Real Time Implementation of Vocalizer For Texts Using Advanced Risc Machine

Dr. S. Saravana Sundaram[1],
Department of Biomedical Engineering,
Hindusthan College of Engineering and Technology, Coimbatore, Tamil Nadu, India

Dr Karuppasamy P[2]
Department of Electronics and Communication,
Adithya Institute of Technology, Coimbatore, Tamil Nadu, India

Abstract— **The Real time text to speech conversion is difficult to attain the similar speech of the native speaker. This conversion plays a major role in real-time applications. Advanced Risc Machine (ARM) has been chosen as platform for implementing this conversion. Our system aim is to make a simple conversion from text to speech initially. So, the operands that are to be recognized are the alphabets and the numbers. The main objective is to provide any valid text at the input via keyboard and the output will be obtained as a voice representation of the inputted text. The steps involves are text tokenization, parts of speech tagging, prosodic tagging, phrasing, duration modeling, speech synthesis etc. Cause for this project is that reading aids for the blind, talking and training aid for the vocally handicapped and other commercial applications. Vocally handicapped people can type the text from keyboard and it will be processed in ARM microcontroller and synthesized using EMIC voice synthesizer. Finally the output is heard through speaker.**

I. Introduction

The available TTS systems which uses large size computers which imposes the problem of carrying larger weights. The large sized systems cannot be used in all the places where required. Since, the existing algorithms would use various OS related functions, porting them on to standalone system faces many difficulties. The existing system runs on a sophisticated OS which runs on a personal computers that have all its needs. It uses multiple OS functionalities to accomplish its tasks. They are high memory and energy consuming.The proposed system is a standalone system which is highly mobile and easy to handle. Here voice output can be closer to be natural language. The system can be more scalable towards adding new versions. The algorithms involves are text tokenization, parts of speech tagging, prosodic tagging, phrasing, duration modeling, speech synthesis etc. Since, algorithm is light weighted it can run on minimal memory footprint.The proposed Text-to-Speech Converter accepts a string of 16*2 characters of text (alphabets and/or numbers) as input. In this process, we have interfaced the keyboard with ARM controller and defined all the alphabets as well as digits keys on it. The speech processor has an unlimited dictionary and can speak out almost any text provided at the input most of the times. It has an accuracy of above 90%. Since, It is an ARM based hardware coded in embedded C language and synthesized using additional voice synthesizer (EMIC). Rest of the paper has been organized as follows. The section II highlights the algorithms for text-to-speech conversion. An implementation technique of text-to-speech algorithms on ARM based embedded platform has been described in section III. The experimental setup and its results of this real time embedded system cases has been presented in section IV. Section V gives the user guide for our project. Finally, section VI concludes the paper, VII gives its future scope, VIII about acknowledgment and IX about referred papers.

II. Text to Speech

The TTS conversion involves several steps. The steps used can vary from one TTS system to another. Mostly the front end steps remain same. The back end steps vary in synthesis algorithms. The complete steps for implementing the Text-To-Speech system can be shown as in Fig1.

The steps implemented for TTS on ARM can be explained as below.

A. Text Tokenization

In this step, the given input text is broken into tokens. Tokens are same as words. They are text strings that are separated by spaces, quotations etc.

Ex: Input text = "Delhi is the capital of India."
Token[1]=Delhi, Token[2]=is, Token[3]=the,
Token[4]=capital, Token[5]=of, Token[6]=India

B. Parts Of Speech Tagging

In this step, the parts of speech for each token is identified and marked. Parts of speech tagging is important to get the true pronunciation of the words.

Ex.: This good (n) is very good (adj).

In this sentence, the noun word good is pronounced differently than the adjective word good. Here standard HMM based tagging framework is used.

C. Phrasing

In this step, the entire text is divided into phrases and the phrase marking is done at the end of each phrase. Phrase tagging is important in pause insertion and duration modelling.

Ex: If he comes early, I will go to temple.
Phrase [1] = If he comes early Phrase
[2] = I will go to temple

D. Pause Insertion

In this step, a pause is inserted at the end of each phrase mark. The phrase marking information done in the phrasing step is used in this step. A pause symbol {pau} is appended at the end of each phrase mark.

Ex: {PSR Engineering college} phrase will be {PSR engineering college} {pau}

Pause insertion helps the uttering of the text into meaningful sentences.

E. Prosodic Tagging

The linguistic prosody refers to the rhythm, stress and intonation of speech. In this step, accent and tone related information is added.

F. Duration Modeling

In this step, the duration of the sound for each phone is marked. This step is useful for concatenative synthesis.

III. IMPLEMENTATION

ARM microcontroller STM32F102CB is used for the implementation of the text to speech system. ARM is the most widely used platform for the embedded systems. ARM cortex A9 has been chosen as a platform for the project because of its low power and high speed operations. Cortex A9 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.

A. serial communication with keyboard

Our project uses an old PS/2 keyboard. Since most of the modern keyboards are USB keyboards, PS/2 keyboards are quite cheap and easy to come by. The male side (keyboard side) of the PS/2 interface has a standard connector with pin- out.

B. interfacing with LCD

The LCD requires 3 control lines (RS, R/W & EN) & 8 (or 4) data lines. The number on data lines depends on the mode of operation. If operated in 8-bit mode then 8 data lines i.e. total 11 lines are required. And if operated in 4-bit mode then 4 data lines+ 3 control lines i.e. 7 lines are required. If we have sufficient data lines we can go for 8 bit mode & if there is text data which should be displayed on the screen.

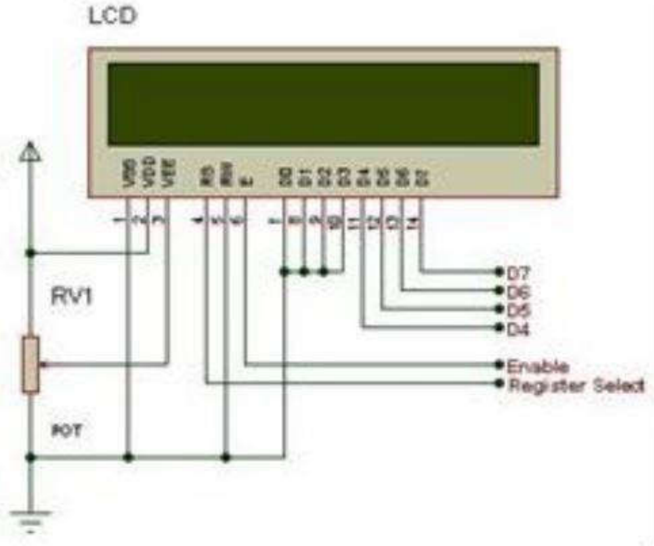

Fig1: *Lcd Interface*

The ARM processor executes the program in memory and is interfaced with EMIC synthesizer for playing the output. Advanced Interrupt Controller (AIC) is the interrupt controller that controls the interrupts during the execution. The binary file that implements the entire text to speech system is loaded into the memory by programming the ARM microcontroller. The ARM processor takes the instructions from the loaded memory and executes them. The program outputs sound signal and this output is played over the speaker through the EMIC module which is synthesizer of natural voice like native speaker shown in Fig 2.

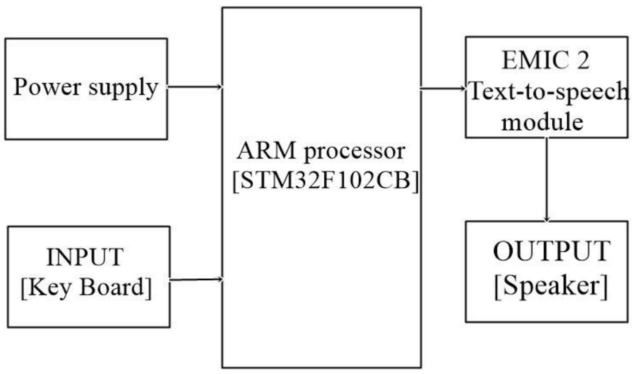

fig *2 The Block Diagram Of System Used For The Implementation Of Text To Speech*

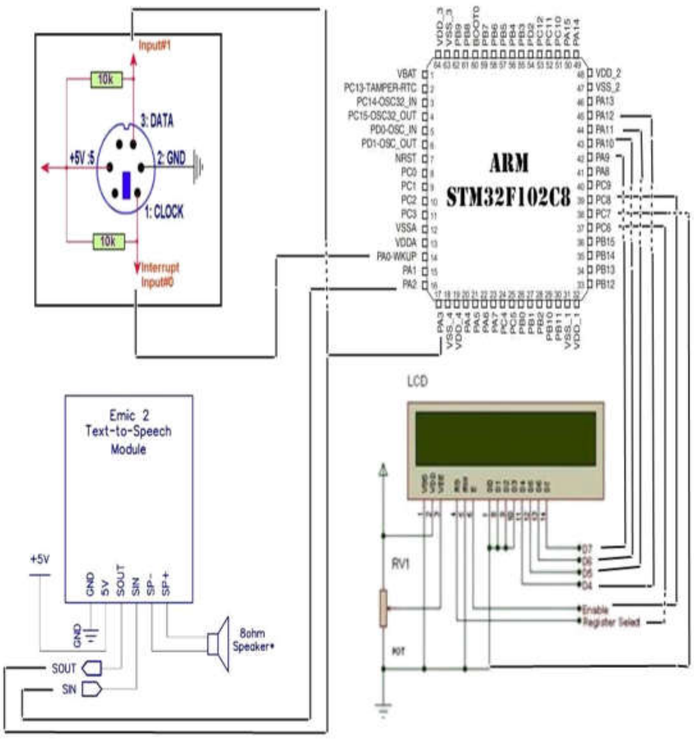

Fig4: *interfacing keyboard, lcd, emic with arm processor.*

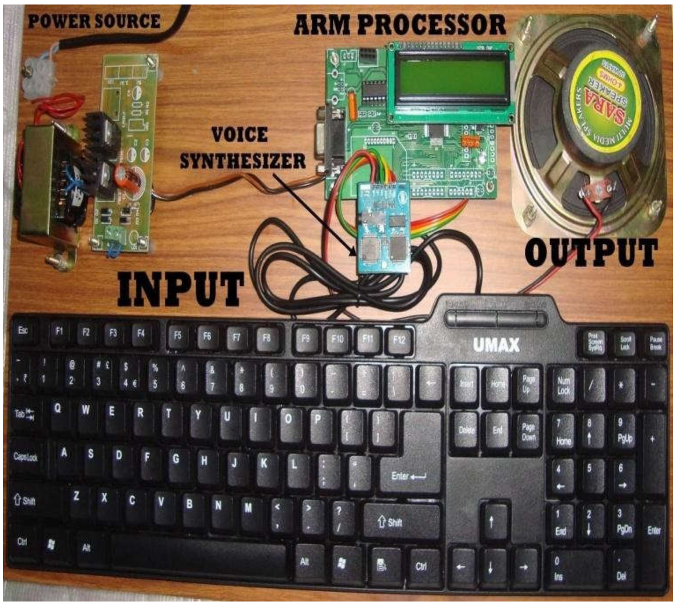

Fig5: *The Complete Architecture Of Our Project*

IV. SIMULATION RESULTS

The ARM implementation output is tested with many test sentences. The output is playedto different readers (students) and the perception of the readers are noted. In most cases a commonly spoken sentences are well understood by many readers and uncommon sentences are also clearly understood by many readers because of EMIC synthesizer's natural voice output. When the power source is switched ON, the system starts with speaking TEXT TO SPEECH KIT which shows the clarity of our project. Then the kit is ready to take the user's inputs via keyboard. The keyboard used is a normal PC interfacing USB keyboard. Hence, it can take any combination of words formation and any numerals symbols can be generated using this. Therefore any of the user' idea can be generated using the keyboard interfaced in our project. It reaches the ARM processor at first. The letters and symbols are converted into their ASCII code using the keyboard and arm interfaces. The code generated are compared with the diphone voices assigned in the coding level of our project. The user's sentences are compared with the diphones andseparated as phrases, idioms, nouns, verbs and so on. After these algorithm is finished, then the voice output is getting ready to spea

zzz

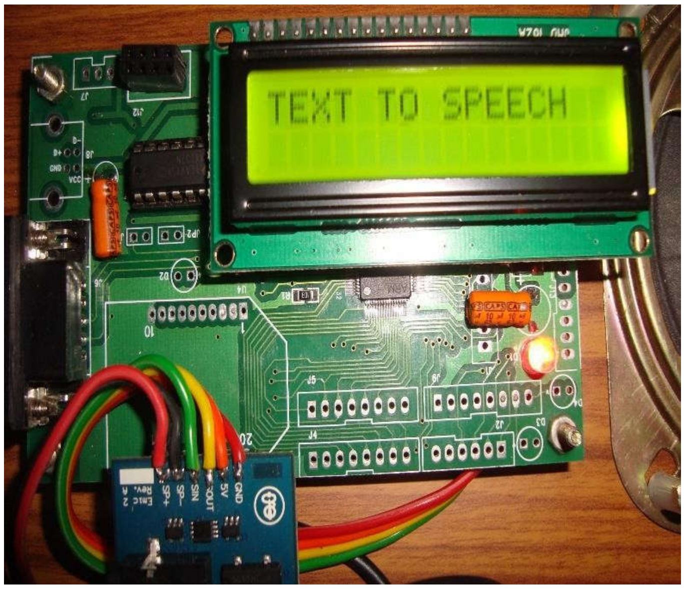

Fig6: *when power source is on*

Fig7: *When The Input Is Given*

Fig8: *state at which the output is spoken*

V. USER GUIDE TO IMPLEMENT

Step 1: Switch on the machine and let the System start.
Step 2: Speech the starting text as "Test to speech kit"
Step 3: Enter the characters want to speech.
Step 4: Confirm that the text doesn't crosses 32 spacesshown in LCD.
Step 5: Press "ENTER".
Step 6: Wait for the system to process it and give a voice output.
Step 7: Done.

VI. CONCLUSION

The paper described a way of implementing the text to speech system on ARM microcontroller. 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. It can be easily extendable for new languages also. Hence the system developed here is a robust text to speech system.

ACKNOWLEDGMENT

We would like to thank our respected Principal Dr. B.G. Vishnu ram of PSR College of Engineering and Head of Department of Electronics and Communication, Mr. C.K. Ramar for supporting us and giving us facilities and immense guidance. We would also like to thank Mrs. K. Ramalakshmi, Asst. prof. ECE for encouraging us in such co-curricular activities.

REFERENCES

[1] Abdul Rawoof, "ARM based implementation of text to speech conversion", IEEE conference on signal processing 2014.

[2] Shin, C. and Sproat, R. "Issues in text-to-speech conversion for Mandarin", Computational Linguistics and Chinese Language Processing, pp. 35-82, August 2012.

[3] The FLITE website. [Online]. Available : http://www.speech.cs.cmu.edu/flite/

[4] Taylor, P. and Black, A. " Assigning Phrase Breaks from part-of-speech Sequences" Computer Speech and Language 12, 99-117,2007

[5] DE Rose, S. "Grammatical category disambiguation by statistical optimization". Computational Linguistics, pp. 31-39, 2010

[6] Black, A., Lenzo, K. and Pagel, V. "Issues in Building General Letter to Sound Rules" 3rd ESCA Workshop on Speech Synthesis, pp. 77-80, 2012

[7] Pagel, V., Lenzo, K. and Black, A. "Letter to sound rules for accented lexicon compression" ICSLP98, pp 2015-2020, 2011.

[8] Black, A. "Predicting the intonation of discourse segments from examples in dialogue speech",ATR Workshop on Computational modeling of prosody for spontaneous speech processing, 2008.

[9] Black, A. "CLUSTERGEN: A Statistical Parametric Synthesizer using Trajectory Modeling", Interspeech ICSLP, pp. 1763-1765, 2013.

[10] Keiichi, K., Takao, K. and Satoshi, I. "Speech Parameter Generation From HMM Using Dynamics Features" , European Conference on Speech Communication and Technology, 2005