Implementing an Audio Compressor Effect with AGC on a Microcontroller

Szabolcs Bodnár
Obuda University
Alba Regia Faculty
Székesfehérvár, Hungary
szabolcs.bodnar.1120@gmail.com

Bertalan Beszédes
Obuda University
Alba Regia Faculty
Székesfehérvár, Hungary
beszedes.bertalan@uni-obuda.hu
https://orcid.org/0000-0002-9350-1802

Abstract- **This paper presents the design and implementation of an audio compressor effect based on the Automatic Gain Control method using a microcontroller platform. The proposed system continuously monitors the input signal amplitude and dynamically adjusts the gain to maintain a consistent output level, thereby reducing distortion and enhancing audio clarity. The architecture combines analog front-end circuitry with digital signal processing executed on the microcontroller, ensuring low latency and efficient resource utilization. Experimental results demonstrate that the developed prototype effectively suppresses sudden volume fluctuations while preserving the natural quality of the audio signal. The solution provides a cost-effective and compact alternative to traditional compressor units, making it suitable for portable audio devices, embedded systems, and educational applications.**

Keywords - Audio Compressor, Automatic Gain Control, Microcontroller, Embedded Systems, Audio Signal Processing, Real-Time Processing, Low-Cost Implementation

I. INTRODUCTION

The conversion of raw audio signals from electric instruments is an essential step in producing a usable output [1]. With the advent of digital audio effects [2], several methods have been developed to digitally model and manipulate analog audio signals [3]. The compressor effect controls the volume fluctuations of the input audio signal using various parameters [4, 5]. In the proposed implementation, the base signal is duplicated: one copy remains in analog form, while the other is fed into a microcontroller. After evaluation, the microcontroller generates a corresponding control signal that regulates the duplicated analog signal path [6].

II. THEORETICAL BACKGROUND

When using the compressor effect, amplitude regulation is performed. This process is known as Automatic Gain Control (AGC) [7]. A threshold voltage level is set with a potentiometer. If the input signal exceeds this threshold, the Arduino initiates amplitude regulation, adjusting the signal strength back to the threshold level. The ramp-up and ramp-down characteristics of the control can be manually adjusted using the potentiometer corresponding to the attack parameter. The input signal is fed to a potentiometer, and by applying a sine wave as the input, the amplitude can be easily varied, allowing direct observation of the control circuit's behavior [8].

A. Analog circuitry

The unity-gain preamplifier stage (see at Fig. 1.) ensures impedance matching. A low-impedance input signal must be applied to the Arduino Uno analog input to ensure that the sampling capacitor charges correctly and provides accurate sampling. This circuit reduces the high input impedance to approximately 100 Ω, ensuring proper operation of the Arduino sampling capacitor. The DC component of the input sine signal is removed with a coupling capacitor, yielding a pure AC waveform. This signal is then shifted to 2.5 V DC using a KF-25 DC-DC converter IC, aligning the average signal level with the midpoint of the microcontrollers 0–5 V input range – follow at Fig. 2.

The microcontroller used is an Arduino Uno, employed exclusively for control. Regulation is carried out by an MCP41100 digital potentiometer. The control signal from the Arduino is applied to the digital potentiometer, which adjusts the amplitude of the analog signal using the principle of voltage division (follow at Fig. 3.). The output signal of the potentiometer is then fed into the output amplifier circuit, which is based on a bipolar junction transistor. This stage is identical to the Electro-Harmonix LPB-1 booster circuit [9]. Unlike the emitter-follower configuration used for preamplification, the common-emitter amplifier employed here provides high voltage amplification [10]. The amplitude of the final output signal can be adjusted using a potentiometer, which functions as the volume control [11, 12].

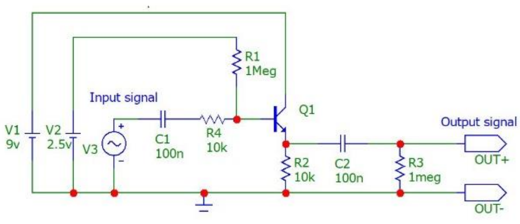

Figure 1: Unity gain schematic

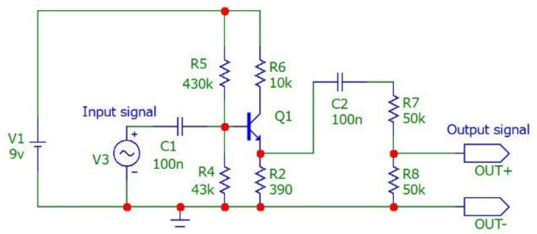

Figure 2: LPB-1 booster schematic

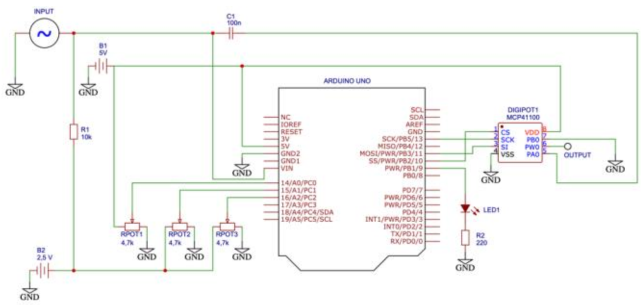

Figure 3: Integration of Arduino Uno into the analog circuit

B. Software realisation

SPI communication is used for data transfer. SPI is a master–slave serial communication protocol that allows for high-speed communication [13]. In this implementation, SPI is employed to control the outputs. Pin 10 is configured as an output; its initial state is set high to disable the SPI command, since the protocol is active low. The SPI bus is then started and the serial port initialized [14].

The control pin for the MCP41100 digital potentiometer is declared. The potentiometer is connected through output pin 10, which delivers the control signal. An indicator LED is also defined, connected to output pin 9. The LED provides visual feedback of the regulation process: its brightness is directly proportional to the control signal. The LED is connected to ground through a 220 Ω resistor, ensuring a current of approximately 9 mA, which is within the safe operating range of the component.

The analog inputs are defined as follows: the base signal is fed to port A0, the threshold potentiometer to port A1, and the attack potentiometer to port A2. An Exponential Moving Average (EMA) algorithm is implemented to determine the ramp-up and ramp-down characteristics of the control [15]. The resulting envelope signal is compared against the threshold to specify the required degree of regulation. Instead of a true moving average, the algorithm computes a decaying envelope that follows the signal peaks [16]. Ramp-up and ramp-down are handled separately, improving robustness against sudden dynamic changes [17].

The variable "emaPeak" is computed as the sum of two weighted terms: the current sample multiplied by a weighting factor, and the previous average multiplied by the complementary factor [18-20]. The weighting factor, adjustable with the A2 potentiometer, controls how strongly the most recent sample influences the peak value. When the algorithm detects a decreasing voltage, its sensitivity is reduced by a precalculated factor, based on empirical data, by applying a scaled version of the attack parameter, thereby smoothing the release phase [20-23].

The main program acquires the signals connected to the analog input pins and scales them into voltage values. The Arduino performs analog-to-digital conversion with 10-bit resolution, corresponding to a scale of 1024 discrete units. A voltage of 0 V is represented as 0, while 5 V corresponds to 1023. To convert the digital values back to voltage, each signal is divided by 1023 and multiplied by the upper limit of the desired scale. In this implementation, the input signal is scaled to 5 V, while the potentiometer values are scaled to 2.5 V. Scaling to 2.5 V is appropriate for the threshold parameter because the input signal is offset to 2.5 V, giving it a theoretical AC range of ±2.5 V. Considering that a highly dynamic guitar signal typically has an amplitude of about 0.5 V, this range provides a suitable reference.

The attack parameter is scaled independently, and its absolute voltage range is less relevant. For consistency in graphical representation, both control potentiometers are scaled to a maximum of 2.5 V, which facilitates direct comparison of their positions. In addition, the attack potentiometer is normalized to a range between 0.01 and 1, enabling its direct use as a weighting factor in the EMA algorithm. The previously mentioned attack/release ratio is also evident in this part of the program.

In the subsequent stage, the amplitude of the input is calculated by removing the nominal 2.5 V offset, effectively centering the waveform around 0 V. This adjusted signal is used to determine the voltage of the envelope curve, which is compared against the threshold to evaluate whether regulation is required. If the EMA voltage exceeds the threshold, a control signal is generated with a magnitude proportional to the difference between the two values. The control output is then converted to 8-bit resolution, yielding a scale of 0-255 for driving the potentiometer and the LED. At this stage, the scaling of the output signals is also performed. The LED is controlled via a PWM signal, while the digital potentiometer operates through a resistance ratio applied between pins 5 and 6, thereby realizing voltage division of the base signal.

Finally, the relevant signals – including the input signal, threshold level, envelope average, and attack level – are displayed graphically. The flowchart of the software can be followed at Fig. 4.

III. PRACTICAL REALISATION

During testing, the threshold was set to 0.5 V, monitored via the Arduino IDE graphical interface. The objective was to observe and document the signal waveforms at various measurement points as the input amplitude varied. The expectation was that once the input exceeded the threshold, both the regulated base signal and the amplified signal from the LPB-1 would remain at a constant level.

Measurements were performed using an XR2206 function generator and a Siglent SDS 1104X-E oscilloscope. Three probes monitored the input, regulated, and amplified signals, each at a resolution of 500 mV per division. In addition to the oscilloscope readings, changes in the LED brightness provided a qualitative indicator of regulation. The results confirmed expectations: as the input amplitude increased beyond the threshold, the regulated and amplified signals stabilized at a fixed level, while their waveform shape remained unchanged. The experimental environment and the measurements of the developed system can be followed in Fig. 5. to Fig. 8.

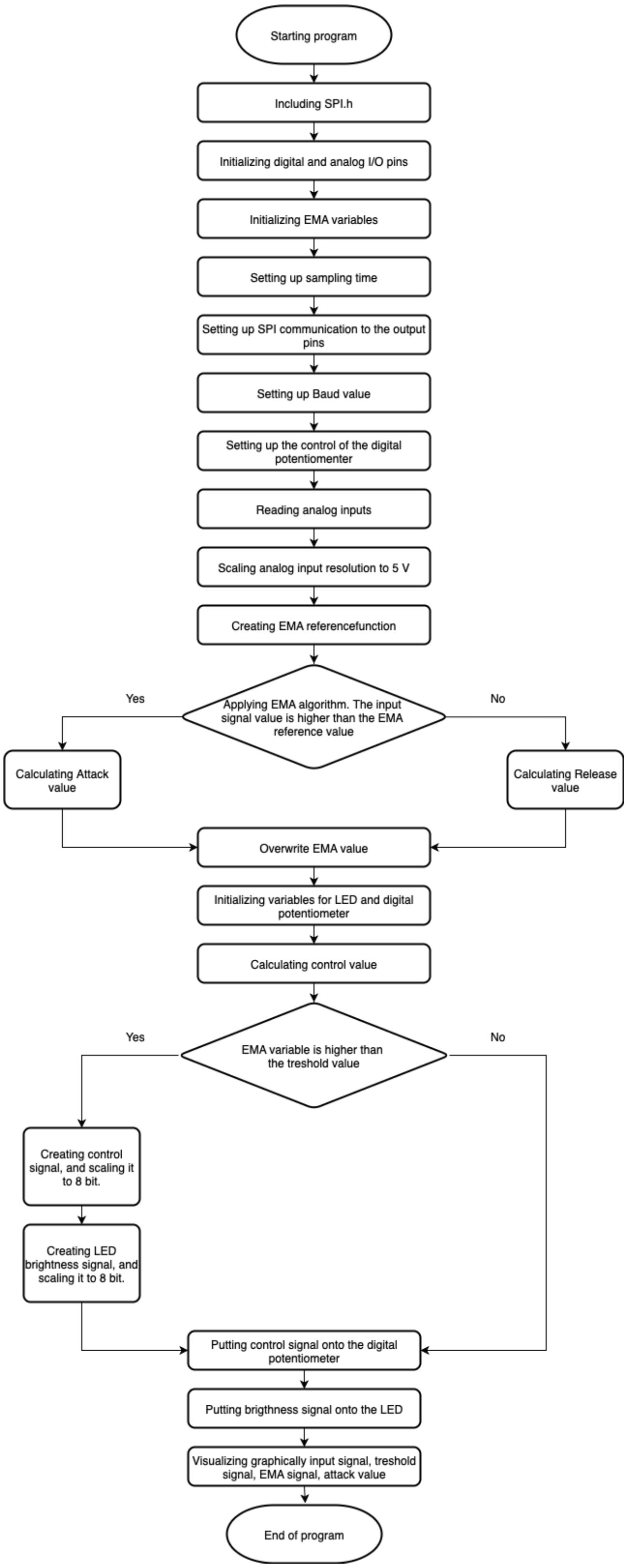

Figure 4: Program flowchart

Figure 5: Input signal level is lower than treshold level

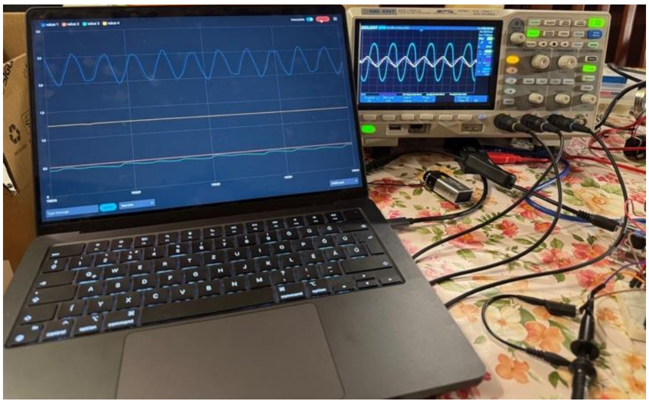

Figure 6: Input signal level equals to treshold level

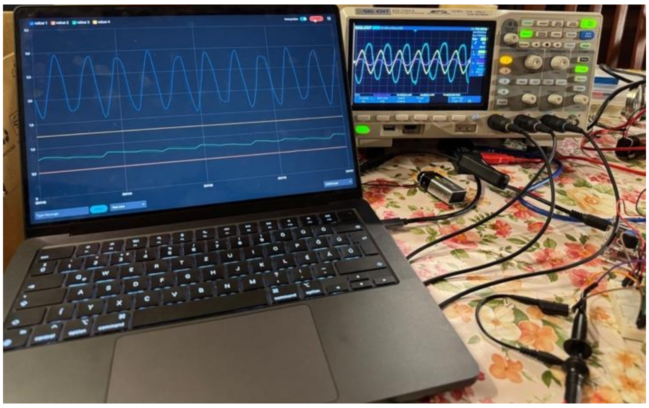

Figure 7: Input signal level is higher than treshold level v1

Figure 8: Input signal level is higher than treshold level v2

IV. EDUCATIONAL RELEVANCE

The implementation of an audio compressor effect with Automatic Gain Control on a microcontroller not only demonstrates a practical engineering solution but also provides substantial educational value in the training of electrical engineers. Embedded systems and digital signal processing are key areas of modern engineering curricula, and project-based assignments such as the presented system foster both theoretical understanding and practical skills.

By working on such projects, students gain hands-on experience with microcontroller programming, analog and digital circuit design, and real-time signal processing. The task integrates knowledge from multiple subjects, including electronics, systems theory, programming, and measurement techniques, thereby reinforcing an interdisciplinary engineering perspective. Moreover, troubleshooting during hardware implementation and software debugging develops essential problem-solving abilities and critical thinking, which are indispensable for professional engineering practice.

The project also supports the development of soft skills, such as project planning, documentation, and technical communication, since students are required to present results, interpret measurement data, and justify design decisions. These competencies are directly transferable to industrial applications, where embedded systems are widely employed in audio technology, telecommunications, and control engineering.

Consequently, the described implementation serves not only as a low-cost and effective solution for audio processing but also as a valuable educational tool. It enhances the engineering capabilities of students by combining theoretical knowledge with practical experimentation, preparing them for the challenges of modern embedded and signal processing systems.

CONCLUSION

The study demonstrated the successful implementation of an audio compressor effect using the Automatic Gain Control method on a microcontroller platform. The proposed system effectively stabilized output levels, reduced sudden volume fluctuations, and preserved the natural quality of the audio signal. By combining simple analog circuitry with efficient digital signal processing, the solution achieved low latency and reliable performance while maintaining low hardware costs. These results highlight the potential of microcontroller-based designs as practical and compact alternatives to conventional audio compressor units.

ACKNOWLEDGMENT

The authors would like to thank all the faculty staff and member of Obuda University Alba Regia Faculty, that provide help and assistance throughout the project completion.

REFERENCES

[1] Sung Du, Mo (2025). What place do Analog Hardware EQs and Compressors have in modern studio environments?. Institute of Art, Design + Technology. Thesis. https://hdl.handle.net/10779/iadt.28597088.v1

[2] Giannoulis, Dimitrios, Michael Massberg, and Joshua D. Reiss. Digital dynamic range compressor design—A tutorial and analysis. Journal of the Audio Engineering Society 60.6. 2012. pp. 399-408.

[3] R. Giampiccolo, A. Bernardini, and A. Sarti. Virtualization of Guitar Pickups Through Circuit Inversion. IEEE Signal Processing Letters, vol. 30, pp. 458–462, 2023.

[4] M. Black and M. Zeytinoglu. Computationally efficient wavelet packet coding of wide-band stereo audio signals. 1995 International Conference on Acoustics, Speech, and Signal Processing. Detroit, MI, USA, 1995, pp. 3075-3078 vol.5, doi: 10.1109/ICASSP.1995.479495.

[5] N. S. Jayant and E. Y. Chen. Audio compression: Technology and applications. in AT&T Technical Journal, vol. 74, no. 2, pp. 23-34, March-April 1995, doi: 10.1002/j.1538-7305.1995.tb00398.x.

[6] H. Huang, H. Shu and R. Yu. Lossless audio compression in the new IEEE Standard for Advanced Audio Coding, 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 2014, pp. 6934-6938, doi: 10.1109/ICASSP.2014.6854944.

[7] D. E. Garcia, J. Hernandez, and S. Mann. Automatic Gain Control for Enhanced HDR Performance on Audio. IEEE Conference on Signal Processing and Systems, pp. 1–8, 2020.

[8] H. Huang, H. Shu and R. Yu. Lossless audio compression in the new IEEE Standard for Advanced Audio Coding. 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Florence, Italy, 2014, pp. 6934-6938, doi: 10.1109/ICASSP.2014.6854944.

[9] A. Belyaev, I. Tutov, and D. Butuzov. Analysis of Noisy Signal Restoration Quality with Exponential Moving Average Filter. 2016 International Siberian Conference on Control and Communications (SIBCON), pp. 1–6, 2016.

[10] White, Frank, and Timothy Dick. Analog compressor modeling. University of Rochester ECE272 Audio Digital Signal Processing. 2016.

[11] D. Sheng and G. Fazekas. Feature Design Using Audio Decomposition for Intelligent Control of the Dynamic Range Compressor. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Calgary, AB, Canada. 2018. pp. 621-625. doi: 10.1109/ICASSP.2018.8461513.

[12] G. Gyorok and M. Mako. Configuration of EEG input-unit by electric circuit evolution. 2005 IEEE International Conference on Intelligent Engineering Systems, 2005. INES '05., Spain, 2005, pp. 255-258, doi: 10.1109/INES.2005.1555168.

[13] Aizhan Zhaparova, Dimitry Titov, Alexander Y. Baklanov, *Gyorgy* Gyorok. Study of the Effectiveness of Switching-on LED Illumination Devices and the Use of Low Voltage System in Lighting. Acta Polytechnica Hungarica (1785-8860 1785-8860): 12 5 pp 71-80 (2015)

[14] Attila Sáfár, Bertalan Beszédes. Educational Aspects of a Modular Power Management System. In: Orosz, Gábor Tamás (szerk.) AIS 2019 : 14th International Symposium on Applied Informatics and Related Areas organized in the frame of Hungarian Science Festival 2019 by Óbuda University. Székesfehérvár, Magyarország : Óbudai Egyetem (2019) pp. 163-166. , 4 p.

[15] G Vakulya, É Hajnal, P Udvardy, G Simon. In-depth development of a versatile rumen bolus sensor for dairy cattle. Sensors (Basel, Switzerland) 24 (21), 6976

[16] A Varga, T Jancsó, P Udvardy. Typical Errors Accuracy Classes and Currently Expected Accuracy of Inertial Measurement Units. Acta Avionica (1335-9479 1339-9853): XXIII 44 (1), 8-14

[17] Éva Hajnal, Gergely Vakulya, Péter Udvardy. Heart Rate (HR) Detection by Cattle Rumen Bolus. 2023 IEEE 27th International Conference on Intelligent Engineering Systems (INES) pp. 155-160.

[18] G. E. Bartos, S. Ünaldı and N. Yalçin. Impact of Image Augmentation on Deep Learning-Based Classification of Granite Tiles. 2024 9th International Conference on Computer Science and Engineering (UBMK), Antalya, Turkiye, 2024, pp. 796-799, doi: 10.1109/UBMK63289.2024.10773433.

[19] Ediboglu Bartos, Gaye and Özmen Akyol, Serel. Deep Learning for Image Authentication: A Comparative Study on Real and AI-Generated Image Classification. in 18th International Symposium on Applied Informatics and Related Areas (AIS 2023), 2023.

[20] G. Ediboğlu Bartos, Y. Hoscan, A. Kauer, and É. Hajnal. A Multilingual Handwritten Character Dataset: T-H-E Dataset. Acta Polytech. Hungarica J. Appl. Sci., vol. 17, no. 9, 2020.

[21] S. Kimathi, and B. Lantos, "Simultaneous attitude and position tracking using dual quaternion parameterized dynamics," in IEEE 22nd World Symposium on Applied Machine Intelligence and Informatics, Stará Lesná, Slovakia, January 2024.

[22] S. Kimathi, and B. Lantos, "Integrated path tracking, and control of a fixed wing UAV based on dual quaternion parameterized dynamics," in 47th MIPRO ICT and Electronics Convention, Opatija, Croatia, May 2024.

[23] S. Kimathi, and B. Lantos, "Robust adaptive formation control of fixed wing UAVs using disturbance estimators", in IEEE 24th International Symposium on Computational Intelligence and Informatics, Budapest, Hungary November 2024.