International Journal of Research
Available at https://edupediapublications.org/journals

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 10
September 2017

Implementation of Text-To-Speech for Real Time Embedded System Using ARM Processor

M. Naresh & G.V Maha Lakshmi

1M.Tech, DSCE, Sreenidhi Institute of Science & Technology, Telangana, India

2Professor, Dept. ECE, Sreenidhi institute of science and technology, Telangana, India

ABSTRACT:

The real time hardware implementation of Text-To-Speech system has been drawing attention of the research community due to its various real time applications. These include reading aids for the blind, talking aid for the vocally handicapped and training aids and other commercial applications. All these applications demand the real time embedded platform to meet the real time specifications such as speed, power, space requirements etc. In this context the embedded processor ARM has been chosen as hardware platform to implement Text-To-Speech conversion. This conversion needs algorithms to perform various operations like parts of speech tagging, phrase marking, word to phoneme conversion and clustergen synthesis. We are using ARM 7 Processor for the implementation.

Index-terms: *ARM Processor, Text-to-Speech(TTS), Reading aids, Talking aid, Embedded systems.*

I. INTRODUCTION:

An implanted framework is a devoted PC framework intended for maybe a couple particular capacities. This framework is implanted as a piece of an entire gadget framework that incorporates equipment, for example, electrical and mechanical segments. The implanted framework is dissimilar to the

Universally useful PC, which is designed to deal with an extensive variety of preparing undertakings. Since an inserted framework is built to play out specific undertakings just, outline designers may upgrade estimate, cost, control utilization, dependability and execution. Inserted frameworks are normally created on expansive scales and offer functionalities over an assortment of situations and applications.

I.I. TEXT-TO-SPEECH

Content to-discourse (TTS) is the issue of Automatic transformation of content into discourse that Resembles, as nearly as could reasonably be expected, a local speaker of the dialect perusing that content [1]. In TTS Systems a subjective content is changed over into discourse which can be heard boisterously.

TTS change includes a few stages. Extensively they can be named front end and back end steps. In front End steps all the dialect related handling is finished. A portion of the front end related strides are content to tokenization, content standardization, word to telephone Conversion and so on. In back end steps discourse related Signal preparing is finished. A portion of the back end Steps are pitch stamping, span displaying, discourse Synthesis and so forth. The execution subtle elements of the considerable number of Steps for TTS are shrouded in this paper. The Implementation displayed here is a non OS based Implementation. The consequences of execution

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 10
September 2017

II. SYSTEM ARCHITECTURE:
2.1 BLOCK DIAGRAM:

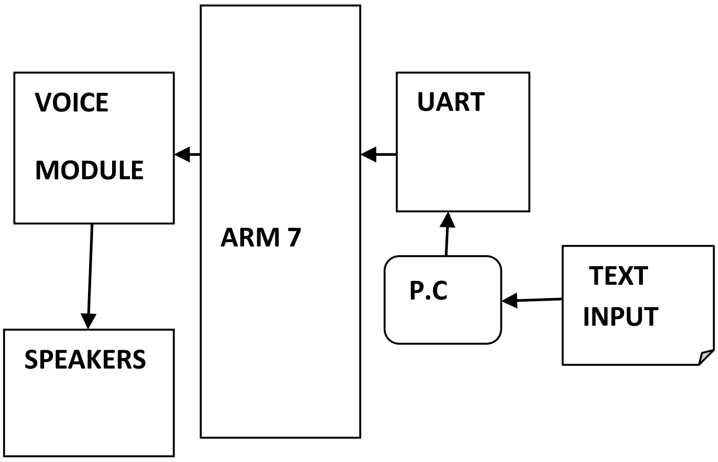


Figure-1: block diagram of project

2.2. EXISTING METHOD

Synthesized speech can be created by concatenating part of recorded speech which is stored in a database. The mainly significant qualities of a speech synthesis system are naturalness and Intelligibility

Naturalness expresses how intimately the output sounds like human speech, whereas intelligibility is the easiness with which the output is understood

[1] " Assigning Phrase Breaks from part-of-speech

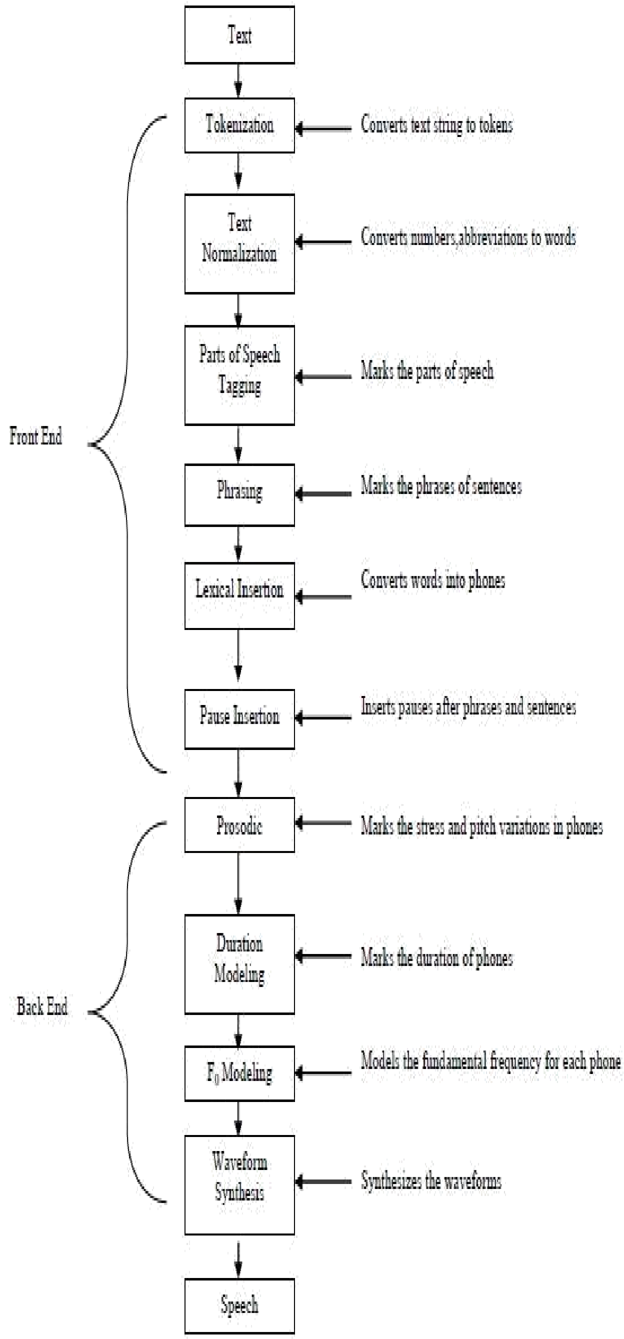
2.3 PROPOSED METHOD

The proposed method is used to overcome the drawback present in existing method. The design of this project involves text to speech. Here whatever we are giving input from any keyboard corresponding output will get in the form of voice means speech. The development board with ARM architecture is selected as the hardware platform. Start-up codes, OS kernel and user's application programs are together stored in a NAND FLASH. Application programs run in 64MB SDRAM, which can also be used as the room of various data and the stack

III.IMPLEMENTATION:

3.1

ARM microcontroller is used for the implementation of the text to speech system. ARM is the most widely used platform for the embedded systems. ARM has been chosen as a platform for the project because of its low power and high speed operations.LPC2148 has been specifically chosen for the implementation of text to speech conversion because it has SDRAM memory, and it supports image sensor interfacing which could be used for embedded system development for visually challenged readers which is our future scope for research.



International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 10
September 2017

Fig. 1 The block diagram showing the step for implementing the Text to Speech system.

The binary file that implements the entire Text to Speech system should be compatible with the ARM instruction set. The generation of the binary file is done with the compilation of the source code with ARM compatible compiler arm-elf-gcc. The arm-elf-gcc is an open source ARM compiler that can be installed with the 'gnu arm' package.

The microcontroller is interfaced with K4S561632E CMOS SDRAM 256Mb die and AD1981 audio codec IC. The interfacing of the microcontroller is done with the help of the microcontroller interfacing codes provided by ATMEL.

The source coding of different steps of the text to speech system is done using the source code of open source FLITE text to speech system. Changes to the source code of FLITE is done so that the microcontroller executes the program without any errors. Significant modifications were change of floating precision to double precision, change of the default of utterance speed (through stretch), change in the output code etc. Without the above modifications the execution was not successful and was not perceptible for the readers.

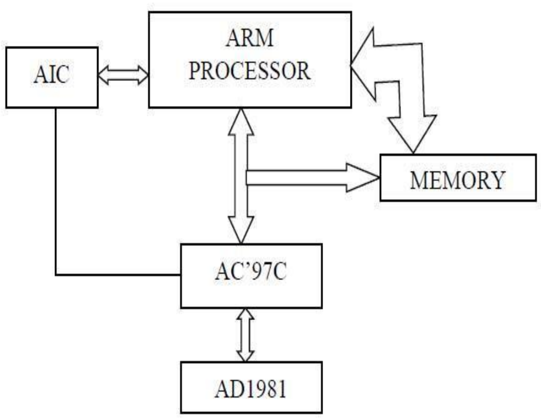


Fig. The block diagram of the system used for the Implementation of Text to Speech

ARM 7 LPC2148

3.2 DESCRIPTION OF PROJECT:

Since decades, real time hardware implementation of Text-To-Speech system has been drawing attention of the research community due to its various real time applications. These include reading aids for the blind, talking aid for the vocally handicapped and training aids and other commercial applications. All these applications demand the real time embedded platform to meet the real time specifications such as speed, power, space requirements etc. In this context the embedded processor ARM (Advanced RISC Machine), has been chosen as hardware platform to implement Text-To-Speech conversion. This conversion needs algorithms to perform various operations like parts of speech tagging, phrase marking, word to phoneme conversion and clustergen synthesis. These algorithms are coded and developed in C using eclipse IDE and finally implemented on commercially available ARM11 microcontroller. Experiments have been performed on ARM microcontroller using test cases. It has been observed that the performance of the ARM based implementation is very close to x86 implementation. In this way we can design text to speech by using

ARM board and **Embedded Linux**.

Table-1: Table of components

S.NO	HARDWARE	COMPONENT SPECIFICATION
1	Microcontroller	LPC 2148
2	LCD	16x2
3	Voice module	APR33A
4	Compiling tool	KEIL IDE

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 10
September 2017

ARM11 microcontroller is used for the implementation of the text to speech system. ARM is the most widely used platform for the embedded systems. ARM has been chosen as a platform for the project because of its low power and high speed operations. ARM11 has been specifically chosen for the implementation of text to speech conversion because it has SDRAM memory and it supports image sensor interfacing which could be used for embedded system development for visually challenged readers which is our future scope for research.

IV. EXPERIMENTAL RESULTS:

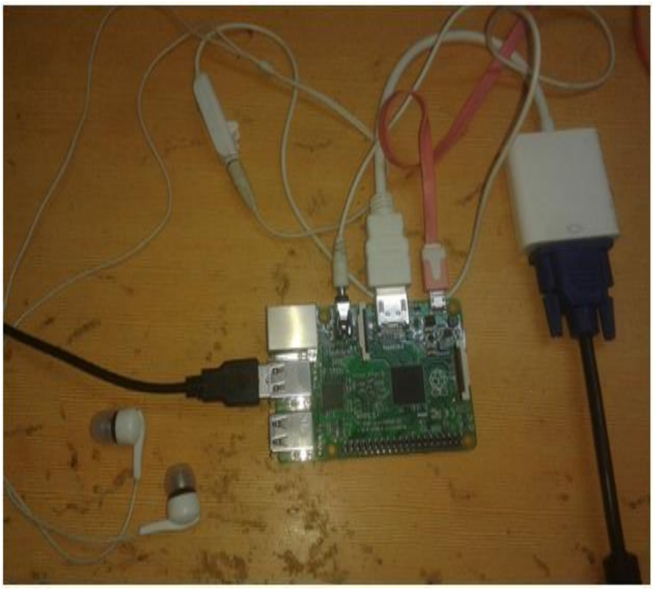


Figure-3: Hardware implementation

Fig. Input by the user

A. Input text"hello" :

The word "hello" is given as input for the ARM based system and x86 based system and the results of the output are as shown in Fig.3

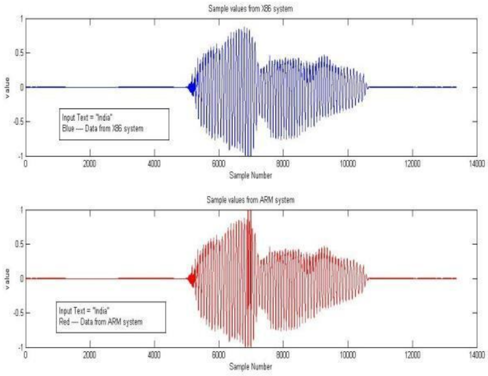


Figure-5: voice Output

V. CONCLUSION

The paper described a way of implementing the text to speech system on ARM microcontroller. Among different speech synthesizing algorithms CLUSTERGEN synthesis is chosen for implementing a real time text to speech system. The results of the implementation are presented. These results shows that the x86 based implementation and ARM based implementation are very close. From the results it can be concluded that a complete low cost real time embedded system can be built with the implementation presented in the paper. This embedded system can be used in many applications like reading aid for the visually disabled persons, talking aid for vocally handicapped etc.

The source coding of the text to speech system can be easily modified for implementing new synthesis algorithms. So the text to speech system developed here can be easily upgraded to the latest synthesis technique for speech. Even though the text to speech system developed here is for English language, it can be easily extendable for new languages also. Hence the system developed here is a robust text to speech system.

VI. REFERENCES

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 10
September 2017

[1] Shin, C. and Sproat, R. "*Issues in text-to-speech conversion for Mandarin*", Computational Linguistics and Chinese Language Processing, pp. 35-82, August 1996.
[2] The FLITE website. [Online]. Available : http://www.speech.cs.cmu.edu/flite/
[3] Taylor, P. and Black, A. " *Assigning Phrase Breaks from part-of-speech Sequences"* Computer Speech and Language 12, 99-117,1998
[4] DeRose, S. "*Grammatical category disambiguation by statistical optimization*". Computational Linguistics, pp. 31-39, 1988
[5] Black, A., Lenzo, K. and Pagel, V. "*Issues in Building General Letter to Sound Rules"* 3rd ESCA Workshop on Speech Synthesis, pp. 77-80, 1998
[6] Pagel, V., Lenzo, K. and Black, A. "*Letter to sound rules for accented lexicon compression"* ICSLP98, pp 2015-2020, 1998.
[7] Black, A. "*Predicting the intonation of discourse segments from examples in dialogue speech*", ATR Workshop on Computational modeling of prosody for spontaneous speech processing, 1995.
[8] Black, A. "*CLUSTERGEN: A Statistical Parametric Synthesizer using Trajectory Modeling",* Interspeech ICSLP, pp. 1763-1765, 2006

[9] Keiichi, K., Takao, K. and Satoshi, I. *"Speech Parameter Generation From HMM Using Dynamics Features" ,* European Conference on Speech Communication and Technology, 1995
[10] Satoshi, I. "*Cepstral Analysis Synthesis on the MEL Frequency Scale*" , IEEE International Conference on ICASSP, pp. 93-96, 1985.