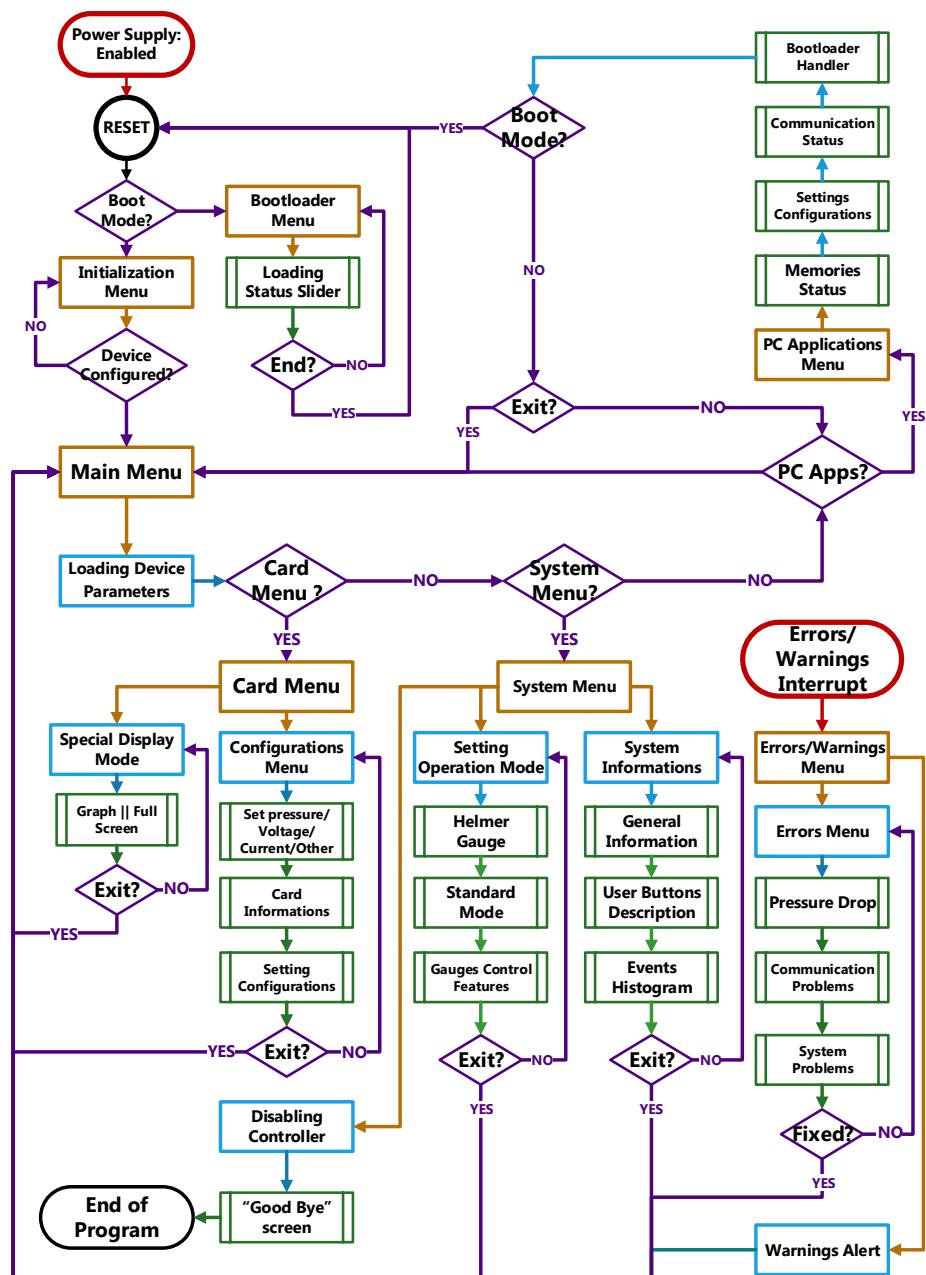
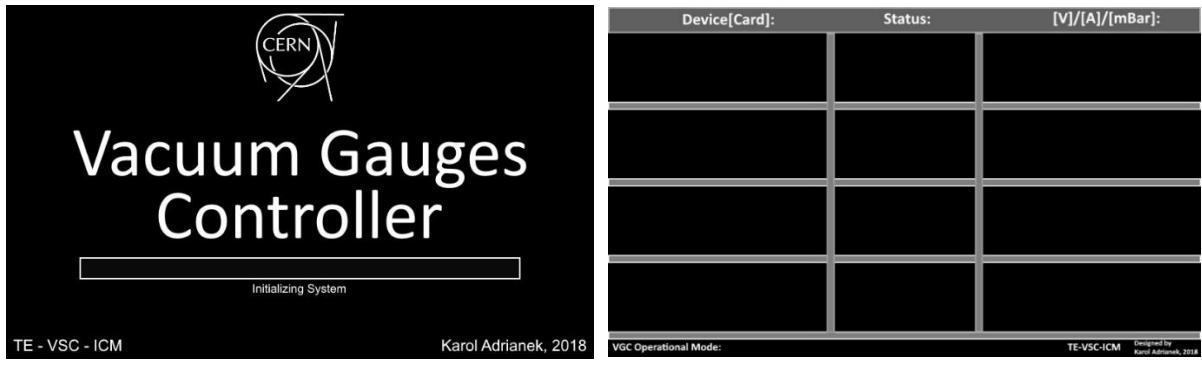


Figure 64. The GUI flow chart [100].

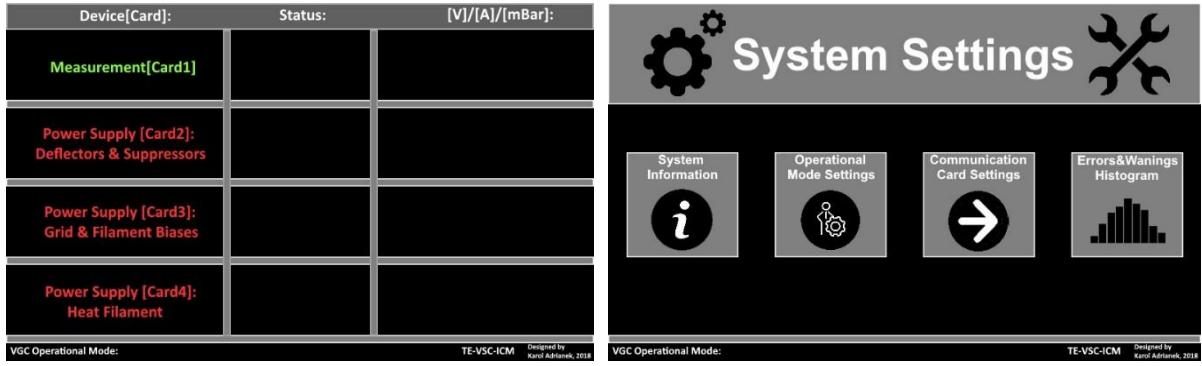
6.5 GUI Templates

The controller's architecture assumes that the graphical effects will be generated using designed GUI templates, saved as standard image formats. Figure 65, Figure 66 and Figure 67 present designed controller's GUI templates, with a short description of the main functionalities. The images have been designed accordingly to the GUI architecture (the chapter 6.4) and implemented in the STM32 peripheral memories.



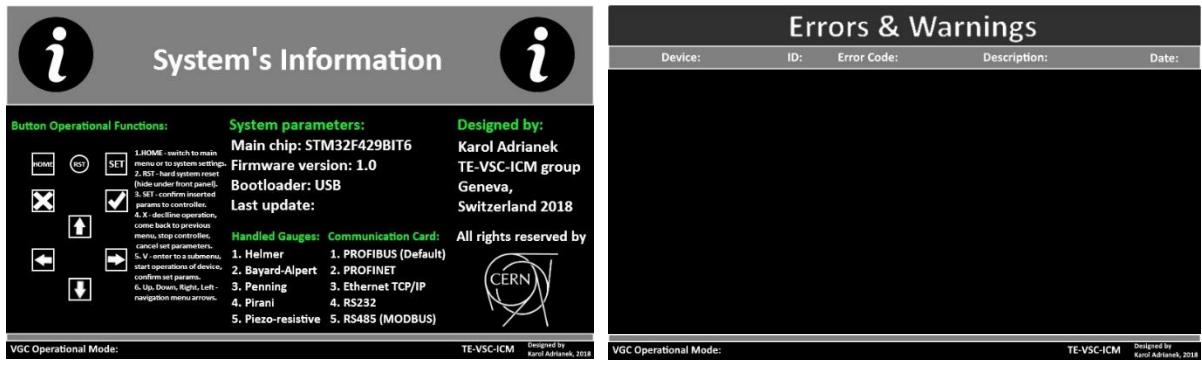
Welcome menu with the status bar indicating controller's initialisation progress.

The main menu displayed after the initialisation (dedicated for the standard operational mode of the controller).



The main menu dedicated for the Helmer gauge only.

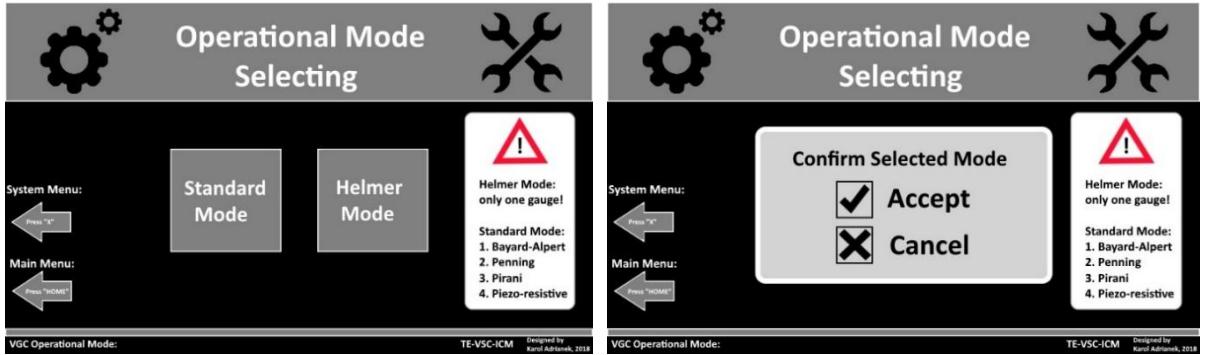
The main system settings, with sub-menus.



General system information menu.

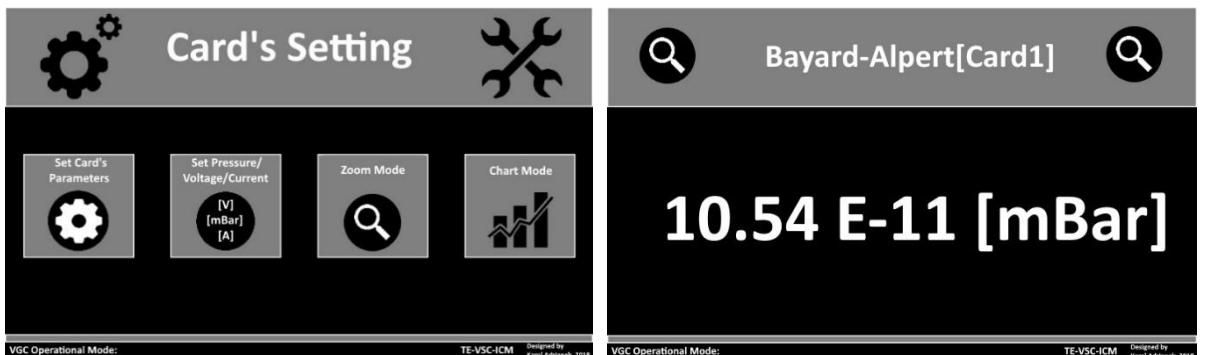
The menu dedicated for displaying information about stored critical system's events, errors and warnings.

Figure 65. The GUI Templates –first part [100].



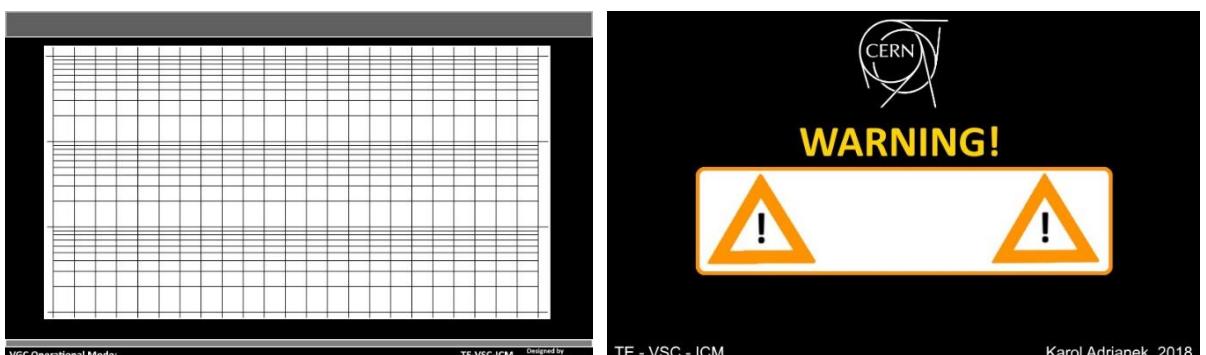
The operation mode-selection menu.

The confirmation window of selected operational mode.



The card's settings menu, with options of display mode, properties, setting and general parameters.

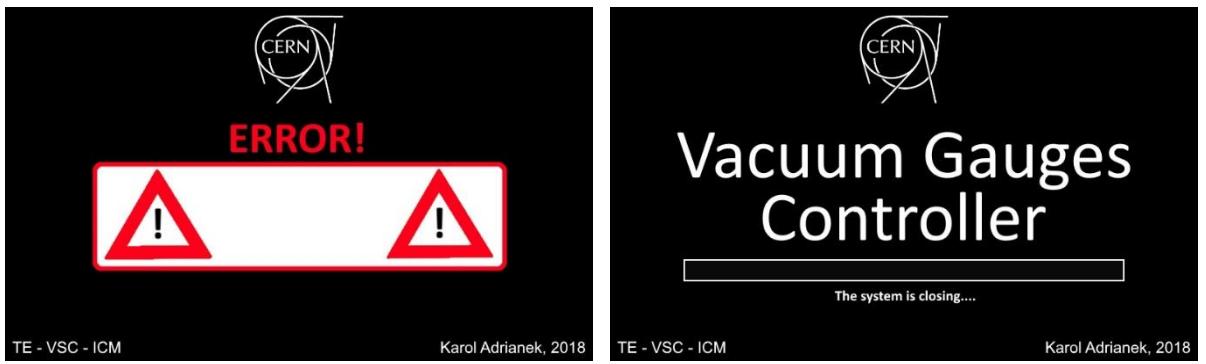
Dedicated “zoom” mode.



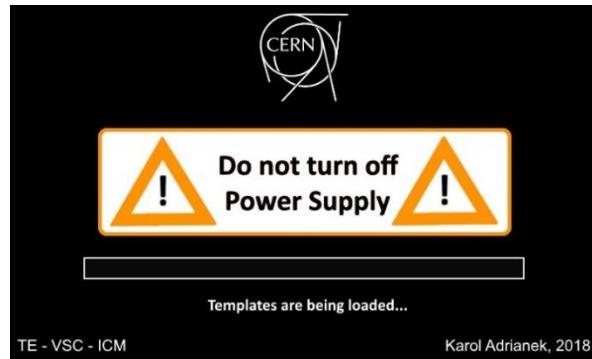
Logarithmic scale data display mode.

The menu dedicated for displaying information about system's warnings.

Figure 66. The GUI Templates – second part [100].



System's error displays only (when critical event will occur). Closing menu (with the progress bar).



The PC application menu, dedicated for loading the GUI templates from a PC to the controller (with status bar).

Figure 67. The GUI Templates – third part [100].

All presented templates have been designed by the master thesis author in the Paint.NET®, free of charge graphic software. It allows for effortless changing of the controllers menu items, GUI styles, positions and placement of elements, logos or dedicated non-standard images.

6.6 PC C++ console application

Following chapter presents a short overview of the PC C++ console application dedicated to control the VGC from standard PCs or laptops. The application main operational view is presented on Figure 68.

Figure 68. The C++ console application [100].

The main goal of the application is that the GUI templates, saved as standard graphical file formats (.PNG and .JPG), are converted to a required format and afterwards loaded into the external Flash and SDRAM. Additionally, the application can be used to configure basic parameters of the VGC controller. The TFT LCD templates loading is used for easily exchanging and creating the GUI and its features.

The conversion of images is done with the use of the Stbi open source library (free of charge released on the MIT licence). The author is Sean T. Barrett [93].

The block diagram of the implemented application algorithm is presented on Figure 69. It can automatically find the TFT LCD template file folder by local path. The program file must be placed in the parent folder of the folder that contains the template files. In other case, the program will not start. The communication is based on a default serial port of the motherboard containing a self-defined name, recorded in the internal memory of the STM32. This name is used to find and select the default serial port. When a default port is found, it will be automatically opened.

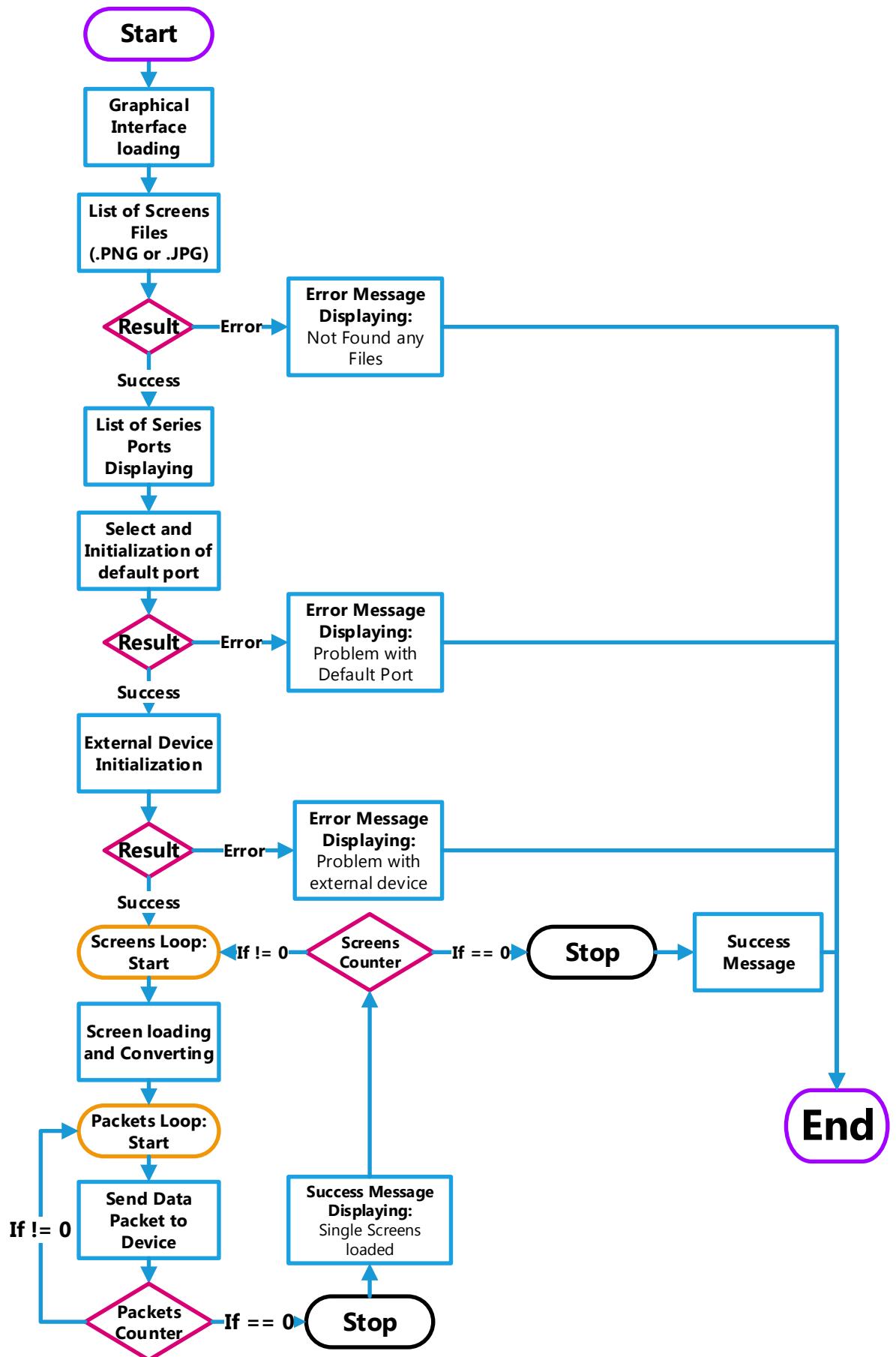


Figure 69. The console application flow chart [100].

6.7 Software development tools

During the design process of the software part, a couple of different software development tools have been used (Figure 70). The majority of used software tools are free and open-source for commercial projects. Moreover, the CERN collaborates with multiple different corporates and institutes, which provide multiple commercial software tools for CERN employees.



Figure 70. The logos of used software development tools [85] [94] [95] [96].

The software development tools used in this project are:

- **Atolic TrueStudio** – IDE for multiple microcontrollers' platform, used to design and implement the firmware for the STM32 main microcontroller in C.
- **Visual Studio Community 2017** – Microsoft's IDE, dedicated to design PC applications in multiple languages (e.g. C/C++, C#, Python). This IDE has been used to write the PC console application in C++ (described in the chapter 6.6).
- **QT Creator** – IDE for the C++, dedicated to create window application for multiple platforms and operating systems. The QT Creator is free for non-commercial project, but free version is relatively limited.
- **GIT** – the version control system, used to track changes of program files. The GIT is open-source and free.

7 VGC controller project; summary and tests

This chapter presents short overviews of the final test, project management, documentation of the PCBs designs, the prototype with photo documentation and the information about used tools. Moreover, the chapter describes performed tests of the software, hardware driver and hardware layers.

7.1 Methodology of project development

Figure 71 shows the block diagram of the project management structure, used for designing and manufacturing of the VGC controller prototype. Each engineering project has to be developed accordingly to a detailed plan, with set priorities and right order of tasks. The VGC has been done in respect of steps shown in Figure 71. Presented architecture made possible to achieve a complete and operational prototype (see chapter 9.4). The success of the majority of engineering projects highly depend upon right management and correct methodology.

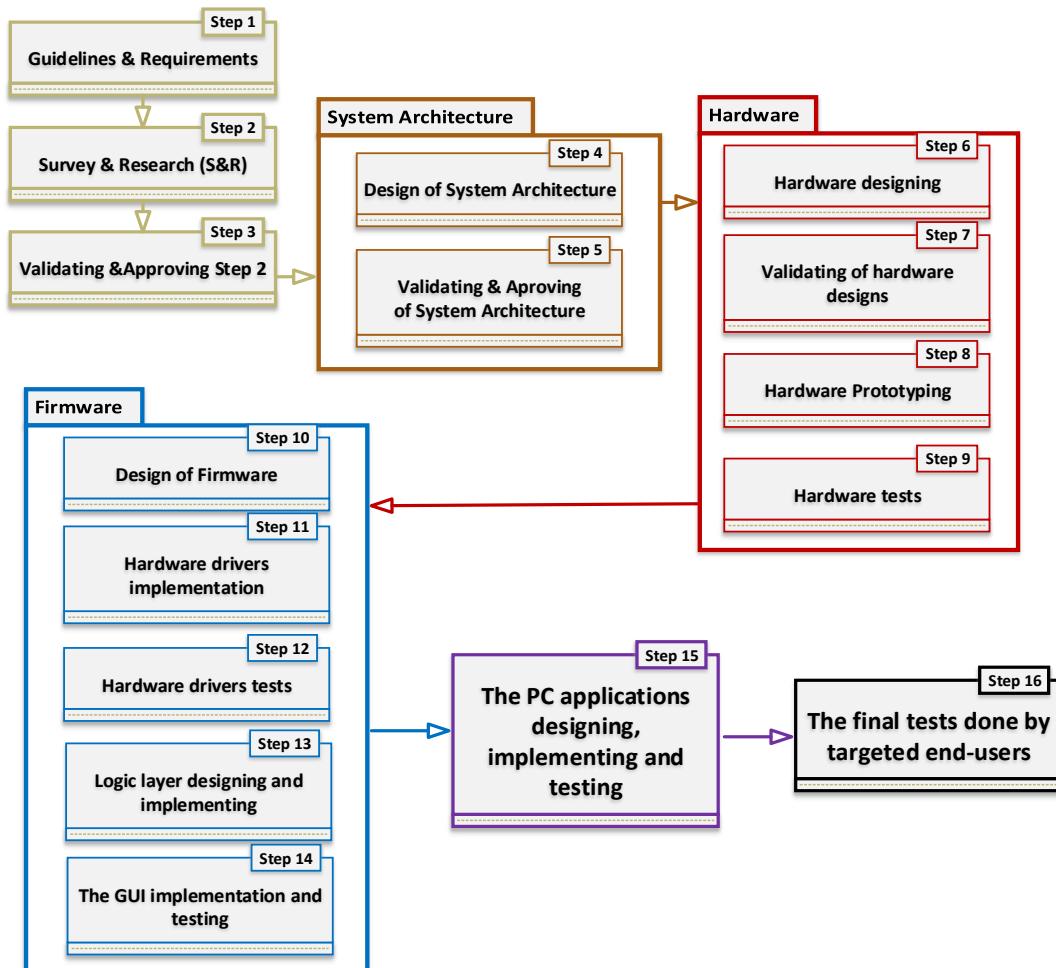


Figure 71. The project managing block diagram [100].

7.2 Electronics design and the prototype

Following chapter presents the documentation (with photos and screenshots) of designed PCBs and self-made prototype of the VGC. The figures (see 9.3), illustrate all the designs of PCBs and 3D models, that have been prepared in Altium Designer 17 program.

The motherboard is the four-layer PCB, where two layers are the power supply layers (the ground and +3.3VDC, placed onto the internal layers of the PCB), and next two are signal layers (placed on the external layers of the PCB). The thickness of the motherboard PCB is standard 1.55mm. The backplane board is standard two-layer board with a thickness of 3mm.

The PCB boards have been done with the use of a solder oven and standard soldering tools. The front panel of the prototype has been done with the use of a CNC milling machine. The design of the front panel has been designed in the AutoCAD 2017 program. The motherboard and the backplane boards are mounted to the mechanical chassis through the M3 screws and standoffs.

The complete prototype of the Vacuum Gauges Controller is placed in the mechanical 42HP 3U Plug-in-Unit (see 9.4). Moreover, the project has been done accordingly to industrial standards, theory of the high signal processing and good engineering practises. Additionally, complete schematics of the motherboard and the backplane board are in the attachments (see 9.1 & 9.2).

Taking into account internal CERN regulations, the source code of the controls' systems must be confidential and prevented from unauthorized access (only CERN employers can have access to the source code). The controller's software was not published in a public repository server and it is not available for external users. For this thesis purposes, only the hardware part (electronics schematics, PCBs' designs, the prototype's photos) has been released.

7.3 Tests system architecture

The test architecture, shown in Figure 72, consists of three main parts: hardware, hardware drivers of the STM32, and the firmware together with a PC software. The main goal of the plan was to achieve the highest performance of testing. Those tests have helped in detection of hardware and software problems with as little delay as possible.

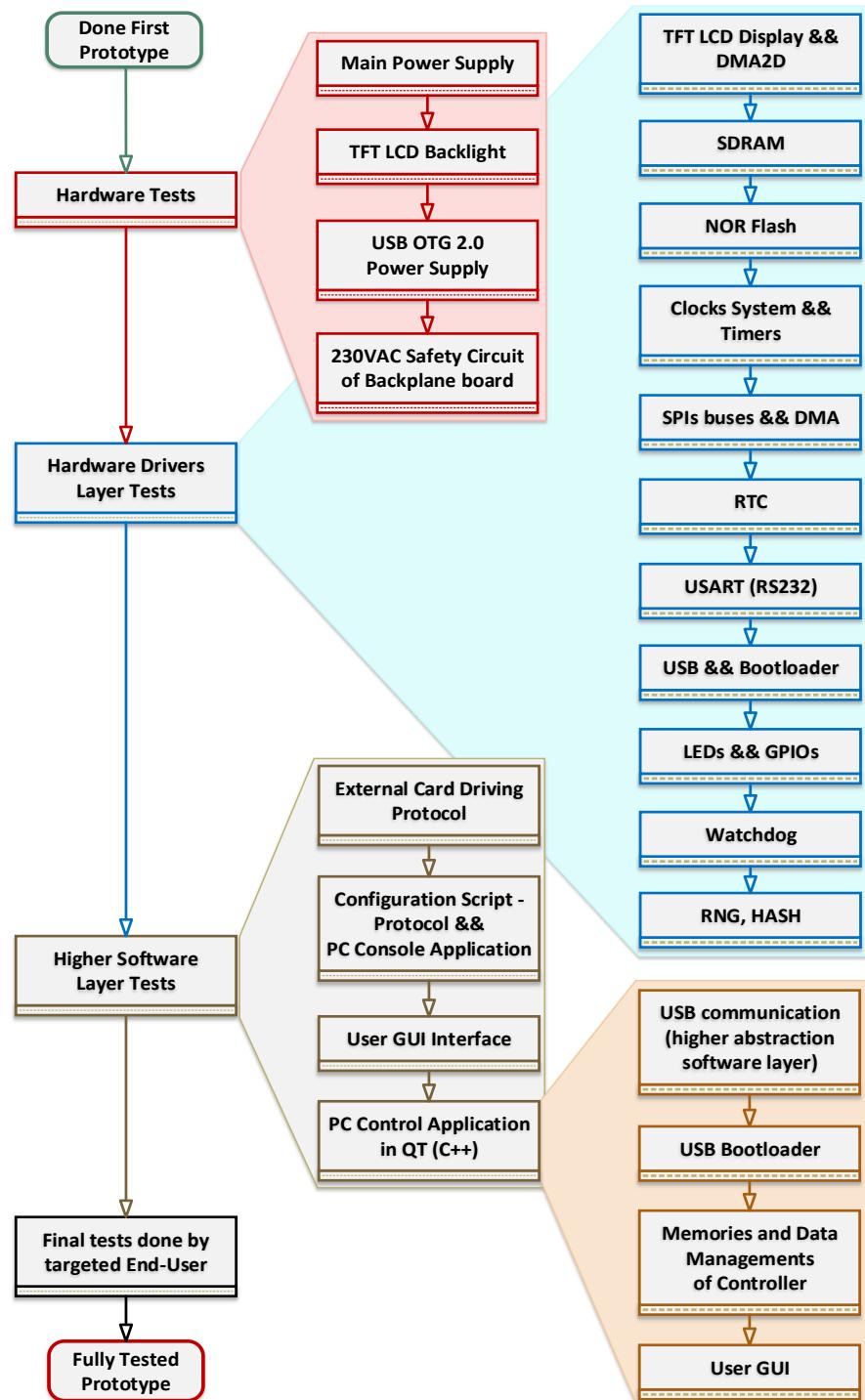


Figure 72. The test architecture block diagram [100].

Prepared tests architecture has significantly decreased the development time. All the problems have been successfully eliminated step-by-step for consecutive test parts. Detected and fixed issues of the one test part did not affect other following test parts. This approach has helped to eliminate the subset of problems sources.

7.4 Hardware tests

The Table 3 describes the hardware tests. It summarises details and used methods with descriptions of solved problems. The hardware tests have checked only the basic electrical parameters of PCBs, such as conductivity of traces or voltage correctness of power supply circuits (DC parameters, not AC properties).

Table 3. The hardware tests [100].

The main Power Supply Circuit	
Test description	➤ “ <i>Trial of fire</i> ” - test of the main step-down DC/DC converter.
Solutions & Results	➤ Test has been finished successfully. Voltmeter shown correct value of the output voltage (+3.3VDC).
The TFT LCD Backlight	
Test description	➤ “ <i>Trial of fire</i> ” - test of the boost DC/DC converter.
Problems	➤ Incorrect output voltage (approx. +3VDC instead of +19VDC). ➤ Incorrect common choke footprint on the PCB. ➤ Incorrect footprint of the common choke contributed to inverse polarisation of chip. As consequence, the chip has been burned.
Solutions & Results	➤ The switching regulator chip has been changed for new piece. ➤ The inductance of inductor (L4) of the regulator was too low and it has been changed for the 47uH inductor.

The USB Power Circuit	
Test description	<ul style="list-style-type: none"> ➤ The test of the USB boost DC/DC converter. ➤ The test of the linear voltage regulator circuit.
Problems	<ul style="list-style-type: none"> ➤ Incorrect output voltage (+5VDC). The output was approx. +0.3VDC. ➤ The DC/DC converter had high-consumption current (approx. 300mA) without any load. Additionally, it has been dissipated huge thermal power.
Solutions & Results	<ul style="list-style-type: none"> ➤ The inductor of the USB boost converter had incorrect inductance (1uH instead 2.2uH). The inductor has been changed. ➤ Short-circuit under chip. The chip had to be changed.

7.5 Hardware drivers layer tests

Table 4 presents hardware drivers tests. This table contains descriptions in details and methods used to solve problems. All of the performed tests have contributed significantly in achieving the information about the correctness of implicit AC parameters such as: clocks system frequencies and communication interface timings.

Table 4. The hardware drivers tests [100].

The TFT LCD	
Tests	<ul style="list-style-type: none"> ➤ Unit tests of random colour displaying. ➤ Unit test of random picture displaying. ➤ Unit test of random text, polyline, figures displaying. ➤ Test of all connections between the TFT LCD and the STM32. ➤ Test of automatic TFT LCD data performing by the DMA2D such as blending, dithering and automatic conversion formats.
Problems	<ul style="list-style-type: none"> ➤ Problem with uniform colour displaying – snowing effect. ➤ Problem with random picture displaying – corrupted pixels. ➤ Problem with data transmission by DMA2D – no visible data. ➤ Problem with some unconnected signals. ➤ Problem with configuration of the DMA2D interrupt.
Solutions & Results	<ul style="list-style-type: none"> ➤ Unsoldered IO ports – checked and repaired. ➤ The unit test of data displaying from the internal memory instead of external memory – detected problem with the SDRAM driver. ➤ Snowing effect and data transmission problems – caused by incorrect configurations of the DMA2D and SDRAM drivers.

The SDRAM	
Tests	<ul style="list-style-type: none"> ➤ Unit tests of random values writing into random memory cells. ➤ Advanced memory test, designed for looking for bug sectors.
Problems	<ul style="list-style-type: none"> ➤ Unit test problem - written and read values were not the same. ➤ Problems with frequency of the clock (60Mhz instead of 90MHz). ➤ Problems with unconnected signal lines. ➤ Problem with driving of older signals byte of the data bus.
Solutions & Results	<ul style="list-style-type: none"> ➤ Unconnected ports and a broken signal trace – detected and solved. ➤ The broken signal wire – the temporary solution has been used. ➤ Problem with the chip driving – incorrect bank address of the FMC.
The NOR Flash	
Tests	<ul style="list-style-type: none"> ➤ Unit tests of random values writing into random memory cells. ➤ Advanced memory test, designed for looking for bug sectors. ➤ Unit test of the ID chip checking. ➤ Test of data erasing and keeping.
Problems	<ul style="list-style-type: none"> ➤ Unit test problem - written and read values were not equalled. ➤ Problems with achieving correct chip's ID. ➤ Problems with unconnected signals. ➤ Problem with chip erasing.
Solutions & Results	<ul style="list-style-type: none"> ➤ Couple unconnected IO ports – detected and solved. ➤ Problem with the chip driving – incorrect chip's driver configuration.
The STM32 clock system	
Tests	<ul style="list-style-type: none"> ➤ Test of the PLL, based on the external 8 MHz source. ➤ Test of the RTC clock, based on the external 32.768kHz source.

Problems	<ul style="list-style-type: none"> ➤ Problem with the PLL driving. ➤ Problem with the RTC driving. ➤ Problem with footprints – exchanged quartz footprints with each other.
Solutions & Results	<ul style="list-style-type: none"> ➤ The placements of the clock sources have been changed with each other. ➤ The exchange of the clock source has been solved any problems.
SPI buses with DMAs	
Tests	<ul style="list-style-type: none"> ➤ Unit tests of random values sending and receiving.
Problems	<ul style="list-style-type: none"> ➤ Problem with unit test passing - different sent and received values. ➤ Problem with correct clock frequency achieving. ➤ Problem with communication card SPI driving. ➤ Problem with the CRC hardware unit.
Solutions & Results	<ul style="list-style-type: none"> ➤ Unconnected IO port of STM32 and buffers – detected and solved. ➤ The two-directional buffer of communication card has been repaired. ➤ Incorrect configurations of SPI drivers – detected and solved.
RTC	
Tests	<ul style="list-style-type: none"> ➤ Tests of backup register value keeping during absence power supply. ➤ Tests of the A and B type alarms with interrupts.
Problems	<ul style="list-style-type: none"> ➤ Problem with setting and changing of time and date registers. ➤ Problem with the alarm interrupt generating. ➤ Problem with running and configuration of the RTC PLC clock.
Solutions & Results	<ul style="list-style-type: none"> ➤ The backup registers hold values during absence of the main power. ➤ Exchange of clock source has been resolved other problems.

USART	
Tests	<ul style="list-style-type: none"> ➤ Unit test of random data sending. ➤ Advanced test of communication with use of the PC console app. ➤ Tests of the DMA streams.
Problems	<ul style="list-style-type: none"> ➤ Problem with unit test - incorrect received values. ➤ Problem with the DMA streams configurations. ➤ Problem with achieving of correct timing parameters. ➤ Problem with shifted data of the receiver channel.
Solutions & Results	<ul style="list-style-type: none"> ➤ Incorrect configuration of the baudrate registers – detected and solved. ➤ Shifted bytes of the receiver channel - solved by the “flushing” serial port (suitable communication flags clearing).
Watchdog	
Tests	<ul style="list-style-type: none"> ➤ The STM32 fake suspending states generating.
Problems	<ul style="list-style-type: none"> ➤ Too more generated reset signals than expected.
Solutions & Results	<ul style="list-style-type: none"> ➤ Problem with configuration of the timing register.

7.6 Software test

This chapter presents a short description of the software tests, dedicated to check the performance of all implemented software parts, such as: an external cards protocol, a PC application, the GUI and the USB features.

7.6.1 External card protocol

Figure 73 shows the block diagram of an external card protocol test. An additional test board (the STM32F429 Discovery board, shown in Figure 18), with only four SPIs (limitation of test board) has been used to emulate external cards.

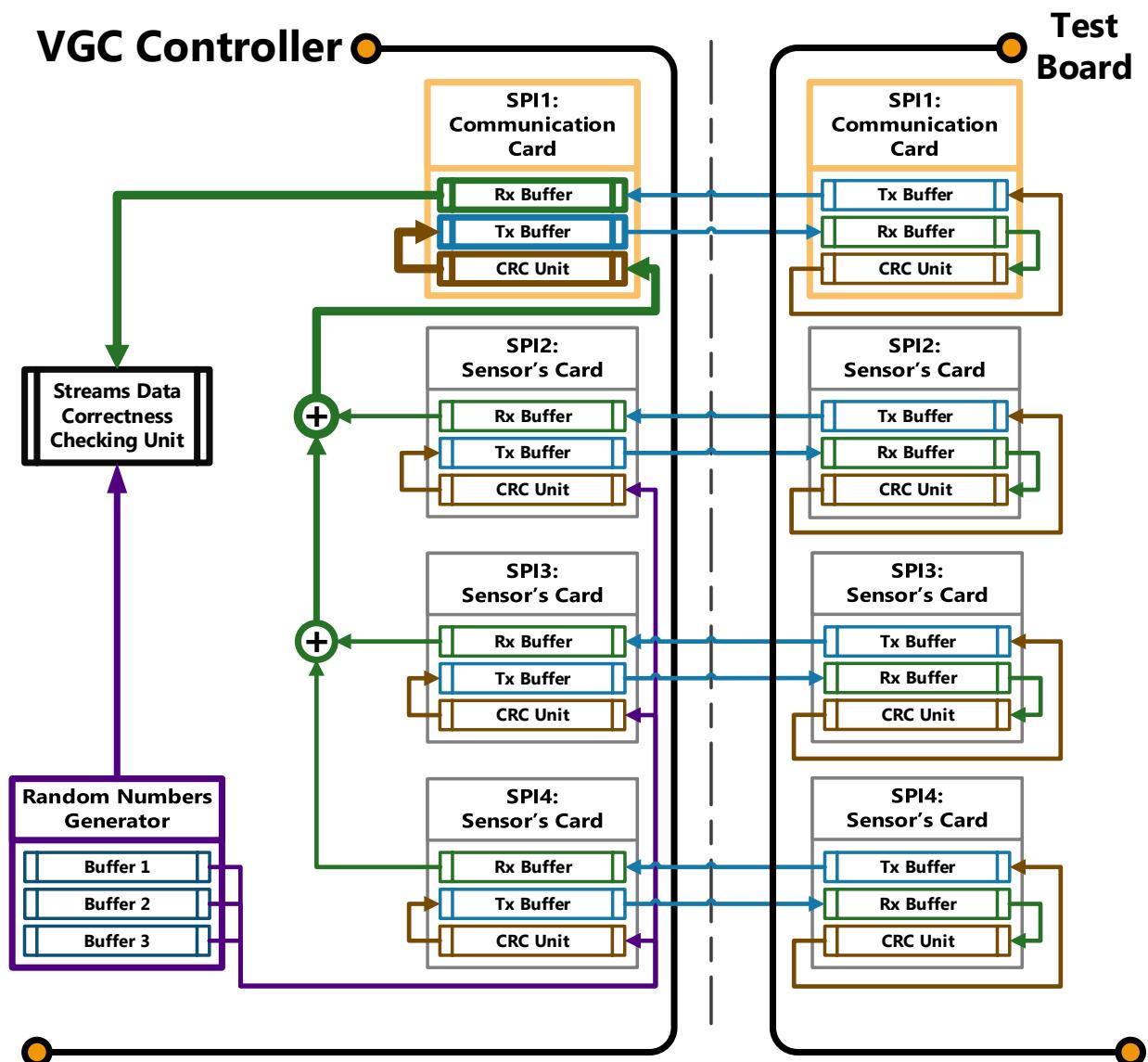


Figure 73. The external card protocol test block diagram [100].

Vacuum measurements do not require higher sampling frequency than 100 – 1000Hz, however the test has been done for 10 kHz sampling frequency. The clock speed of the SPI, dedicated to emulate a communication card, has been two times higher than other SPIs (respectively approx. 3MHz and 1.5MHz for other cards).

The continuous streams of data have been used to verify the correctness of the implemented SPI protocol. The test has been performed for eight hours. During the tests, the numbers of correctly and incorrectly received data frames have been displayed on the TFT LCD (Figure 74). Feedback streams of the test board are always shifted by one frame backwards (details in the chapter 6.2).

During testing, a couple of problems have occurred:

- losing single frames or bytes,
- receiving random values,
- shifted bits and bytes of receiving frames,
- problem with driving SPI lines for higher frequencies,
- unexpected triggering of SPIs interrupts.

A dedicated setup (Figure 74) has been set up using long ribbon cables (not recommended for higher frequencies). With these cables, it was not possible to drive SPIs lines with the maximum possible speed. The noises and oscillation levels have been too high to achieve reliable communication, however decreasing the maximum SPI clock frequency a couple of times allowed to solve the majority of communication related problems.

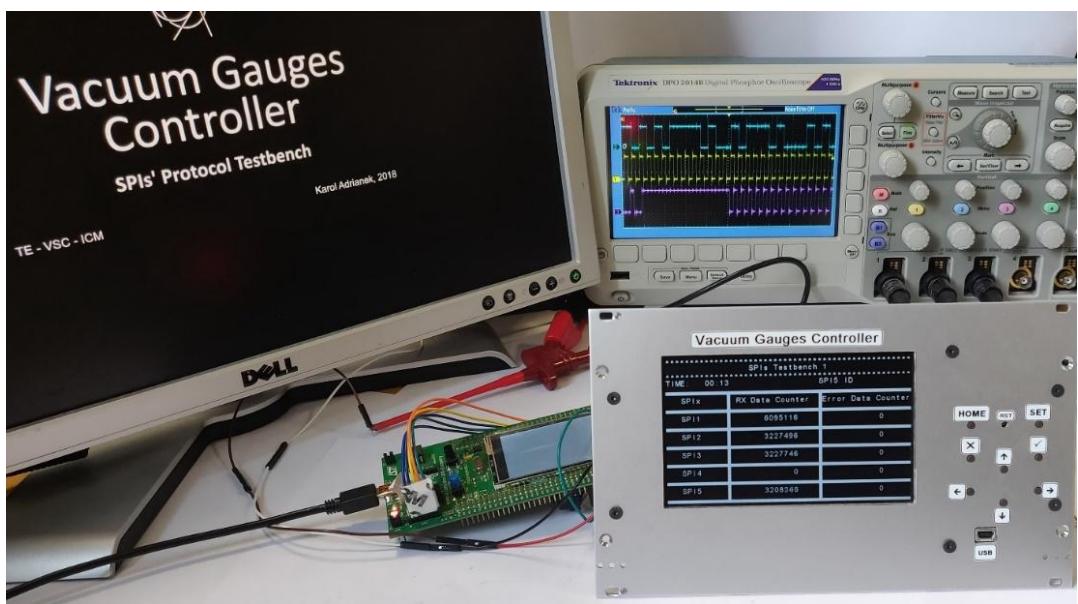


Figure 74. SPIs' protocol testbench setup (during testing) [100].

7.6.2 PC C++ console application

The multithreading console application has been verified by the continuous stream of the TFT LCD templates data, loaded to the controller and displayed on the TFT LCD matrix. The tests have been performed also for eight hours. The reliability, communication correctness and standard graphic files conversion have been checked. The test has been performed for the maximum default baud rate, which equals 256kbps.

During testing, a couple of problems has occurred:

- losing random single bytes or data frame,
- receiving shifted data into frame structure,
- losing received char flag during overlapped event (PC side),
- problems with synchronisation between receiving and transmitting threads,
- problems with entering and leaving the critical section for the port handler (PC side),
- receiving incorrect values of the CRC sum byte,
- problems with the app stack caused by the Stbi library during debugging mode,
- problems with the PC system events (overlapped error handlers),
- the receiver and transmitter threads blocking each other,
- errors with creating and handling system events and files (for PC system, the communication port and its events are considered as a file),
- problems with interrupting the DMA stream.



Figure 75. The PC console application testbench setup [100].

A user of the Stbi library should be very careful when applying this library. The image conversions to the pixel array is performed by the allocating stack, this has been problematic in the debugging mode [93]. Each application of a PC system has a maximum operational stack size. For larger pictures, the problem with stack overflow might occur. For the TFT LCD, the application operates correctly for the Window system.

7.6.3 GUI

The GUI test main goal was to check, if all implemented features of the GUI software are compliant with the architecture presented in the chapter 6.4.

During testing the following GUI features and properties have been checked such as:

- the “welcome” loading bar during initialisation of the controller,
- the visibility of data displaying, fonts, menu items backlight,
- transparency of the TFT LCD operational layers dependency on the menu items,
- the correctness of the XY coordinates on the TFT LCD matrix dependency on the menu items,
- the quality of the loaded TFT LCD templates to the controller,
- correctness of the defined operational functions for each user button.

The test results have given precious feedback about the coherency of the GUI software implementation with designed architecture. During tests, some insignificant problems have occurred and they were solved quickly.

7.7 Final tests

The final set of tests have been performed to verify functional user-friendliness of the controller. Moreover, the details of displayed messages (such as fonts size, visibility, colours, transparency, backlight of menu items), GUI, user buttons and a PC application functionalities have been verified. The end-users (technician or engineer of the TE-VSC maintenance department at CERN) have given their feedbacks if all the requirements are fulfilled and which options/menu should be changed. Based on the tests results, some small modifications were implemented. The first controller’s version has covered majority of the targeted end-users requirements. The tests have contributed in achieving a complete VGC prototype.

8 Summary

The master thesis presents the new version of the Vacuum Gauges Controller, designed and made for the LHC vacuum installations. The aims of this graduate thesis “*The design and prototype of the Vacuum Gauges Controller for the high and ultra-high vacuum measurements for the Large Hadron Collider at CERN*” were fulfilled successfully.

This graduation thesis is composed of six chapters, dealing with different aspects of designing and prototyping of an universal controller for the largest vacuum installation in the world [7]. Chapter 1 is an introduction describing the CERN accelerator complex. Chapter 2 briefly reviews the vacuum system and the place of the designed controller in the vacuum controls hierarchy. Chapter 3 defines the architecture of the VGC controller. Chapter 4 concentrates on the used hardware components and solutions, with short descriptions of advantages, disadvantages and operational principles. Chapter 5 and 6 deal with the software implementation with dedicated software architecture, algorithms and protocols used. Chapter 7 presents short descriptions of performed tests on the controller’s hardware, firmware and software parts. The documentation part of the controller (electronics schematics, PCB designs, 3D models, the prototype’s photos) is available in the attachments (chapter 9).



Figure 76. The Vacuum Gauges Controller –final version [100].

The practical part of this thesis was conducted from September 2017 up to September 2018 at CERN. The VGC prototype is operational and can serve existing and future vacuum controls. Additionally, it is a good starting point for further developments of newer controllers. Thanks to its versatility of architecture, methodology and applied solutions, it provides an amazing opportunity for further upgrades.

The author suggests that not all implemented features and functions are operating with the highest performance and some aspects could be improved. Unfortunately, some planned features have not been implemented, due to highly limited development time.

Bibliography

- [1] CERN, "About CERN," CERN, [Online]. Available: <https://home.cern/about>. [Accessed 17 04 2018].
- [2] CERN, "A Brief history of CERN," [Online]. Available: <http://cds.cern.ch/journal/CERNBulletin/2011/10/News%20Articles/1331510>. [Accessed 17 04 2018].
- [3] CERN, "The Large Hadron Collider," [Online]. Available: <https://home.cern/topics/large-hadron-collider>. [Accessed 02 05 2018].
- [4] CERN, "Subatomic particles," [Online]. Available: <https://home.cern/about/physics/subatomic-particles>. [Accessed 01 08 2018].
- [5] CERN, "Member states," [Online]. Available: <https://home.cern/about/member-states>. [Accessed 02 08 2018].
- [6] CERN, "Powering CERN," [Online]. Available: <https://home.cern/about/engineering/powering-cern>. [Accessed 02 08 2018].
- [7] CERN, "Cern Accelerator Complex," [Online]. Available: <http://te-epc-lpc.web.cern.ch/te-epc-lpc/machines/pagesources/Cern-Accelerator-Complex.jpg> . [Accessed 02 05 2018].
- [8] CERN, "A vacuum as empty as interstellar space," [Online]. Available: <https://home.cern/about/engineering/vacuum-empty-interstellar-space>. [Accessed 02 08 2018].
- [9] CERN, "Cryogenics: Low temperatures, high performance," [Online]. Available: <https://home.cern/about/engineering/cryogenics-low-temperatures-high-performance>. [Accessed 02 08 2018].
- [10] CERN, "Vacuum, Surfaces and Coatings," [Online]. Available: <http://te-dep.web.cern.ch/content/vacuum-surfaces-and-coatings>. [Accessed 02 05 2018].
- [11] CERN, "Technology Department," [Online]. Available: https://edms.cern.ch/ui/file/1887397/1/TE_August_2018.pdf. [Accessed 02 05 2018].
- [12] International Organization for Standardization, "Standard catalogue. 31 - Electronics.," [Online]. Available: <https://www.iso.org/ics/31/x/>. [Accessed 12 05 2018].
- [13] IPC, "IPC Standards Tree," [Online]. Available: https://www.ipc.org/4.0_Knowledge/4.1_Standards/SpecTree.pdf. [Accessed 14 05 2018].

- [14] STMicroelectronics, "STM32F427xx STM32F429xx. Datasheet.,," [Online]. Available: www.st.com/resource/en/datasheet/stm32f429ve.pdf. [Accessed 18 06 2018].
- [15] Midas Components Limited, "MCT050TC12W800480LML Technical Documentation.,," [Online]. Available: www.midasdisplays.com . [Accessed 02 06 2018].
- [16] Schroff, "Main Catalog. Frame type plug-in units - Kits.,," [Online]. Available: https://schroff.nvent.com/wcsstore/ExtendedSitesCatalogAssetStore/Attachment/SchroffAttachments/Documents/7_3_Kassetten_e.pdf. [Accessed 27 04 2018].
- [17] Eurocircuits, "PCB Design Guidelines," [Online]. Available: [https://www.eurocircuits.com/pcb-design-guidelines/](http://www.eurocircuits.com/pcb-design-guidelines/). [Accessed 13 05 2018].
- [18] CERN (Geneve) and IHEP (Protvino), "The control system of CERN accelerators vacuum.,," [Online]. Available: <http://accelconf.web.cern.ch/AccelConf/icalepcs2011/papers/mopms016.pdf>. [Accessed 14 05 2018].
- [19] P. Gomes, "Vacuum Controls - Portuguese Language Teachers Program," [Online]. Available: https://indico.cern.ch/event/577125/contributions/2676180/attachments/1518403/2372512/VacuumControls_PTteachers_2017.pdf. [Accessed 12 05 2018].
- [20] The Institute of Measurement and Control, Guide to the Measurement of Pressure and Vacuum., London: National Physical Laboratory, 1998.
- [21] P. Collon, "Introduction to vacuum gauges.,," 13 11 2008. [Online]. Available: [https://www3.nd.edu/~nsl/Lectures/urls/Introduction_to_vacuum_gauges.pdf](http://www3.nd.edu/~nsl/Lectures/urls/Introduction_to_vacuum_gauges.pdf). [Accessed 15 05 2018].
- [22] Pfeiffer, "TPG 300, basic unit," [Online]. Available: <https://www.pfeiffer-vacuum.com/en/products/measurement-analysis/measurement/modulline/controllers/?detailPdoId=3407>. [Accessed 05 05 2018].
- [23] Pfeiffer, *The TPG300 - Schematics (Confidential for CERN)*, Pfeiffer, 1987.
- [24] Volotek SA, "The VGC1000. Ultra High Precision Vacuum Gauge Controller.,," [Online]. Available: <http://www.volotek.com/component/virtuemart/high-precision-instruments/extreme-high-vacuum-measurement/vgc1000-detail?Itemid=0>. [Accessed 29 04 2018].
- [25] Volotek SA, *VGC1000 - Vacuum Gauge Controller - Schematics E040212-C*, Geneve: Volotek SA, 2006-2008.
- [26] PROFIBUS and PROFINET International, "Profibus and Profinet. Organization & Community.,," [Online]. Available: [https://www.profibus.com](http://www.profibus.com). [Accessed 17 03 2018].

- [27] Cisco, "Ethernet Technologies," [Online]. Available: http://docwiki.cisco.com/wiki/Ethernet_Technologies. [Accessed 29 07 2018].
- [28] RS Components Ltd, "Ventilated Plug-in Unit with Handles, 3U, 42hp, 167mm Deep," [Online]. Available: <https://uk.rs-online.com/web/p/rackmount-enclosures/4421653/>. [Accessed 15 05 2018].
- [29] Farnell element14, "05.03.401 - Standoff, Zinc Plated, Steel, M3, Hex Female, 40 mm, 05.03 Series," [Online]. Available: <http://uk.farnell.com/ettinger/05-03-401/spacer-m3x40-vzk/dp/1466833> . [Accessed 05 05 2018].
- [30] Farnell element14, "ESQ-110-44-T-D - Board-To-Board Connector.," [Online]. Available: <http://uk.farnell.com/samtec/esq-110-44-t-d/socket-2-54mm-vertical-tht-20way/dp/1930327>. [Accessed 05 05 2018].
- [31] Farnell element14, "TSW-110-21-T-D - Board-To-Board Connector," [Online]. Available: <http://uk.farnell.com/samtec/tsw-110-21-t-d/header-2-54mm-tht-vert-20way/dp/2029035>. [Accessed 05 05 2018].
- [32] Farnell element14, "Harting 09 03 232 6825 - DIN 41612 Connector," [Online]. Available: http://uk.farnell.com/harting/09-03-232-6825/socket-din41612-c-32way/dp/1096908?ost=09+03+232+6825&ddkey=http%3Aen-GB%2FElement14_United_Kingdom%2Fsearch. [Accessed 05 05 2018].
- [33] Budnick Converting Inc., "Tape101 a comprehensive guide to adhesive tape properties and selection.,," [Online]. Available: https://www.budnick.com/cms_uploads/2016/11/tape101-ebook.pdf. [Accessed 27 07 2018].
- [34] Rapid Electronics Limited, [Online]. Available: <https://www.rapidonline.com/st-stm32f429bit6-microcontroller-32-bit-arm-cortex-m4-180mhz-2048kb-lqfp-208-73-3443> . [Accessed 16 05 2018].
- [35] STMicroelectronics, "UM1562 User manual. Getting started with software and firmware environment for the STM32F3DISCOVERY Kit.,," 09 2012. [Online]. Available: https://www.st.com/resource/en/user_manual/dm00062662.pdf. [Accessed 17 06 2018].
- [36] STMicroelectronics, "UM1734. User manual. STM32Cube USB device library.,," 05 2015. [Online]. Available: https://www.st.com/resource/en/user_manual/dm00108129.pdf. [Accessed 01 07 2018].
- [37] K. K. J. Scott Gardner, A Guide to CPU Cores and Processor IP. FOcusing on CPU and GPU. Fourth E, The Linley Group, 2013.
- [38] T. P. Morgan, "AMD To Unify X86, ARM Systems With SkyBridge," EnterpriseTech. Tabor Communications Inc., 04 05 2014. [Online]. Available: <https://www.enterprisetech.com/2014/05/05/amd-unify-x86-arm-systems-skybridge/>. [Accessed 09 05 2018].

- [39] Xiaomi Corporation, "Xiaomi Redmi 5A Specs.," [Online]. Available: <https://www.mi.com/en/redmi-5a/specs/>. [Accessed 09 05 2018].
- [40] Qualcomm, "Snapdragon 425 Processor," [Online]. Available: <https://www.qualcomm.com/products/snapdragon/processors/425>. [Accessed 09 05 2018].
- [41] ARM, "Cortex-M4. Technical Reference Manual.," 2009, 2010. [Online]. Available: http://infocenter.arm.com/help/topic/com.arm.doc.ddi0439b/DDI0439B_cortex_m4_r0p0_trm.pdf. [Accessed 14 06 2018].
- [42] STMicroelectronics, "RM0090 Reference manual.," [Online]. Available: https://www.st.com/resource/en/reference_manual/dm00031020.pdf. [Accessed 15 06 2018].
- [43] STMicroelectronics, "UM1670. User manual. Discovery kit with STM32F429ZI MCU.," 09 2017. [Online]. Available: https://www.st.com/resource/en/user_manual/dm00093903.pdf. [Accessed 12 06 2018].
- [44] STMicroelectronics, "Discovery kit with STM32F429ZI MCU," [Online]. Available: <https://www.st.com/en/evaluation-tools/32f429idiscovery.html> . [Accessed 16 05 2018].
- [45] Newark element14 Electronics, "MCT050TC12W800480LML - DISPLAY, LCD TFT, 5", 800X480, LANDSCAPE," [Online]. Available: <http://www.newark.com/midas/mct050tc12w800480lml/display-lcd-tft-5-800x480-landscape/dp/01AC6905>. [Accessed 12 06 2018].
- [46] STMicroelectronics, "AN4861 Application note. LCD-TFT display controller (LTDC) on STM32 MCUs.," 02 2017. [Online]. Available: https://www.st.com/resource/en/application_note/dm00287603.pdf. [Accessed 25 05 2018].
- [47] Cypress, "S29GL064N, S29GL032N, 64 Mbit, 32 Mbit 3 V Page Mode MirrorBit FLash.," 26 05 2017. [Online]. Available: <http://www.cypress.com/file/202426/download>. [Accessed 14 06 2018].
- [48] M. G. Howard Johnson, High-Speed Digital Design: A Handbook of Black Magic., Prentice Hall, 1993.
- [49] Toshiba, "NAND vs. NOR Flash Memory. Technology Overview.," [Online]. Available: http://maltiel-consulting.com/NAND_vs_NOR_Flash_Memory_Technology_Overview_Read_Write_Erase_speed_for_SLC_MLC_semiconductor_consulting_expert.pdf. [Accessed 02 06 2018].
- [50] Micron, General DDR SDRAM Functionality, Technical Note, TN-46-05., Micron, 2001.

- [51] Micron, "MT48LC16M16A2P-6A Datasheet.,," [Online]. Available: <https://www.micron.com/products/datasheets/5917c8ee-3afc-4f8b-8e28-c2d65f00cc08>. [Accessed 15 06 2018].
- [52] Microchip, "24FC512 512 Kbit, I2C Serial EEPROM.,," 2010. [Online]. Available: <http://www.farnell.com/datasheets/575340.pdf>. [Accessed 14 06 2018].
- [53] NXP Semiconductors, "UM10204. I2C-bus specification and user manual.,," 04 04 2014. [Online]. Available: <https://www.nxp.com/docs/en/user-guide/UM10204.pdf>. [Accessed 14 06 2018].
- [54] STMicroelectronics, "AN3371 Application note. Using the hardware real-time clock (RTC) in STM32 F0, F2, F3, F4 and L1 series of MCUs.,," 09 2012. [Online]. Available: https://www.st.com/resource/en/application_note/dm00025071.pdf. [Accessed 28 05 2018].
- [55] USB Implementers Forum, "USB On-The-Go and Embedded Host," [Online]. Available: <http://www.usb.org/developers/onthego>. [Accessed 16 06 2018].
- [56] Texas Instruments, "Integrated USB Power Switch with Boost Converter - TPS2500. Datasheet.,," 10 2006. [Online]. Available: www.ti.com/lit/ds/symlink/tps2500.pdf. [Accessed 17 06 2018].
- [57] Texas Instruments, "Fast-Transient Response 500mA LDO Voltage TPS776xx - Datasheet.,," 03 2009. [Online]. Available: www.ti.com/lit/ds/symlink/tps776.pdf. [Accessed 18 06 2018].
- [58] Texas Instruments, "Understanding the Efficiency of an LDO," [Online]. Available: <http://www.ti.com/lit/ml/slup239/slup239.pdf>. [Accessed 19 06 2018].
- [59] W. H. Paul Horowitz, The Art of Electronics. Third Edition., New York, USA: Cambridge University Press, 2015.
- [60] Wurth Electronics, "How Common Mode Chokes Work," [Online]. Available: https://www.we-online.com/web/en/passive_components_custom_magnetics/blog_pbcm/blog_detail_electronics_in_action_61439.php. [Accessed 16 06 2018].
- [61] P. N. A. Wright, Electric Fuses. Third Edition., London, United Kingdom: The Institution of Engineering and Technology, 2004.
- [62] STMicroelectronics, "The L5972D 2A switch step down switching regulator.,," 10 2007. [Online]. Available: <https://www.st.com/resource/en/datasheet/l5972d.pdf>. [Accessed 16 06 2018].
- [63] S. Roberts, DC/DC Book of Knowledge. Practical tips for the User. Third Edition., RECOM Engineering GmbH, 2016.

- [64] Linear Technology, "LT3466 - Datasheet.,," [Online]. Available: www.analog.com/media/en/technical-documentation/data-sheets/3466fa.pdf. [Accessed 19 06 2018].
- [65] USB Implementers Forum, "Introduction to USB On-The-Go," [Online]. Available: USB Implementers Forum. [Accessed 16 06 2018].
- [66] Maxim Integrated, "USB On-the-Go Basics," [Online]. Available: <https://www.maximintegrated.com/en/app-notes/index.mvp/id/1822>. [Accessed 18 06 2018].
- [67] ON Semiconductor, "The STF202-22T1G USB Filter with ESD protection.,," 09 2012. [Online]. Available: [https://www.onsemi.com/pub/Collateral/STF202-22T1-D.PDF](http://www.onsemi.com/pub/Collateral/STF202-22T1-D.PDF). [Accessed 16 06 2018].
- [68] ON Semiconductor, "The NZF220TT1 EMI Filter with ESD Protection.,," 03 2006. [Online]. Available: <http://www.onsemi.com/pub/Collateral/NZF220TT1-D.PDF>. [Accessed 16 06 2018].
- [69] ARM, "ARM1136JF-S and ARM1136J-S. Technical Reference Manual.,," 2009. [Online]. Available: http://infocenter.arm.com/help/topic/com.arm.doc.ddi0211k/DDI0211K_arm1136_r1p5_trm.pdf. [Accessed 14 06 2018].
- [70] STMicroelectronics, "UM1075 User manual. ST-LINK/V2 in-circuit debugger/programmer for STM8 and STM32.,," 03 2016. [Online]. Available: [https://www.st.com/resource/en/user_manual/dm00026748.pdf](http://www.st.com/resource/en/user_manual/dm00026748.pdf). [Accessed 14 06 2018].
- [71] ARM, "Serial WIre Debug and the CoreSight Debug and Trace Architecture.,," [Online]. Available: [https://www.arm.com/files/pdf/Serial_Wire_Debug.pdf](http://www.arm.com/files/pdf/Serial_Wire_Debug.pdf). [Accessed 14 06 2018].
- [72] Texas Instruments, "The SN74LVC2T45 2-Bit Dual Supply Transceiver with Configurable Voltage-Level Shifting and 3-State Outputs," [Online]. Available: <http://www.ti.com/product/SN74LVC2T45>. [Accessed 16 04 2018].
- [73] Texas Instruments, "The SN74LVC2G17 Dual Schmitt-Trigger Buffer datasheet (Rev. N.),," [Online]. Available: <http://www.ti.com/product/SN74LVC2G17?keyMatch=SN74LVC2G17DBVR&tisearch=Search-EN-Everything>. [Accessed 16 04 2018].
- [74] Electronics StackExchange, "What is the purpose of a buffer gate?," [Online]. Available: <https://electronics.stackexchange.com/questions/236666/what-is-the-purpose-of-a-buffer-gate>. [Accessed 26 08 2018].
- [75] STMicroelectronics, "STM32F7 - RCC. Reset and clock control.,," [Online]. Available: https://www.st.com/content/ccc/resource/training/technical/product_training/group0/c

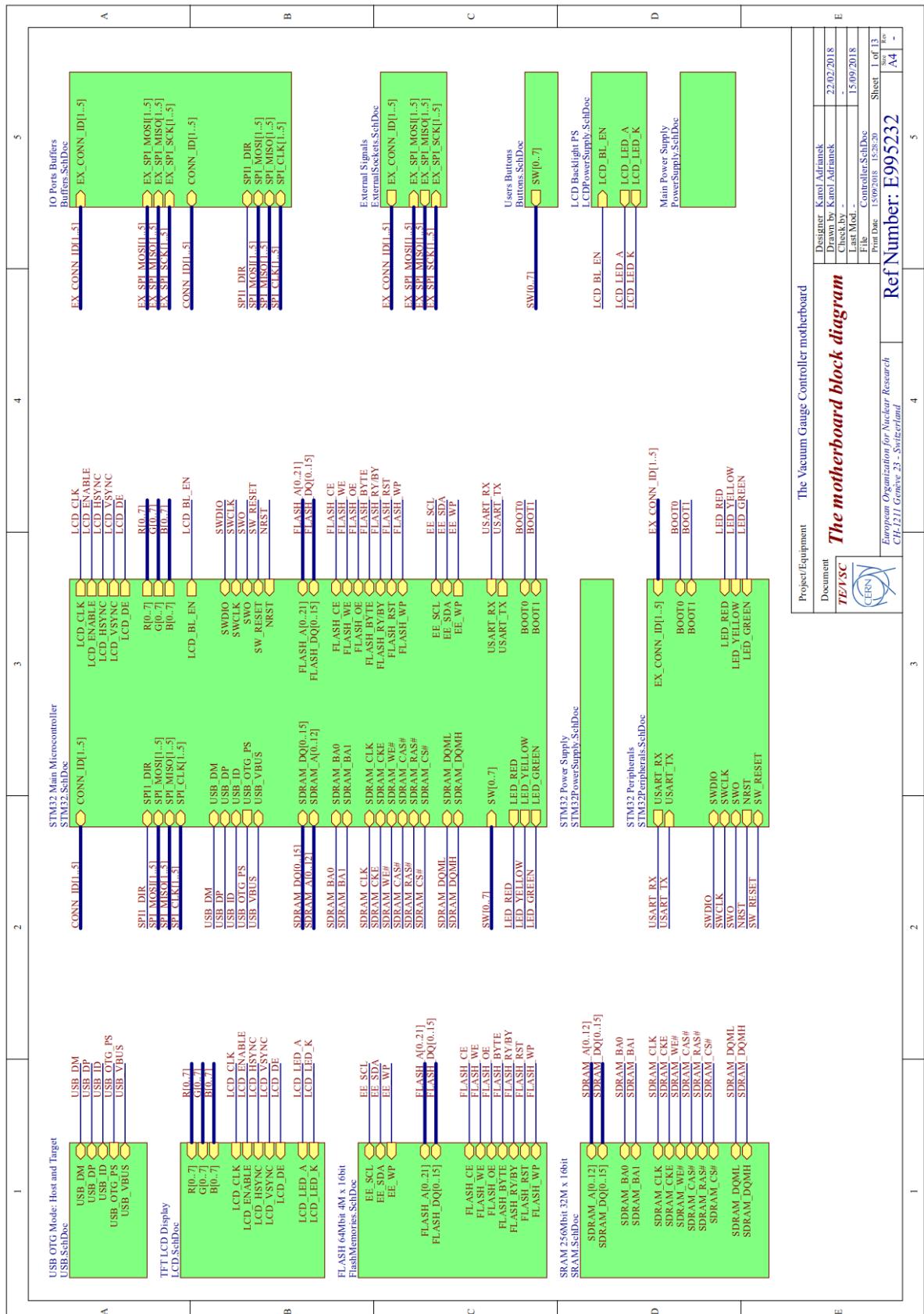
- 8/9e/ff/ac/7a/75/42/d1/STM32F7_System_RCC/files/STM32F7_System_RCC.pdf/_j_cr_content/translations/en.STM32F7_System_RCC.pdf. [Accessed 23 06 2018].
- [76] Texas Instruments, "Crystal Oscillator and Crystal Selection for the CC26xx and CC13xx Family of Wireless MCUs.," 12 2015. [Online]. Available: <http://www.ti.com/lit/an/swra495f/swra495f.pdf>. [Accessed 17 06 2018].
 - [77] Qantek Technology Corporation, "QC5CA Series - Datasheet.," [Online]. Available: https://www.qantek.com/tl_files/products/crystals/QC5CA.pdf. [Accessed 19 06 2018].
 - [78] Qantek Technology Corporation, "QTC4 Series - Datasheet.," [Online]. Available: https://www.qantek.com/tl_files/products/crystals/QTC4.pdf. [Accessed 19 06 2018].
 - [79] Maxim Integrated, "MAX6816/MAX6817/ MAX6818 Datasheet," [Online]. Available: <https://datasheets.maximintegrated.com/en/ds/1896.pdf>. [Accessed 01 06 2018].
 - [80] J. G. Ganssle, "A Guide to Debouncing," 2004. [Online]. Available: <https://pubweb.eng.utah.edu/~cs5780/debouncing.pdf>. [Accessed 10 06 2018].
 - [81] E. Nisley, "Contact Bounce: Why Capacitors Don't Fix It," 13 07 2012. [Online]. Available: <https://softsolder.com/2012/07/13/contact-bounce-why-capacitors-dont-fix-it/>. [Accessed 05 06 2018].
 - [82] R. C. Martin, *Clean Code : A Handbook of Agile Software Craftsmanship*, Pearson Education (US) , 2009.
 - [83] STMicroelectronics, *STM32CubeMX. STM32F429BIT6x Clock Configuration.*, STMicroelectronics.
 - [84] J. Corbet, A. Rubini and G. Kroah-Hartman, "Chapter 10. Interrupt Handling," in *Linux Device Drivers, Third Edition.*, O'Reilly Media, 2005, p. 258.
 - [85] Atolic, "TrueSTUDIO," [Online]. Available: <https://atollic.com/truestudio/>. [Accessed 17 06 2018].
 - [86] STMicroelectronics, "PM0214 Programming manual.," 10 2017. [Online]. Available: https://www.st.com/resource/en/programming_manual/dm00046982.pdf. [Accessed 14 06 2018].
 - [87] STMicroelectronics, "STM32F7 - SPI.," [Online]. Available: https://www.st.com/content/ccc/resource/training/technical/product_training/group0/3e/ee/cd/b7/84/4b/45/ee/STM32F7_Peripheral_SPI/files/STM32F7_Peripheral_SPI.pdf/_jcr_content/translations/en.STM32F7_Peripheral_SPI.pdf. [Accessed 17 06 2018].
 - [88] STMicroelectronics, "STM32L4 - USART.," [Online]. Available: https://www.st.com/content/ccc/resource/training/technical/product_training/4f/cc/bb/f7/04/99/41/66/STM32L4_Peripheral_USART.pdf/files/STM32L4_Peripheral_USA

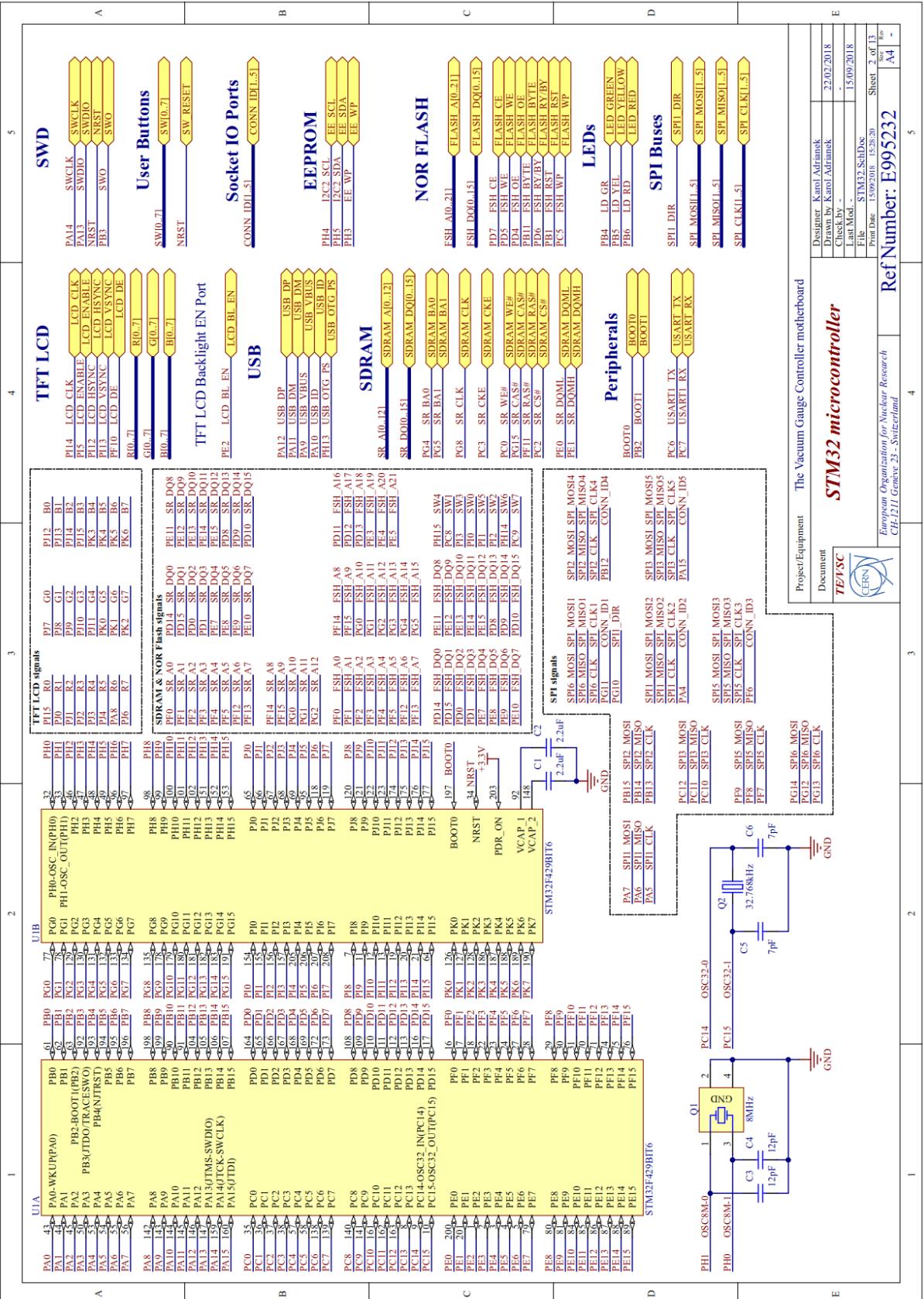
- RT.pdf/jcr:content/translations/en.STM32L4_Peripheral_USART.pdf. [Accessed 07 06 2018].
- [89] STMicroelectronics, "UM0424. User manual. STM32 USB-FS-Device development kit.,," 12 2012. [Online]. Available: https://www.st.com/resource/en/user_manual/cd00158241.pdf. [Accessed 20 06 2018].
- [90] STMicroelectronics, "AN2606. Application note. STM32 microcontroller system memory boot mode.,," [Online]. Available: https://www.st.com/resource/en/application_note/CD00167594.pdf. [Accessed 21 06 2018].
- [91] STMicroelectronics, "AN3156 Application note. USB DFU protocol used in the STM32 bootloader.,," 03 2017. [Online]. Available: https://www.st.com/resource/en/application_note/cd00264379.pdf. [Accessed 27 05 2018].
- [92] Microsoft Corporate, "Context Switches," Microsoft Corporate, 31 05 2018. [Online]. Available: <https://docs.microsoft.com/en-us/windows/desktop/ProcThread/context-switches>. [Accessed 01 08 2018].
- [93] S. T. Barrett, "Stb single-file public domain libraries for C/C++," [Online]. Available: <https://github.com/nothings/stb>. [Accessed 03 06 2018].
- [94] Git, "Git logos.,," [Online]. Available: <https://git-scm.com/downloads/logos>. [Accessed 29 06 2018].
- [95] Qt, "Qt Company," [Online]. Available: <https://www.qt.io/company>. [Accessed 29 06 2018].
- [96] Microsoft Corporation, "Visual Studio 2017 logo and wordmark," [Online]. Available: https://commons.wikimedia.org/wiki/File:Visual_Studio_2017_logo_and_wordmark.svg. [Accessed 29 06 2018].
- [97] B. K. Marcy Stutzman, "Deep UHV gauges. Theory and Practice.,," 21 10 2008. [Online]. Available: https://www.jlab.org/accel/inj_group/docs/2008/Deep-UHV-gauges.pdf. [Accessed 23 05 2018].
- [98] MKS Instruments, "The SRG-3 Spinning Rotor Gauge System.,," [Online]. Available: <https://www.mksinst.com/docs/ur/srg-3ds.pdf>. [Accessed 23 05 2018].
- [99] Farnell element14, "STM32F429I-DISC1 - Development Board," [Online]. Available: <https://uk.farnell.com/stmicroelectronics/stm32f429i-disc1/dev-board-advanced-line-mcu/dp/2506924?MER=sy-me-pd-mi-alte>. [Accessed 18 07 2018].

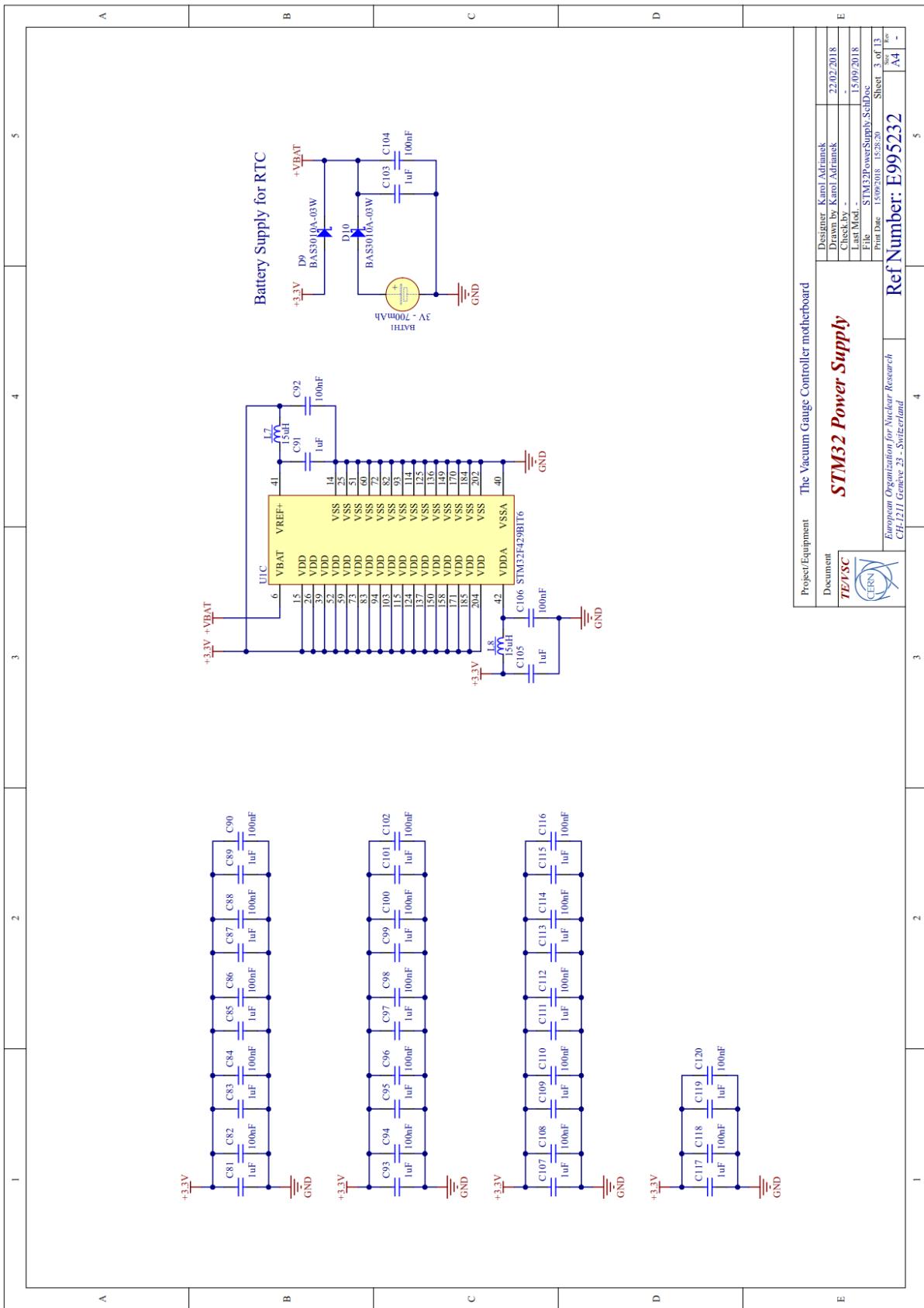
[100] Adrianek Karol, *Own elaboration*, Geneva, Switzerland, 2017-2018

9 Attachments

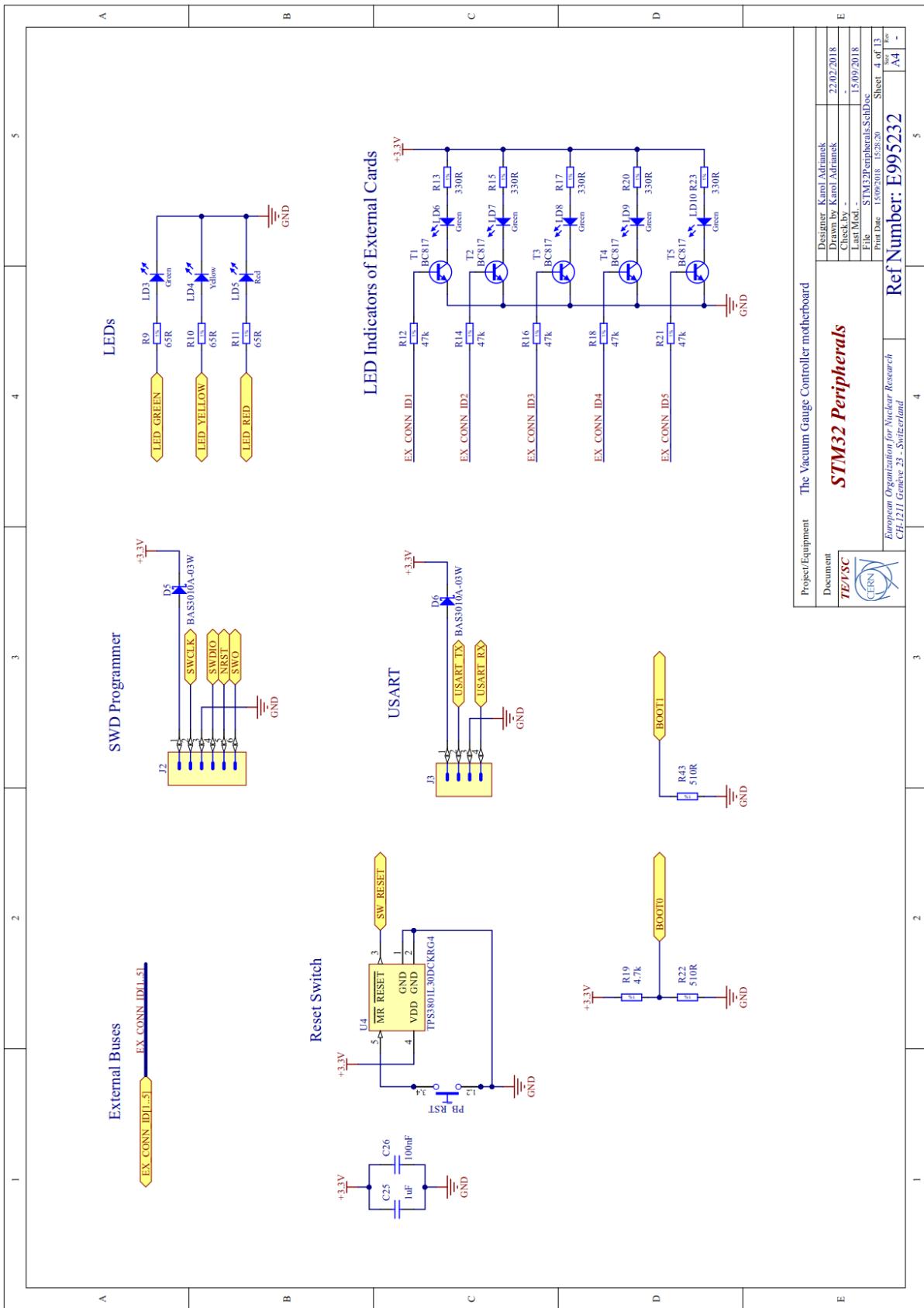
9.1 Motherboard schematics

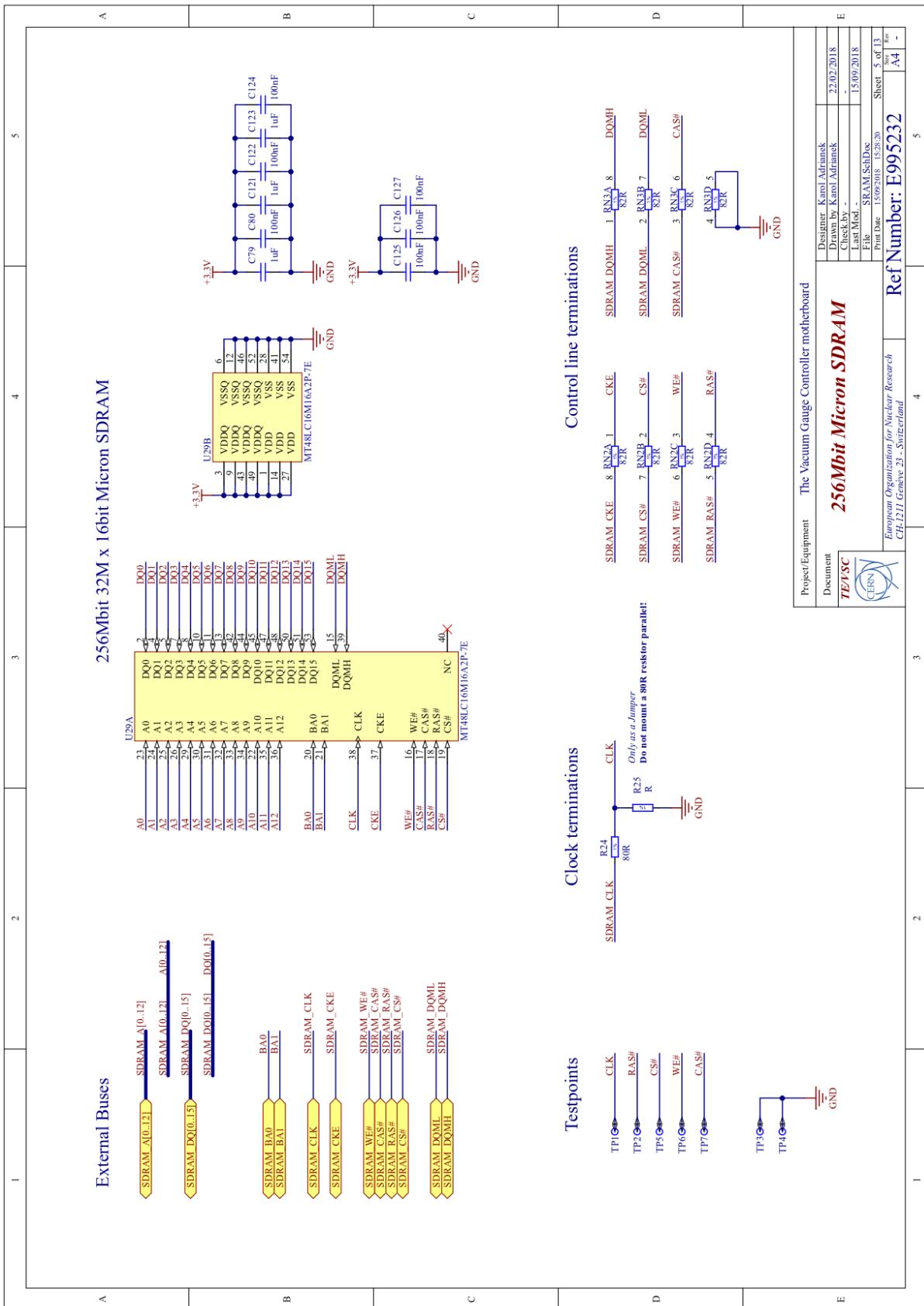


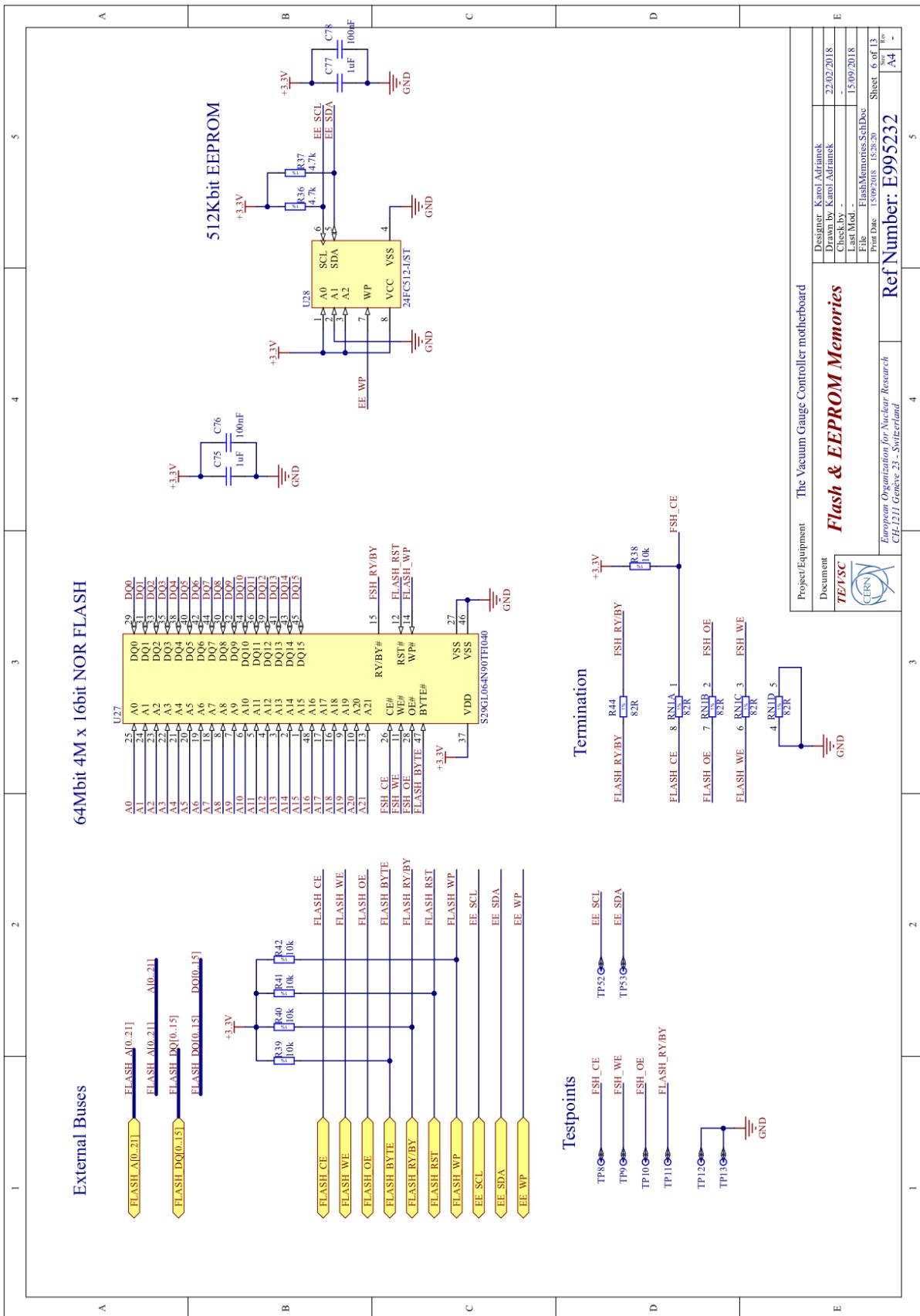


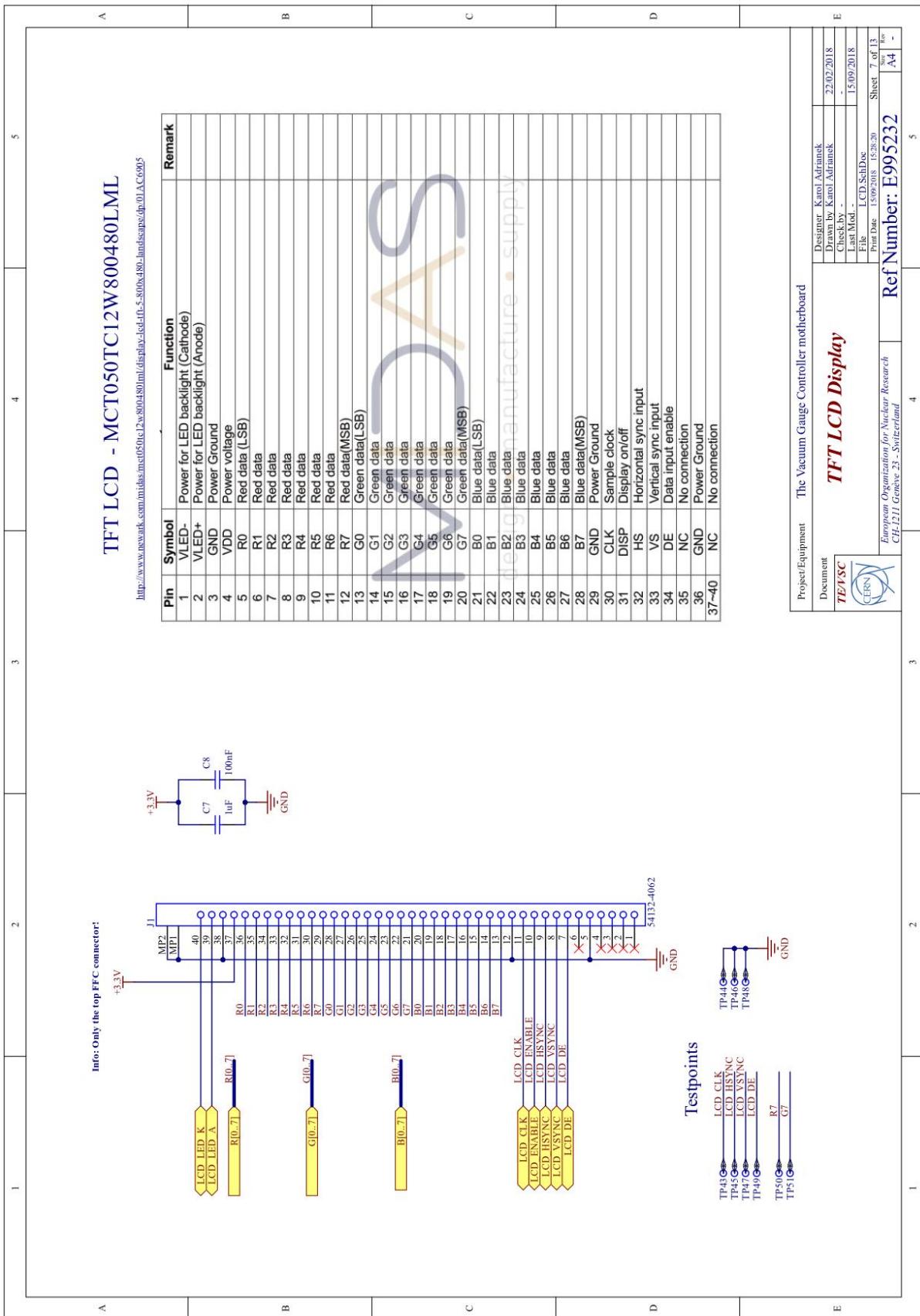


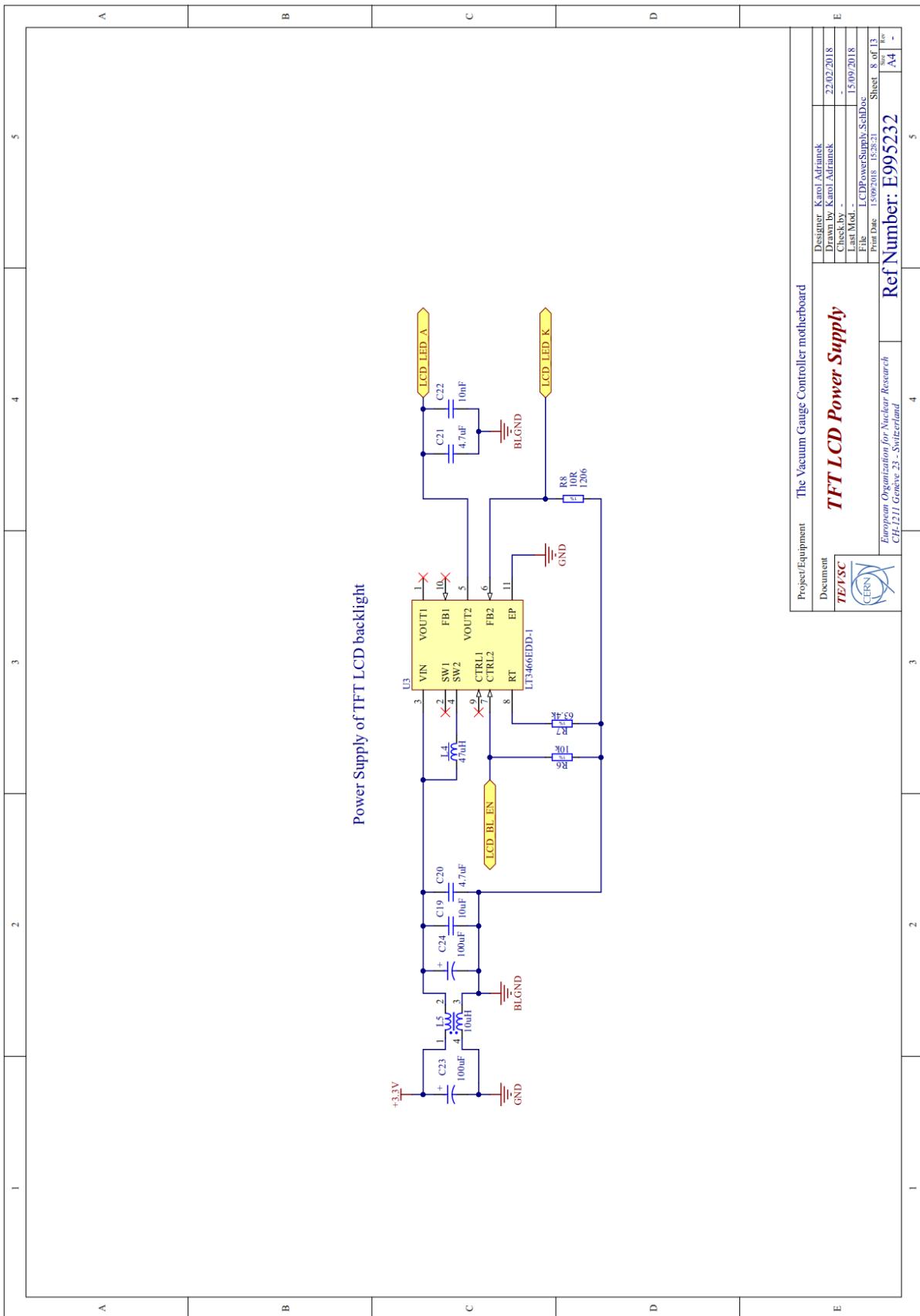
Project/Equipment		The Vacuum Gauge Controller motherboard	
Document	STM32 Power Supply	Designer: Karol Adiranek	22/02/2018
TEVSC	CERN	Drawn by: Karol Adiranek	-
		Checked by: -	22/02/2018
		Last Mod.: -	15/09/2018
		File: STM32PowerSupply-SchDoc	Sheet 3 of 13
		Print Date: 15/09/2018 15:28:20	A4
		Ref Number: E995232	5

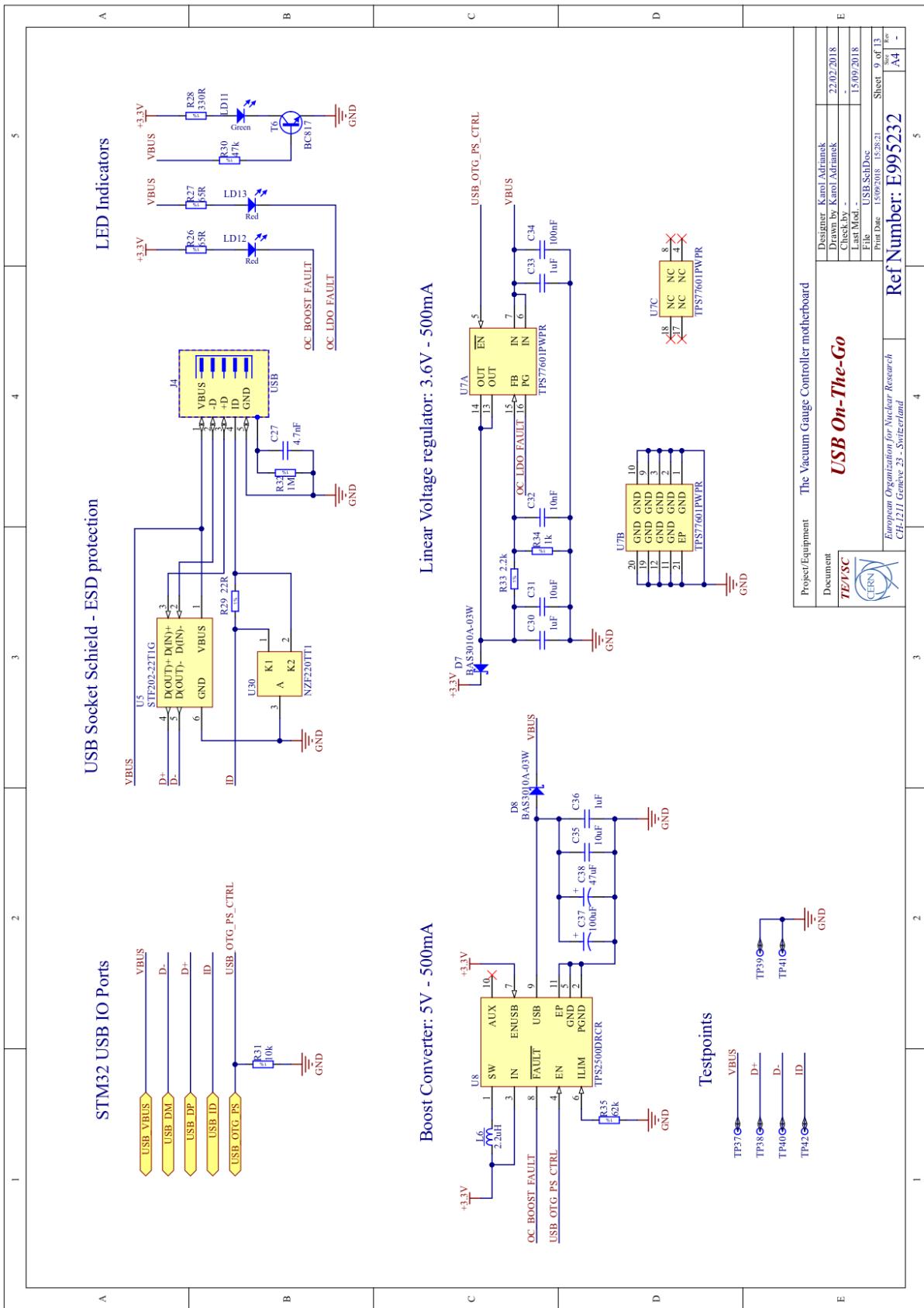


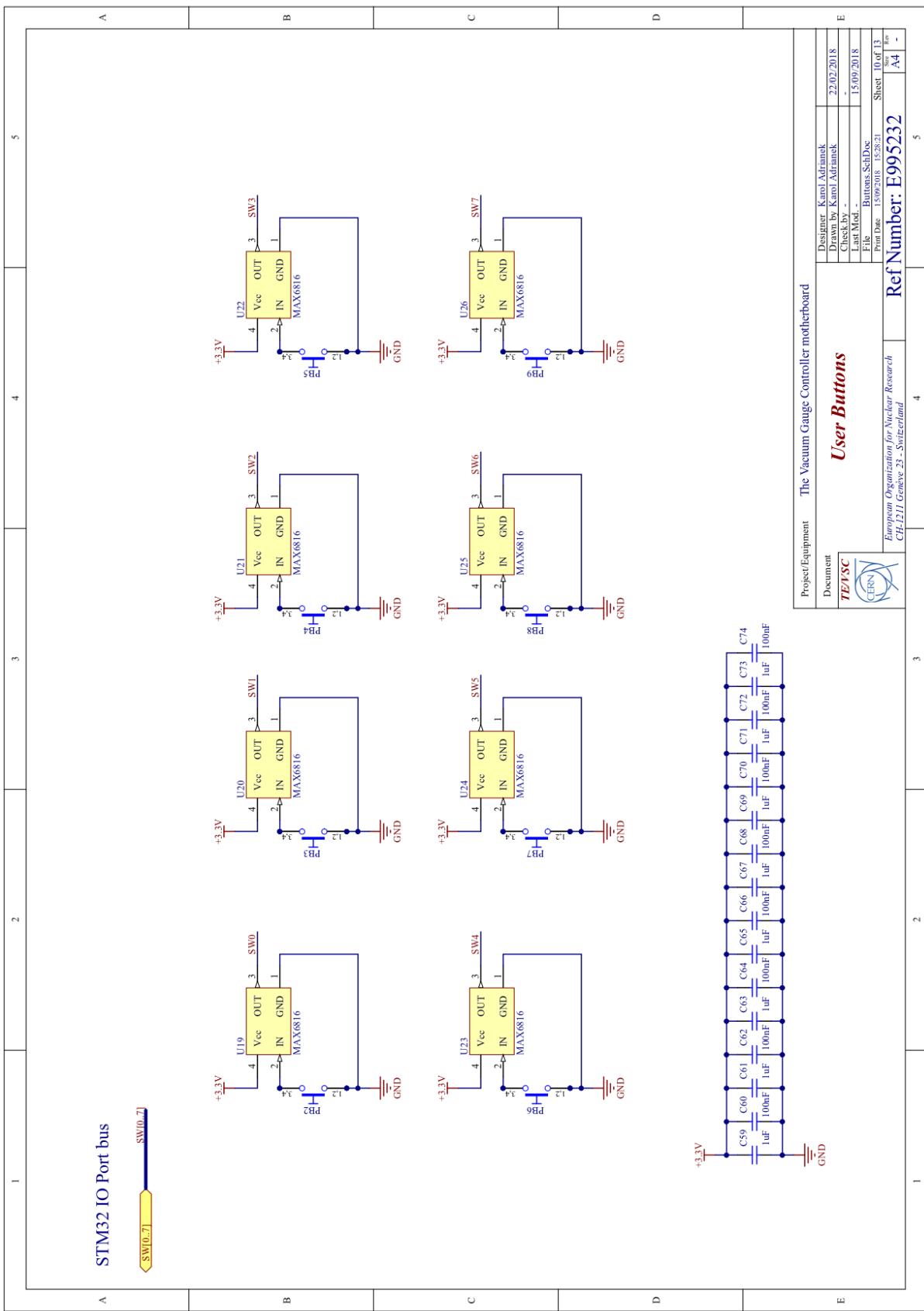


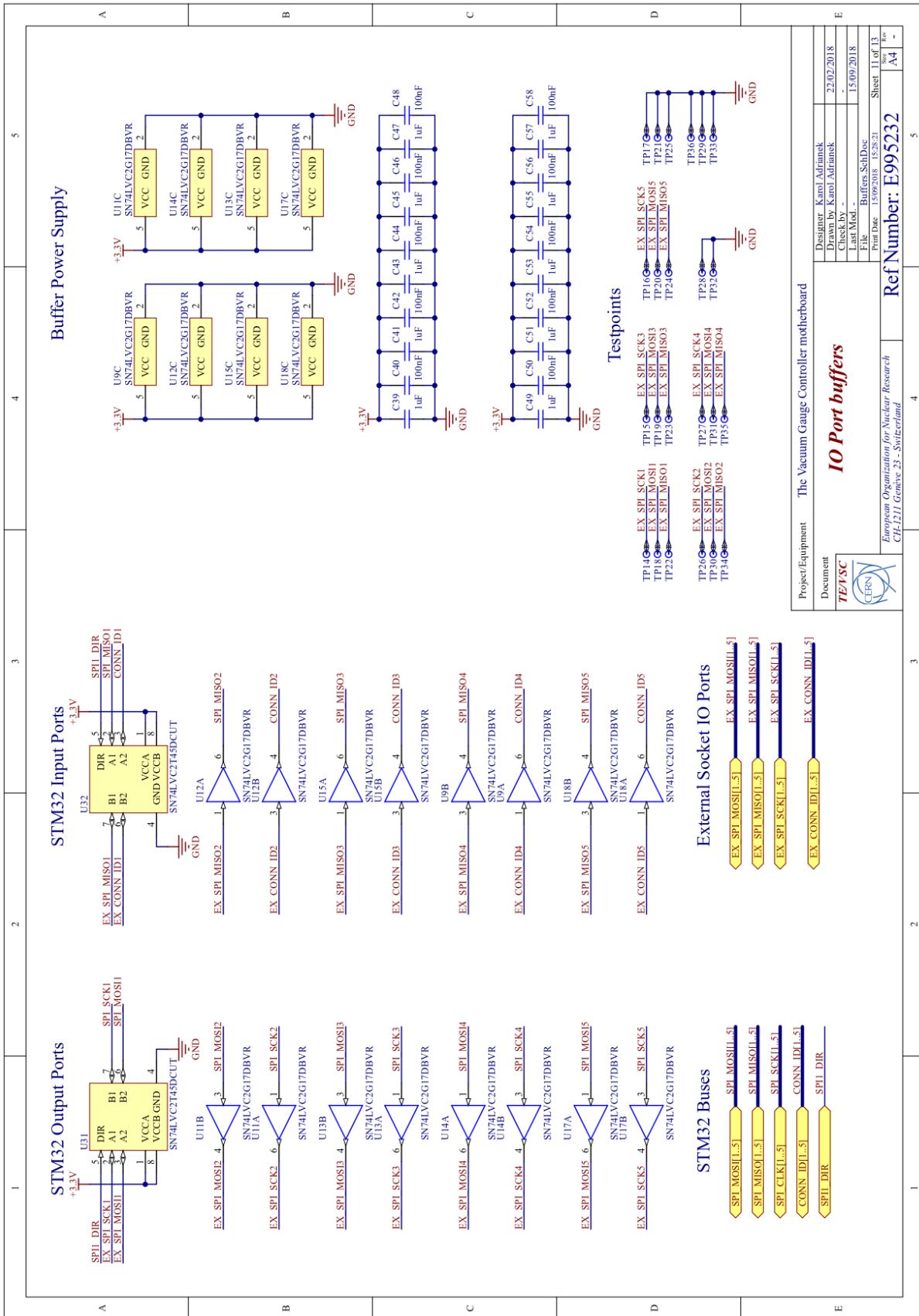


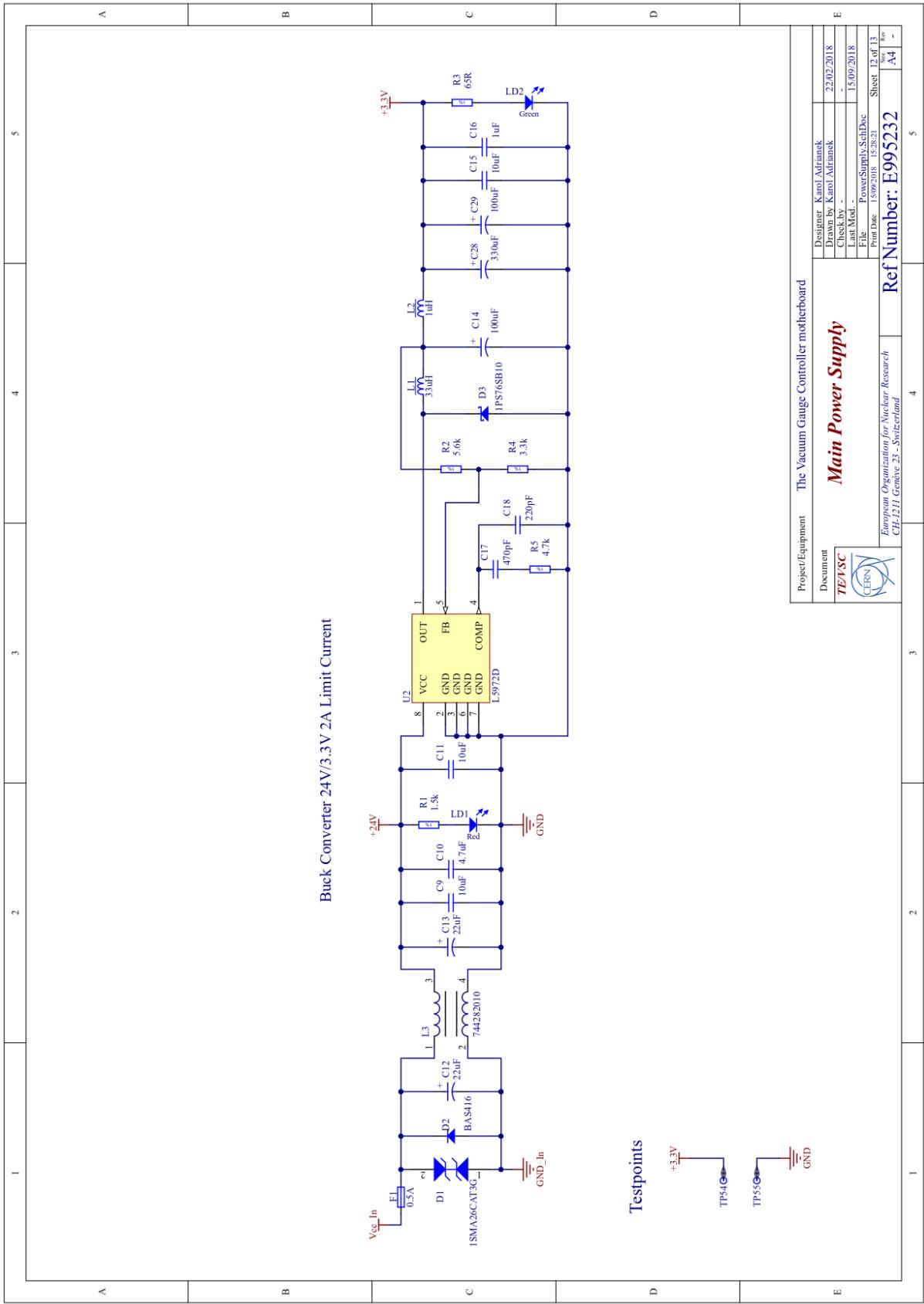


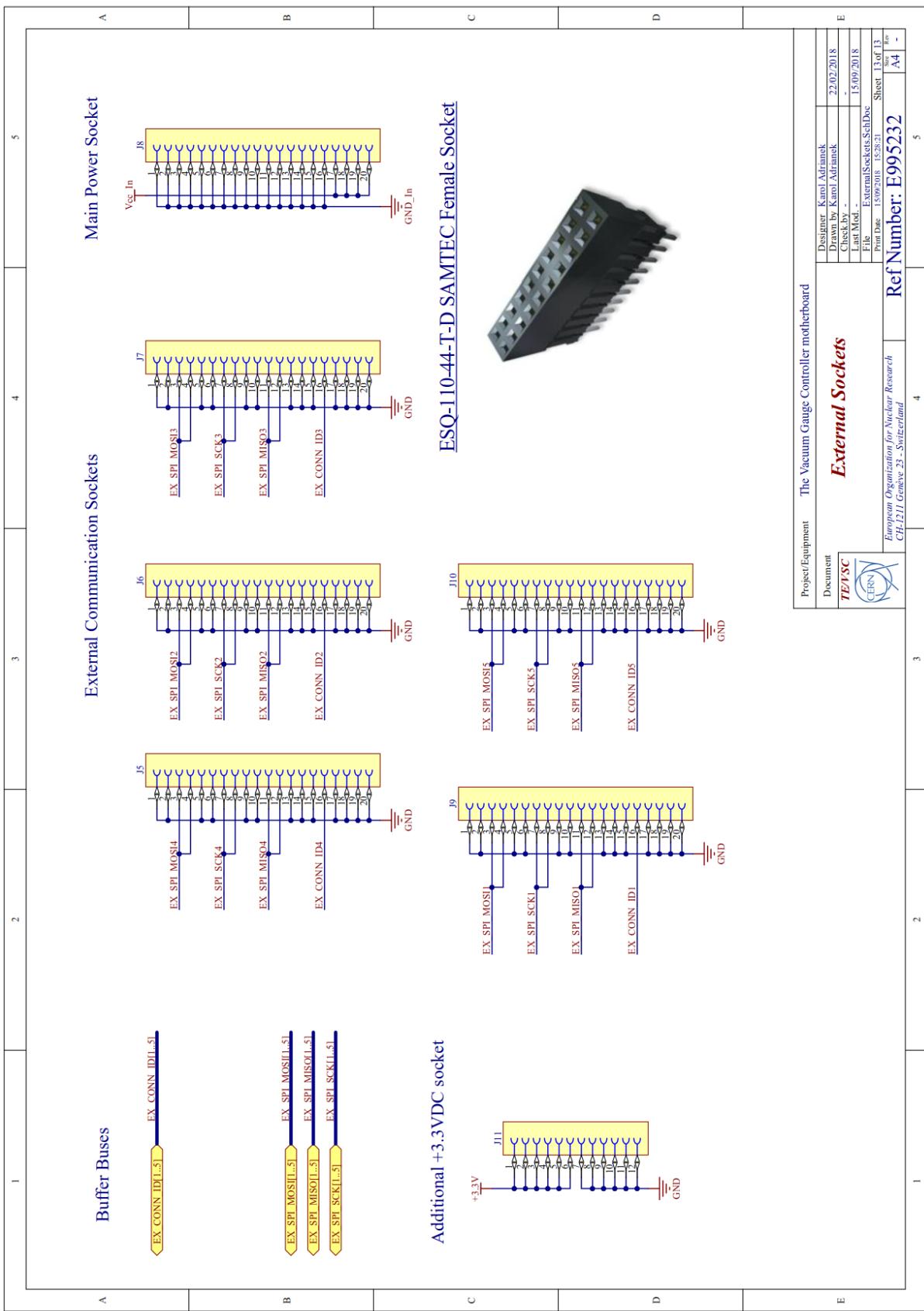




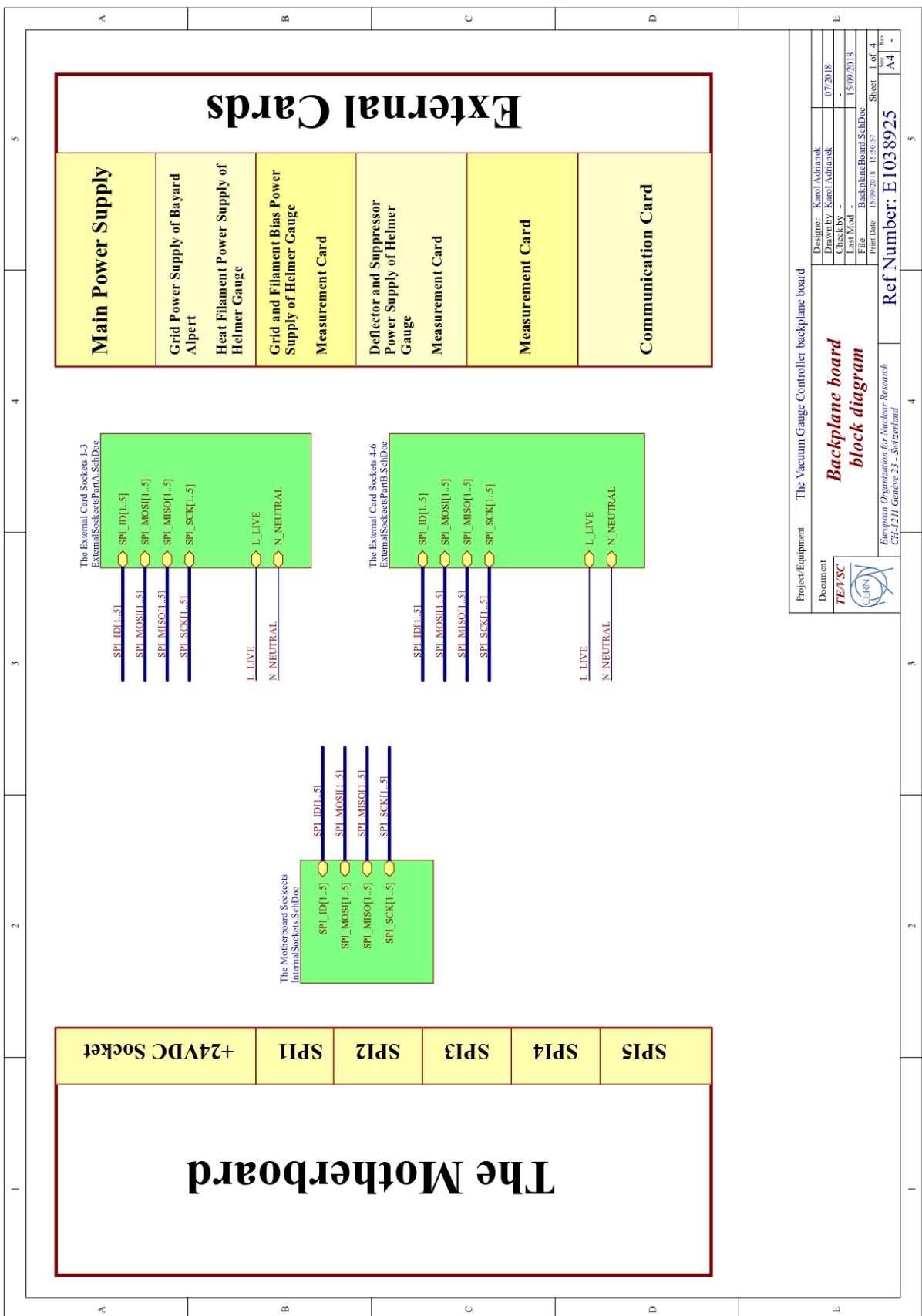


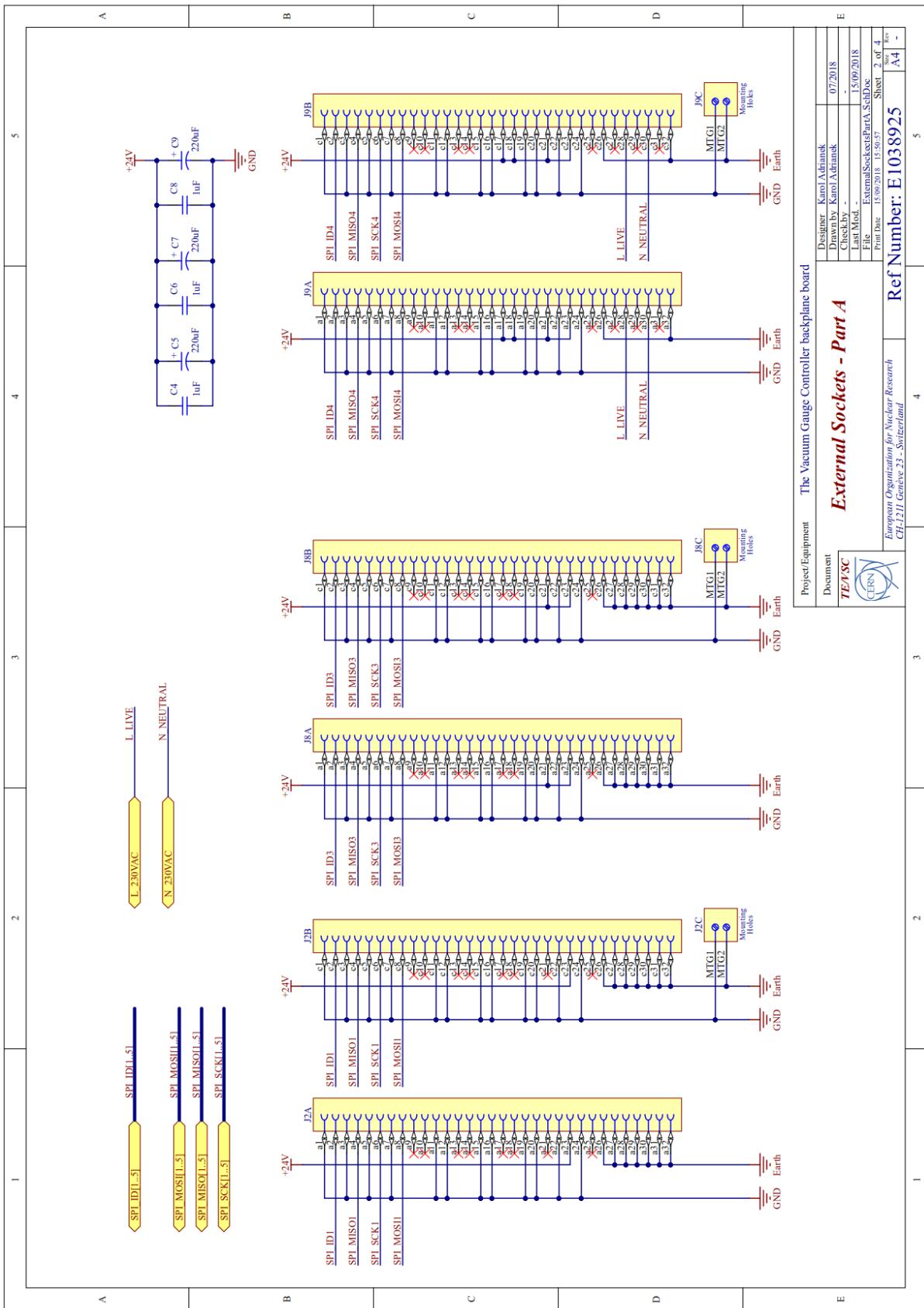


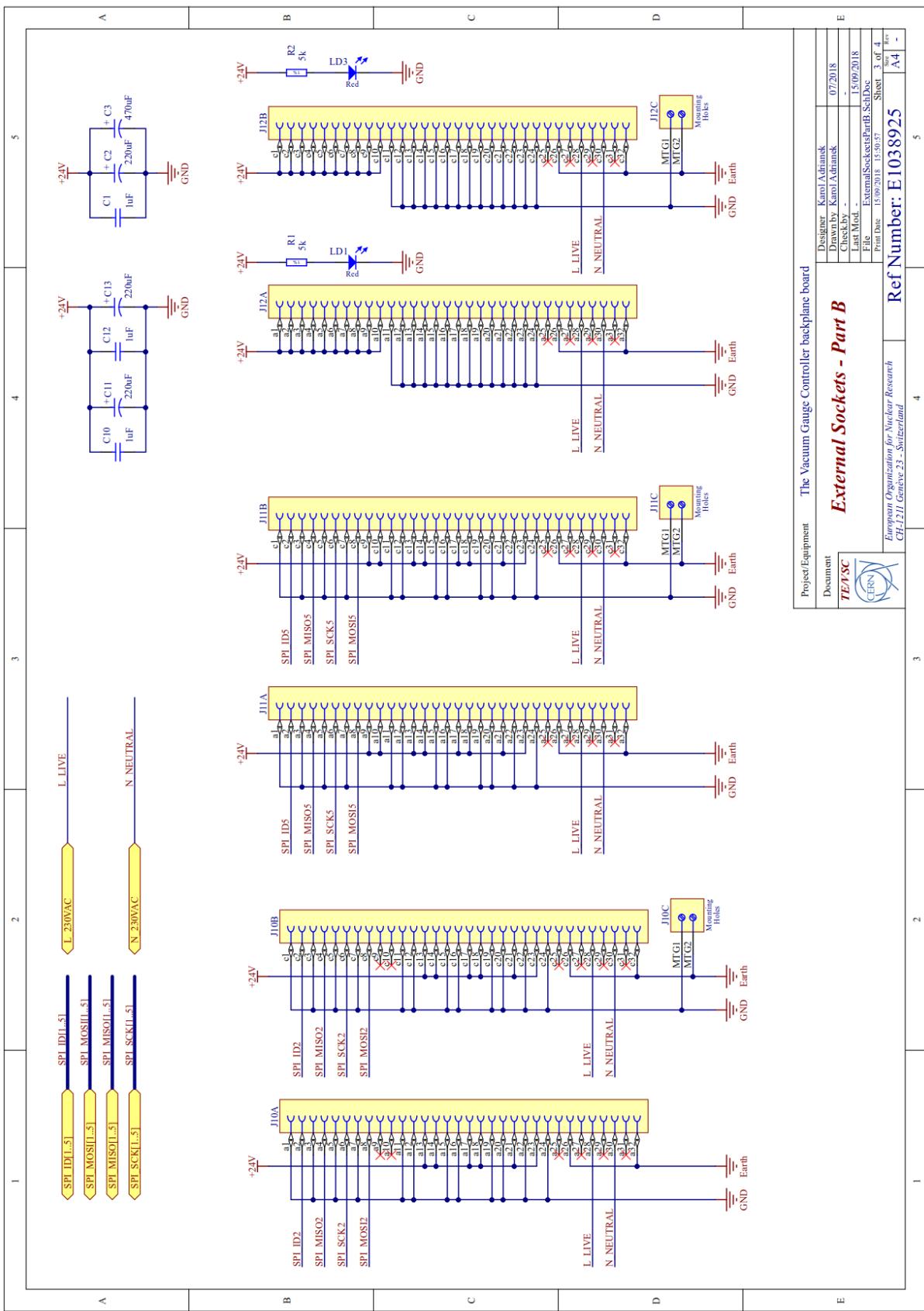




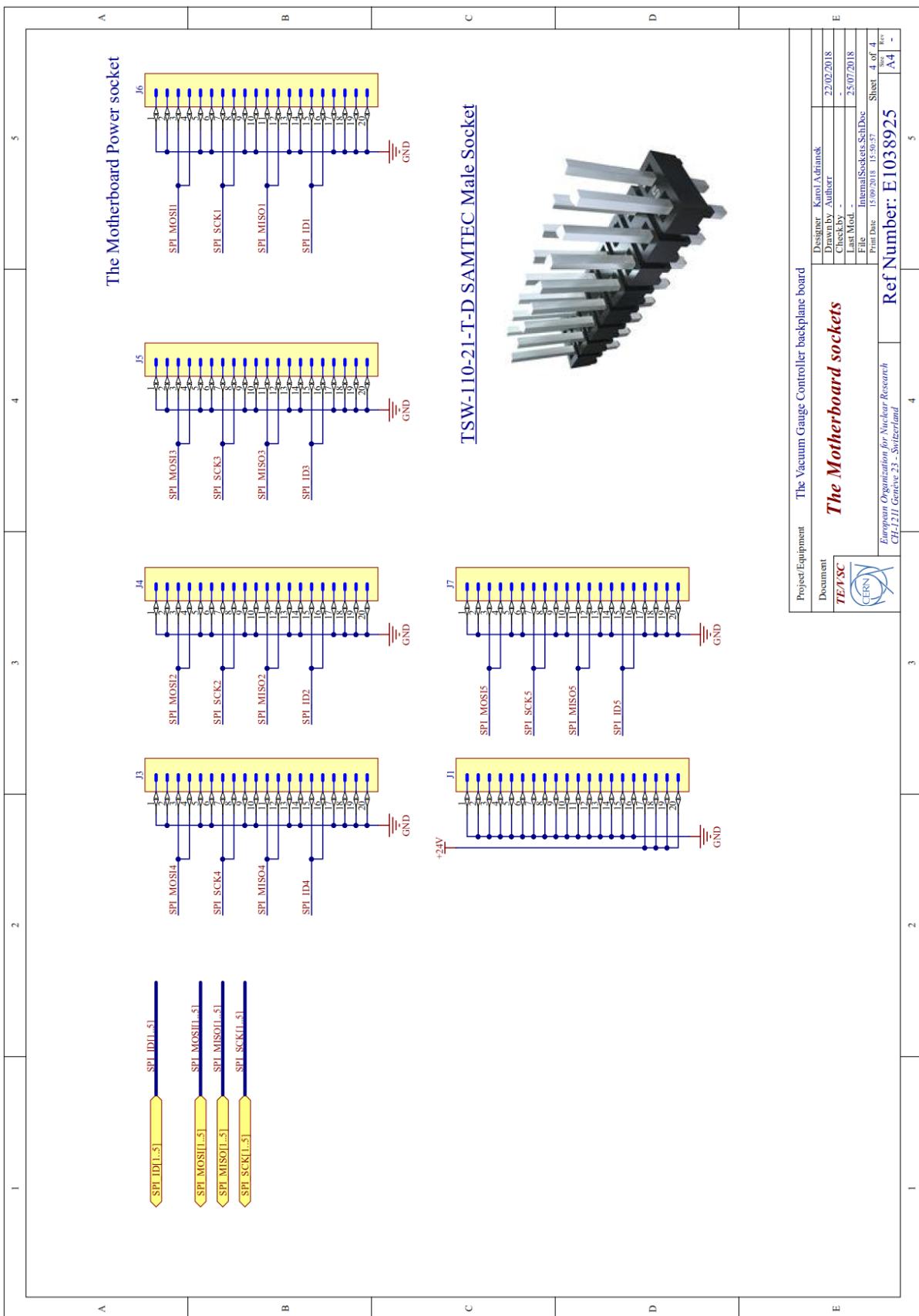
9.2 Backplane board schematics







Project/Equipment		The Vacuum Gauge Controller backplane board	
Document	External Sockets - Part B	Designer Karol Adršák	07/2018
CERN			
E	Drawby Karol Adršák	-	-
	Checkby -	-	-
	Last Mod. -	-	-
	File ExternalSocketsPartB SchDoc	15/09/2018 15:50:57	Sheet 3 of 4
	Print Date		A4
	Ref Number:	E1038925	
			5



9.3 PCB designs

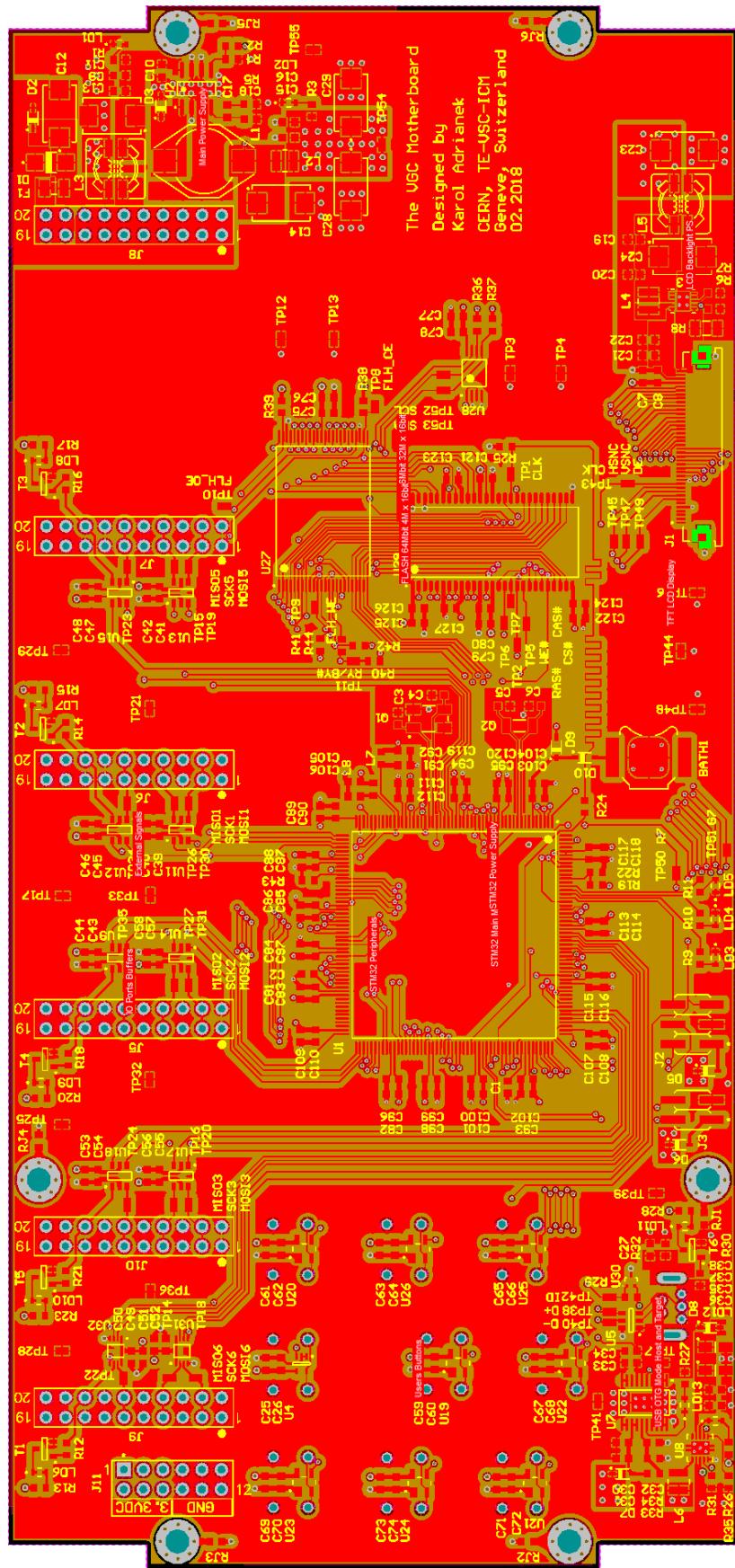


Figure 77. The motherboard design in Altium Designer (top view) [100].

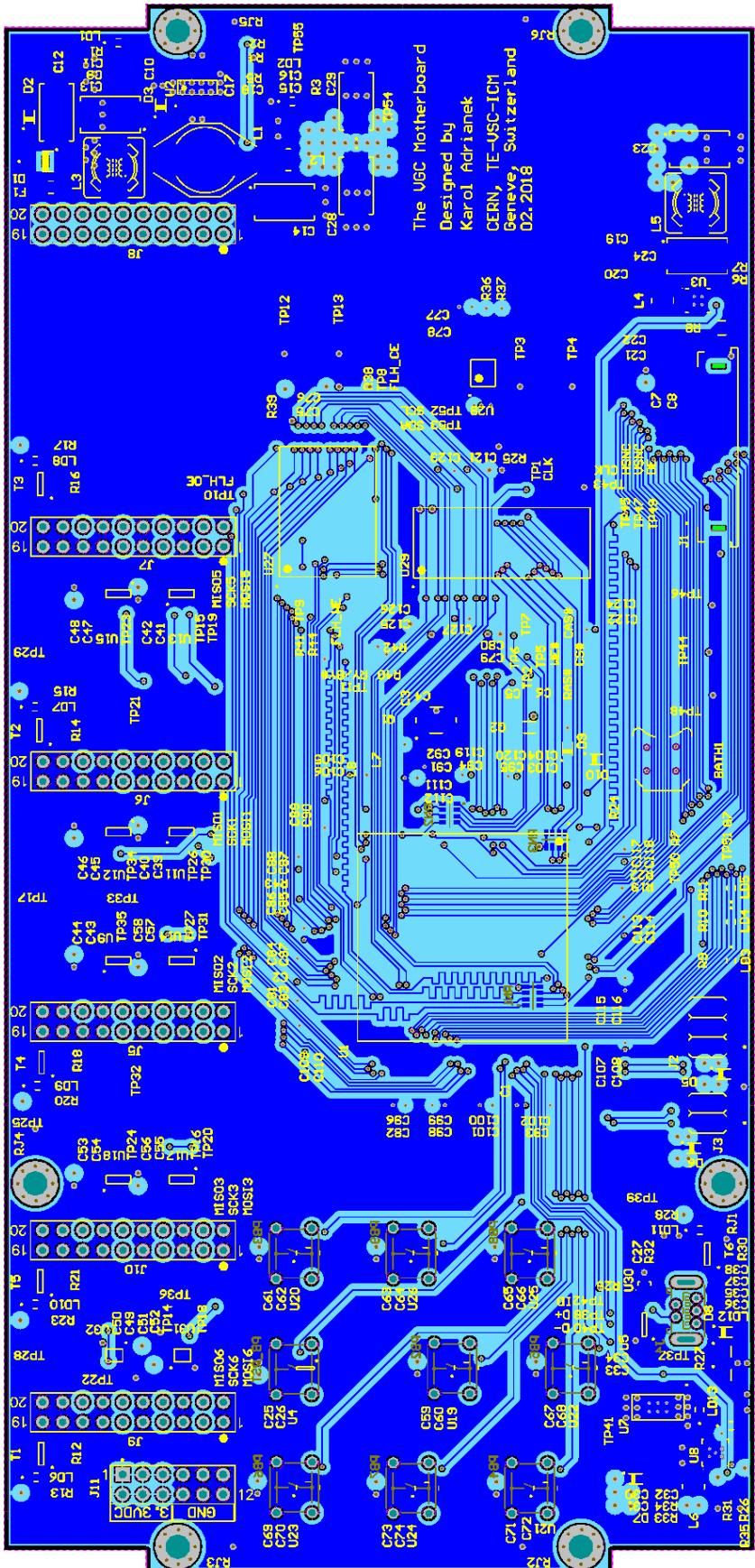
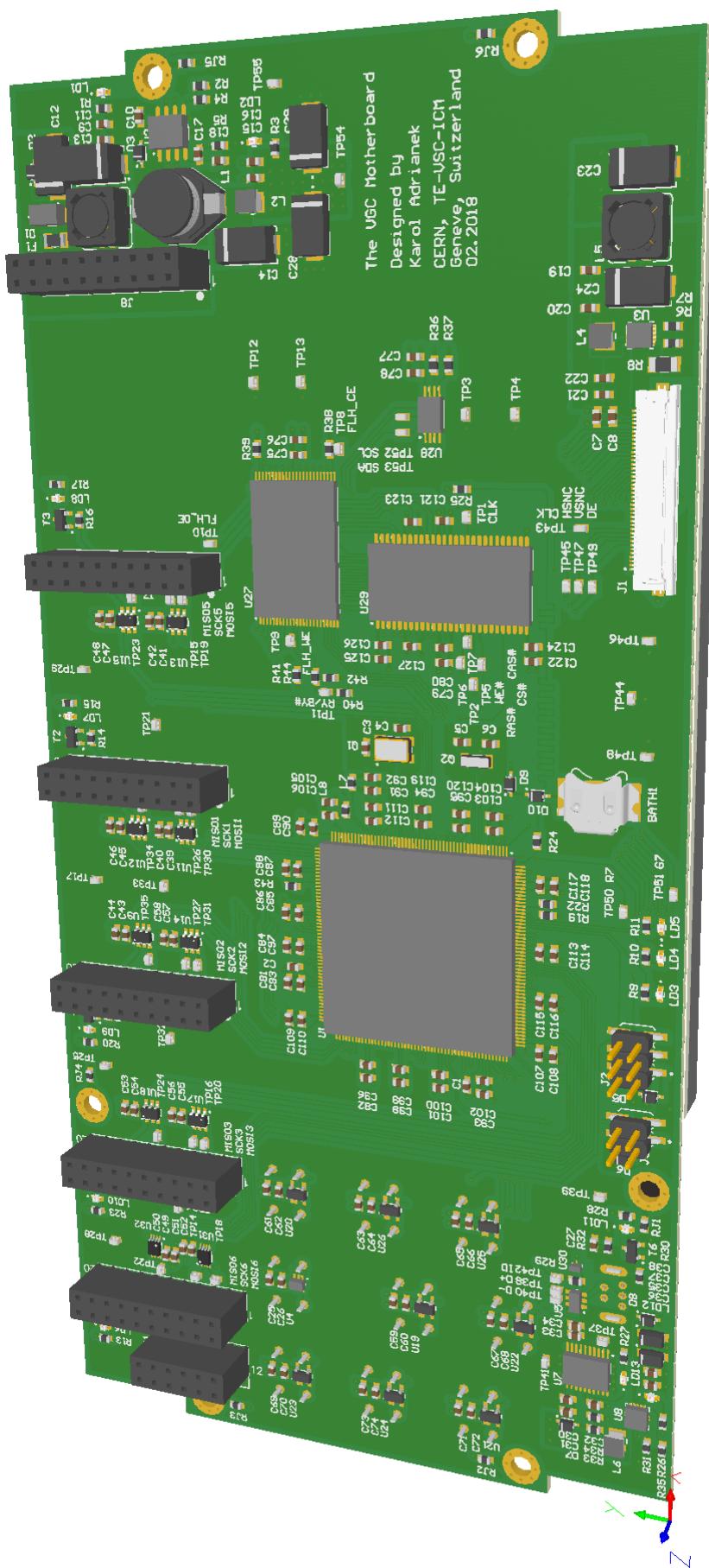


Figure 78. The motherboard design in Altium Designer (bottom view) [100].



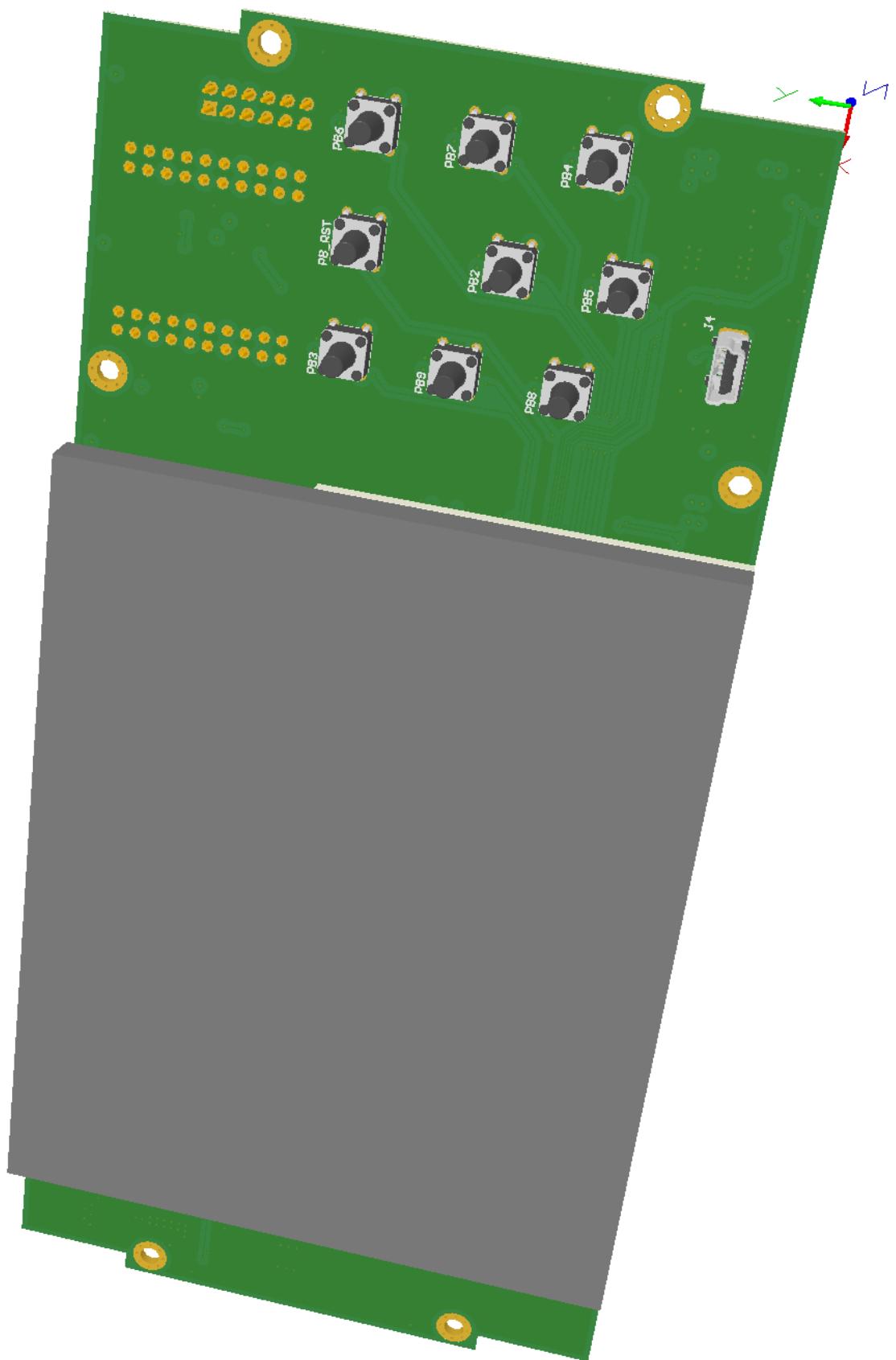


Figure 80. The 3D motherboard design in Altium designer (bottom view) [100].

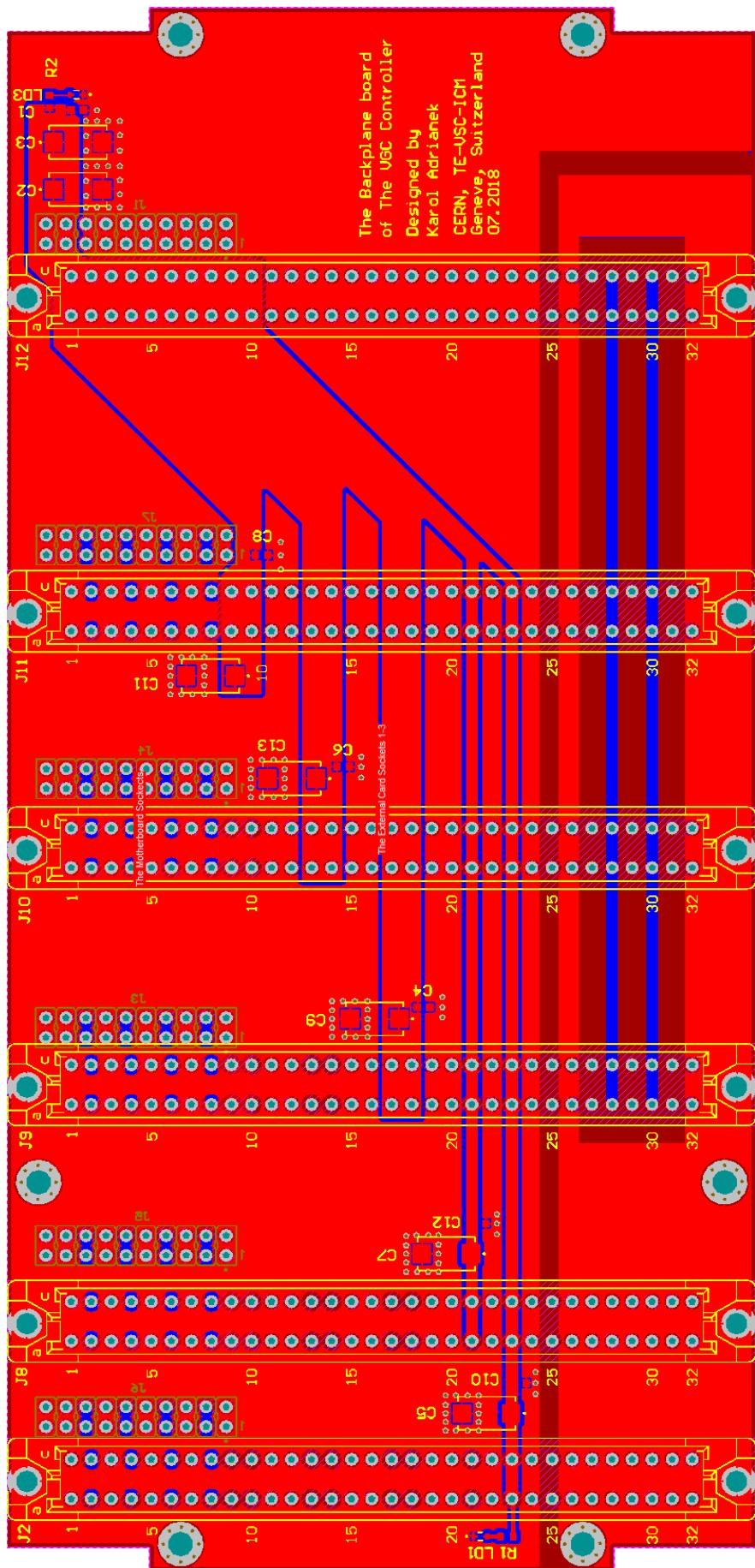


Figure 81. The Backplane board design in Altium Designer (top view) [100].

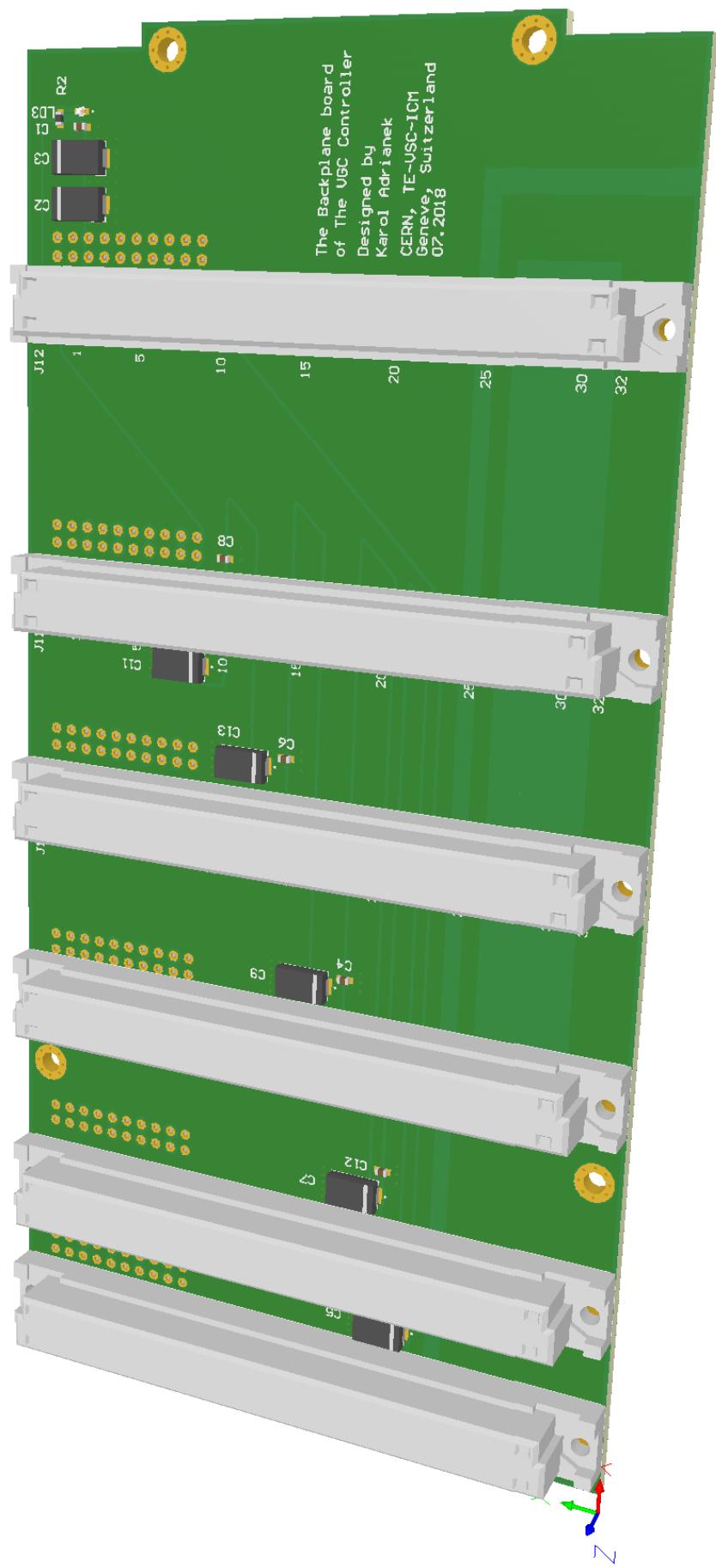


Figure 82. The 3D backplane board design view in Altium designer [100].

9.4 Prototype photos

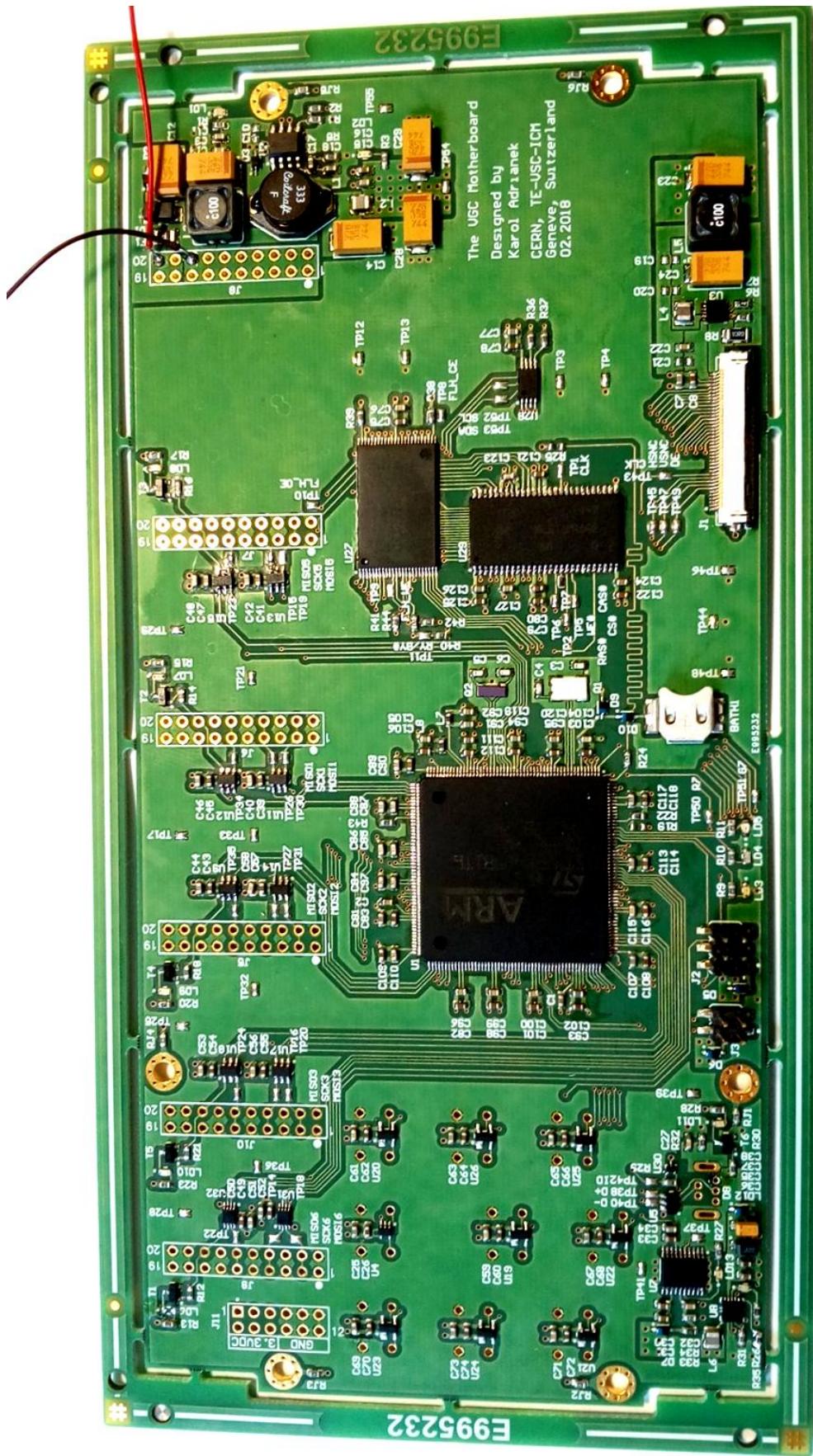


Figure 83. The motherboard PCB top view (after soldering) [100].

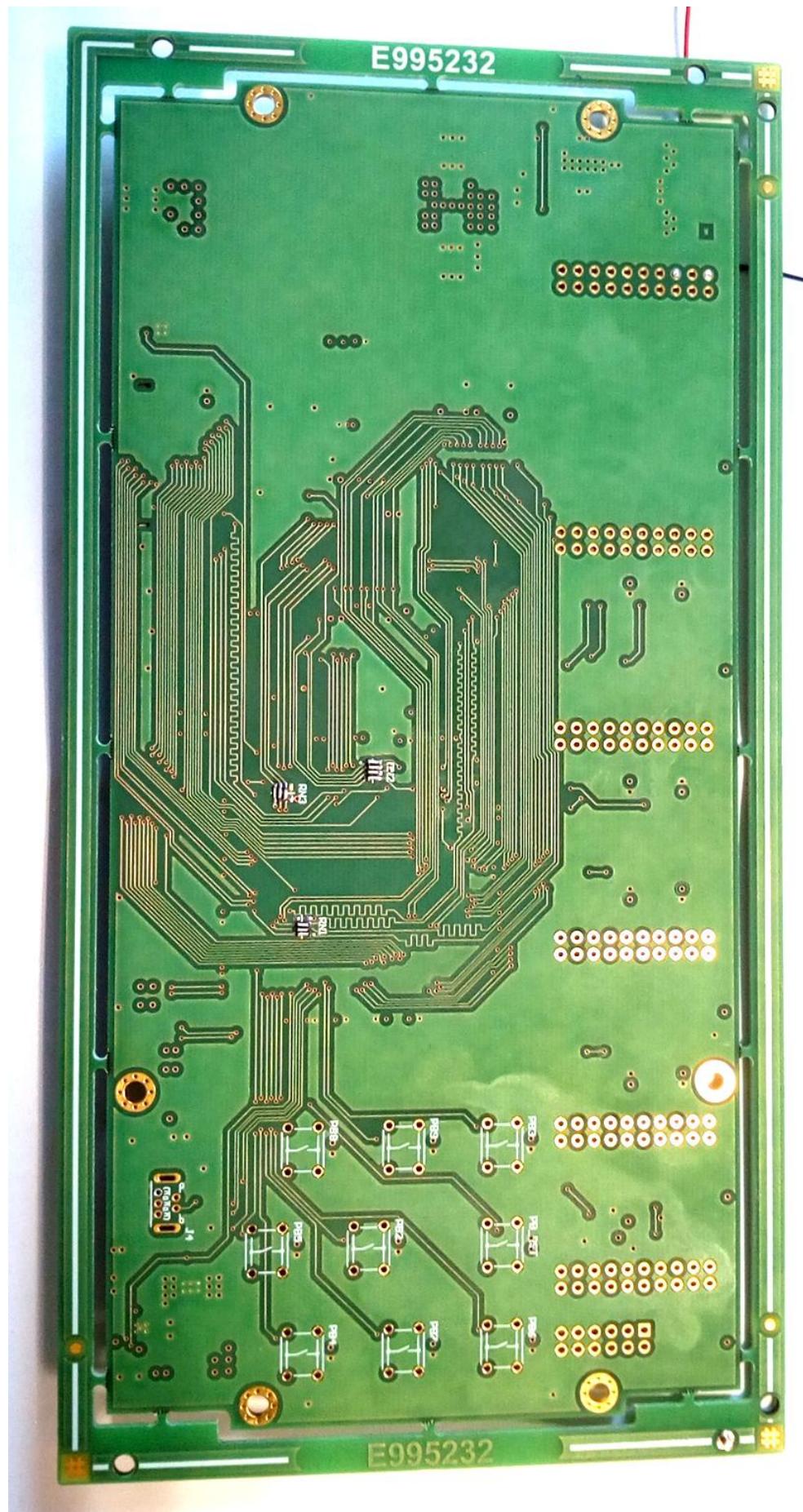


Figure 84. The motherboard PCB bottom view (after soldering) [100].

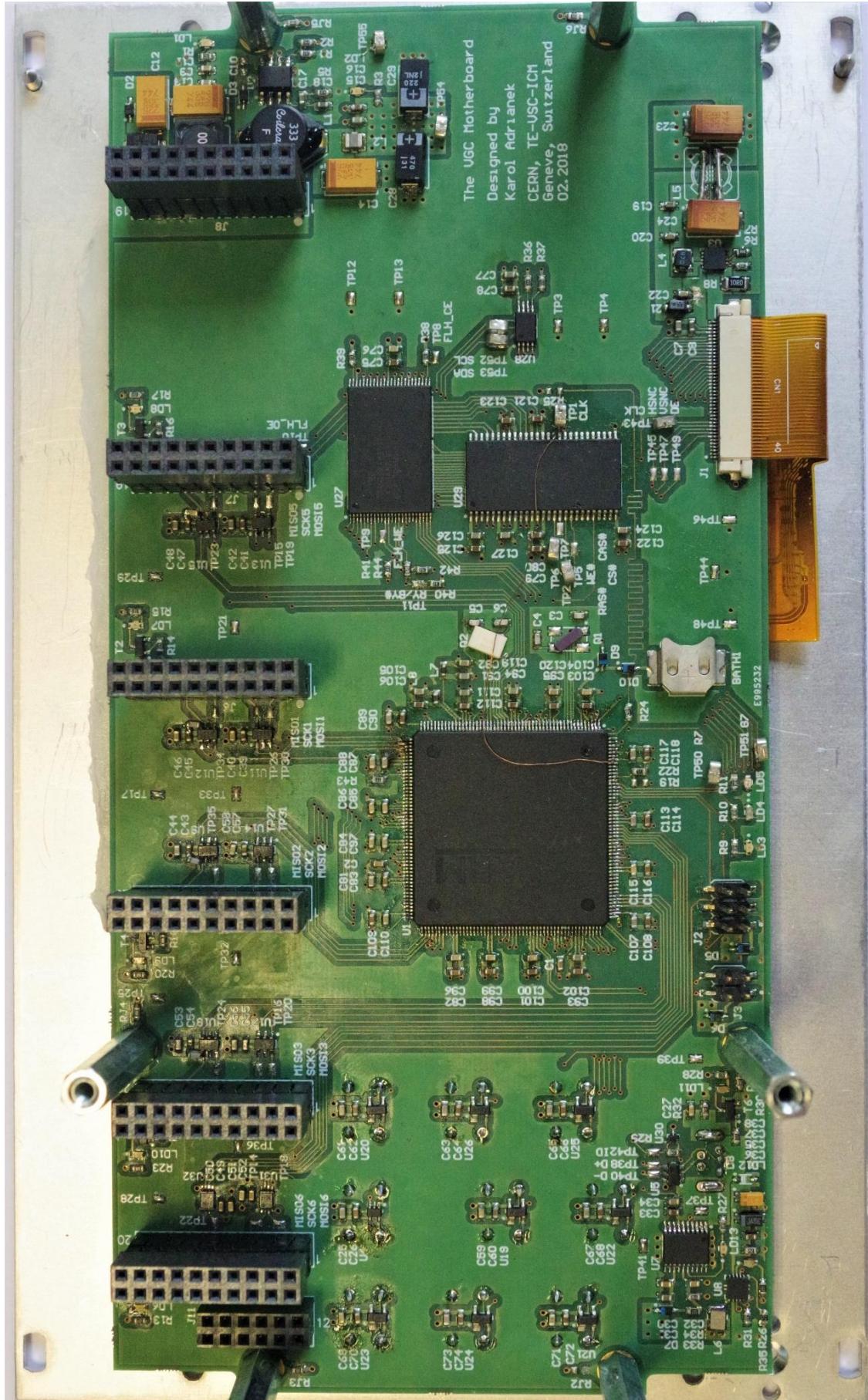


Figure 85. The Motherboard fixed to the mechanical chassis front panel [100].

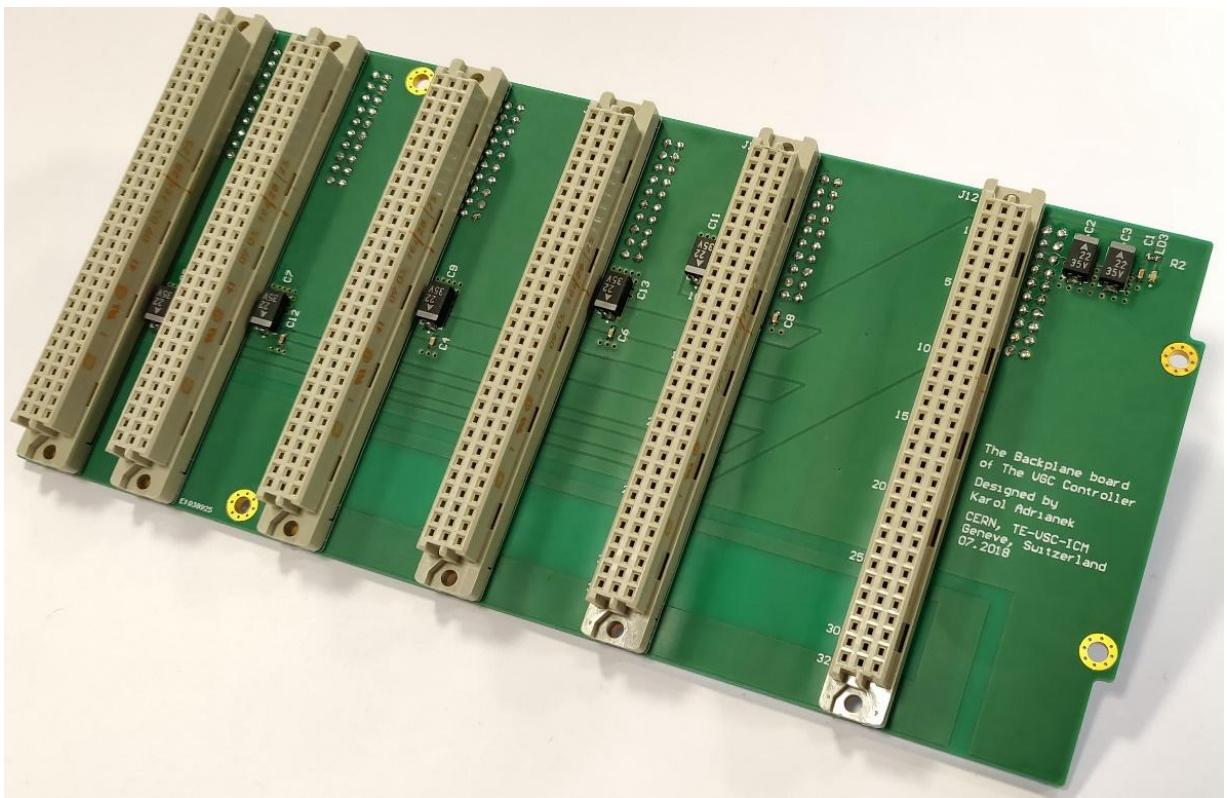


Figure 86. The backplane PCB top view (after soldering) [100].

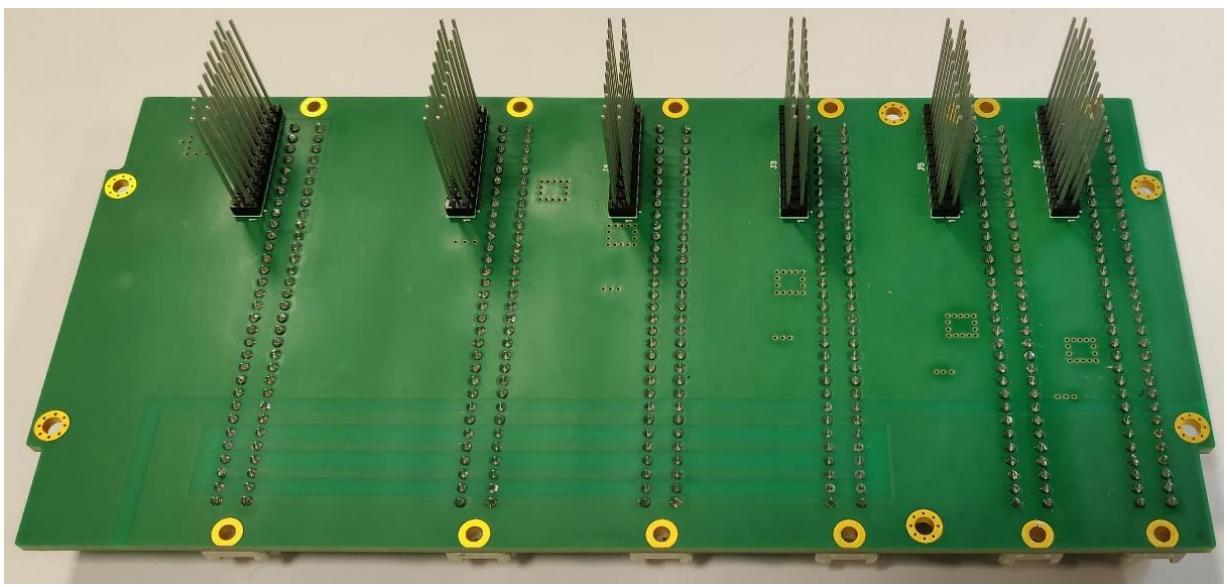


Figure 87. The backplane PCB bottom view (after soldering) [100].

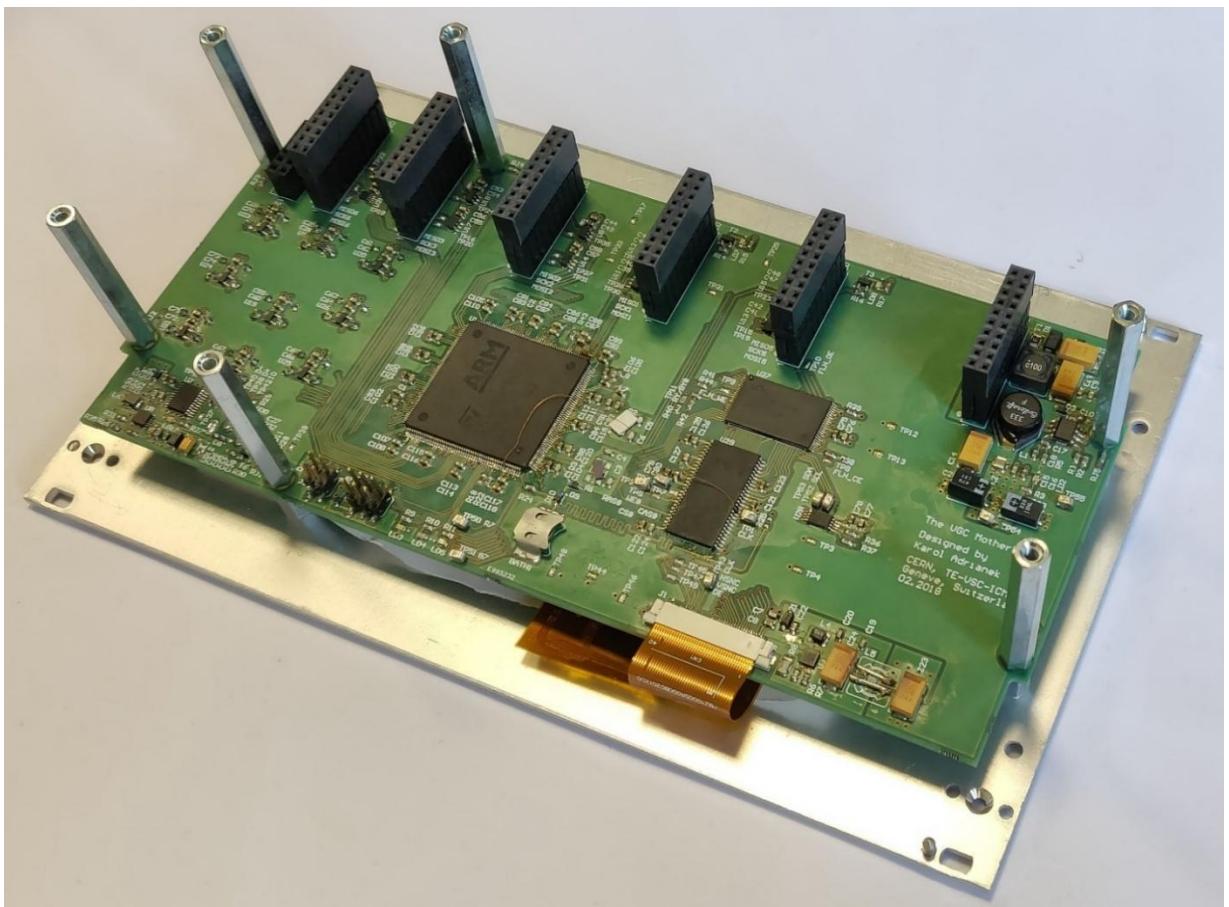
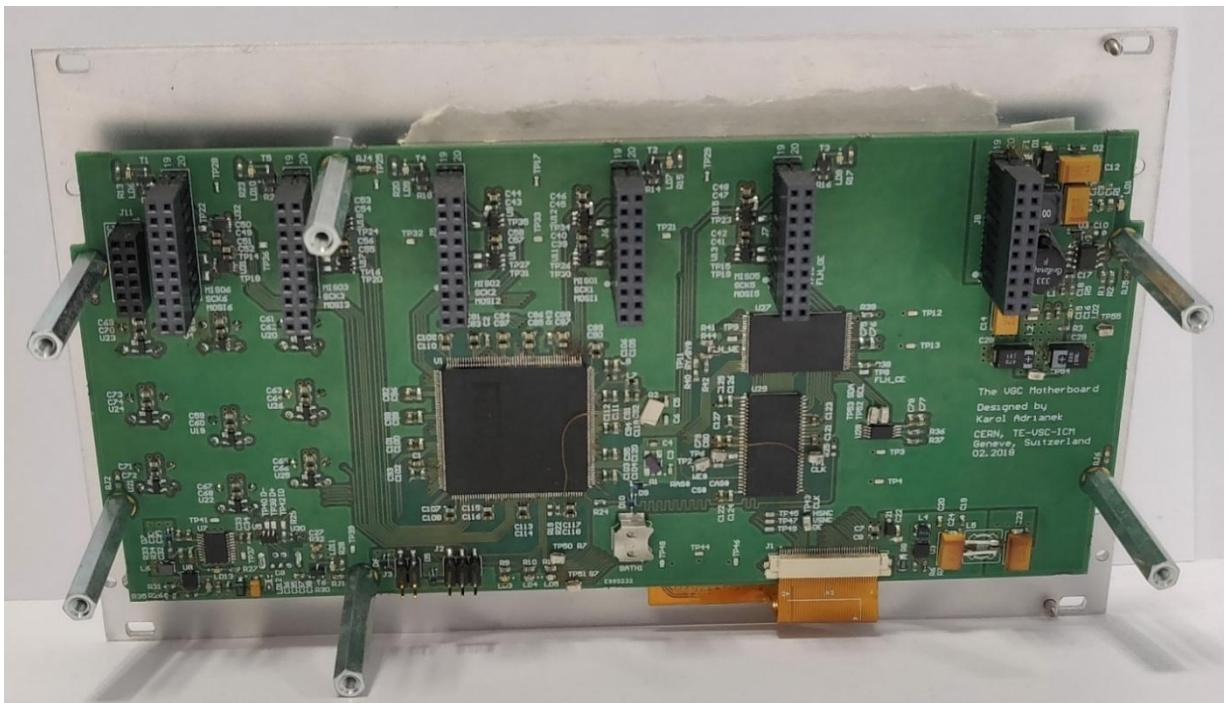


Figure 88. The VGC motherboard fixed to the chassis front panel (perspective) [100].



Figure 89. The VGC motherboard fixed to the backplane board [100].

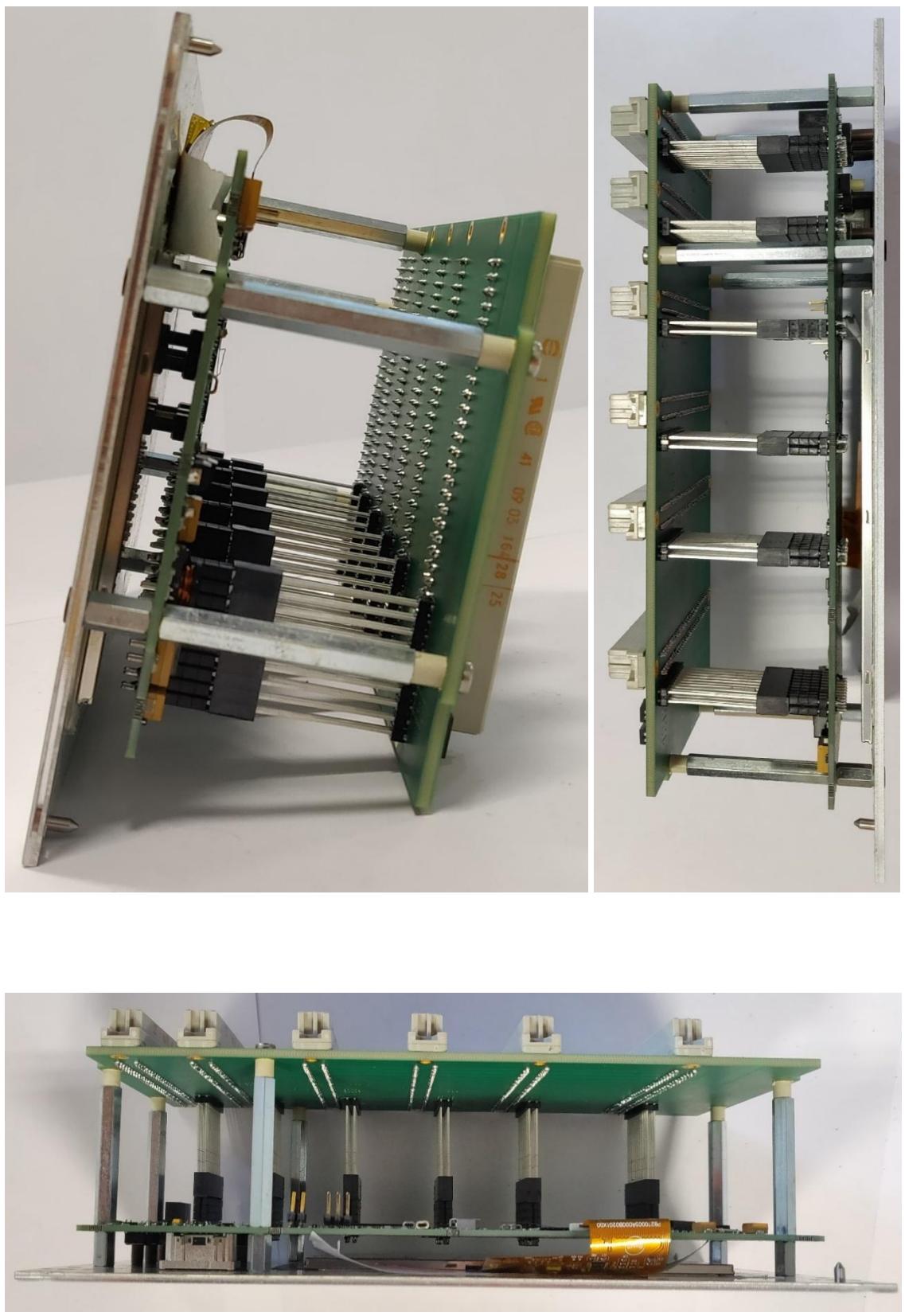


Figure 90. The VGC controller with mounted the backplane board (side view) [100].



The welcome menu with the status bar.



The PC console application menu (during the GUI templates loading).



The main system settings menu.



General system information menu.

Figure 91. The GUI screens (selected) [100].

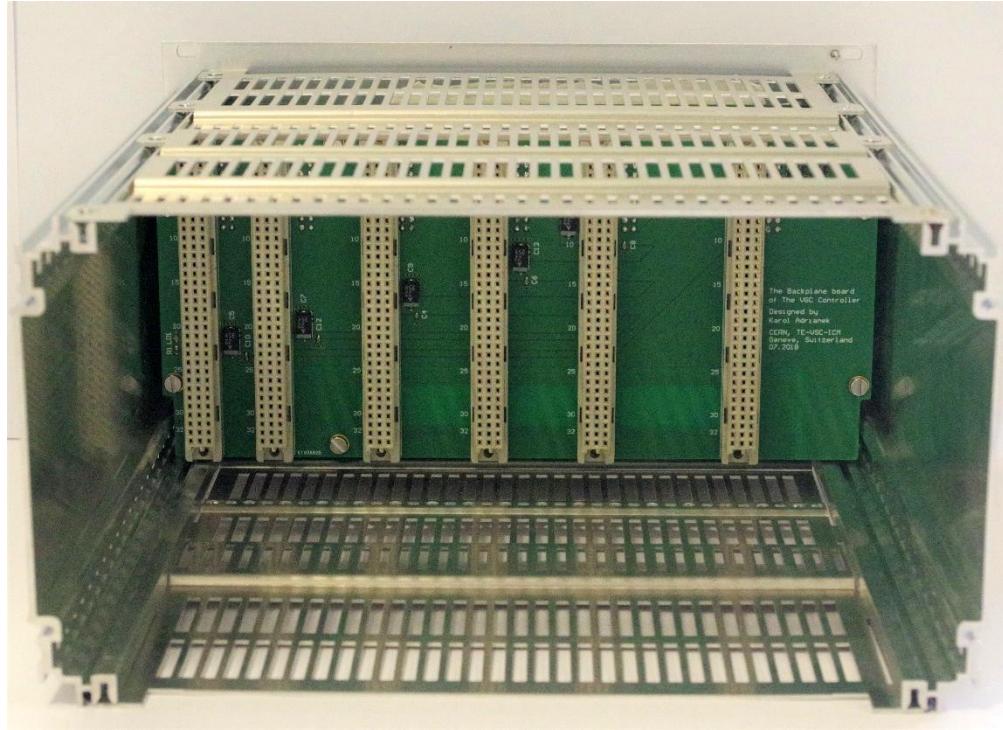


Figure 92. The VGC controller prototype in the chassis (rear perspective) [100].

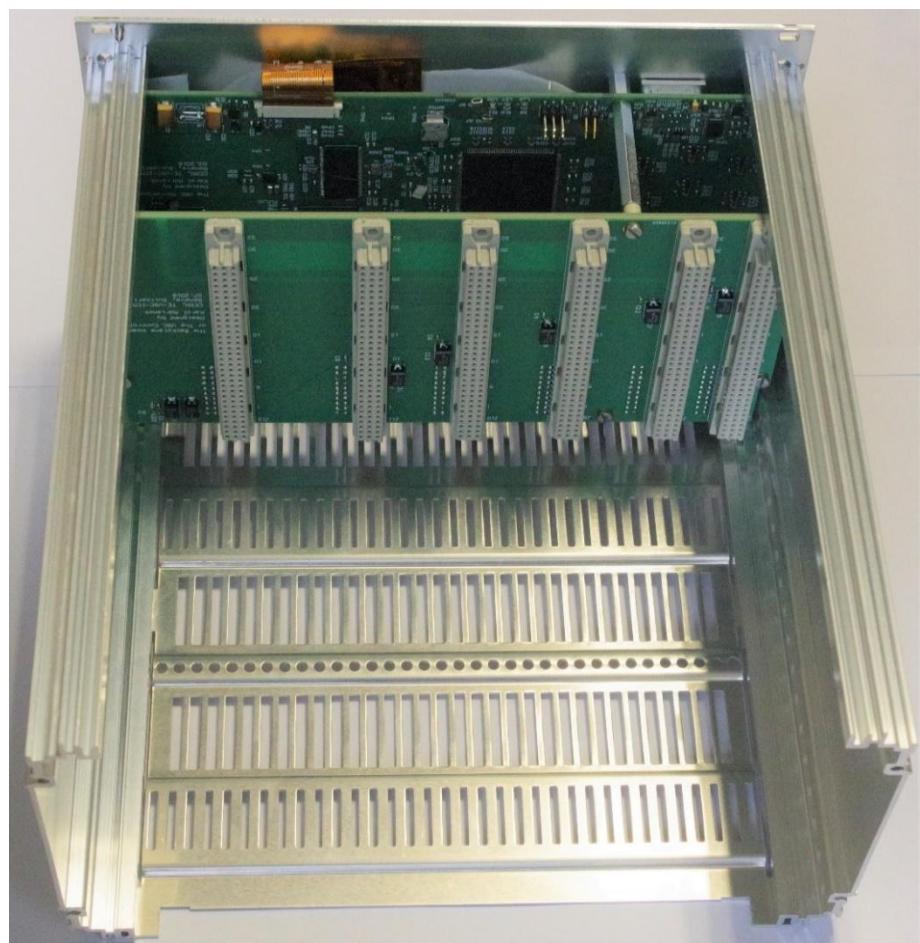


Figure 93. The VGC controller in the 42HP 3U housing (perspective) [100].

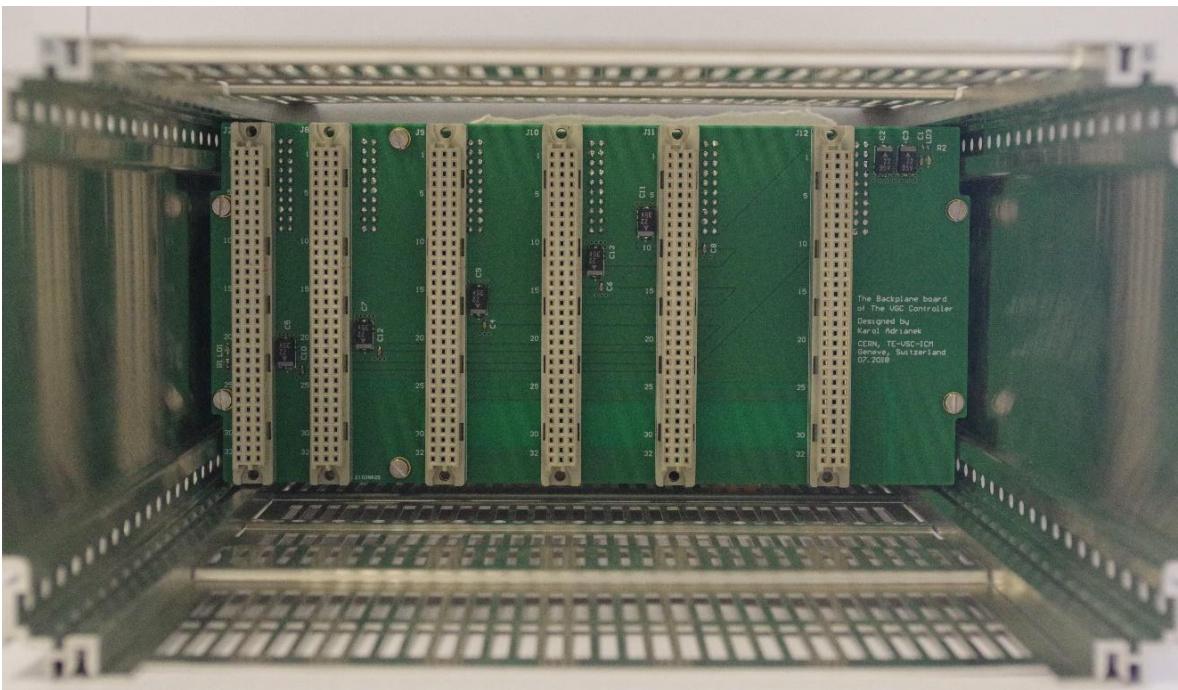
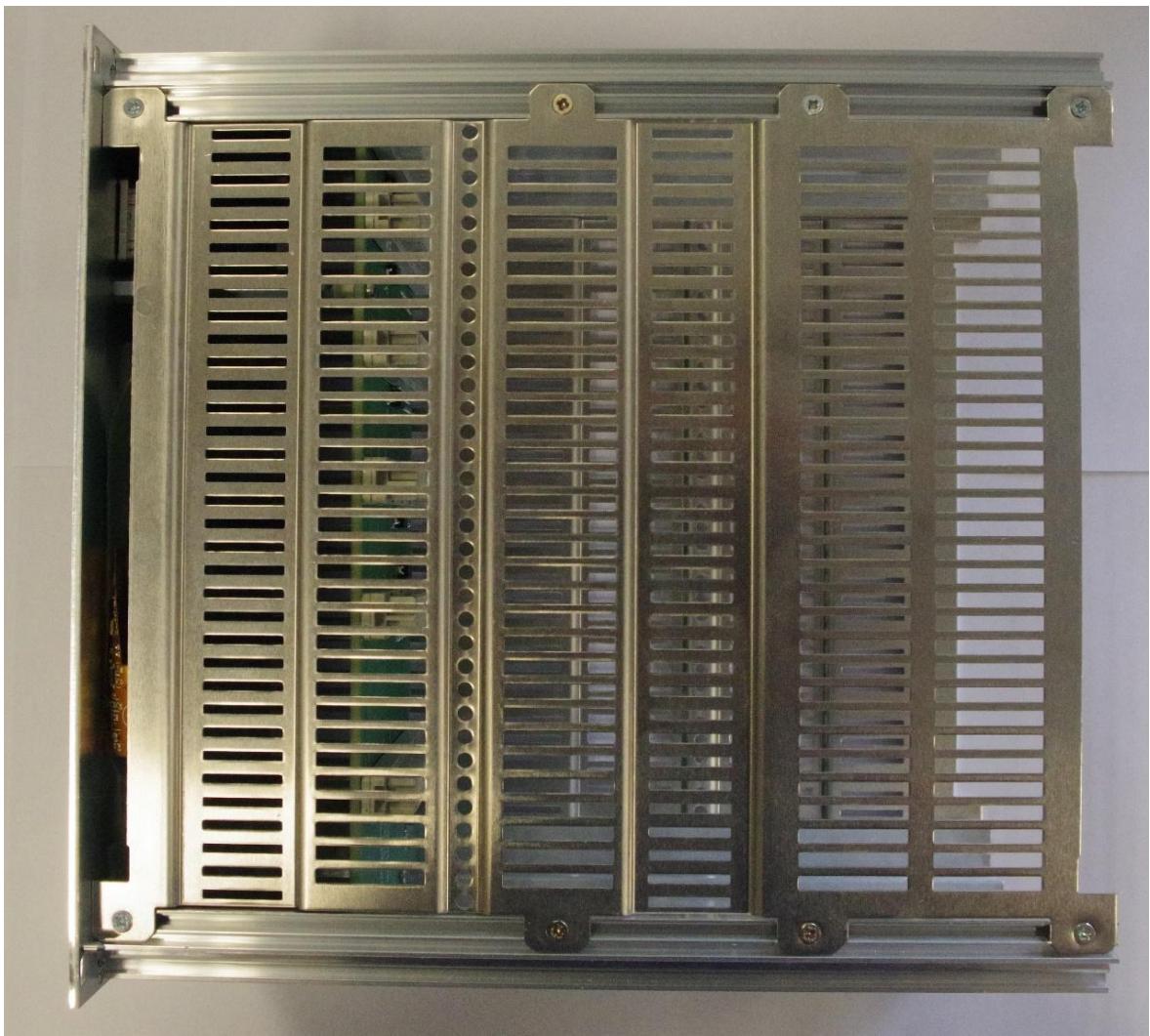


Figure 94. The VGC controller in the mechanical chassis (side and rear view) [100].

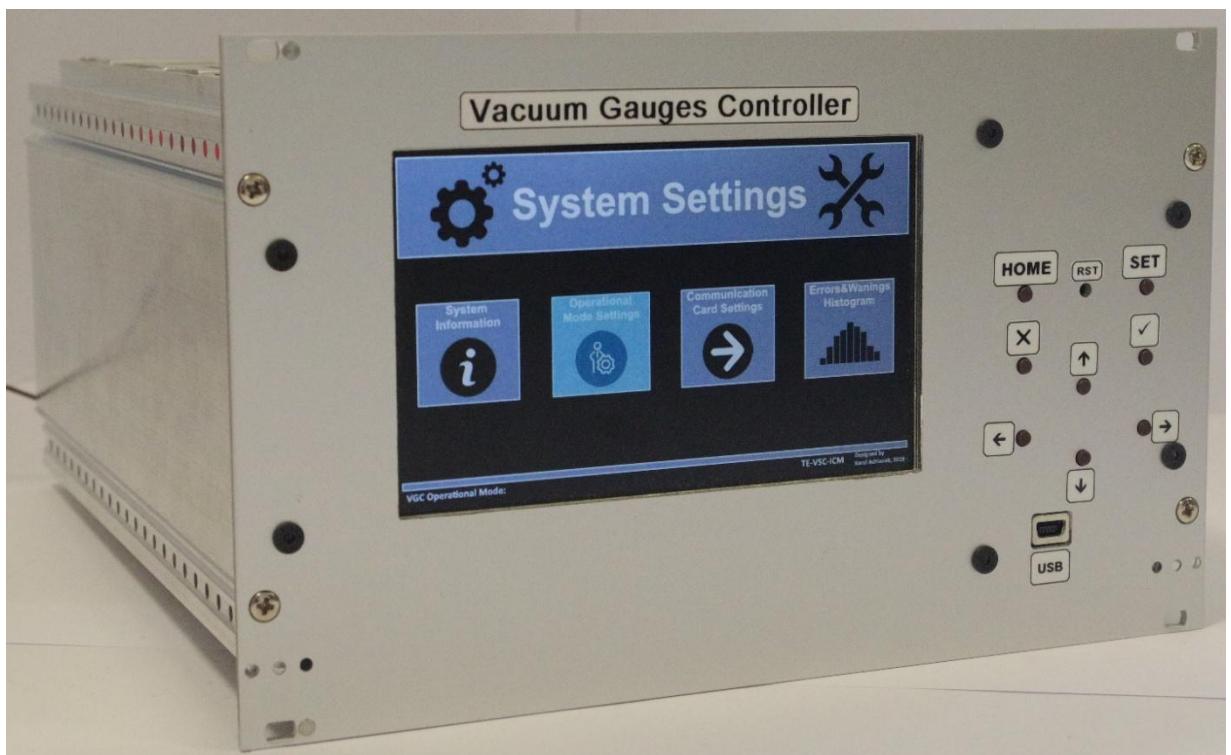


Figure 95. The final Vacuum Gauges Controller [100].