DOI 10.31489/2022No4/58-66

UDC 629.7.05; 629.783

DEVELOPMENT OF ONBOARD CONTROL SYSTEM ARCHITECTURE FOR NANOSATELLITES

Alipbayev K.[1], Sarsenbayev Y.[2], Mussina A.[2]*, Nurgizat Y.[1]

[1] Institute of Telecommunications and Space Engineering, Almaty University of Power Energy and Telecommunications named after G.Daukeev, Almaty, Kazakhstan
[2] Ghalam Limited Liability Partnership, Astana, Kazakhstan, a.mussina@ghalam.kz

This article presents development of the architecture of the onboard control system. In particular, the hardware and circuit solutions in the development of the module are presented. Technical solutions, the concept of the mechanical layout of the onboard control system and block diagrams of the presented modules of the onboard complex are also discussed. Main functions of the on-board software are described in details, which will help in design and development of ultra-small artificial satellites. This article serves as a new approach for how to effectively configure and operate the on-board control system.

Keywords: onboard computer, cube sat, software defined radio, Field-Programmable Gate Array.

Introduction

Nowadays satellite technologies and components allow designing very small sized satellites (nanosatellites) to reduce overall cost of the space systems. The nanosatellites belong to the class of small satellites weighing less than 10 kg and this type of small satellite is used in most Low Earth Orbit (LEO) missions with different payloads. Most of them uses CUBESAT class standard and their numbers in orbit is rapidly grows [1]. In this frame the development multifunctional control systems for nanosatellites in compact form factor is getting relevant meet the growing demand in small space systems.

Currently accepted, a OBCS is a set of modules and software designed to control attitude and orbit of the satellites and operations of on-board equipment. The classic OBCS plays central role in the operation of a spacecraft and its functions are:

- Attitude and orbit control (AOCS units, OBC);
- Executing realtime & time-tag telecommands (OBC);
- Telemetry data management (OBC);
- Platform & Payload housekeeping (OBC);
- On-board time synchronization (GPS receiver);
- Failure detection, isolation and recovery (OBC);
- Reception, validation, decoding and distribution commands to other subsystem in Emergency modes (TT&C subsystem);
- Collection, processing and analysis of control and diagnostic information;
- Telecommand uploading and execution, telemetry downlinking to Ground Control station.

The aim of considered development is to provide integrated solution for small/nano satellite systems which effectively implements abovementioned OBCS functions in a single on-board module and furthermore covers functions of other subsystems in a satellite platform:

- Payload computer function with flexible interface for the most payload;
- High-speed data transceivers up to 6 GHz based on SDR technology.

This article presents development of architecture of On-Board Control System (OBCS) with a unified payload interface for nanosatellites implemented using software-defined FPGAs. The technological approach reduces weight, power consumption, size and cost of the satellite. The design is suitable as to CubeSAT and nano/micro satellite design in terms of mechanical and electrical compatibility. The development is patented with patent for Spacecraft onboard control system utility model #6912 in Republic of Kazakhstan. [2]

The concept of building OBCS were developed based on the requirements of a systematic approach to the design of on-board controls and the actual practice of flight control of spacecraft (S/C) of various classes and purposes.

1. Development of the architecture of the OBCS hardware

The section describes the main components of the OBCS, the corresponding modules and computational functions. Figure 1 shows a generalized architecture and a diagram of the use of OBCS. The considered onboard control system consists of 3 functional modules physically presented in the form of printed circuit boards:

– Processor module - the main component of the OBCS, in which computing functions are concentrated, as well as memory modules for storage housekeeping data, data from the payloads are located in dedicated memory (NAND);

– RF module - a board for converting the high frequency (HF) signal into digital form for subsequent processing, as well as the generated digital signal into analog form in the corresponding HF range. Filters are used in the path to select the desired working band. Through high-frequency connectors and HF cables, signals are sent / received to the HF path unit, where amplification occurs to the required level;

– Interface board - a board for connecting external interfaces to the OBCS (except HF).

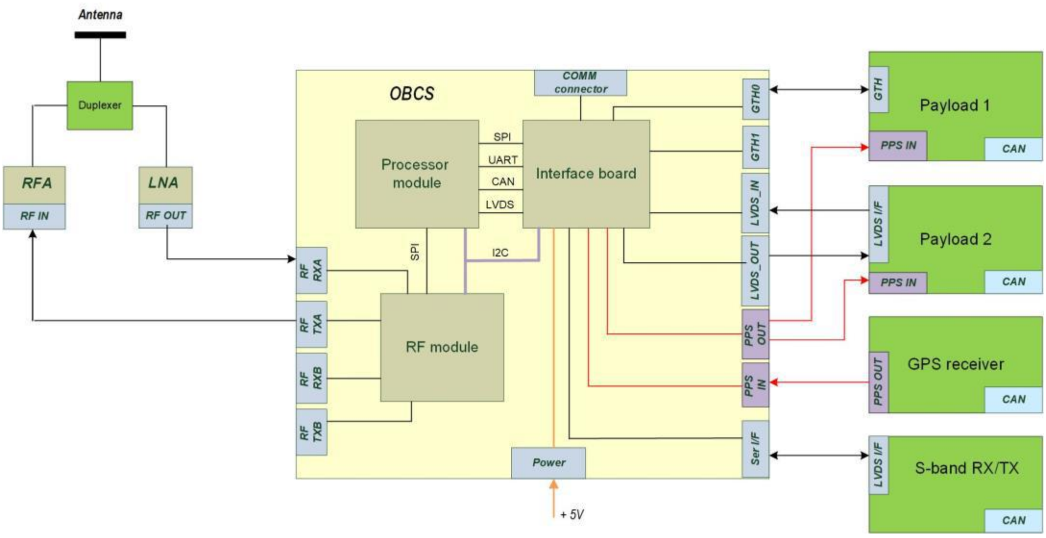

Fig. 1. OBCS usage diagram.

During interfaces definition, the usage of OBCS was taken into account to solve a wide class of tasks in such missions as remote sensing of Earth, radar and scientific ones. Table 1 provides a list of the main interfaces that can be used for most nanosatellite platforms and CUBSAT projects.

Table 1. Definition of OBCS interfaces.

Interface	Type of signals	Protocol	Quantity	Transfer rate per 1 interface
LVDS I/O	LVDS33	Serial+Clock	32	622 Mbit/s
GTH transceiver	LVDS33	Serial+Clock	4	6 Gbit/s
CAN I/O	LVCMOS33	CAN SU2	2	500 Kbit/s
UART	LVCMOS33	Own	4	115.2 Kbit/s
I2C	LVCMOS33	I2C	2	400 Kbit/s
SPI	LVCMOS33	SPI	2	25 Mbit/s
PPS I/O	LVDS33	-	4	1 Hz

The concept of mechanical design is presented in Figure 2 and based on a ”sandwich” layout, as in CUBESAT, when the boards form in a multi-tiered structure, but unlike standard connectors, using highly reliable inter-board connectors of the Razor Beam type, which were used in the previous model of the on-board computer in the KazSTSat project [3].

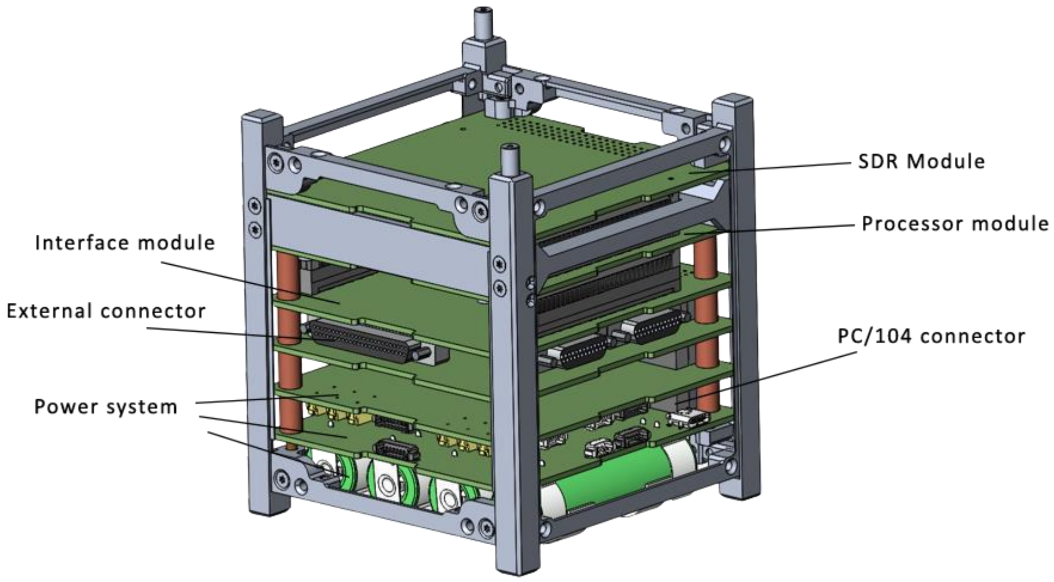

Fig. 2. The concept of the mechanical layout of the OBCS.

2. OBCS Software Architecture

In frame of development process of the architecture of the On-Board Software (OBSW), options for implementing architectural solutions of several satellite manufacturers were studied, as well as information from scientific sources in the field of S/C design and OBSW development were analyzed.

The traditional architecture of the S/C OBSW includes three main levels: the low or basic software level, the service level and the application level [4]. The low-level layer consists of a Real-Time Operating System (RTOS) and interface device drivers. The service layer is a layer of data handling and memory services. The application layer includes system management, subsystem management, and payload applications.

The research takes advantage of using orbit and design data from two different missions KazEOSat-1 and KazSTSat [5] for the analysis. The architecture of the onboard software of the KazEOSat-1 [6] and KazSTSAT [3] remote sensing SC has a basic structure corresponding to the traditional static architecture of the OBSW, considered in [7] and [8].

Widely used in the design and development of OBSW ultra-small Earth satellites flight software commercial frameworks consist of three main layers. At the same time, in order to maintain flexibility and applicability both to the tasks of various S/C and to various platforms, the layers differ from the traditional OS architecture: the service level includes additional blocks of system components and libraries for compatibility with the Platform Support Package level (PSP). [9]

Applying the results of the analysis of the OBSW design solutions, as well as the experience in the software development of the universal on-board computer OBCARM, the technological payload of the space system for scientific and technological purposes KazSTSAT, the architecture of the OBSW was developed (Figure 3), taking into account the OBCS hardware. The presented architecture consists of three main levels: the level of hardware and basic software, the level of services and the application level.

The application layer performs the main modes of the SC and mission control, monitoring of the technical condition of the SC, management of subsystems and the system level FDIR (Failure Detection, Isolation and Recovery). In order to support the platform and payload operations, On-Board Control Procedures (OBCP) [8] provide the capability to interact with the OBSW by executing script files previously uploaded on board by operators of the ground station [7].

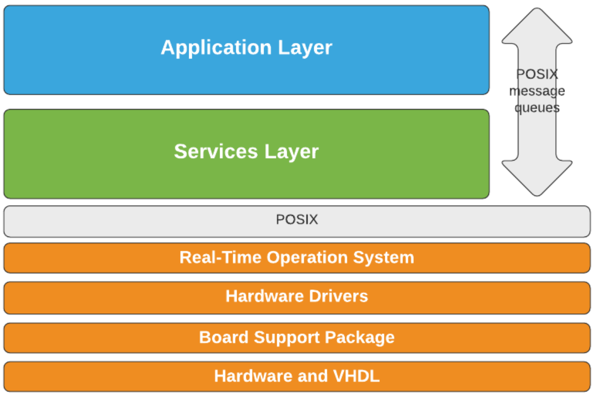

Fig. 3. On-board software architecture.

The service layer contains general-purpose data processing services, as well as services of the centralized Package Usage Standard (PUS) [10]. In addition, this layer includes hardware controllers offering interfaces to the input/output (I/O) boards of the OBCS and communication buses, such as CAN [11], and serial channels. Server-level software objects (CAN server, serial data transfer server, etc.) implement communication protocols, provide interfaces to the CAN bus, serial ports, etc.

The low-level software layer includes a real-time operating system (RTOS) and the on-board computer hardware drivers that provide access to dual-redundant central processing unit, peripherals and I/O boards. The core of the real-time operating system and the Board Support Package (BSP) provides the main services for the on-board computer software (scheduling, message sending, task manager, etc.), and also provides support for interfaces for various hardware modules.

BSP is a set of software used to initialize hardware devices on a board with a Zynq UltraScale+ processor module and implement procedures related to this board that can be used by the kernel and the device drivers.

The OBSW function blocks located on top of the RTOS use POSIX message queues for inter-process communication. The architecture of the service levels, services and applications allows to expand the system.

The main on-board software of the OBCS runs on the Zynq UltraScale+ SoC processor module located on the application processor board and performs the main functional tasks of the on-board computer/payload module. The main function of the OBSW is to provide management, control and data processing to support the operation of the S/C using the various space communications protocols, as well as performing the following main functions:

- data management and ground control and control support;
- S/C modes management;
- orientation and orbit control;
- platform management;
- payload monitoring.

The processes for performing these functions are mainly implemented at the services and applications level. The architecture of the OBSW with an expanded structure of the application and services layers is shown in Figure 4, where CTRL is the control vector, St_V is the status vector, AOCS_TLM is the AOCS telemetry vector, AOCS_CTRL is the AOCS control vector, PL_CTRL is the Payload control vector, PL_TLM is the Payload telemetry vector, Sat_CTRL is the SC subsystems control vector, MP_St_V is the SC subsystems telemetry vector.

Processes in the system are grouped according to functional blocks based on data input/output paths, data processing/formatting, protocol stacks and algorithmic elements. OBSW uses an event-driven architecture in real-time for I/O processes using the preemption mechanisms present in the operating system, while applications and services operate in round-robin scheduling slots within their priority group. OBSW communicates with SC devices (star tracker, GPS receiver, transponders, transmitters, payload instruments, etc.) via data bus and serial channels (LVDS interface, CAN, I2C interface, PPS, UART and SPI interfaces). Exchange of telemetry and telecommands with the ground segment based on the PUS (Package Utilization Standard) standard [10].

.

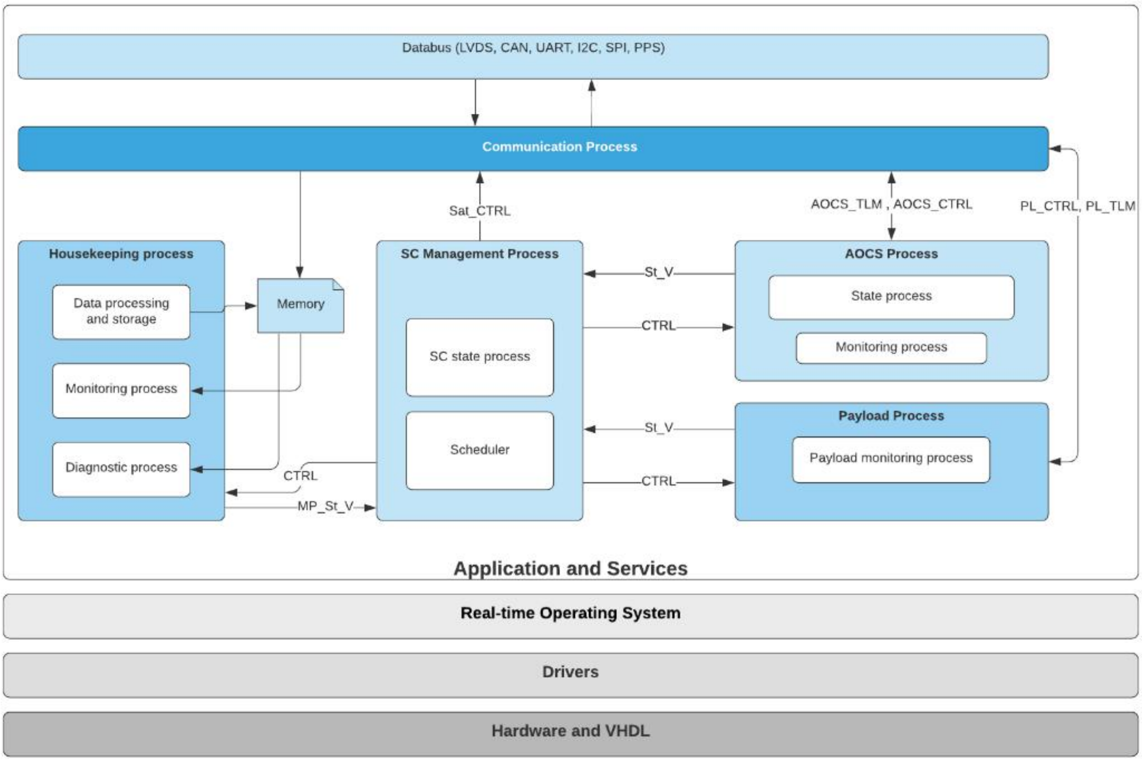

Fig. 4. OBSW architecture. Application and Services level.

In Figure 5 working algorithm of the Satellite management process is shown, which is the key software component mastering other software components in OBSW. Satellite management process functions include telemetry file recording, registration, data compression, automatic file cleanup, processing of S/C operating modes and task scheduling. The task scheduler is an execution unit which function is to issue an ordered list of telecommands.

The telecommands list is generated from SKED (schedule) files and varies depending on the S/C operating mode.

Satellite management process supports up to 6 operating modes:

1. Initialization (INITIALIZATION);
2. Test mode (TEST);
3. Launch and Early Orbit Procedure mode (LEOP);
4. Nominal mode (NOM);
5. FDIR mode;
6. Safe Mode.

First steps, the algorithm go through to initialization of the process and configure satellite parameters. Depending on AIT strap status (removed before flight), the algorithm ether to Test mode or flight one. Test mode is supposed to perform all functional tests within AIT campaign. In case AIT strap removed, the algorithm enters to LEOP mode in which all necessary deployment and configuration activities are performed according to LEOP SKED file. After successful LEOP ending indicated by LEOP flag, the algorithm goes to Nominal Mode where Scheduler executes either telecommands coming from Ground station or command lines which are extracted form NOM SKED file. If an error occurred during command execution or coming telemetry shows fault status of hardware the algorithm goes to FDIR (Fault Detection, Isolation, Recovery) mode which handles and mitigates occurred anomaly in spacecraft. If the algorithm doesn't find appropriate FDIR mode by condition, the last point is to go to Safe mode, which uses minimum hardware to keep satellite in live condition. Result of the satellite management process execution is telecommand packets generation to control satellite regarding on satellite current mode and hardware status. Input and output data format for the process are framed in standard forms, which are defined in [10].

Matrix of possible transitions from mode to mode is shown in Table 2, where CMD means transition by the command and CND – by condition.

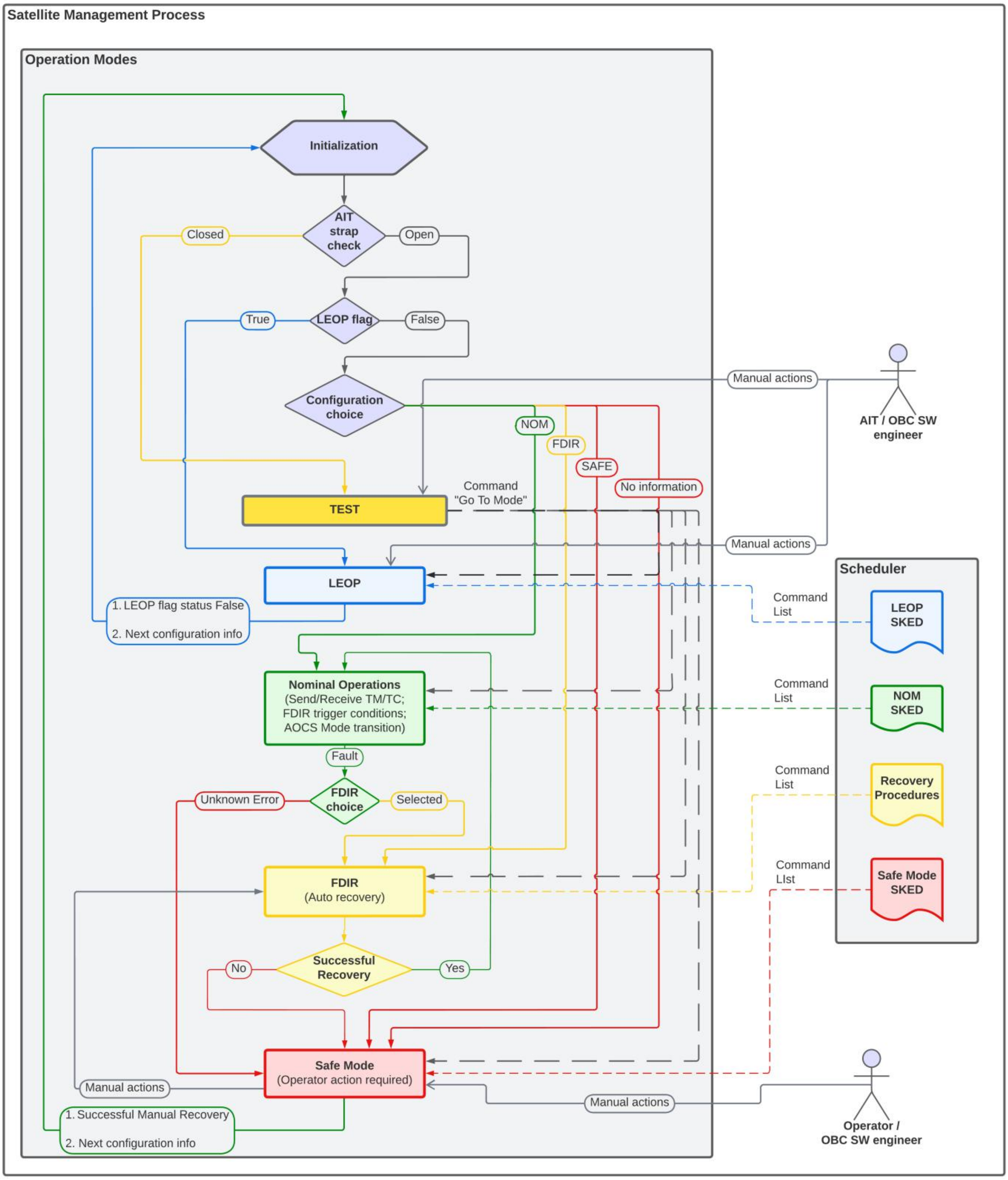

Fig. 5 Algorithm of satellite management process.

Table 2. Modes Transition Variations

From mode	To mode					
	INIT	**TEST**	**LEOP**	**NOM**	**FDIR**	**SAFE**
INIT		CND	CND	CND	CND	CND
TEST			CMD	CMD	CMD	CMD
LEOP	CND/CMD			CND/CMD		CND/CMD
NOM					CND/CMD	CND/CMD
FDIR	CMD			CND		CND /CMD
SAFE	CND CMD			CMD		

3. Circuit design solutions for the development of on-board control system modules

In frame of the development, work was carried out on the formation of circuit solutions at the level of functional blocks and components. As it was shown in the previous sections, the OBCS consists of 3 functional modules physically presented in the form of printed circuit boards:

- Processor module;
- SDR module;
- Interface board.

The enlarged schematic diagram of the processor module is shown in Figure 6. The main component of the processor module is the Zynq ZU9EG chipset - system-on-a-chip from Xilinx.

The Zynq EG series have a quad-core ARM® Cortex-A53 processor with a frequency of up to 1.5 GHz. Combined with the Cortex-R5 dual-core real-time processor, the Mali-400 MP2 GPU and 16 nm FinFET+ programmable logic, the EG devices have an architecture focused on applications in 5G wireless networks, cloud computing, as well as aerospace and defense applications. All the peripherals of the processor module are connected to it, such as: QSPI, DDR4, NAND memory chips, reference generators and an oscillator, as well as voltage converters. An I2C node based on the MSP430 microcontroller is used to monitor the processor module.

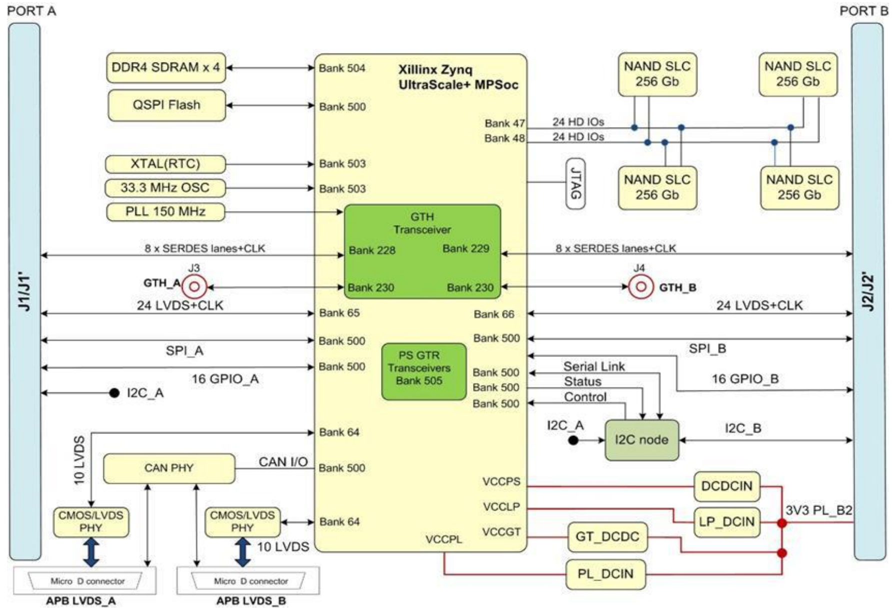

Fig. 6. Block diagram of the OBCS processor module.

Its tasks include collecting telemetry of the state of the processor module, as well as, if problems are detected in the operation of the Zynq system-on-chip, performing module reconfiguration both in automatic mode and in direct command execution mode. One of the features of the OBCS being developed is the availability of a high-speed data transmission channel to the ground station based on SDR. At the same time, Zynq resources are used for demodulation, mixer functions and signal filtering, which significantly unloads the hardware design of the OBCS as a whole, and only direct conversion to analog form and back to digital is performed on the SDR module. The functional schematic diagram of the SDR module is shown in Figure 7.

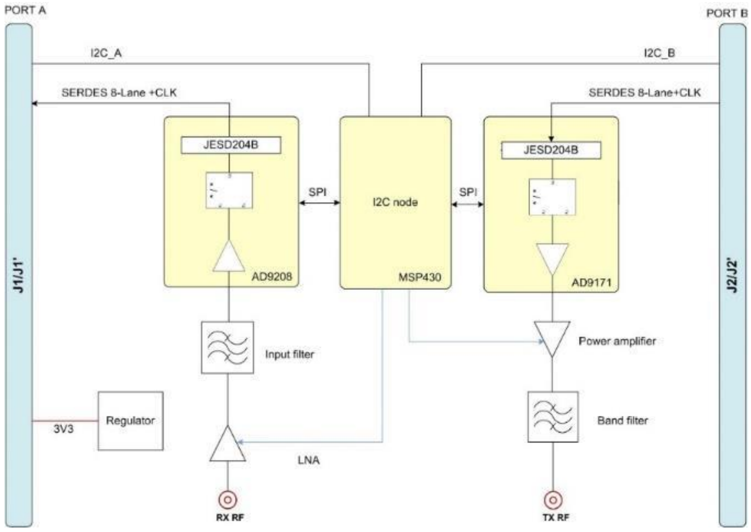

Fig.7. Block diagram of the SDR module of the OBCS.

The board uses high-performance analog-to-digital and digital-to-analog converter chips providing up 6 GSPS sampling rate. For receive path a signal is pre-amplified along the receiving path to the desired signal level and fed to the input filter to allocate the operating frequency bands, then the signal is fed directly to the analog-to-digital converter, which converts to a digital stream via 8 LVDS lines. For the transmitting part, conversions are performed in reverse order. The design feature of the considered OBCS is its application both on the CUBESAT platform and on a small satellite platform for mostly used configurations. Figure 8 shows an enlarged schematic diagram of the interface board, which provides this capability. The external connectors may not be soldered with the CUBESAT configuration, and the interface with the rest of the modules is provided via the PC/104 pad.

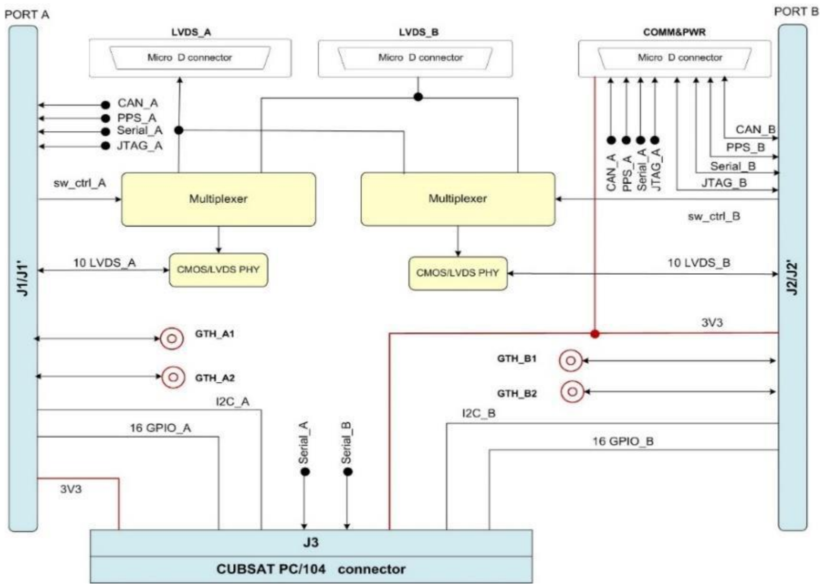

Fig. 8. Block diagram of the OBCS interface board.

.

Conclusion

This development will dramatically simplify designing a nanosatellite architecture by using modular flexible structure and integrated solution which effectively implements several functions based on modern technologies (SoC, SDR) allowing in short terms and low prices to build small satellite systems.

As part of the research work, an analysis was carried out in the field of the OBCS design methodology, the characteristics of existing models of OBCS for nanosatellites and CUBESAT platforms were considered. The variety of models available on the markets indicates the trend of rapid growth of this segment against the background of great interest in projects based on nanosatellites and their prospects for application in most missions. As a result of the analysis, it can be concluded that the proposed development is relevant. The key difference from similar OBCS will be multifunctionality, adaptive integrability with various payload and the use of SDR technology for high-speed data transmission.

Work has also been carried out on the development of OBCS electronics circuitry in the form of block circuit diagrams, which are the basis for further development of detailed design of printed circuit boards.

Acknowledgments (or Funding)

This research is funded by the Aerospace Committee of the Ministry of Digital Development, Innovations and Aerospace Industry of the Republic of Kazakhstan (BR109018/0221) "Development and testing of domestic key technologies for the creation and control of spacecraft".

REFERENCES

1 Arseno D., Edwar E., Harfian A.R., Salsabila J.N. Characterization of On Board Data Handling (OBDH) Subsystem. *Proceeding of the IEEE 13th Intern. Conf. on Telecommunication Systems, Services, and Applications (TSSA)*, 2019. doi: 10.1109/TSSA48701.2019.8985466

2 Sarsenbayev Y., Mussina A., Ismailov U., Bychkov A. Spacecraft onboard control system. *Detailed information: Utility model № 6912*. 2022, p.1. Available at: gosreestr.kazpatent.kz/Utilitymodel/Details?docNumber=351783.

3 ESA KazSTSAT (Kazakh Science and Technology Satellite), 2022. Available at: www.eoportal.org/ satellite-missions/kazstsat#list-of-payloads-on-the-spaceflight-sso-a-rideshare-mission.

4 Omran E.A., Murtada W.A., Serageldin A. Spacecraft on-board real time software architecture for fault detection and identification. *Proceeding of the 12th Intern. Conf. on Computer Engineering and Systems (ICCES)*, 2017. doi:10.1109/ICCES.2017.8275379

5 Ten V., Oralmagambetov B., Murushkin S., Bekembayev A. Absolute passive mode pecularities and applications for LEO missions. *Proceeding of the 67th I Intern. Astronautical Congress*, 2016. doi: IAC-16.D1.IP.5.x35477. Available at: www.iafastro.directory/iac/archive/browse/IAC-16/D1/IP/35477/

6 Verdict Media Limited. *KazEOSat-1 Earth Observation Satellite*. 2022. Available at: www.aerospace-technology.com/projects/kazeosat-1-earth-observation-satellite.

7 Wenker R., Legendre C., Ferragutoz M. On-board software architecture in MTG satellite. *Proceeding of the IEEE Intern.Workshop on Metrology for AeroSpac*e, 2017. doi:10.1109/MetroAeroSpace.2017.7999588

8 Tipaldi M., Ferraguto M., Ogando T., Camatto G., Wittrock T., Bruenjes B., Glielmo L. Spacecraft autonomy and reliability in MTG satellite via On-board Control Procedures. *Proceeding of the IEEE Intern. Workshop on* Metrology for Aerospace, 2015, pp. 155 – 159. doi:10.1109/MetroAeroSpace.2015.7180645

9 Miranda DJF, Ferreira M, Kucinskis F, McComas D. A Comparative Survey on Flight Software Frameworks for 'New Space' Nanosatellite Missions. *J Aerosp Technol Manag*, 2019, V.11: e4619. Available at www.doi.org/10.5028/jatm.v11.1081

10 European Cooperation for Space Standardization Ground systems and operations - telemetry and tele-command packet utilization. Euro pean Cooperation for Space Standardization. Available at: ecss.nl/standard/ecss-e-st-70-41c-space-engineering-telemetry-and-telecommand-packet-utilization-15-april-016/. ECSS-E-70-41C, 2016.

11 ISO. Road vehicles - Controller area network (CAN). Part 3: Low-speed, fault-tolerant, medium-dependent interface. Available at: www.iso.org/standard/36055.html. ISO 11898-3, 2006.

Article accepted for publication 31.10.2022