Development of a MIDI synthesizer for test signals to a wireless acoustic sensor network

Snezhana Pleshkova, Kostadin Panchev, Alexander Bekyarski
Faculty of Telecommunication
Technical University
Sofia, Bulgaria
snegpl@tu-sofia.bg,
k.v.panchev@abv.bg

***Abstract*—In practice exist many devices that generate MIDI data. Most widely used are: MIDI musical instruments (synthesizers) and MIDI controllers. Synthesizers are available in many appearances. The feature that apart them from MIDI controllers, is their ability to make both audio and MIDI data. These synthesizers can be integrated into scientific experiment to generate acoustic signals to test the functionality of a wireless acoustic sensor network.**

Keywords—MIDI, synthesizer, Wireless Acoustic Sensor Network, acoustic signals

I. Introduction

The standard MIDI synthesizer is an electronic keyboard that resembles a small piano. When a key on the keyboard is pressed, a tone is heard. The new models are produced with many varieties of sounds and effects. When a MIDI synthesizer key is pressed, in addition to creating a beep, MIDI data is also created. If the keyboard is connected to a computer, this MIDI data can be recorded in a serial program. Also, this keyboard can be connected to a drum machine or sampler, and it can be controlled via MIDI commands [1].

There are many typical instruments converted into MIDI synthesizers- guitar synthesizers that have touch pads instead of thresholds, wind instrument synthesizers that look like plastic clarinets, but plays tones like a saxophone.

The main difference between MIDI controller and MIDI synthesizer is that the first does not emit a standalone sound. It only creates pure MIDI data, which is processed and played by a computer or by a MIDI synthesizer with audio capability.

It is interesting that exist special pedals and sophisticated control consoles with dozens of knobs and faders for professional mixing quality. There are even specific MIDI console devices for controlling the lighting on the stage of a show or a concert [1].

The proposed in this article MIDI synthesizer will be applied in the future research works to generate sounds for testing an acoustic sensor network and to help getting an acoustic picture of a closed room.

II. Devloping Functional Scheme of a Proposed Configuration of MIDI Synthesizer

Figure 1 shows the block scheme of a proposed configuration of MIDI synthesizer.

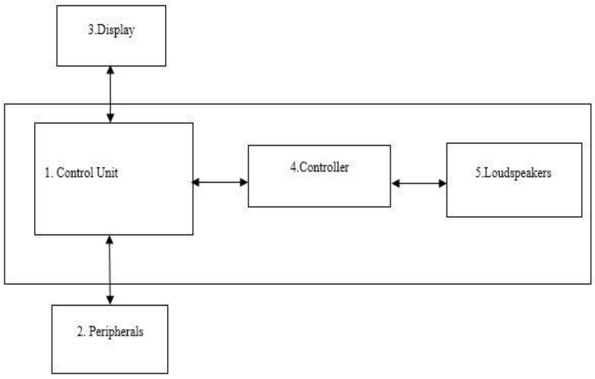

Fig. 1 Block scheme of a proposed configuration of MIDI synthesizer.

The Control Unit is a single-board computer for control of the controller and processing of the data received from it.

Raspberry Pi 4 is the control unit in this case. It is better than the other controllers, beacause it is faster, and adds Wi-Fi and Bluetooth Low Energy connectivity without the need for additional modules [3]. These innovations allow remote access and management of other modules and boards connected to the Raspberry Pi 4. Supports: Raspbian (Debian), Ubuntu, Windows 10 IoT Core, RISC OS, OSMC, OpenELEC, etc. Block “Peripherals” includes mouse, keyboard, USB hub, etc., used to control and set commands on the single-board computer. Display connects to the single-board computer and serves for information visualization and control (in case the display is touch-sensitive). Single-chip synthesizer is used for writing, reading and processing MIDI files. Supports a large database of sounds. The Loudspeakers are very important- to reproduce the result of the synthesized sound to the acoustic sensor network [2].

In the next paragraph we will review some softwares and audio programmes for Raspberry Pi 4 and suitable for Linux OS. Than we will choose one of them which is the most useful for our future work.

III. Software for the Implementation of the Developed MIDI Synthesizer Through a Standard Raspberry-Pi Module

In the following sections, we will review several software products that are virtual synthesizers running on the Linux operating system.

A. *Software system "amsynth" for realization of the developed MIDI synthesizer*

Amsynth is a basic virtual analog synthesizer, made for Linux. It is polyphonic (maximum 16 voices). Each voice has two oscillators, a 12 or 24 dB octave resonant filter and dual ADSR generators. All of them can be modulated with a low frequency oscillator (LFO). The synthesizer also has distortion and reverberation effects [4]. Amsynth is a good starting point for research, as it is easy to set up and use software. It can work independently (JACK, ALSA or OSS) or as a plug-in (DSSI, LV2, VST). When “amsynth” starts, it automatically searches for the “JACK” audio server (a standard audio server for working with professional Linux audio). If it cannot find a “JACK” server, it switches to “ALSA” (Advanced Linux Sound Architecture) audio.

To install "amsynth" on "RaspberryPi" it is necessary to enter the following command in "Linux":

- sudo apt-get install amsynth- this command "extracts" (downloads) and installs "amsynth" and the libraries it needs.

To test how the program works with ALSA, two command windows must be opened. In the first, the "amsynth" command is executed to start the software. The program control panel is displayed on the screen [4] (Fig. 2).

Fig. 2 The control panel of “Amsynth”.

B. *“FluidSynth” software system for realization of the developed MIDI synthesizer*

“FluidSynth” itself does not have a graphical user interface, but it works with several applications, embedded systems and is even used in some mobile applications. It is a command line program that accepts MIDI input from the keyboard of a MIDI controller or from the software MIDI "sequencer".

It needs a file that contains an instrument samples. It plays the incoming notes using the selected SoundFont instruments. FluidSynth supports sixteen MIDI channels (default) [5].

There are three main modes of use:

- interactive command mode;

- "single line" mode;

- server mode.

If you enter the command "fluidsynth" in the command line, the program starts in its interactive mode, ie. accepts and interprets its own commands. This option makes it easy to study the configuration of FluidSynth and because of that it is possible to write complex combinations.

Single-line mode launches the software without going into interactive mode. This mode unlocks many of the command line options (Fig. 3).

- `-C, --chorus`: Turn chorus ON or OFF
- `-R, --reverb`: Turn reverb ON or OFF
- `-K, --midi-channels`: Set the number of MIDI channels
- `-j, --connect-jack-outputs`: Connect JACK outputs
- `-F, --fast-render`: Render MIDI to an audio file
- `-O, --audio-file-format`: Audio file format for fast rendering
- `-r, --sample-rate`: Set the sample rate
- `-T, --audio-file-type`: Audio file type for fast rendering
- `-i, --no-shell`: Don’t run in interative mode
- `-S, --server`: Start FluidSynth as a server process

Fig. 3 Window with commands of “FluidSynth”

“FluidSynth” searches a command line options, SoundFont file and a list of MIDI files. By entering the command from fig. 4, a MIDI file is played through the audio port- 3.5 mm stereo jack of Raspberry Pi [5].

```
fluidsynth -a alsa -n -i /usr/share/sounds/sf2/FluidR3_GM.sf2 EvilWays.mid
```

Fig. 4 A command line in “FluidSynth”.

The "-a" option selects "JACK" and the "-j" option tells it to connect the "FluidSynth" audio output to the system's audio output.

The "-t" option specifies the file format, the "-f" option specifies the name of the output file. The visualization process takes some time.

Server mode is required when FluidSynth needs to be used as a standalone server [5].

C. *“Qsynth” software system for realization of the developed MIDI synthesizer*

It is based on a Qt framework and a set of tools for designing and implementing a user interface (Fig. 5). It is used as a software synthesizer with MIDI controller or "sequencer" [5].

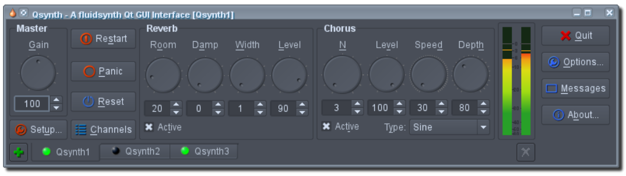

Fig. 5 User interface of “Qsynth”

Qsynth browses for the JACK server and connects audio to it. The panel controls the main gain and the effects of

Identify applicable funding agency here. If none, delete this text box.

reverberation and chorus. There are options to view / edit the settings, stop "stuck" notes, restart "FluidSynth", reset the settings. [5].

To connect to a MIDI controller or MIDI keyboard, select the "Connect" button on the control panel of the software "QjackCtl" (Fig. 6).

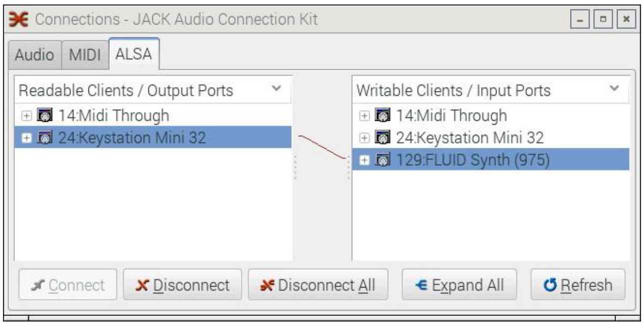

Fig. 6 Control panel of the software.

"FluidSynth" appears as a destination in the right column instead of "Qsynth" because "Qsynth" is a graphical interface for "FluidSynth" (software synthesizer) running in the background. The MIDI controller must communicate with the software synthesizer.

Then select the "Setup" button on the "Qsynth" control panel. The setup window appears, which has four tabs: "MIDI", "Audio", "Soundfonts" and "Settings" (Fig. 7).

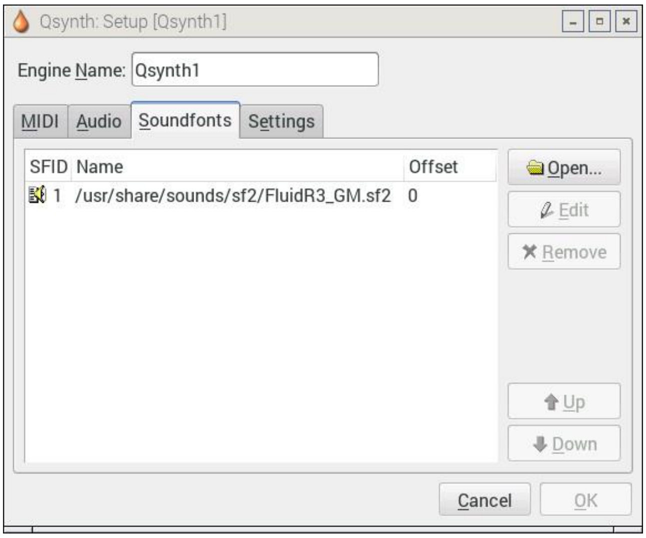

Fig. 7 Control panel of the software

The "Soundfonts" tab displays the audio loaded in the synthesizer.

If "Qsynth" is started with "General MIDI SoundFont" and notes on MIDI channel 1 are played, a piano sound will be heard. By pressing the "Channels" button on the control panel, a menu opens, from where the channel is edited and the sounds are changed (Fig. 8).

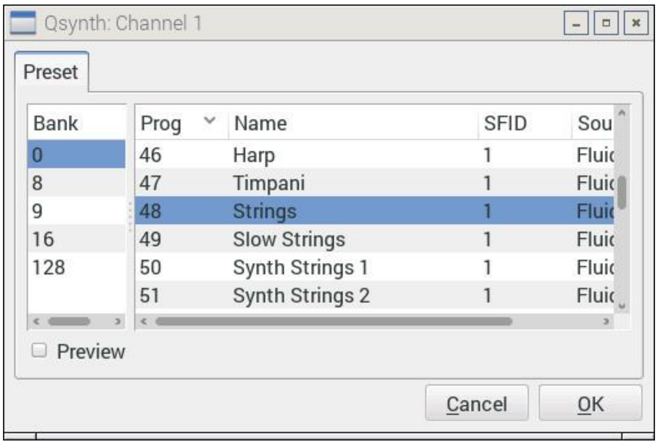

Fig. 8 Channel menu of the software

Figure 8 shows a list of SoundFont voices and sounds. The votes are ordered and picked in a conventional way. The voice does not change until the OK button is pressed to confirm the change. To review and test votes, the "Preview" box must be checked. When the preview function is activated, Qsynth changes the voice, allowing a movement away from the controller to hear the voice before it is changed [5].

IV. VERIFICATION OF THE DEVELOPED SCHEME AND SOFTWARE

Figure 9 shows the terminal window in "Linux" and the command which starts the installation of the software "amsynth".

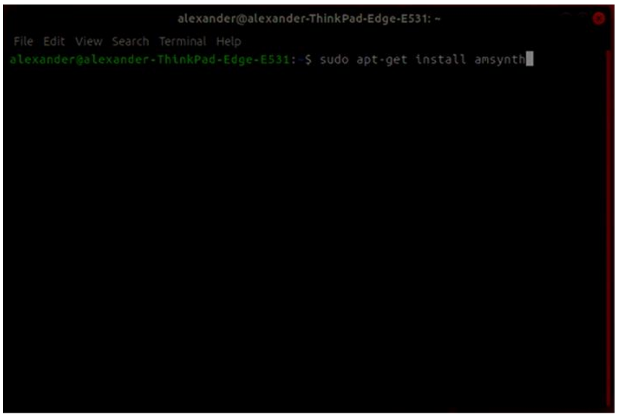

Fig. 9. Terminal window in "Linux"

The next two figures- fig. 10 and 11 present the stages of downloading and installing the software system.

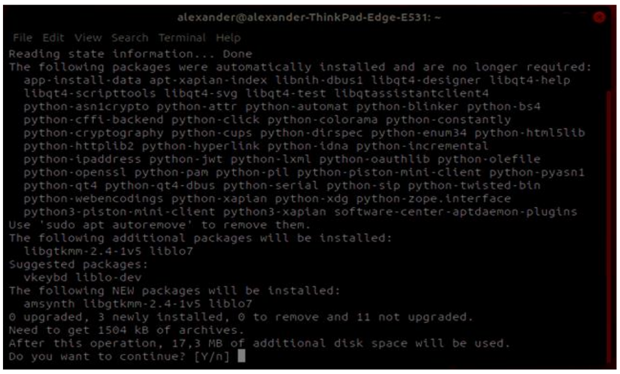

Fig. 10.

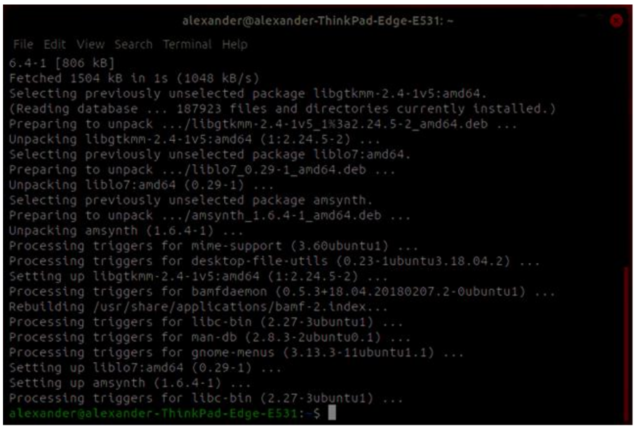

Fig. 11.

After the program is successfully installed, by entering the command "amsynth" (Fig. 12), we start it and the control panel is displayed on the screen (Fig. 13).

Fig. 12

Fig. 13

In order to access and manage the "JACK" audio server, the "JACK Audio Connection Kit" software interface must also be installed.

After installation, open the "JACK Audio Connection Kit". Its interface resembles an audio player, with several specific additional buttons (Fig. 14).

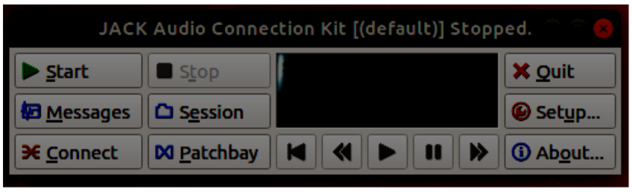

Fig. 14

From the "Setup" button you enter the program settings, from where values of various parameters, driver selection and external MIDI source are set (Fig. 15).

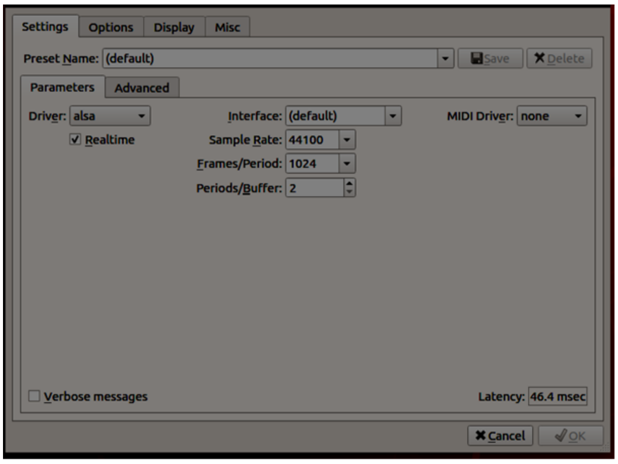

Fig. 15

Once the settings are made and all the necessary parameters are entered, we start playback. There is an easy control interface - standard buttons "Play", "Pause" - and a display that displays the necessary and useful information (Fig. 16).

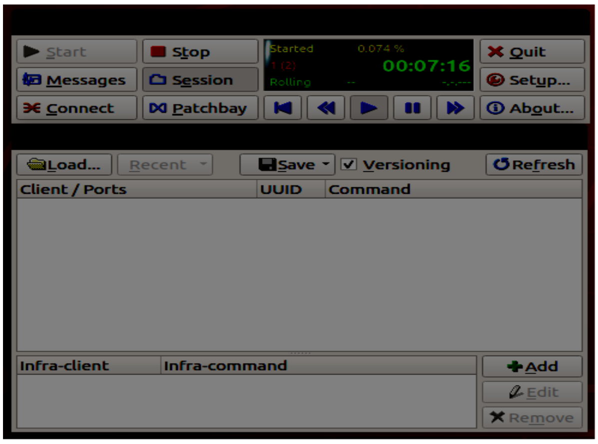

Fig. 16

V. CONCLUSION

After testing the experimental setup (hardware and software), we can conclude that the system works well. The software is easy to install and has a simple management interface. Communication with a hardware synthesizer (MIDI keyboard) or MIDI controller is easy to implement and without delay. Also, the developed MIDI synthesizer is used for generating test signals to a wireless acoustic sensor network, described in the article “Acoustic Sensor Network for Acoustic Measurements in Closed Rooms” [6], and the loading and playback of sounds and melodies is without almost any delay and the sound quality is very good.

ACKNOWLEDGMENT

This work was supported by Technical University – Sofia inner program to research projects under 202PD0030-07 “Development of modules for acoustic sensor networks based on the Internet of Things and artificial intelligence”.

REFERENCES

[1] Сн. Плешкова. Аудио системи. Издателство на ТУ- София, 2016. https://www.midi.org/specifications/category/gm-specifications

[2] Raspberry pi MIDI Synthesizer. https://www.raspberrypi.org/blog/pi-synthesisers/

[3] https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/

[4] http://sandsoftwaresound.net/rpi-soft-synth-get-started/

[5] http://sandsoftwaresound.net/tag/raspberry-pi/

[6] Pleshkova Sn., Panchev K., Bekyarski Al., Acoustic Sensor Network for Acoustic Measurements in Closed Rooms. 2020- INTERNATIONAL CONFERENCE ON HIGH TECHNOLOGY FOR SUSTAINABLE DEVELOPMENT, HiTech 2020, October 2020, Sofia, Bulgaria. IEEE Conference, wait for print