International Journal of Electrical and Computer Engineering (IJECE)
Vol. 2, No. 1, February 2012, pp. 112~119
ISSN: 2088-8708

AVR Microcontroller Implementation for Customized Sound Generation

[1]Tanvir Hussain, [2]Md. Mejbaul Haque
[1]Dept. of Electrical and Electronic Engineering, Rajshahi University of Engineering & Technology, Bangladesh
[2]Dept. of Electrical and Electronic Engineerring, Khulna University of Engineering & Technology, Khulna-9203, Bangladesh.

Article Info

Article history:

Received Nov 1st, 2011
Revised Jan 18th, 2012
Accepted Jan 28th, 2012

Keyword:

DAC
Microcontroller
Op-amp
Sound generation
VSM

ABSTRACT

In this paper, we have proposed a technique by which customized sound can be generated using 8 bit AVR microcontroller. Any sound file chosen from computer can be integrated in microcontroller programming which is written in C and compiled by AVR libc compiler. An 8 bit DAC connected to PORT A of AVR microcontroller takes data of sound file as input and gives converted analog sound signal as output. An audio amplifier based on LM386 Op-Amp circuit amplifies the sound signal which is finally connected to 8Ω speaker for audible sound output. This circuit has been simulated using Proteus ISIS 7 Virtual System Modeling (VSM).

Corresponding Author:

Md. Mejbaul Haque
Dept. of Electrical and Electronic Enginerring
Khulna University of Engineering & Technology
Khulna-9203, Bangladesh.
E-mail: mhmukul.eee05@gmail.com

1. INTRODUCTION

Nowadays, voice and music integration in modern home tools such as dishwasher, washing machine, microwave oven, calculator etc. and in industry, human machine interaction is a common thing. In robotics it is also used extensively. Any customized valid sound file which resides on computer can be integrated to AVR GCC program after extracting the sound file data which can be done using binary file editing software such as *Hex Editor Neo* [1]. We used AVR processor based microcontroller ATmega32 [2]. Clock frequency was chosen as 1MHz and WinAVR software has been used for programming which has built in libc AVR compiler [3]. After writing C program we have compiled it which has generated the object code (HEX file). The sound file data has been kept in the Flash memory of the microcontroller and it goes to DAC through PORT A of the microcontroller. From an audio amplifier circuit, output has been taken where an 8Ω speaker is connected. Fig. 1 shows the complete block diagram for the sound generation by AVR.

2. AUDIO FILE CLASSIFICATION

For transmission and storage purpose, different compression method of audio has been devised for many years [4]. Due to different application and need, there are two types of audio coding method and is given below

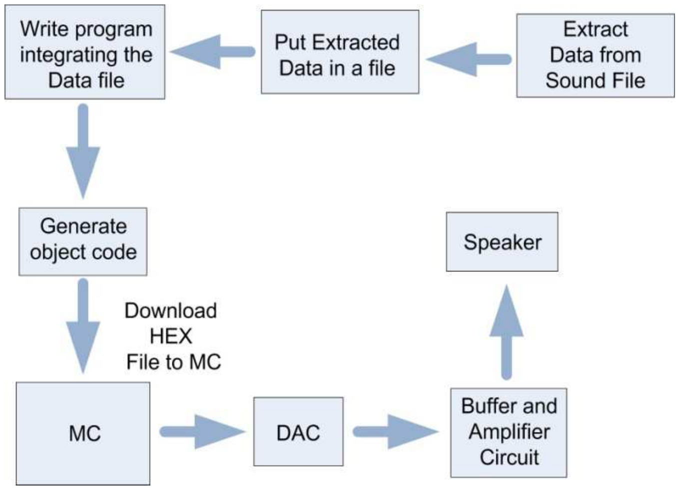

Fig. 1 Complete block diagram of generating sound using microcontroller

2.1 LOSSLESS AUDIO CODING

Lossless audio coding is based on linear prediction followed by entropy coding [4]. Lossless audio coding is divided into two groups as follows

2.1.1 UNCOMPRESSED

There are various files format which applies uncompressed lossless audio format such as WAV, AIFF, AU, RAW etc [5].

2.1.2 COMPRESSED

Lossless compressed file is efficient for storing audio data on disk but it takes more processing time than lossless uncompressed audio file. FLAC, WavPack, ALAC (Apple lossless) etc. file applies this format [5].

2.2 LOSSY AUDIO CODING

Lossy audio coding makes use of a psychoacoustic model of human acoustic perception to quantize and code the audio signal [4]. MP3, Vorbis, Musepack, AAC, ATRAC etc applies lossy audio coding [5].

3. DATA EXTRACTION FROM AUDIO FILE

In our design we have applied WAV file format as in this file format no decoding is needed. To convert any audio file to WAV file format we have used SWITCH Sound File Converter Software [6]. We have used 8 bit DAC and one speaker.That is why, any audio file has to be converted to WAV file containing these specification, uncompressed PCM format, 8-bit Mono and sampling rate is 8 KHz. Here, 8-bit data takes 1 byte space in memory and there are 8000 sample per second. So 1 second sound file takes 8Kbyte space in memory. In this project we have applied 3 second sound as Atmega32 has maximum 32Kbyte Flash memory. Fig. 2 shows the snapshot of Switch sound file converter software. From the menu bar of “Switch sound file converter” software we have to go File>Add File(s) or have to press ctrl+A from keyboard. Selecting the desired file we have to click on Encoder Option button in the left bottom corner of the software. Here a window will pop up where desired changes for the output sound file are selected. Finally pressing the Convert button on right-bottom corner will convert the sound file to our desired sound file format. After converting any sound file to WAV file, we have to open our desired file using Hex Editor Neo. Using this software, we have extracted the data in decimal format and copied it in a text file. Later the extension of the file containing the sound data has been changed to *.h. As this sound has been sampled in 8 kHz 8-bit mono mood, we have sent every sample to the DAC from microcontroller every 125 µs interval as microcontroller is set to run at 1MHz frequency.

Fig. 2 Snapshot of Switch Sound File Converter with wave encoder option

Fig. 3 shows *Hex Editor Neo* software overview. In this software, from menu bar we have to open our desired WAV file from *File> Open > File Open*. After opening the audio file, the binary data of the file will be shown in this window. we can copy our required data from this opened window to a text file. Later the extension of the text file have to be changed from *.txt to *.h.

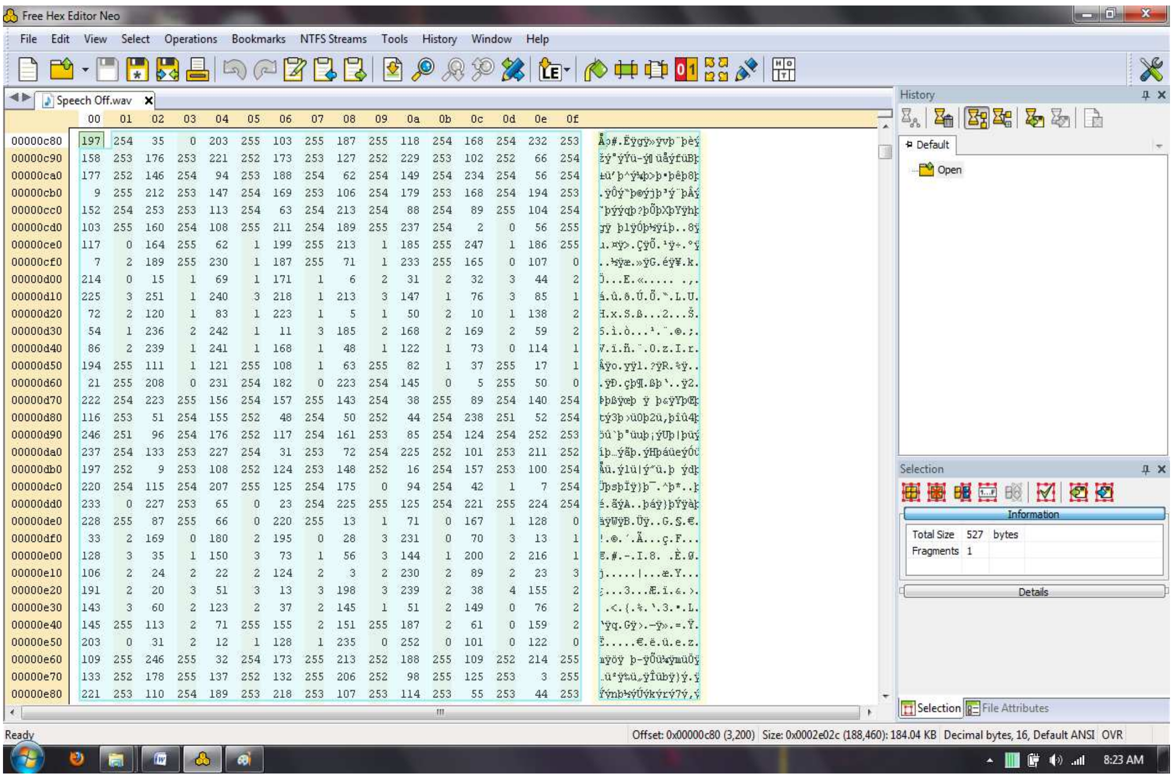

Fig. 3 Hex Editor Neo with wav file opened in window

4. DIGITAL TO ANALOG CONVERTER

In electronics, a digital to analog converter (DAC) converts binary code to analog signal. There are various types and circuits for DAC but in this work we have used R-2R resistor ladder digital to analog converter circuit [7]-[9]. A basic R-2R resistor ladder network is shown in Fig. 4.

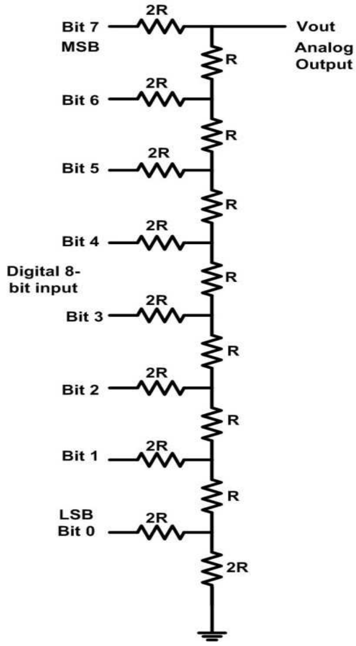

Fig. 4 Resistor ladder digital to analog converter circuit

Bit 7 MSB (most significant bit) to Bit 0 LSB (least significant bit) are driven from microcontroller ports. Ideally, the bits are switched between 0 volts (digital 0) and Vref (digital 1) [10]. The R-2R network causes the digital bits to be weighted in their contribution to the output voltage Vout. In this circuit 8 bits are shown, giving 256 possible outputs. Depending on which bits are set to 1 and which to 0 the output voltage (Vout) will be a stepped value between 0 volts and Vref minus the value of the minimum step, bit 0. The actual value of Vref and 0 voltsn will depend on the technology of the digital logic gates used to drive bit 8-0. In our case we will drive our microcontroller by standard +5 Volt supply and so Vref will be 5V. For a digital value VAL, of a R-2R DAC of N bits of 0 V/V_{ref}, the output voltage Vout is given in [10] as

$$V_{out} = V_{ref} \times \frac{VAL}{2^N} \qquad (1)$$

In this paper we have considered 8 bit DAC, so N=8 and hence 2^N=256 with V_{ref}=5 V. So VAL will vary between 0b0000000 to 0b1111111. So we have from equation (1) the minimum value of VAL will be (00000001 = 1) and maximum value of VAL will be (11111111 = 255)

$$V_{out} = 5 \times \frac{1}{256} = 0.0195 \text{ Votls} \qquad (2)$$

$$V_{out} = 5 \times \frac{255}{2^8} = 4.9804 \text{ Volts} \qquad (3)$$

So the output voltage of the DAC will swing between 0.0195V to 4.9804V.

5. AUDIO AMPLIFIER

The output signal power from DAC is very weak to drive a speaker. To get notable and audible sound, an audioamplifier circuit is used shown in Fig. 5. This simple amplifier shows the LM386 in a high-gain configuration (A = 200). For a maximum gain of only 20, the 10 µF capacitor connected from pin 1 to pin 8 has to be removed. Maximum gains between 20 and 200 may be realized by adding a selected resistor in series with the same 10 µF capacitor. The 10kΩ potentiometer will give the amplifier a variable gain from zero up to the maximum [11].

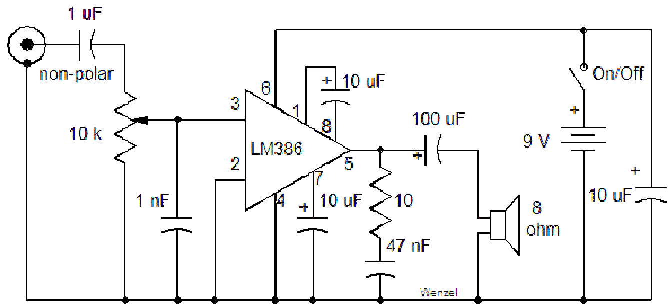

Fig. 5 Schematic circuit diagram for audio amplifier

6. CIRCUIT DIAGRAM

Fig. 6 shows the complete circuit diagram. The resistor ladder R-2R DAC is connected to the microcontroller through PORT A and the “x” marked place shows the output of the DAC. It is then fed to the audio amplifier circuit. This circuit amplifies the power of the input audio signal. Generated sound is produced by an 8Ω speaker. The microcontroller is powered from standard +5V power supply and the audio amplifier needs +9V power supply for smooth operation.

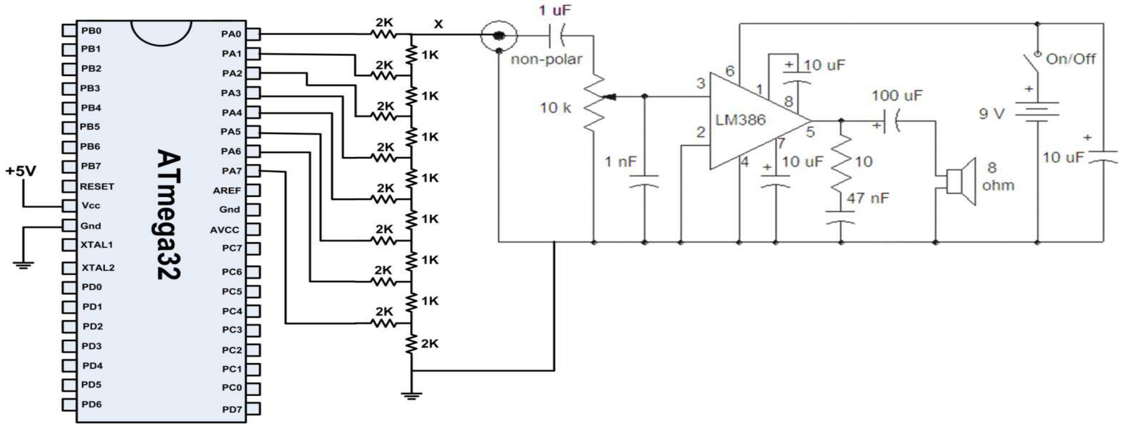

Fig. 6 Complete circuit diagram of the sound generation system using AVR based microcontroller

7. AVR GCC PROGRAM

AVR GCC is the most popular c compiler for programming AVR based microcontroller as it is completely free [12]. It also has some inherent advantages over some traditional c compiler [13]. The program for the microcontroller has been written in C and compiled with AVR GCC. The sound file data has been placed in a file called data.h and included in the program, then a loop reads those data one by one and sends to DAC connected to PORTA after every 125µs, due to sound has been re-sampled at 8 KHz 8 bit WAV mode as described before. Fig. 7 shows the flow chart of the main C program.

8. SIMULATION AND RESULT

DAC circuit connected to ATmega32 can be simulated using Proteus ISIS 7 VSM. Fig. 10 shows the circuit we just described before drawn in Proteus. To check if our circuit really works, first we sent a continuous data from microcontroller to DAC ranging, binary 00000000 to binary data 11111111 with 1 increment. Linearly incrementing the data will also increase the output voltage accordingly as shown in equation (1). If this process loops back than on oscilloscope we will see a triangular wave shown in Fig. 8.

In next step we have loaded our hex file to the microcontroller and the sound output voltage graph shown in Fig. 9 from Proteus ISIS 7 VSM oscilloscope.

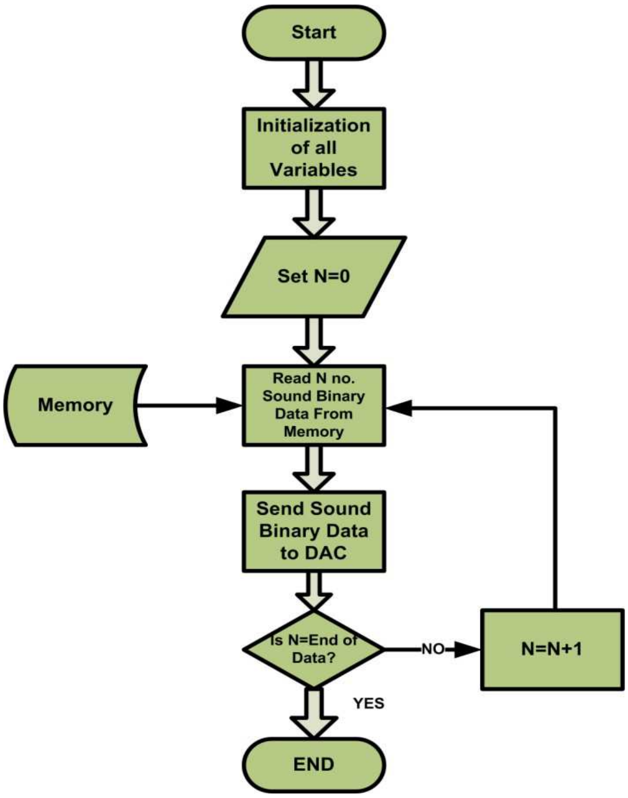

Fig. 7 Flow chart of the main c program

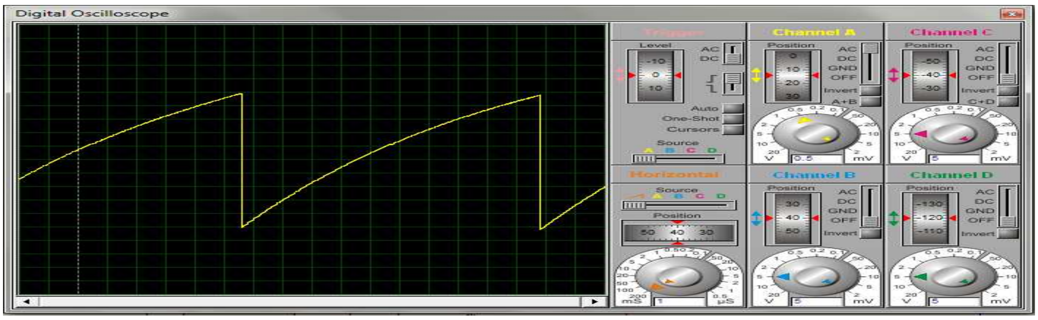

Fig. 8 Triangular wave on proteus oscilloscope when DAC is fed by linear increasingly data

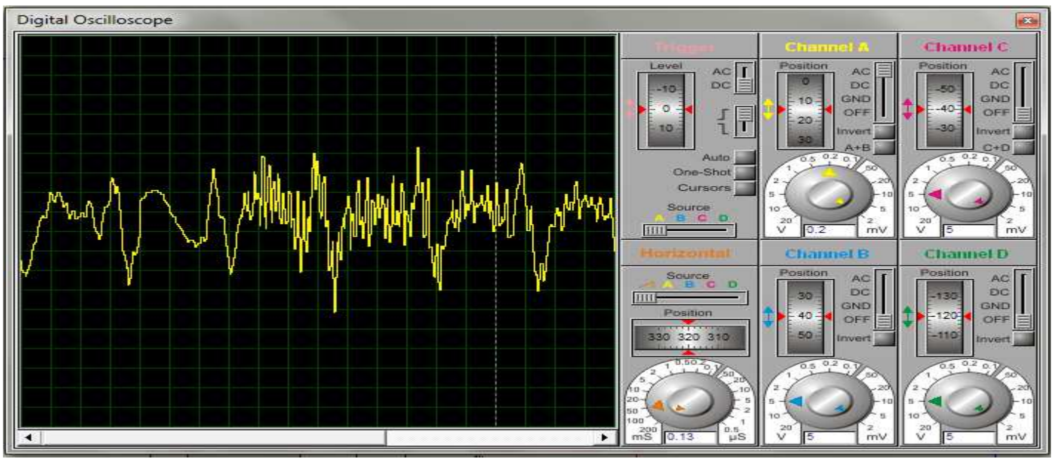

Fig. 9 Sound signal on proteus oscilloscope

In Fig. 10 we have drawn the resistor ladder R-2R DAC with microcontroller and we placed an oscilloscope to view the output wave shape at point V0.

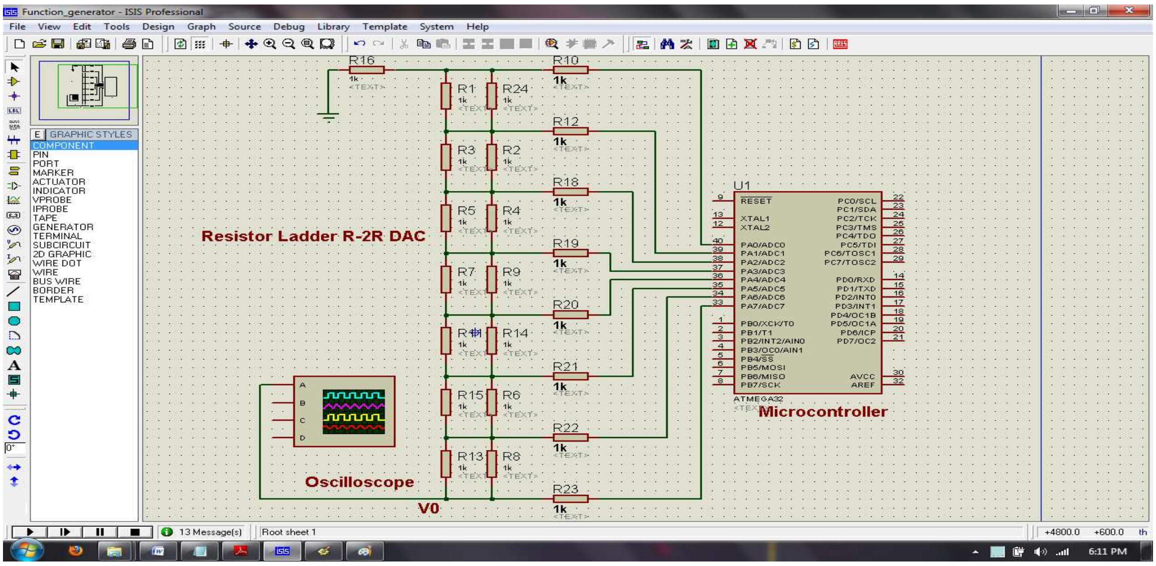

Fig. 10 Simulation of the sound generation system using AVR based microcontroller using Proteus

9. CONCLUSION

In this paper, customized sound generation technique has been shown in detail implementing micrcontroller. Not only ATmega32 microcontroller but also any 8-bit microcontroller can be used for this purpose. PIC microcontroller will also serve the purpose. If it is necessary to store a large sound file, then MMC or SD card can be used. MMC or SD card can communicate with microcontroller with its SPI protocol [14]. In summary the whole process is like this: i. Conversion of our desired sound file format to WAV sound format. ii. Extraction of sound data. iii. Integration of the sound data in main c program. iv. Compiling the c program. v. Programming the microcontroller with the generated HEX file. vi. Connecting the circuit as shown in Fig. 6 vii. Simulation of the sound generation system is shown in Fig. 10 using Proteus viii. Testing of the circuit in real world.

REFERENCES

[1] Hex Editor Neo website. [Online]. Available: http://www.hhd software.com/free-hex-editor.
[2] The Atmel website. [Online]. Available: http://www.atmel.com/ dyn/products/product_card.asp?part_id=2014
[3] WinAVR website. [Online]. Available: http://winavr.sourceforg e.net/
[4] Udo Zolzer, "Digital Audio Signal Processing, "2nd edition, Antony Rowe Ltd., Chippenham, England: Wiley & Sons, 2008
[5] Wikipedia website. [Online]. Available: http://en.wikipedia.org/ wiki/Audio_file_format
[6] The Switch Audio File Converter website. [Online]. Available: http://www.nch.com.au/switch/index.html
[7] Mark Balch, "Complete Digital Design,"*1st edition, New York*, McGraw-Hill, 2003
[8] All About Circuits website. [Online]. Available: http://www.alla boutcircuits.com/vol_4/chpt_13/3.html
[9] Michael Peter Kennedy, "On the Robustness of R-2R Ladder DAC," *IEEE Transactions on Circuits and Systems*, Vol. 47, No. 2, pp. 109-116, Feb. 2000.
[10] Wikipedia website. [Online]. Available: http://en.wikipedia.org/ wiki/Resistor_ladder
[11] Techlib website. [Online]. Available: http://www.techlib.com/e lectronics/audioamps.html
[12] AVR Libc website. [Online]. Available: http://www.nongnu.org/ avr-libc/
[13] T. Fryza, "Basic C Code Implementations for AVR Microcontroller," *14th International Workshop on Signals, Systems and Image Processing*, pp. 434-437, Nov. 2007.
[14] Multimedia Card, *(ADE203-1294C), Hitachi*, Apr. 26, 2002.

BIBLIOGRAPHY OF AUTHORS

Tanvir Hussain was born on Nov 13 1989. He has received his BSc. Engineering from the Dept. of Electrical and Electronic Engineering at Rajshahi University of Engineering &Technology, Rajshahi, Bangladesh.His research interests mainly are Circuits and Systems: theory, design and implementation. He has co-authored 4 publications including two International journal and conference papers.Tanvir Hussain is a member of IEEE Women in Engineering (WIE), IEEE circuits and Systems society, IEEE Power electronics society and IEEE solid-state circuits society. Tanvir Hussain with his team was awarded first in IEEE ethics competetion, IEEE Bangladesh Section, SPAC 2010. He also got merit based technical scholarship from Rajshahi University of Engineering and Technology

Md. Mejbaul Haque was born in Kushtia, Bangladesh on September 27, 1988. He received his B.Sc. Engineering Degree from Rajshahi University of Engineering & Technology (RUET), Bangladesh, in April 19, 2010 in Electrical and Electronic Engineering. He published three conference papers. He is currently working as a Lecturer in Electrical and Electronic Engineering Department, KUET. His research interests are Power system stability, Power quality improvement and Renewable Energy. Md. Mejbaul Haque is the associate member of the IEB of Bangladesh.