



[4] Kopelovich A. *Inzhenernie metody rascheta pri vibore avtomaticheskikh regulyatorov*. M: Metalurgizdat, 1960.

[5] <https://uk.mathworks.com/products/sysid.html>

[6] <https://uk.mathworks.com/products/simulink.html>.

Отримана в редакції 29.04.2023. Прийнята до друку 15.05.2023. Received 29 April 2023. Approved 15 May 2023. Available in Internet 19 June 2023.

---

UDC 004.77:[654.1:316.774]

# DEVELOPMENT OF THE COMPLEX LABORATORY WORKS FOR RESEARCH DIGITAL TELEVISION AND MOBILE COMMUNICATION SYSTEMS BASED ON SDR TECHNOLOGY

Illia Bondarenko, Oleksii Dushenkov, Valerii Loshakov  
Kharkiv National University of Radio Electronics (Ukraine)

Copyright © 2021 by author and the journal “Automation of technological and business – processes”.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0>



DOI: 10.15673/atbp.v15i2.2524o

---

**Abstract.** *Recent decades have marked by revolutionary changes, including the emergence of software-defined radio equipment (SDR). However, there is virtually no training in this area, which does not allow for the full realization of the benefits of new, more advanced communication systems. Therefore, this work aimed at solving this problem. It presents the results of the development of a set of laboratory work on modeling new generations of mobile communication and digital television systems implemented using SDR technology and the methodological instructions necessary for students to perform these works.*

**Keywords:** SOFTWARE DEFINED RADIO, SOFTWARE DEFINED RADIO SYSTEMS, WIRELESS COMMUNICATION, DIGITAL TELEVISION, BASE STATION, LimeSDR.

---

## I. Introduction

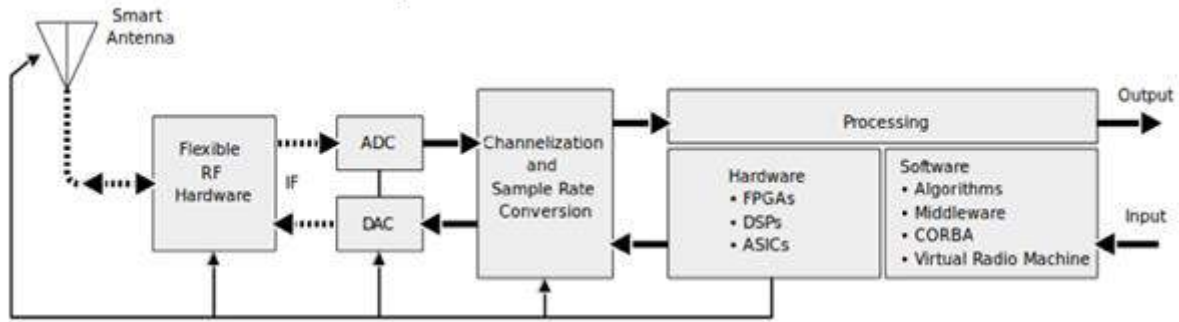
Technology development does not stand still. If earlier a set of various task-oriented electronic circuits was used to process and convert signals, now it is possible to do all this with the help of software. For example, to set or change the frequency range, modulation type, signal output power, etc. At the same time, most of the work of SDR radio platforms performed on a regular personal computer or an FPGA controller that provides the necessary software configuration of the radio system. Due to the flexibility of the technology, it is possible to programmatically implement virtually all known methods of signal generation and processing used in mobile communications, television, radio broadcasting, radiomonitoring and information security systems.

## II. Using sdr technology in mobile communications

### 2.1 Features of SDR technology and its current state

SDR (Software Defined Radio) is radio equipment with software-defined radio frequency parameters. This feature increases the flexibility and versatility of communication systems. At the same time, any changes in signal processing and generation can quickly implemented with just a program that describes the operation of the SDR [1].

In the simplest "ideal" case of an SDR transceiver (Fig. 2.1), the antenna is connected to the input of an analog-to-digital converter (ADC) or the output of a digital-to-analog converter (DAC). The output signal of the ADC converted into the required form of data representation after software digital processing. The DAC input receives a programmatically synthesized digital signal to be transmitted.

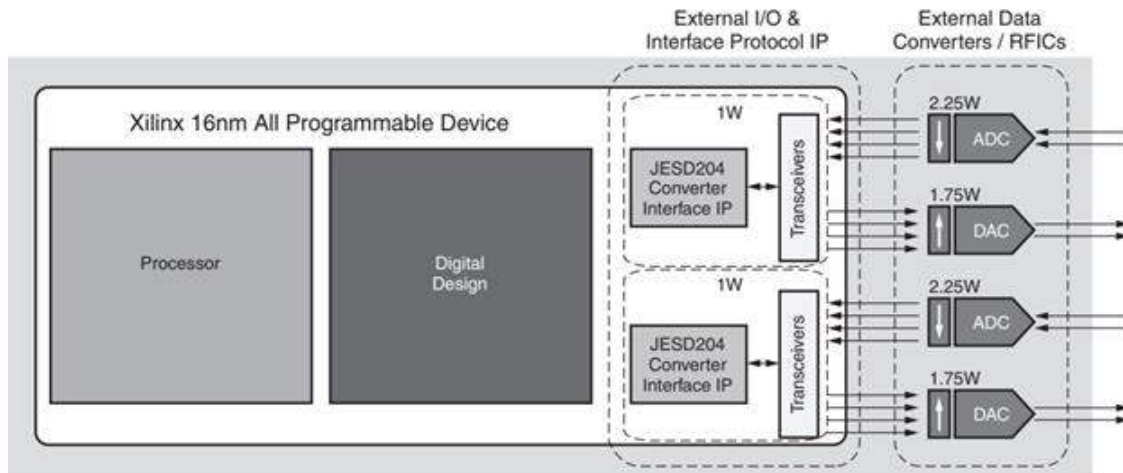


**Fig. 2.1. Typical SDR structure**

The main limiting factors in the implementation of such an "ideal" SDR system are the parameters of the ADC and DAC. Since in the ideal case of SDR (when digitized directly), the frequency range of the input signal is limited by the sampling rate of a particular ADC, which, according to Nyquist's theorem, must be at least twice the frequency of the received signal. The same applies to the transmission path.

To ensure the frequency independence of the SDR system from the parameters of the ADC/DAC between them and the antenna, preliminary signal frequency converters (up/down converters) are usually used, providing at the same time radio frequency selection and amplification

According to the IMT-2020 development plan, fifth-generation mobile networks will operate at ultra-high frequencies up to 60 GHz using MIMO technology, in particular, its development called Massive-MIMO. Massive-MIMO is a two-dimensional array of antennas with 128 to 256 antennas. This technology makes it possible to form a diagram on demand for each subscriber, effectively use re-reflection, and increase spectral efficiency in general. Xilinx aims at practical implementation of such systems and has announced the release of an integrated RFSoc circuit that combines eight DAC/ADCs, FPGAs, and a multi-core ARM Cortex A53 processor (Fig. 2.2) [2].



**Fig. 2.2. RFSoc chip structure**

The chip has 12-bit ADCs with a sampling rate of 4 GHz and 14-bit DACs with a sampling rate of 6.4 GHz. Such characteristics of the converters allow processing directly at the carrier frequency using only a relatively simple microwave part, which simplifies the creation of adaptive antenna arrays that will provide spatial, time and polarization signal processing and auto-compensation of interference. The RFSoc chip requires data stream processing at a speed of up to 300 Gbps to ensure the operation of the ADC/DAC. For this purpose, the chip is equipped with the most productive Xilinx Zynq Ultra Scale + SoC. This SoC also solves the problem inherent in previous generations of SDRs, namely the multi-bit connection of a spaced processor and FPGA. Thus, the RFSoc is an SDR assembled in a "perfect circuit". In this case, the analog part consists only of an antenna, low-noise amplifiers/power amplifiers and filters.

### 2.2 Comparative characteristics of existing SDR platforms

A variety of SDR platforms are currently on the market. The characteristics of the most common of them can be found in according to the efficiency/cost criterion, the LimeSDR radio platform stands out among the most common SDR projects. This project is attractive at a relatively low cost of \$299 for fairly high technical characteristics [3]:

RF Transceiver: Lime Microsystems LMS7002M MIMO FPRF ([Datasheet](#))

FPGA: Altera Cyclone IV EP4CE40F23 - also compatible with EP4CE30F23

Memory: 256 MBytes DDR2 SDRAM

USB 3.0 controller: Cypress USB 3.0 CYUSB3014-BZXC

Oscillator: Rakon RPT7050A @30.72MHz ([Datasheet](#))

Continuous frequency range: 100 kHz – 3.8 GHz

Bandwidth: 61.44 MHz



- RF connection: 10 U.FL connectors (6 RX, 4 TX)
- Power Output (CW): up to 10 dBm
- Multiplexing: 2x2 MIMO
- Power: micro USB connector or optional external power supply
- Status indicators: programmable LEDs
- Dimensions: 100 mm x 60 mm
- Antenna accessory pack pledge:
  - Four antennas with SMA connectors included
  - Four UFL to SMA cables included - 8" (~203mm) cable, not measuring connectors
  - Antennas are omni-directional, and bend to -90, -45, 0, 45, and 90 degrees
  - Antennas tuned to the major bands: 800-960MHz / 1710-2170MHz / 2400-2700MHz

The LimeSDR platform based on the Lime Microsystems LMS7002M radio chip (Fig. 2.3), which contains a MSP, receiver and transmitter mixers, receiver and transmitter filters, receiver gain control circuit, transmitter power control circuit, ADC and DAC. It is important that it is completely open source, i.e., the design of the device and PCB, the source code of the FPGA, microcontroller, and PC client programs are available.

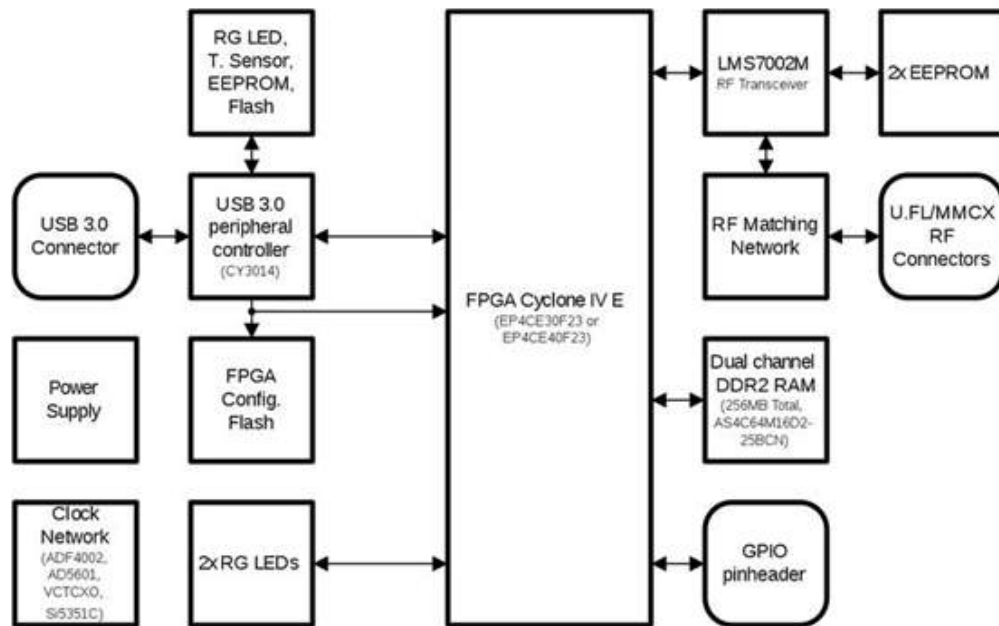


Fig. 2.3 Block diagram of the LimeSDR platform

### 2.3 Practical characteristics of SDR devices

The main technical characteristics of SDR devices are significantly superior to those of radio devices implemented using traditional technologies. Given the variety of characteristics, let us demonstrate this advantage using the example of an SDR receiver.

The most important parameters of a receiving device are: frequency range;

Noise Figure (NF) or sensitivity; dynamic range, which is usually defined as the level of the third-order intersection point (IP3) [4].

It should be noted that the definition of all these parameters for SDR receivers has peculiarities compared to their analog prototypes. Let us briefly consider these features.

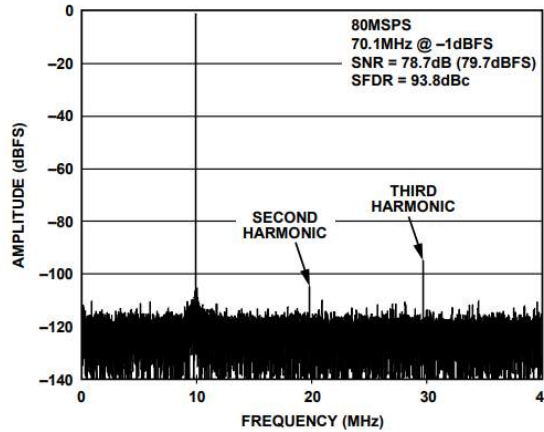
One of the main components of the SDR receiver, which largely determines all the above characteristics, is the ADC. In particular, if the range of input signal amplitudes exceeds the dynamic range of the ADC, it acts as a rigid signal limiter. This leads to a sharp increase in the levels of third-order intermodulation components and other combination components.

To evaluate the above basic parameters of the receiver, the signal-to-noise ratio (SNR) is usually used, which is equal to the ratio of the power of the useful signal to the noise power:

$$SNR = 10 \log_{10} \left( \frac{P_{signal}}{P_{noise}} \right) = 20 \log_{10} \left( \frac{A_{signal}}{A_{noise}} \right),$$

Where:  $P_{signal}, P_{noise}$  are the average signal and noise powers;  $A_{signal}, A_{noise}$  are the rms values of the signal and noise amplitudes measured in the system bandwidth.

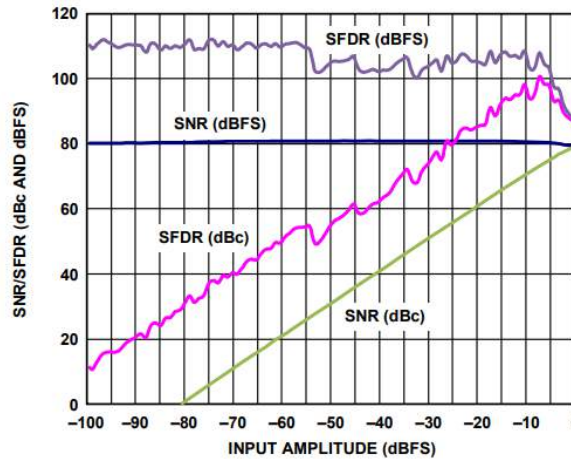
Fig. 2.4 shows the signal spectrum obtained as result of digitizing the useful signal. It can be seen that the useful signal level is 0dB, and the harmonic level (the most powerful parasitic frequency component) is -93.8dB.



**Fig. 2.4. Amplitude spectrum of a digitized sinusoidal signal**

Other parameters related to signal-to-noise ratio and dynamic range are: Spurious-Free Dynamic Range (SFDR) - the dynamic range free of parasitic components; Signal-to-Noise and Distortion Ratio (SINAD) - the ratio of the root mean square value of the signal in the full scale range to the root mean square sum of all noise components; Total Harmonic Distortion (THD) - the ratio of the power of the useful signal to the power of harmonics without noise.

The values of these parameters measured in dBc, when the reference is the level of the useful signal at the carrier frequency, or dBFS, when the reference level corresponds to the full scale of the receiver. In Fig. 2.5 shows the graphs of SNR and SFDR of the receiver depending on the input signal level from -100dBm to 0dBm (lower scale).



*AD9265-80 Single-Tone SNR/SFDR vs. Input Amplitude ( $A_{IN}$ ) with  $f_{IN} = 98.12$  MHz*

**Fig. 2.5. Graphs of SNR and SFDR dependence on the input signal amplitude**

In Fig. 2.5, each graph shows specific characteristics of the receiver, which together give a complete picture of the properties of the receiving device:

SFDR (dBc) (pink curve) relative to the reference signal varies from -12 dBc to -100 dBc (scale on the left) and reflects the excess of the signal level over the noise level;

SFDR (dBFS) is the dynamic range free from parasitic components (purple curve), varies from 110 dBFS to 75 dBFS (left scale);

SNR (dBFS) - signal-to-noise ratio in the full scale of the receiver (blue line), practically does not change (left scale).

### III. Object, subject and methods of research

**The object** of research is methods of implementation of communication systems and digital television using the latest trends and developments in SDR technology.

**The subject** of the research is models of DVB-S/S2 satellite television systems and new generation mobile communication systems implemented on the basis of SDR technology.

**Research methods** - mathematical and physical modeling, experiment, comparative analysis.

**The purpose of the work** is to create educational laboratory works on modeling and research of mobile communication and satellite television systems of new generations based on SDR technology for their use in the educational process in the specialty "Telecommunications and radio engineering".

### IV. Results of development complex of laboratory work on research of digital television and communication systems of new generatins

#### 4.1 Development of software for LimeSDR and architecture of the complex laboratory installation

As result of the analysis of current trends, a list of promising technologies that can implemented on basis of LimeSDR in a complete laboratory setup highlighted:



DVB-T2 standard digital terrestrial broadcasting system;  
DVB-S2 standard digital satellite broadcasting system;  
GSM and LTE standard mobile communication systems.

Appropriate models and software developed to launch and control radio frequency parameters of receivers, transmitters, pico networks and base stations of the abovementioned standards on basis of LimeSDR.

The freeware operating system Ubuntu chosen to implement the software for LimeSDR. Ubuntu is a GNU/Linux operating system that based on Debian packages [5]. Ubuntu is one of the most popular distributions, it actively developed and has a very wide support from the user community around the world and recommended for the first time to get to know Linux. By default, Ubuntu comes with the Unity graphical environment.

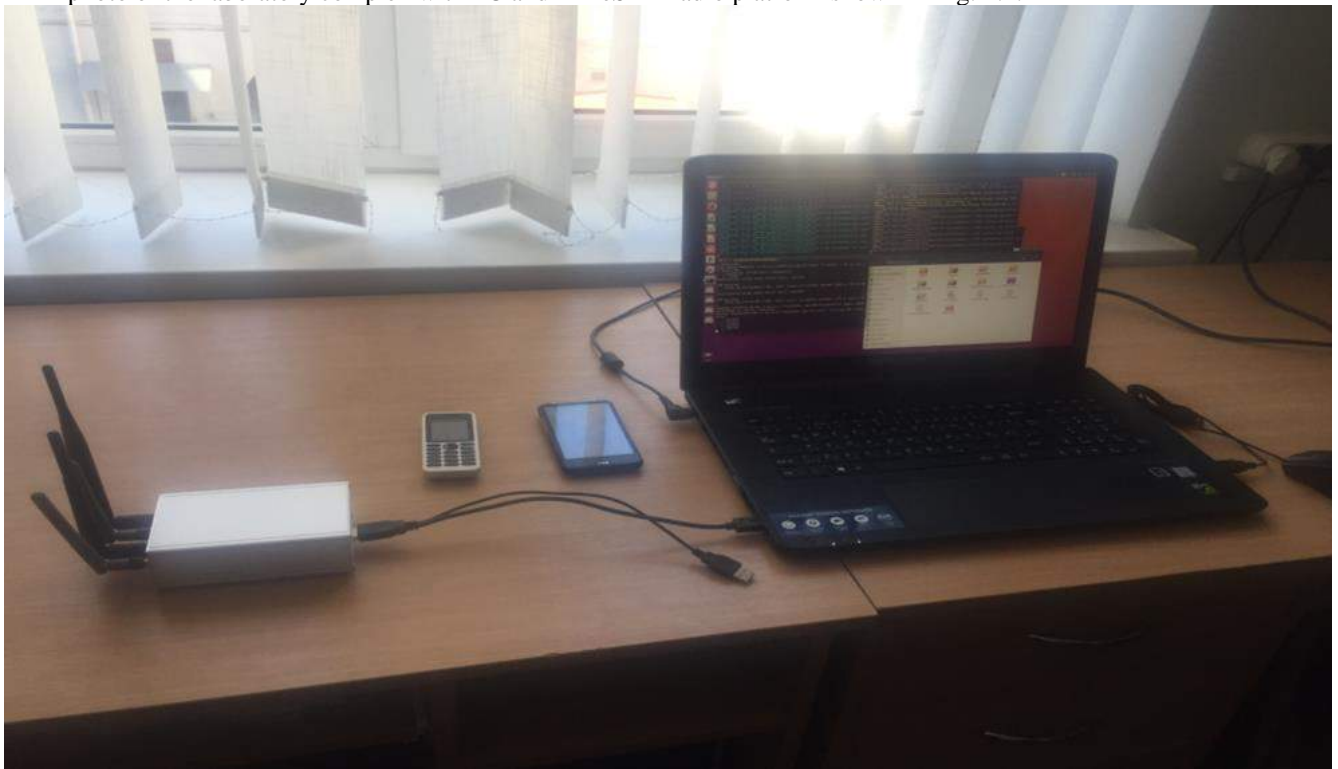
The main working tools will be Terminal, the console utility GNURadio and the visual shell (GUI) for model building - GNURadioCompanion (GRC). Terminal and GRC are located on the left panel of the desktop.

A command in the Linux console is a string of characters consisting of a command name and arguments separated by spaces. The arguments provide the command with additional parameters that determine its behavior. For example, the command echo 12345 will display a string of 12345 characters entered as an argument to the command. Commands that are part of the OS called built-in commands. In addition, the names of executable files used as commands. For example, to run GNURadioCompanion, the command grc must entered. As arguments, commands passed as keys or options consisting of a dash and one or more characters. An example of such a command is:

ls -l. ls is a command to display information about files. When entered without arguments, this command will simply list the files in the current directory. If entered with -l, the list is in long format, showing attributes, owners, date and time. In most cases, you can get help information for using a particular command by typing its name with the --help parameter: ls --help - displays help for the ls command. In most cases, it is acceptable to use the -h switch to get help information.

The source code for each model written in C++ and Python and requires precompilation, building the package and installing it using a standard Terminal tool. All operations with source code are performed under a root account by entering the console command "sudo -s". Before starting work with source code, it is necessary to update all packages of the operating system to the current state by entering the console command "apt-get update". In addition, the graphical package manager Synaptic installed beforehand with the console command "apt-get install synaptic".

A photo of the laboratory complex with PC and LimeSDR radio platform shown in Fig. 4.1.



**Fig. 4.1. Photo of the laboratory complex**

The computer also used as a router for the LimeSDR platform for remote laboratory work and for subscriber access to the Internet for the communication software package.

#### **4.2 Laboratory work of study digital satellite broadcasting system of the DVB-S2 standard**

**Purpose of the work:** Study of the principle of signal formation in the DVB-S2 digital television system. Acquiring basic skills in working with the GNURadio program.

Digital Video Broadcasting - Satellite v.2 (DVB-S2) is the second generation of the digital satellite broadcasting standard. The standard developed at the same time as the introduction of HDTV and H.264, respectively, and improvements made to coding and modulation. A new compression technology also implemented. This made it possible



to broadcast HDTV in the same frequency band previously occupied by the previous DVB-S standard for SDTV broadcasts, with significant power savings.

**How it works:** To generate the signal used in the DVB-S2 standard, you need to install GNURadio according to the following instructions.

Open the terminal with the key combination Ctrl+Alt+T and enter the following commands: `sudo apt-get update;`  
`sudo dpkg --configure -a;` `sudo apt-get install synaptic xterm gnuradio.`

You can open the installed program by searching in the "Show Programs" menu or using the following command: `gnuradio-companion.` In all works for educational purposes, the source file used as a unified source.

Build the DVB-S2 signal modulator circuit shown in Fig. 4.2.

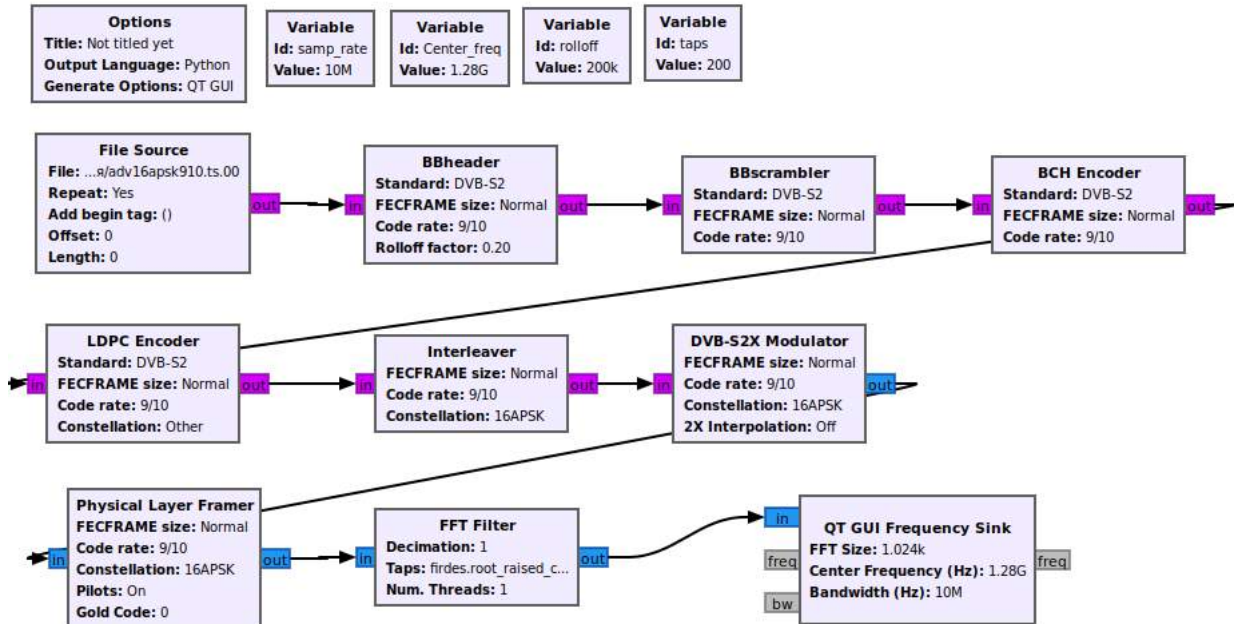


Fig. 4.2. DVB-S2 signal modulator circuit diagram

To build the schematic, it recommended to use the icon of which can found on the toolbar. As a source, specify the location of the file with the .ts extension provided by the teacher. Parameters that specified in several places in the program recommended to be set through the *variable* block, when using which you can enter *id* instead of numbers in other blocks.

Run the assembled circuit by clicking the green triangle. Take screenshots of the signal at several points in time, and summarize the signal parameters in a table. Draw conclusions about the signal parameters and describe what causes them.

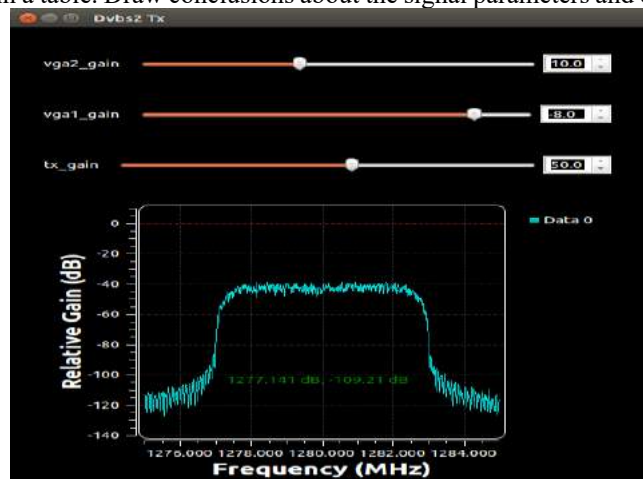


Fig. 4.3. Result of successful DVB-T2 signal transmission

**The report should include:** Title page, purpose of the work, constructed circuit and obtained graphs, description of signal parameters, conclusions.

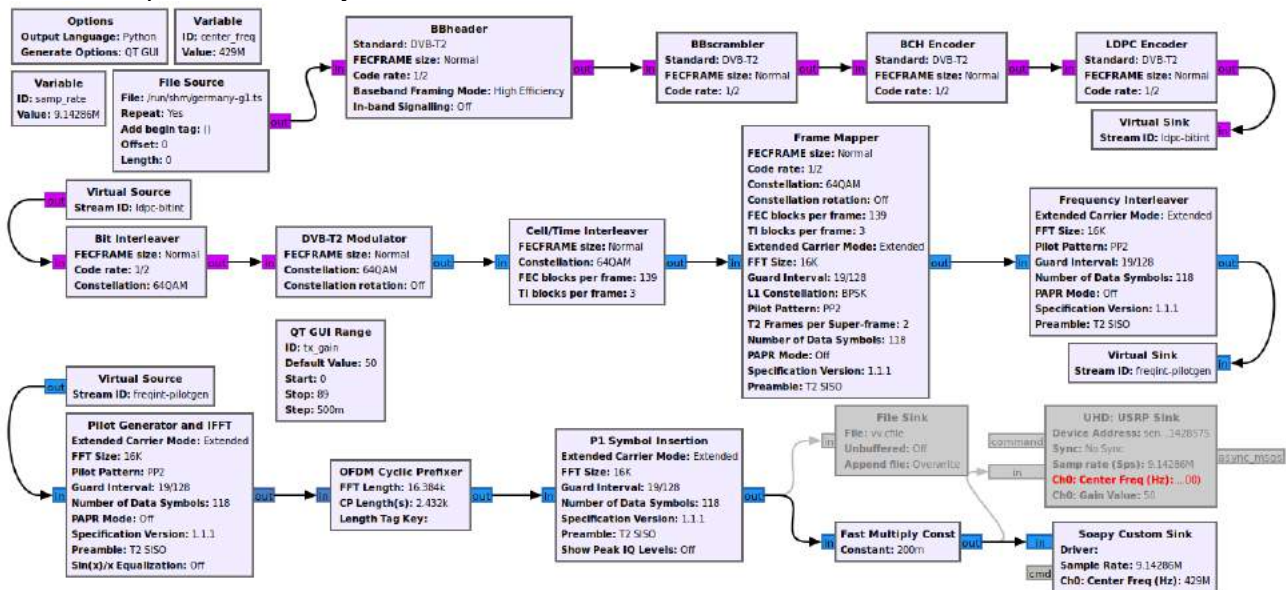
### 4.3 Laboratory work on study the digital terrestrial television standard DVB-T2

**Purpose of the work:** To study the signal in the DVB-T2 digital television transmission system.

DVB-T2 is a second-generation European digital television standard. It has increased network coverage by 30-50% compared to DVB-T, but is not hardware compatible with the latter. The changes affected the OFDM modulation parameters, error correction system, protection intervals, MISO mode support, etc. The list of service capabilities has significantly expanded to include ultra-high-definition television UHD TV and 3DTV.

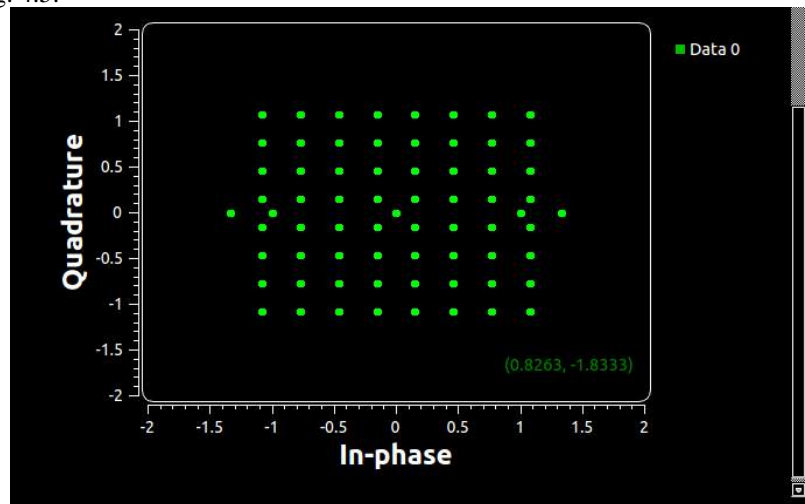


**How it works:** Open the GNURadio program and build the circuit shown in Fig. 4.4. Recommendations for work are similar to the previous laboratory work on the DVB-S2 standard.



**Fig. 4.4. DVB-T2 signal modulator**

Run the scheme and take screenshots of the multi-position signal constellation. An example of the resulting signal constellation shown in Fig. 4.5.



**Fig. 4.5. Example of the obtained constellation**

Change the gain value several times (by changing the tx\_gain value) by moving the corresponding cursor. Describe how the signal changes as the gain value changes.

**The report should include:** Title page, purpose of the work, the constructed circuit, the obtained graphs and their description, conclusions on the work.

#### 4.4 Laboratory work on the study of the communication system of the GSM standard

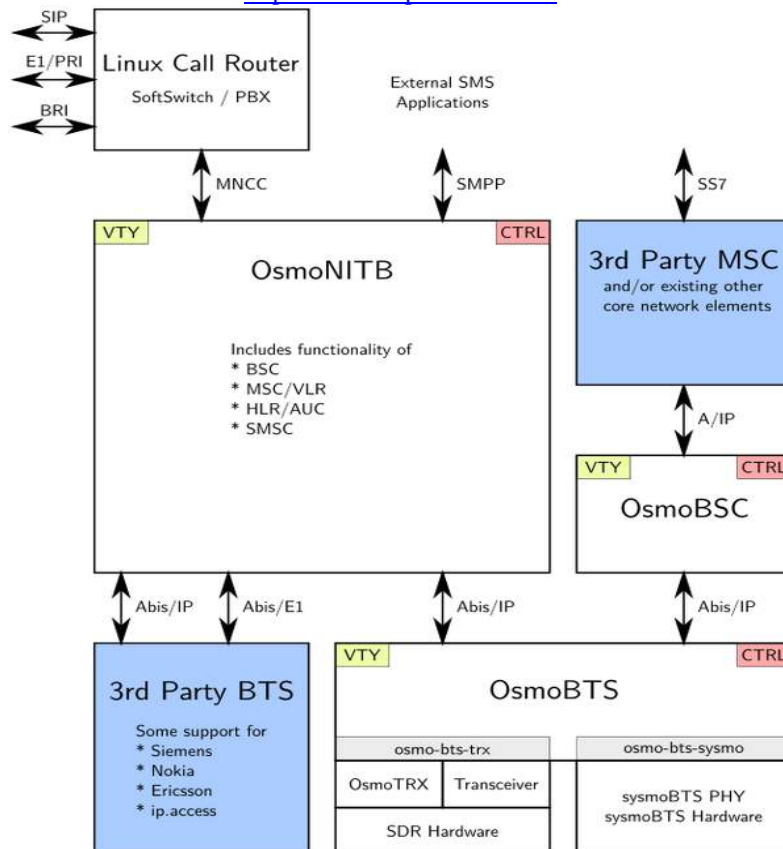
Due to its widespread use and high documentation, the GSM digital mobile cellular communication standard is an ideal candidate for studying the principles of cellular communication.

As part of the work, the basic infrastructure of a GSM picosota base station (BS) deployed. For this purpose, we used the freely distributed osmocomb (Open source mobile communications) software package, which consists of dozens of software tools that can work both separately and form an integral infrastructure [6].

To launch our own BS with the minimum set of parameters required for operation, we used software tools based on a computer running Ubuntu Linux and LimeSDR:

- OpenBSC - GSM base station controller;
- OsmoBTS - software implementation of the BS;
- OsmoTRX - a receiver that implements the physical level of the radio interface in accordance with 3GPP specifications.

The general scheme of the osmocomb software package shown in Fig. 4.6.



**Fig. 4.6. General scheme of the osmocomb software package**

Before you start building the BS, you need to install the related libraries. To do this, open terminal and perform all actions on behalf of the superuser by entering the `sudo -s` command.

Next, we install the libraries with the `apt-get install` command, listing all the necessary libraries: `libtool, shtool, libusb-1.0-0-dev, automake, autoconf, git-core, pkg-config, make, gcc` (Fig. 4.7).

```

root@ubuntu: ~
ubuntu@ubuntu:~$ sudo -s
root@ubuntu:~# apt-get install libtool shtool libusb-1.0-0-dev automake autoconf git-core pkg-config make gcc
Reading package lists... Done
Building dependency tree
Reading state information... Done
autoconf is already the newest version.
autoconf set to manually installed.
automake is already the newest version.
gcc is already the newest version.
libtool is already the newest version.
libusb-1.0-0-dev is already the newest version.
make is already the newest version.
make set to manually installed.
pkg-config is already the newest version.
The following NEW packages will be installed:
  git-core shtool
0 upgraded, 2 newly installed, 0 to remove and 173 not upgraded.
Need to get 151 kB of archives.
After this operation, 579 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
    
```

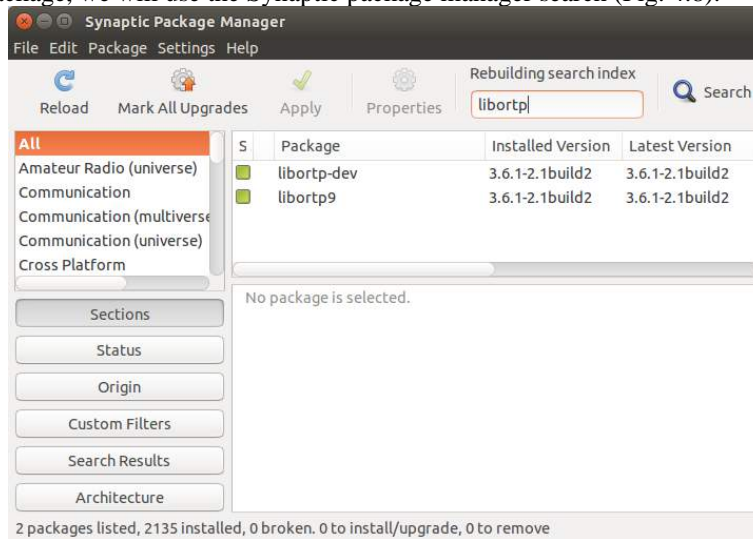
**Fig. 4.7. Installing libraries for the GSM BS**

The procedure for installing all components is as follows:

- a) Download the source code - `git clone "component address"`;
- b) Go to the folder with the source code - `cd "component name"`;
- c) Run the script - `autoreconf -I`;
- d) Compile the source code - `configure`;
- e) Build the package - `make`;
- f) Installing the package - `make install`;
- g) Setting up the binding of dynamic links - `ldconfig -i`;
- h) Exit to the root directory - `cd ...`.



All of the above steps are performed in parallel for the following libraries: libosmocore, libpcsclite-dev, libosmo-abi. To install the ortp package, we will use the Synaptic package manager search (Fig. 4.8).



**Fig. 4.8. Finding and installing the ortp package**

For the osmocore software package to interact with LimeSDR. It is necessary to install the OsmoTRX, which implements the physical layer of the radio interface in accordance with 3GPP technical specifications (TS) [7]: TS 05.01 "Physical layer in the radio communication path"; TS 05.02 "Multiplexing and multiple access in the radio communication path"; TS 05.04 "Modulation"; TS 05.10 "Synchronization of radio subsystems". The location of OsmoTRX in the osmocore architecture shown in Fig. 4.9.



**Fig. 4.9. OsmoTRX in the osmocore architecture**

Using the `trx-bench` command, we start the test and get the measured values of the parameters (bit error rate, frame error rate, time spent on encoding and decoding).

```

[+] Testing: GSM TCH/AFS 7.95 (recursive, flushed, punctured)
[.] Input length : ret = 165 exp = 165 -> OK
[.] Output length : ret = 448 exp = 448 -> OK
[.] Pre computed vector checks:
[.] Encoding: OK
[.] Decoding base:
[.] Decoding SIMD:
[.] Code N 3
[.] Code K 7
[.] Decoding: OK

[.] BER tests:
[.] Testing base:
[.] Input BER..... 0.023492
[.] Output FER..... 0.000100
[.] Testing SIMD:
[.] Input BER..... 0.023337
[.] Output FER..... 0.000000

[.] Performance benchmark:
[.] Testing base:
[.] Encoding / Decoding 10000 bursts on 1 thread(s):
[.] Elapsed time..... 2.029035 secs
[.] Rate..... 2.207946 Mbps
[.] Testing SIMD:
[.] Encoding / Decoding 10000 bursts on 1 thread(s):
[.] Elapsed time..... 0.118375 secs
[.] Rate..... 37.845829 Mbps
[.] Speedup..... 17.140739
  
```

**Figure 4.10. Results of load testing of the GSM picosota BS**

After connecting the LimeSDR and launching the osmocore software package, the spectrum analyzer confirms the operation of the BS at the GSM frequency of 930 MHz.

The IMSI of the subscriber terminals entered into the HLR (Home Location Register), after which the terminals authorized to connect to the created picosota and assigned phone numbers (see Fig. 4.1). After that, a test call made to test the correct operation of the GSM picosota.

**4.5 Laboratory work on the study of the LTE communication system**

To launch the LTE BS, the Open Air Interface 5G (oai) project chosen, created by the EURECOM Communication Systems Research Center, which aimed to develop an alternative OpenSource ecosystem of 3GPP cellular network protocols [8].



The oai project has certain hardware requirements: The PC should be based on Intel architecture; the processor should be of the Intel Core i5/i7 family of the second generation and above. These requirements are justified by the intensive use of entire SSE (Streaming SIMD Extensions) instructions.

Before starting to build the BS, it is necessary to update all operating system packages to the current state by entering the apt-get update console command. To do this, open terminal and perform all actions on behalf of the superuser by entering the sudo -s command.

Install the oai software.

To do this, you need to download the source code from the git distributed version control system using the git clone command and specifying the address <https://gitlab.eurecom.fr/oai/openairinterface5g.git>.

Open the openairinterface5g directory with the cd openairinterface5g command. Set the correct variables with the source oaienv command.

Oai comes with an automated installation script located in the cmake\_targets directory, which can be accessed with the cd cmake\_targets command.

Next, run the oai installation script with the driver for LimeSDR with the command /build\_oai -i --install-optional-packages -w LMSSDR (Figure 4.11).

```
-- checking for one of the modules 'nettle'
-- checking for one of the modules 'xpm'
-- option_HW_lib=-LLMS_SDR -LLMS7002M -LSi5351C -rdynamic -ldl
-- HW_SOURCE=/home/ubuntu/openairinterface5g/targets/ARCH/LMSSDR/USERSPACE/LIB/L
ms_lib.cpp
-- option_TP_lib=
-- TRANSPORT_SOURCE=
-- Configuring done
-- Generating done
-- Build files have been written to: /home/ubuntu/openairinterface5g/cmake_target
ts/lte_build_oai/build
Compiling lte-softmodem
Log file for compilation has been written to: /home/ubuntu/openairinterface5g/cm
ake_targets/log/lte-softmodem.Rel10.txt
lte-softmodem compiled
Log file for compilation has been written to: /home/ubuntu/openairinterface5g/cm
ake_targets/log/oai_lmssdrdevif.Rel10.txt
oai_lmssdrdevif compiled
liboai_device.so is linked to LMSSDR device library
10. Bypassing the Tests ...
root@ubuntu:~/openairinterface5g/cmake_targets#
```

**Fig. 4.11 – Installing oai with the driver for LimeSDR**

Before starting the BS, you need to configure it. In the first window of the terminal, enter LTE\_fdd\_enodeb. In another terminal window, connect to the management interface with the telnet command 127.0.0.1 30000. The help command will display all possible settings and their current values.

After the configuration process is complete, save the settings with the commands write use\_cfg\_file 1 and write use\_user\_file 1.

Start the LTE BS with the LTE\_fdd\_enodeb command. To control the BS parameters, open the second terminal window and connect to the management console with the command telnet 127.0.0.1 30000. To monitor the BS operation, open the third window of the terminal and connect to the debugging interface with the command telnet 127.0.0.1. 4.22.

```
ubuntu@ubuntu:~$ LTE_fdd_enodeb
linux; GNU C++ version 4.8.4; Boost_105400; UHD_003.009.004-0-g2b5a88bb

*** LTE FDD ENB ***
Please connect to control port 30000
[ ]

ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ telnet 127.0.0.1 30000
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
*** LTE FDD ENB ***
Type help to see a list of commands
[ ]

ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ telnet 127.0.0.1 30001
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
01/16/2017 20:04:28.837876 info iface LTE_fdd_enb_interface.cc 787 *** LTE FDD E
NB DEBUG INTERFACE ***
```

**Figure 4.12 - Starting, managing and monitoring LTE BS**

Therefore, after connecting the LimeSDR and running the oai software, the spectrum analyzer confirms the operation of the BS at the LTE standard frequency of 2.69 GHz with a channel width of 10 MHz, which corresponds to Band 7 [9].



## V. Conclusions

Over the past few years, the use of SDR technology has provided a major leap forward in the development of mobile communication networks, attracting interest from manufacturers and customers of telecommunications systems and reducing the cost of modernizing communication facilities. At the same time, it contributed to the rapid development of high-speed DAC/ADC and wide-band (up to several gigahertz) transceivers with reprogrammable characteristics.

Today, SDR technology is used in such areas: Mesh, Smart-Grid, Smart Antenna, MIMO, RFID, WRAN, DVB, DRM, LTE.

Of the main radio platforms considered, LimeSDR is the most advantageous platform for solving educational problems.

Using a laboratory sample, software on Ubuntu Linux developed for LimeSDR operation. The working models of receivers and transmitters of the main standards of digital terrestrial television DVB-T2, satellite broadcasting DVB-S2 and base stations of GSM and LTE standards implemented.

The work of the developed models with LimeSDR tested in the educational process during a series of laboratory works on the study of new generation communication systems.

## VI. References

- [1] DmitrySpb79. (14 May 2019 г.). Software Defined Radio - How does it work? Part 1. Retrieved from HBR: <https://habr.com/ru/post/451674/>
- 1.1. Retrieved from <https://www.xilinx.com/products/silicon-devices/soc.html>
- [2] *List of software-defined radios*. (13 October 2022 г.). Retrieved from Wikipedia: [https://en.wikipedia.org/wiki/List\\_of\\_software-defined\\_radios](https://en.wikipedia.org/wiki/List_of_software-defined_radios).
- [3] 4 *Third-order intercept point*. (13 July 2022 г.). Retrieved from Wikipedia: [https://en.wikipedia.org/wiki/Third-order\\_intercept\\_point](https://en.wikipedia.org/wiki/Third-order_intercept_point)
- [4] Rud, A. (3 May 2022 г.). Ubuntu - description of the OS: current versions, pros and cons. Retrieved from Hyperhost: <https://hyperhost.ua/info/uk/ubuntu-opis-os-aktualni-versii-plyusi-minusi>
- [5] *Osmocom*. (2022, October 15). Retrieved from Wikipedia: <https://en.wikipedia.org/wiki/Osmocom>
- [6] *Specifications & Technologies*. (2022). Retrieved from 3gpp: <https://www.3gpp.org/specifications-technologies>
- [7] EURECOM. (2022). *openairinterface5G*. Retrieved from gitlab: <https://gitlab.eurecom.fr/oai/openairinterface5g>
- [8] *The State of LTE*. (2018). Retrieved from OPENSIGNAL: [The State of LTE \(February 2018\) Report Report | Opensignal](https://www.opensignal.com/insights/the-state-of-lte-february-2018-report)

Отримана в редакції 18.04.2023. Прийнята до друку 16.05.2023. Received 18 April 2023. Approved 16 May 2023. Available in Internet 19 June 2023.

УДК 004.852-029

# CHARACTERISTICS OF THE TECHNICAL TASK FOR THE DEVELOPMENT OF A CHATBOT

Veronika Neliubii<sup>1</sup>, Halyna Lukash<sup>2</sup>

<sup>1,2</sup>Vasył Stus Donetsk National University, Vinnitsa (Ukraine)

ORCID: <sup>1</sup> <https://orcid.org/0009-0003-9447-7407> <sup>2</sup> <https://orcid.org/0000-0002-9777-7718>

E-mail: <sup>1</sup> [neliubii.v@donnu.edu.ua](mailto:neliubii.v@donnu.edu.ua) <sup>2</sup> [g.lukash@donnu.edu.ua](mailto:g.lukash@donnu.edu.ua)

Copyright © 2021 by author and the journal “Automation of technological and business – processes”.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0>



DOI: 10.15673/atbp.v15i2.2525

**Abstract.** The article describes the general principles of the technical task for the development of chatbots of a modern educational institution. It is noted that chatbots are still a new tool that is constantly developing and improving. Almost any company can integrate a chatbot into its work, because today it has become a universal assistant for various tasks. Depending on the actual needs, it can be added to the site, application or messenger. The process of developing chatbots aimed at ensuring the effectiveness of documentation and communication activities of an educational institution has been studied. Technical tasks for chatbot development are described. The purpose of the study is to consider IT tools and chatbots to activate the process of increasing the efficiency of the documentation and communication activities of the