IMPLEMENTATION OF TURKISH TEXT TO SPEECH SYNTHESIS WITH RC8660 VOICE SYNTHESIZER

A MASTER'S THESIS

in

Electrical & Electronics Engineering

Atilim University

by

TIMUR KARAMEHMET

JULY 2015

IMPLEMENTATION OF TURKISH TEXT TO SPEECH SYNTHESIS

WITH RC8660 VOICE SYNTHESIZER

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

OF

ATILIM UNIVERSITY

BY

TIMUR KARAMEHMET

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

JULY 2015

Approval of the Graduate School of Natural and Applied Sciences, Atilim University.

Prof. Dr. İbrahim AKMAN

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Elif AYDIN

Head of Department

This is to certify that we have read the thesis "Implementation of Turkish Text To Speech Synthesis with RC8660 Voice Synthesizer" submitted by "Timur KARAMEHMET" and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. İbrahim Baran USLU	Asst. Prof. Dr. Hakan TORA
Co-Supervisor	Supervisor

Examining Committee Members

Asst. Prof. Dr. Fikret ARI

Asst. Prof. Dr. Kutluk Bilge Arıkan

Asst. Prof. Dr. Erol Özçelik

Asst. Prof. Dr. Hakan TORA

Asst. Prof. Dr. İbrahim Baran USLU

Date: 14 July 2015

I declare and guarantee that all data, knowledge and information in this document has been obtained, processed and presented in accordance with academic rules and ethical conduct. Based on these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Timur KARAMEHMET

Signature:

ABSTRACT

IMPLEMENTATION OF TURKISH TEXT TO SPEECH SYNTHESIS WITH RC8660 VOICE SYNTHESIZER

Karamehmet, Timur

M.S., Electrical & Electronics Engineering Department

Supervisor: Asst. Prof. Dr. Hakan Tora

Co-Supervisor: Asst. Prof. Dr. İbrahim Baran USLU

July 2015, 61 pages

This thesis examines the text-to-speech (TTS) synthesis problem and the adaptation of the RC8660 Embedded System to Turkish. RC8660 is a system loaded with English phonemes and in the thesis the aim is to make this card synthesize Turkish. Firstly Turkish phonemes from their corresponding English phonemes are defined. For this purpose IPA: International Phonetic Alphabet was used. Because the syllabic structures of Turkish and English are different, there occurred a need for defining an Exception Dictionary using the *text* and *phoneme* modes of the system. The rules for correct syllabification were added to the dictionary one by one. Stress and intonation rules were also defined for making the synthesized Turkish speech as natural as a native speaker's. Different phonemes were used for text and alphanumeric characters. The effects of the *Speed*, *Expression*, *Pitch*, *Formant Frequency*, *Tone*, *Delay*, and *Articulation* adjustments in the system's software: RC Studio were also tested on the speech. The quality of the synthesis was evaluated by the Mean Opinion Score (MOS) test.

Keywords: Text to Speech, RC8660 voice synthesizer, phoneme, syllabification, MOS test

ÖZ

RC8660 SES SENTEZLEYİCİ İLE

TÜRKÇE METİNDEN KONUŞMA SENTEZLEME

Karamehmet, Timur

Yüksek Lisans, Elektrik Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Yrd. Doç. Dr. Hakan Tora

Ortak Tez Yöneticisi: Yrd. Doç. Dr. İbrahim Baran USLU

Temmuz 2015, 61 sayfa

Bu çalışma, metinden konuşma sentezleme problemini ve RC8660 gömülü sisteminin Türkçe'ye uyarlanması için yapılan çalışmaları incelemektedir. Tezde, İngilizce fonemlerle yüklü gelen RC8660 kartının Türkçe konuşma sentezlemesi hedeflenmiştir. Bu amaçla, öncelikle İngilizce'de bulunan fonemlere karşılık gelen Türkçe fonemler tanımlanmıştır. Bunun için IPA: International Phonetic Alphabet'den yararlanılmıştır. Türkçe ve İngilizce'nin hece yapıları farklı olduğu için, kartın sahip olduğu *metin* ve *fonem* modlarından yararlanılarak yeni bir harici sözlük tanımlamaya ihtiyaç duyulmuştur. Doğru heceleme için gerekli kurallar, oluşturulan bu sözlüğe tek tek eklenmiştir. RC8660'ın bir konuşmacı doğallığında Türkçe konuşması için vurgu ve tonlama kuralları da tanımlanmıştır. Üretilen konuşmanın düzgün ve anlaşılabilir olması amacıyla karakterler ve sayılar için farklı fonemler kullanılmıştır. Kartın yazılımı olan RC Studio'da yer alan *hız, ifade, perde, formant frekansı, ton, gecikme* ve *telaffuz* ayarlarının konuşma üzerindeki etkileri de test edilmiştir. Üretilen konuşmanın kalitesi ortalama görüş skoru (MOS) testi ile ölçülmüştür.

Anahtar Kelimeler: Metinden Konuşma Sentezleme, RC8660 ses sentezleyici, fonem, heceleme, MOS testi

To My Parents

ACKNOWLEDGMENTS

I express sincere appreciation to my supervisor Asst. Prof. Dr. Hakan Tora for his guidance and insight throughout the research. Thanks also go to my co-supervisor Asst. Prof. Dr. İbrahim Baran USLU. To my wife, kid and whole family, I offer sincere thanks for her continuous support and patience during this period.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ADC	Analog to Digital Converter
EVM	Evaluation Kit / Development Kit
FD-PSOLA	Frequency Domain Pitch Synchronous OverLap and Add
GPS	Global Positioning System
GUI	Graphical User Interface
HMM	Hidden Markov Model
I/O	Input/Output
IPA	International Phonetic Alphabet
MOS	Mean Opinion Score
PDA	Personal Digital Assistant
TTS	Text to Speech
PC	Personal Computer
PSOLA	Pitch Synchronous Overlap and Add
LP-PSOLA	Linear Predictive Pitch Synchronous Overlap and Add
RELP	Residual Excited Linear Prediction
SOLA	Synchronous Overlap and Add
TD-PSOLA	Time Domain Pitch Synchronous OverLap and Add
UART	Universal asynchronous receiver and transmitter

CHAPTER 1
INTRODUCTION

The aim of this thesis is automatic conversion from natural language sentences in text form to spoken form. The produced speech is supposed to be as intelligible and natural as possible. The field of speech synthesis has attracted a great attention from researchers, because text to speech conversion finds intensive applications especially for visually and orally handicapped people. These kinds of services are feasible in some areas such as reading emails, web pages, announcement systems, robotics, talking OCR systems, ATM machines, talking pagers and PDAs, GPS navigation systems, vending and ticketing machines, remote diagnostic reporting, dial-up information systems, handheld barcode readers, electronic test and measurement, industrial controllers, security and warning systems.

Speech is one of the ways for human beings to communicate with each other. It is the combination of sounds gathered together according to the language rules and also carries information. The speech is composed of units called phonemes. They are the smallest units that make a difference in the meaning of words. Therefore, phonemes must be properly used to create sound patterns for a relavent pronounciation. In this concept, phonemes are required to be represented with symbols [2]. Each language has its own phonetic alphabet which consists of the list of these symbols. A text to speech technology effectively exploits the phonetics knowledge.

A text-to-speech (TTS) system converts text into speech and reads the text aloud. A computer system used for this purpose is called a speech synthesizer, and can be implemented in software or hardware products. In the recent studies TTS systems are also realized in embedded form due to the rapidly increasing advances in the signal processing and integrated circuit (IC) technologies. There exists IC chips that can synthesize intelligible speech as well. RC8660 Voice Synthesizer is one of them and employed in our study. It can be utilized in multilanguage applications.

Literature survey on Turkish TTS is done for the years between 1991 and 2015 in graduate studies TTS. Table 1 shows the conducted studies. As can be seen there are rare works employing IC based TTS solutions. We are then motivated to develop an IC chip based Turkish synthesizer and fill the gap in that area.

The studies 1 to 29 in Table 1 are borrowed from [1] and the rest shows the latest works.

Table 1 Graduate Studies Conducted for Turkish TTS between 1991 and 2015

No	Author	M.Sc / PhD	Title	Method
1	Alper GERÇEK	M.Sc, 1991	"A TMS 5220 based speech synthesis development system"	Linear predictive coding based hardware implementation
2	Karen BÜYÜKAŞIKOĞLU	M.Sc, 1992	"Konuşma işaretlerinin analiz ve sentezi"	Linear predictive coding was implemented
3	Enis Sezai BAŞARA	M.Sc, 1992	"Yapay ses üretim yöntemleri"	Linear predictive coding was implemented
4	İlhan Yaşar ÖZÜM	M.Sc, 1993	"A speech synthesis system for Turkish language based on the concatenation of phonemes taken from a speaker"	Phoneme and diphone concatenation (linear interpolation and implementation with TMS 320 C25)
5	Kamil GÜVEN	M.Sc, 1994	"PC based speech synthesis for Turkish"	Formant synthesizer for Turkish vowels
6	Murat Servet ERER	M.Sc, 1994	"Karma söz üretme yöntemi ile Türkçe yazılı metinden söze geçme"	Syllable based two different concatenation methods
7	Kerem AYHAN	M.Sc, 1998	"Text to speech synthesizer in Turkish using non parametric techniques"	Syllable concatenation at vowel overlaps using PSOLA and its derivatives
8	Özgül SALOR	M.Sc, 1999	"Signal processing aspects of text to speech synthesizer in Turkish"	Syllable concatenation at consonant overlaps using LP-PSOLA (RELP)
9	Barış BOZKURT	M.Sc, 2000	"Reading aid for visually impaired (A Turkish text-to-speech system development)"	TD-PSOLA based implementation
10	Ömer ESKİDERE	M.Sc, 2000	"Yazılım tabanlı söz sentezleyici tasarımı"	Formant synthesizer for Turkish words
11	Çağla ÖNÜR	M.Sc, 2001	"Concatenative speech synthesis based on a sinusoidal speech model"	Sinusoidal model based implementation
12	Erkan ABDULLAHBEŞE	M.Sc, 2001	"Fundamental frequency contour synthesis for Turkish text to speech"	F0 contour synthesis simulations based on syntactic features
13	Şifa Serdar ÖZEN	M.Sc, 2002	"Türkçe metinden konuşma sentezleme"	LP-PSOLA based diphone concatenation

14	Barış EKER	M.Sc, 2002	"Turkish text to speech system"	Diphone concatenation at voiced parts
15	Ömer ŞAYLİ	M.Sc, 2002	"Duration analysis and modeling for Turkish text-to-speech synthesis"	Linear additive, mean phoneme/triphone and triphone-tree duration models
16	Banu OSKAY	M.Sc, 2002	"Automatic modelling of Turkish prosody"	Intonation models in sentences
17	Esra VURAL	M.Sc, 2003	"A prosodic Turkish text-to-speech synthesizer"	Word stress patterns and allophone duration analysis using the Festival synthesizer
18	Haşim SAK	M.Sc, 2004	"A corpus based concatenative speech synthesis system for Turkish"	Corpus based harmonic coding concatenation
19	Ozan AKTAN	M.Sc, 2004	"A single chip solution for text-to-speech synthesis"	Concatenation of LPC coded letter recordings using a single IC
20	Asude KARLI	M.Sc, 2005	"Örnek bir dizi cümle için Türkçe metinden konuşma sentezleyici"	Diphone concatenation using prototype waveform interpolation
21	Özlem ÖZTÜRK	PhD, 2005	"Modeling phoneme durations and fundamental frequency contours in Turkish speech"	Regression trees for duration and F0 models
22	İlker ÜNALDI	M.Sc, 2007	"Taşınabilir cihazlar için Türkçe metinden konuşma sentezleme sistemi"	LP-PSOLA based diphone concatenation on a mobile device
23	Cansel DEMİR	M.Sc, 2009	"Konuşma tanıma sentezleme sistemlerinin okul öncesi dönem yabancı dil eğitiminde kullanılması"	Microsoft Speech Server TTS based implementation for foreign language learning
24	Kenan GÜLDALI	M.Sc, 2009	"Türkçe metin seslendirme"	Rules for Turkish syllable concatenation
25	Zeliha GÖRMEZ	M.Sc 2009	"Implementation of a text-to-speech system with machine learning algorithms in Turkish"	Unit selection based concatenative synthesis using TD-PSOLA
26	Yücel BİCİL	M.Sc, 2010	"Türkçe metinden konuşma sentezleme"	TD-PSOLA based concatenative synthesis
27	Cavit ERDEMİR	M.Sc, 2010	"Türkçe metin seslendirme için doğal konuşma sentezleme"	Waveform concatenation using at most two-letter syllables
28	Tuncay ŞENTÜRK	M.Sc, 2010	"Türkçe metin seslendirme"	Two-phone and syllable based concatenation
29	Güray ARIK	M.Sc, 2011	"Görme engelliler için bilgisayar kullanımının etkinleştirilmesi, erişilebilirlik ve bir Türkçe hece tabanlı konuşma sentezleme sisteminin geliştirilmesi"	Syllable based concatenation system for the visually handicapped

30	İbrahim Baran USLU	PhD 2012	Konuşma işleme ve Türkçenin dilbilimsel özelliklerini kullanarak metinden doğal konuşma sentezleme	Natural speech synthesis using duration, pitch and energy modifications
31	İLHAMİ SEL	M.Sc 2013	Türkçe metinler için hece tabanlı metinden konuşma sentezleme sistemi	Word synthesis using SOLA methods
32	EKREM GÜNER	M.Sc 2013	A hybrid statistical/unit-selection text-to-speech synthesis system for morphologically rich languages	Unit Selection and HMM based TTS
33	ABDULKADİR KARACI	PhD 2013	Ses sentezleme ve tanıma teknolojilerini kullanarak Türkçenin ana dil olarak öğretimi için zeki öğretim sistemi geliştirilmesi	Clever Learning System design using speech synthesis and recognition technologies
34	AMİR MOHAMMADİ	M.Sc 2014	Speaker adaptation with minimal data in statistical speech synthesis	Statistical speech synthesis and speaker adaptation

In this thesis, a Turkish Text to Speech synthesis system is implemented with RC8660. For this purpose, phoneme mapping between English and Turkish is first accomplished since the synthesizer is originally designed for English language. Then phonetic rules are described to utter the Turkish sounds as naturally as possible by exploiting the Turkish grammer structure. Furthermore the prosodic features, pitch and amplitude, are determined for proper stress and intonation.

This theses is organized as follows: After this Introduction chapter, in Chapter 2 Text to Speech Synthesis issues are covered in detail. Chapter 3 describes the RC8660 Voice Synthesizer. In Chapter 4, Turkish phonemes and pronounciation rules are defined. Tests and Results are presented in Chapter 5. Finally in Chapter 6 Conclusion is discussed.

CHAPTER 2

TEXT TO SPEECH SYNTHESIS

Text to speech synthesis (TTS) is the automatic conversion of any text (Internet, Email, Word Document, E Books, Newspaper, Book, Magazine) to spoken form. It can specifically be viewed as the mimic of human speech by using computing abilities. All the stages of this process are discussed in detail in this chapter.

2.1 TTS Blocks

Basic parts of TTS are shown in Figure 1. As seen, fundamental components are: Text pre-processing, Prosody and Synthesis. In such systems text is always an input and speech is always an output. The followings describe each block of the TTS system.

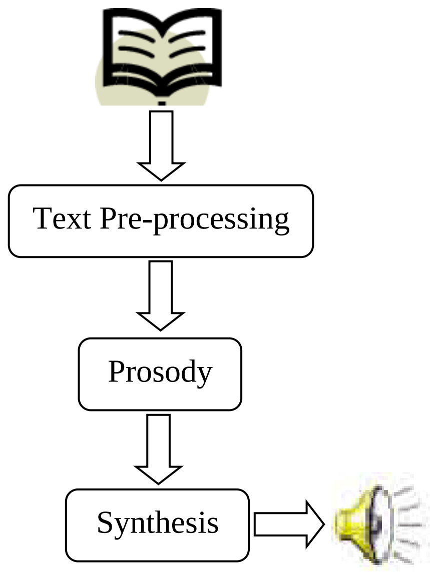

Figure 1 Text to Speech Synthesis (TTS) System

A. Text pre-processing

This block first accepts the text input. This text passes through some normalization processes. In this step, all the punctuations, abbreviations, numbers, and some special characters are identified. In addition to this, sentence and word boundaries are determined. After this, each word is the decomposed into its syllables. The last step of text pre-processing is an assignment of phonemes to each syllable. This is called text-to-phoneme conversion. For this purpose, speech synthesis systems use two basic approaches. The first approach is the *dictionary-based* approach, where a large dictionary containing all the words of a language and their correct pronunciation is stored by the program. Specifying the correct pronunciation of each word is done by looking up each word in the dictionary and replacing the letters with the pronunciation specified in the dictionary. The second approach used for text-to-phoneme conversion is the *rule-based* approach, where the pronunciations of words are found according to the structure of the words.

RC8660 Voice Synthesizer which is used in our thesis, has some text pre-processing features. The text-to-speech modes of the RC8660 utilize an English lexicon and letter-to-sound rules to convert text it receives into speech. Exception dictionaries make it possible to alter the way the RC8660 interprets character strings it receives. This is useful for correcting mispronounced words. The pronunciation rules determine which sounds, or phonemes, each character will be matched with based on its relative position within each word. The integrated DoubleTalk text-to-speech engine analyzes text by applying these rules to each word or character, depending on the operating mode in use. Exception dictionaries help us in redefining the built in text to phoneme conversion rules.

In English there are approximately 44 sounds, corresponding to 26 letters, with some differences regarding style of pronunciation and articulation. On the other hand there are 55 recorded different phonemes to make RC8660 speak out in English. In order for RC8660 to produce Turkish speech, 37 from these 55 phonemes were selected. The list of these phonemes is given in Table 2 in Section 2.3.

B. Prosody

Prosody is the suprasegmental properties of an utterance. It deals with intonation, tone, stress and rhythm in the field of linguistics.

In terms of speech signal processing, prosodic features are related with energy, pitch and duration. Energy is related with the amplitude of the sounds. Pitch frequency (fundamental frequency) is the characteristic feature of the speaker and a measure of how bass/treble a speaker's voice. Rhythm is related with the duration of and the pauses in and between words. Stress and intonation patterns are strongly influenced by these three parameters.

Before synthesis, the prosodic labeling must be performed in this step. The stressed syllable(s) in a word should be provided after a prosodic analysis. Similar to word stress, some parts of a sentence such as phrases or question words must be assigned a proper intonation. In addition to this, the effects of the punctuation marks, namely the required pauses, must also be considered.

The electronic integrated circuit used in this work is capable of controlling the prosodic features mentioned above. One can set each feature to a desired value to produce a human like sound. The settings: Volume, Speed, Pitch, Expression, Reverb, Formant Freq, Tone, Articulation, and Delay come with RC8660 voice synthesizer. All of these settings can be adjusted within predefined ranges. Hence RC8660 allows to play with the recorded sounds in the way we want. Note that, pitch frequency modifications can only be applied to the voiced parts of the speech. So vowels are good segments for pitch changes. Increasing volume at the end of the words or increasing pitch at the question words give stress and it helps understanding the speech better.

C. Synthesis

Synthesis part of a TTS system can be implemented according to three approaches: Concatenative, Articulatory, Formant synthesis. In this section the three methods are examined in detail.

i. Concatenative Synthesis

Concatenative synthesis uses pre-rocorded human voices. Therefore most natural sound comes with this type of synthesis. The voices are consecutively added one after another with proper overlaps. Phonemes, diphones, syllables or words can be the units for concatenation. The disadvantage of this method is the requirement of a recording database. RC8660 voice synthesizer adopted concatenative synthesis approach to produce natural sounds. It has 55 pre-rocerded phonemes.

ii. Articulation Synthesis

Articulation synthesis is based on modeling of the human vocal tract compromised of tong, lips, teethes, and jaw. In a way this approach can be seen as creating an artificial vocal tract and because of this it has many difficulties. The vibration produced by the vocal cords is travelling through the glottal organs and takes its final form as an audible sound.

iii. Formant Synthesis

Unlike concatenative synthesis, formant synthesis does not use any human voices. However it uses some variables for synthesis. These variables are frequency, amplitude, and tone. Sometimes this method is known as rule based synthesis. Basically this method tries to mimic the acoustic waveform using the formant frequencies.

2.2 Speech Synthesizer Systems

Around the 1950s the first computer based text to speech synthesis system was built and the first developer was Noriko Umeda in Electrotechnical Laboratory in 1968. In Bell Laboratory John Larry Kelly and Louis Gerstman around the 1961, they used IBM 704 for text to speech synthesis.

Electronics life started around in the 1970s for text to speech synthesis. First developer was Telesensory Systems for text to speech system. First text to speech system was done for the blind people in 1976. After couple years in 1978 Texas instruments produced educational product.

In 1979 first chess computer started to speak, after then in 1980 speech synthesis implement to games by electronic company.

Afterwards Apple, Amiga, and Microsoft started to use text to speech sytems in their hardware system. Texas instruments, national semiconductor, and silicon system used text to speech system in their hardware system.

For now we are using text to speech system online on internet, on Adroid smartphones or on Apple smartphones, and on opensource systems. Latest most of the smartphones, Microsoft operating systems, and Apple Operating systems are using text to speech system in their systems.

2.3 Proposed Turkish Synthesis System

The RC8660 used in the thesis is a complete text to speech synthesizer. In other words, it implements each block shown in Figure 1. The synthesis system that we proposed in this study significantly relies on adapting English phonemes to Turkish language since the synthesizer employed in the thesis is designed for English. Table 2 shows how Turkish letters are related to the phonemes both built in RC8660 voice synthesizer and in International Phonetic Alphabet (IPA). For example: Turkish "i" is uttered as in "s<u>ee</u>" in English, Turkish "a" is uttered as in "<u>fat</u>her", Turkish "ı" is uttered as in "bot<u>tom</u>", Turkish "ç" is pronounced as in "<u>ch</u>air" and Turkish "ş" is pronounced as in "<u>sh</u>ow".

The Turkish letter 'ğ' is not paired to any of the English phonemes. Eventhough that letter is not pronounced in our system we reflect its effect by increasing the duration of pronounciation of the vowel immediately before 'ğ'. Thus, one can hear the word with 'ğ' as if 'ğ' were pronounced. Moreover, some of the vowels are necessarily

Table 2 Phonemes List

No	Turkish Letter	RC8660 Phonemee Symbol	English Example Word	IPA	Alternative phonemee Symbols
1	a	aa	Cot	/a/	ah (cut)-ay(bite)
2	ı	ax	Bottom	/ ɯ/	--
3	e	ey	Bake	/e/	eh (bet)
4	i	iy	Beet	/i/	ih (bit) ix (rabbit)
5	o	ow	Boat	/o/	oy (boy)
6	ö	er	Bird	/ø/	--
7	u	uh	Book	/u/	uh(book)
8	ü	yy uh	Pure	/y/	uh(book)
9	b	b	Bib	/b/	--
10	c	j	Age	/ʤ/	--
11	ç	ch	Church	/ʧ/	--
12	d	d	Dad	/d/	--
13	f	f	Fee	/f/	--
14	g	g	Garage	/g/ , /ɟ/	--
15	ğ	--	Previous Vowel Extended	/ː/, /‿/, /ʲ/	--
16	h	h	Hat	/h/	--
17	j	zh	Measure	/ʒ/	--
18	k	k	Cut	/k/, /c/	--
19	l	l	Label	/ɫ/, /l/	--
20	m	m	Man	/m/	--
21	n	n	Nine	/n/	--
22	p	p	Pitch	/p/	--
23	r	r	Ring	/r/	--
24	s	s	See	/s/	--
25	ş	sh	Shell	/ʃ/	--
26	t	t	Time	/t/	-
27	v	v	Valve	/ʋ/, /v/	--
28	y	yy	Yes	/j/	--
29	z	z	zoo	/z/	--

paired to several phonemes in order to get the pronounciation of a word understood much better. For example, the sound of the letter ‘a’ in ‘sahne’ (stage) is produced by the phoneme ‘ah’ while the first ‘a’ in ‘sayaç’ (counter) is pronounced by ‘ay’. In order to determine the correct pronunciation of Turkish letters, a lot of Turkish texts were tested and the most relevant pairs and rules were obtained. That process was exhaustive.

To use the RC8660 for any language other than English, an Exception Dictionary is required. Turkish pronounciation rules, numbers, abbreviations, special characters and the phoneme pairs in Table 2 should be included in this Exception Dictionary. Hence the RC8660 is ready to use for Turkish languge.

CHAPTER 3
RC8660 VOICE SYNTHESIZER

RC8660 is a general purpose voice synthesizer, and this synthesizer is controlled by computer using RCStudio software. In this chapter, the properties of RC8660 are presented.

3.1 General Description

The RC8660 is a multilanguage voice synthesizer, having a text-to-speech (TTS) processor, recording sound and playback capabilities, tone generators for musical sound, telephone numbers dialling and Analog to Digital converter. Using a RS232 standard serial communication or bus interface, ASCII text is able to be sent automatically into the RC8660 speech processor.

Also RC8660's TTS processor includes DoubleTalk technology. This DoubleTalk technology is based on a voice concatenation technique and uses real human voice samples. This processor allows us to control the speech parameters which are Volume, Speed, Pitch, Expression, Reverb, Formant Freq, Tone, Articulation, and Delay.

RC8660 has 33 minutes recording ability and up to 7.5 MB nonvolatile memory for the storage. It has also greeting messages and the system can store and play them automatically.

Programmable dictionary allows the user to redefine any character and pronounce the character. Therefore RC8660 is able to be used for multilanguages by the help of this dictionary.

3.2 RC8660 Block Diagram

RC8660 board has two surface mounted devices and these devices operate with +3.3V or +5V power supply. The block diagram of RC8660 in Figure 2 shows the processing route.

The operation of the board is as follows:

- Start with RCStudio on the computer and connect to RC8660,
- Use V-POD for communication with computer. V-POD is equipped with RS232 communication chip.
- After connection with V-Stamp over the ASYNC SERIAL I/F with RXD, TXD processing goes to 8KB INPUT BUFFER and DOUBLETALK TEXT-TO-SPEECH PROCESSOR.
- Doubletalk text to processor uses 16KB Exception Dictionary for other languages and processing goes to 2KB AUDIO BUFFER, after this point processing goes to ANALOG AUDIO I/F then goes to audio power amplifier for output with speaker.

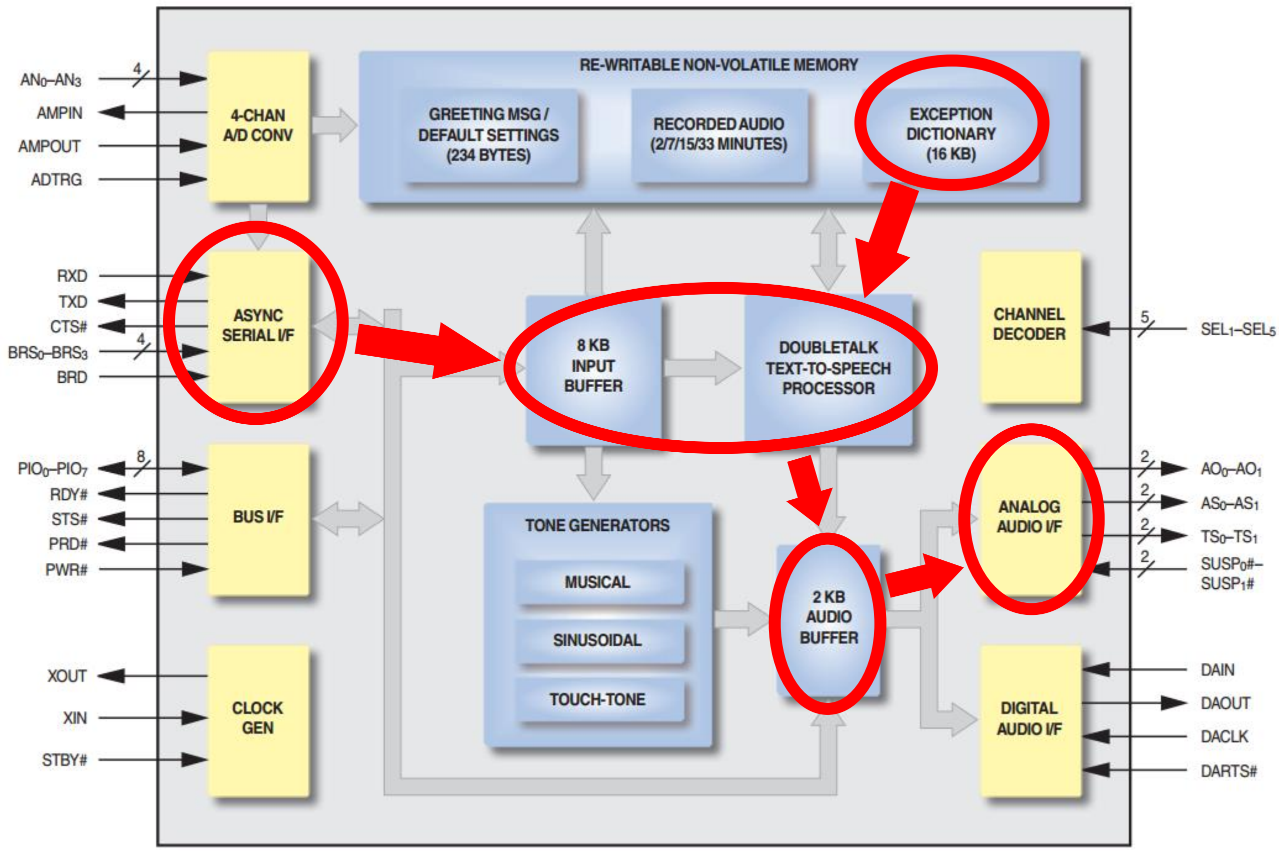

Figure 2 RC8660 Block Diagram

3.3 Alternative Usage Options for RC8660

Before explaining V-POD and V-Stamp which are used for this work, there are other ways for the user to install V-Stamp with other peripherals. The card can be interfaced to microcontrollers and FPGA cards.

Microcontrollers are cheap and easily accessible in the market. They can be used with the V-Stamp card. Depending on the application RS232 or USB communication can be preferred. RC8660 is able to send data or receive data over the RX and TX pins. Microcontrollers are able to control RC8660 through those pins.

FPGAs are more complicated devices. They can also be used with RC8660. There are different kinds of ARM processors to use with RC8660 voice synthesizer. The user can implement different combining methods like HMM, TD-PSOLA or FD-PSOLA on the FPGAs.

3.4 V-POD & V-STAMP

In our work we used both V-POD and V-STAMP. V-POD is the communication part between the computer and embedded system, and V-STAMP is the Text to Speech Processor.

3.4.1. V-POD

Figure 3 shows the V-POD. The V-POD is a development platform and programming adapter for the RC Systems' RC8660 and V-Stamp voice synthesizer modules. It is ready-to-use and cheapest way to develop product and connect with any PC using serial connection. Thus development items such as RCStudio and RCLink can become able to be used with RC8660. The V-Pod is also able to be used for creating and downloading data files as greeting messages, exception dictionaries (for other languages), and sound files to RC8660.

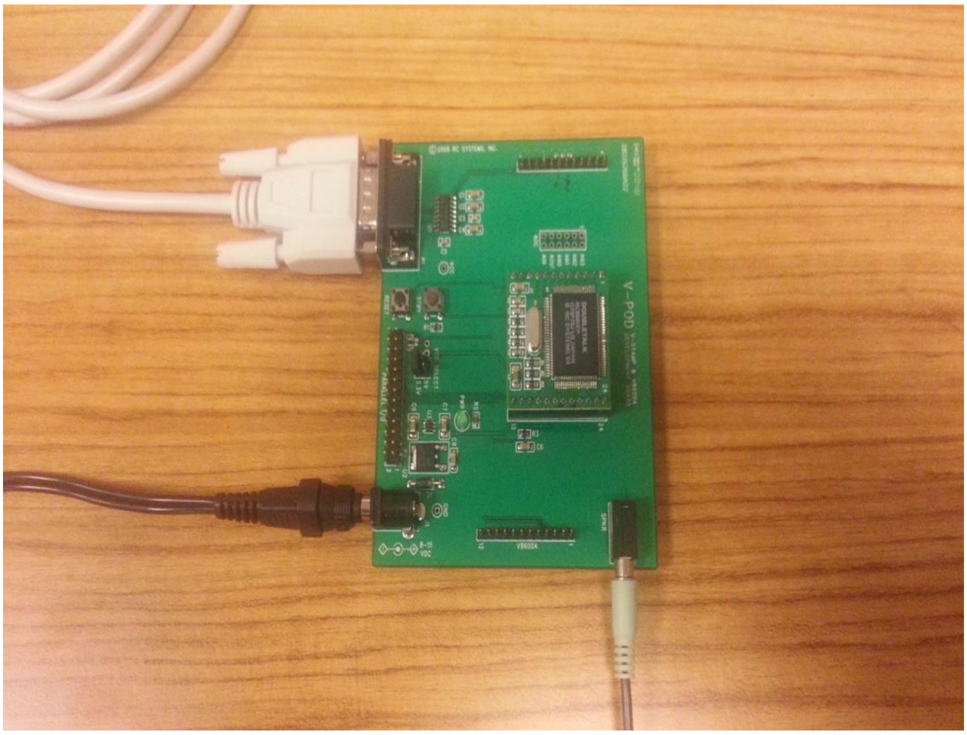

Figure 3 V-Pod

There is a docking board as a daughter board to support speaker output through the 3.5 mm speaker jack (SPKR). You can use stereo speakers, but you will hear mono sound. This system supports serial communication and there is serial communication interface (P1) on the V-Pod. This serial communication interface has DB-9 connector and connects the computer with serial communication cable using serial (COM) port on your PC as seen in Figure 4 .

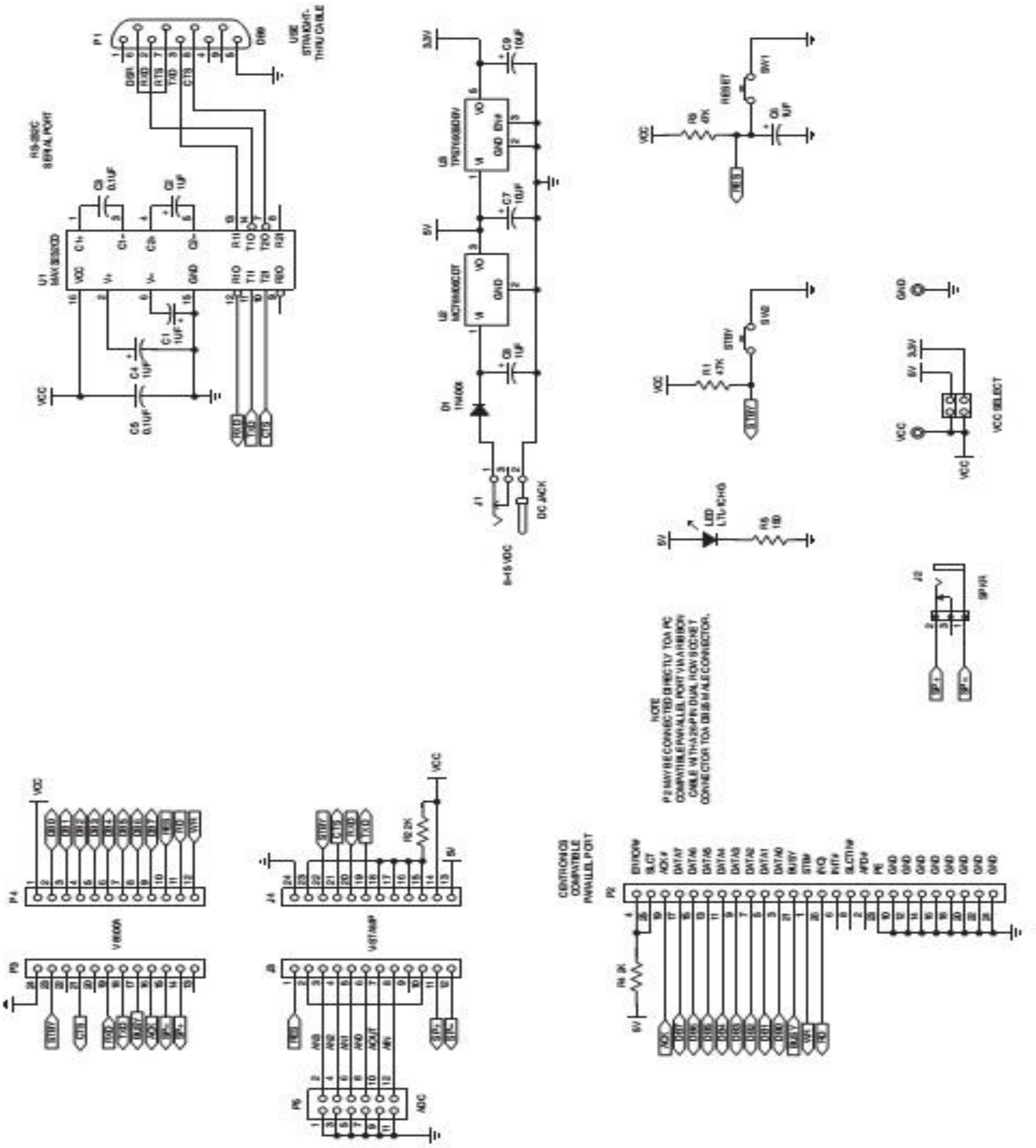

Figure 4 V-POD Docking Board Schematic

3.4.2. V-STAMP

Figure 5 shows the V-Stamp board. The V-Stamp card is of size 11cm^2. Because of this size and special controls, which does not need any other electronic devices, V-Stamp becomes good in voice synthesizer category. V-Stamp is also capable of audio recording, playback, tone generation and analog to digital converter channels.

V-Stamp takes power from V-Pod. It has 24 pin PC board connectors. Also there are two different types of V-Stamp, one of which works with 3.3V power and the other works with 5V power. However 33 minutes recording capacity type of V-Stamp, implemented for this work, works with 3.3V power.

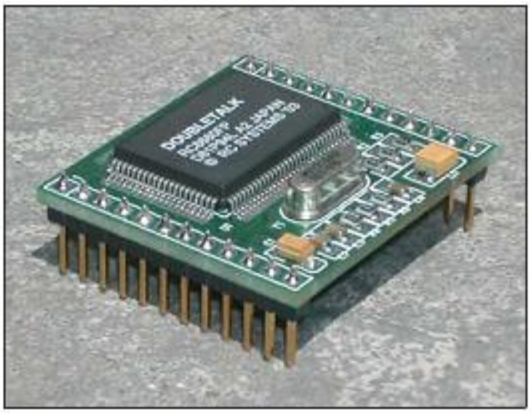

Figure 5 V-Stamp Board

Figure 6 shows a typical application circuit.

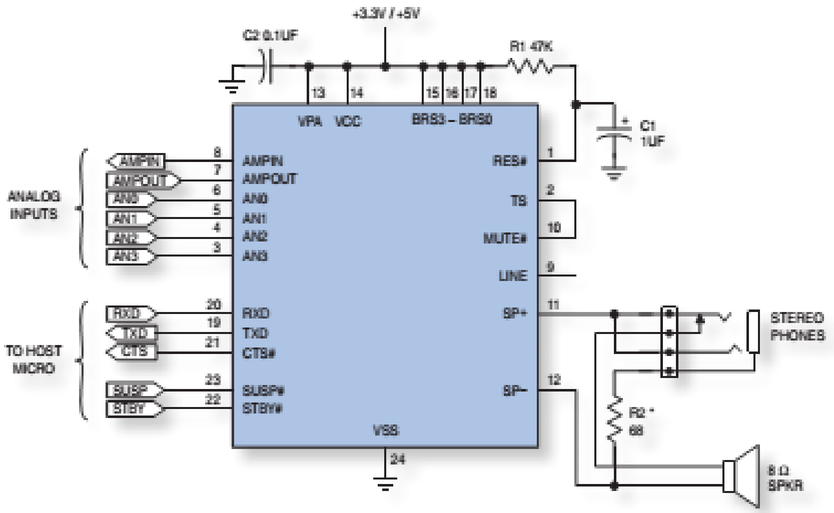

Figure 6 Application Circuit

3.5 Control Parameters of TTS

The RC8660 card has got many prosody control features which can implement the prosody block in Figure 1. We can control the output voice with TTS Controls. The TTS controls are: Volume, Speed, Pitch, Expression, Reverb, Formant Freq, Tone, Articulation, and Delay. Each control can be adjusted between minimum and maximum values. Figure 7 shows the control panels of RC8660.

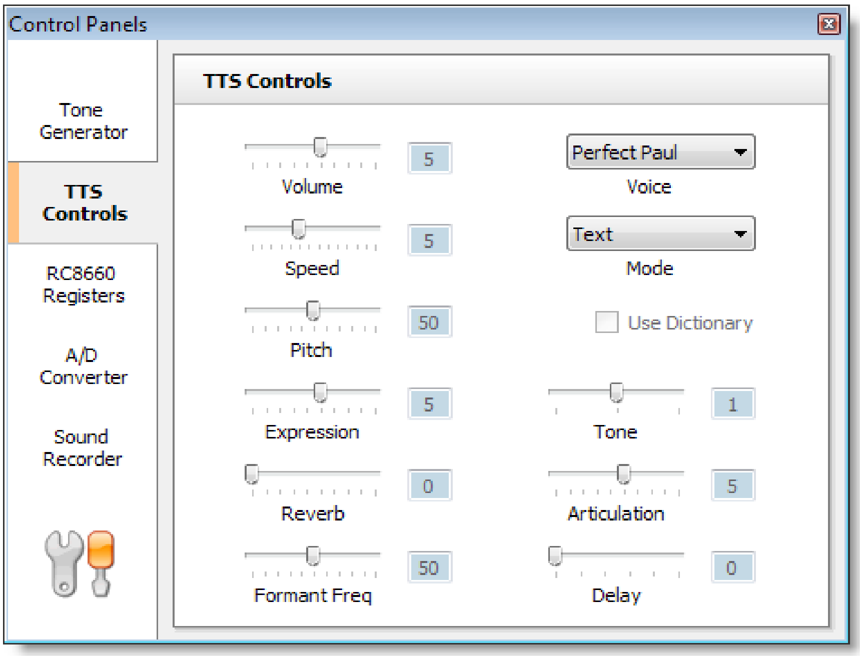

Figure 7 TTS Control Panels

3.5.1 Volume

The volume is used to set the output sound level with changing the voice amplitude in 10 steps (default 5). As the step number increases the sound level gets higher.

3.5.2 Speed

This control sets the synthesizer's reading speed. It changes both the inter-word and inter-syllable delays. It has a scale from 0 to 13 (default 5). When the scale increases, the speech gets gradually faster, and when the scale is decreased, the speech gets gradually slower. If the speed is sufficiently high, understanding the words becomes very hard.

3.5.3 Pitch

Pitch is the fundamental frequency of the speaker’s voice. It determines the main characteristic of the speaker’s voice. RC8660 has a slider to scale the pitch from 0 to

99 (default 50). As the scaling factor increases the voice goes gradually from bass to treble.

3.5.4 Expression

This control sets the reading manner of the synthesizer. Expression is directly related with the intonation. RC8660 has a scale for the *expression* from 0 to 9 (default 5). As the scaling factor increases, speech intonation gets away from monotonic reading.

3.5.5 Reverb

Reverb is echo of the voice. Reverb controls reverberation of the speech. RC8660 has a slider scale for reverb from 0 to 9 (default 0). As the scaling factor increases the echo starts.

3.5.6 Formant Frequency

This control affects characteristic of the voice by adjusting vocal tract formant frequencies. Slider scales the formant frequency from 0 to 99 and the default value is 50. As the scaling factor increases sound can be fine adjusted.

3.5.7 Tone

This control determines the tone of the voice coming from the voice synthesizer. There are three levels of the tone: Bass (0), Normal (1), and Treble (2).

3.5.8 Articulation

This controls how fast the phonemes are concatenated. This is also related with the articulation model of the glottis. When this control is low, sound is slurred; when this control is high, the voice becomes choppy. RC8660 has a slider scale for reverb from 0 to 9 (default 5).

3.5.9 Delay

This control adjusts how much delay is inserted between words in text mode and character mode. Slider bar adjusts the delay from 0 to 15 and default value of the delay is 0.

3.6 RCStudio

RCStudio is developed by RC System for debugging and setting RC8660. This application can be used as a development software for preparing Turkish Exception Dictionary for RC8660. RCStudio supports RC8660 chipsets including the V8600 series, V-Stamp and DoubleTalk LT.

In RCStudio it is possible to define Exception Dictionaries, create greeting messages and musical tones. In Figure 8 you can see RCStudio's screenshot, and RCStudio has 3 default screens: TTS control panel, Text Box, and Phonetic Output. Basically you can type your text in Text Box, from where the system takes words and after processing prepares the speech output. You can control voice prosody with nine parameters in TTS Control panel. You can see the phonetic representation of the pronounciation in Phonetic Output panel.

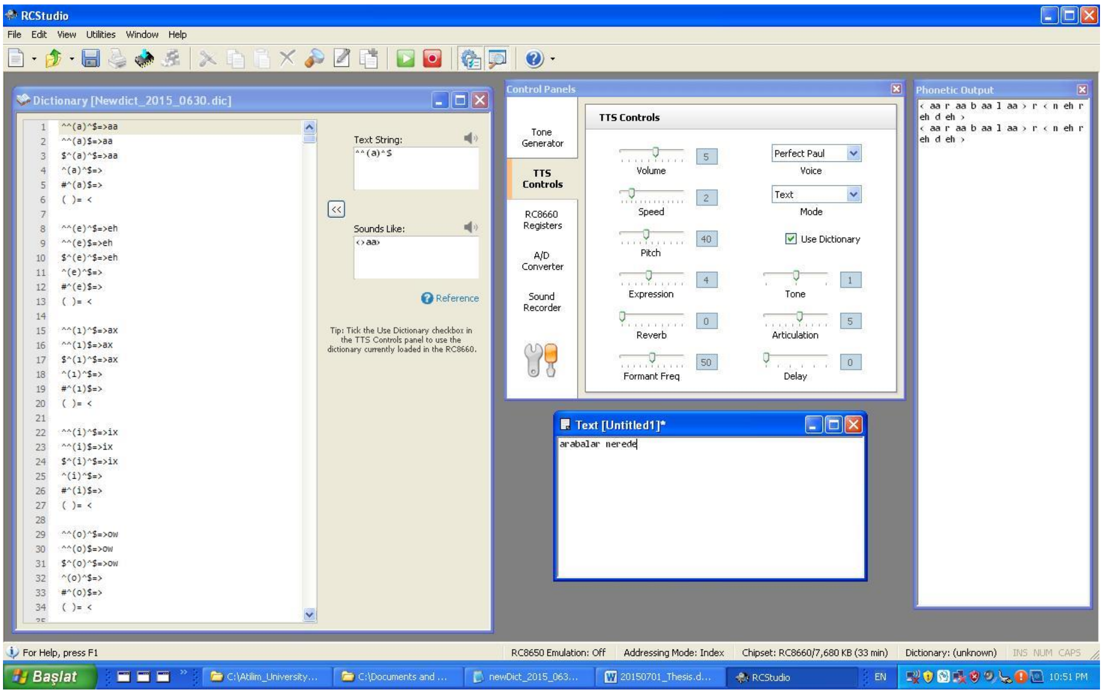

Figure 8 RCStudio Graphic User Interface

In Figure 9, you can see a sample of Turkish Exception Dictionary. Each line was directly typed on the dictionary text editor. This editor enables us to try individual phonemes via “Sound Like” and “Text String” boxes.

RCStudio helps creating Exception Dictionary. Once that dictionary is completed will have a file with an extention of (.dic). in order to send that dictionary file to the speech processor it needs to be compiled. Compilation results in a file with an extension of (.dix) which is used by the processor. The size of the .dix file is limited to 16KB. In our study Exception Dictionary developed for Turkish was 22KB, and corresponding .dix file was 12KB. In order to apply our Exception Dictionary to synthesizer we should activate “use dictionary” box. The system is now ready to pronounce the text written in “text window” in Turkish. In the mean time pronounced text appears as a phonetic form in the “phonetic output” window. Apart from this, using the mode dropbox menu, textbox window can be functionated in either phonetic mode or text mode.

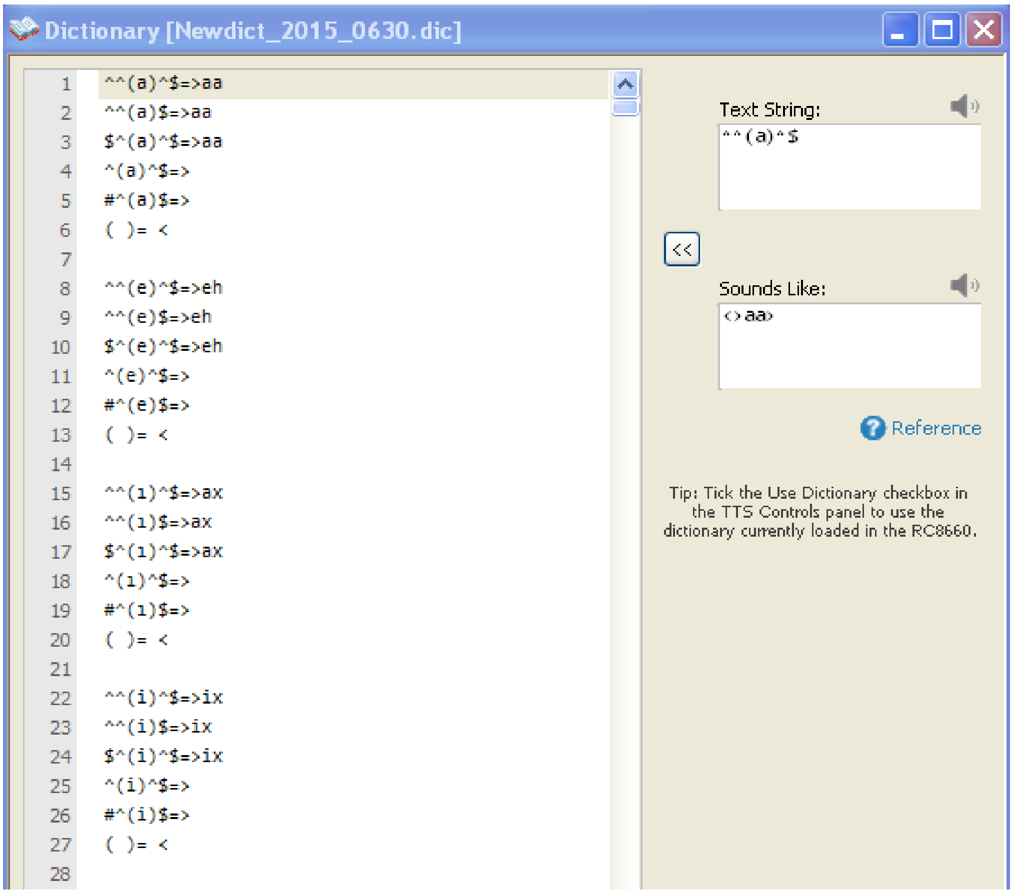

Figure 9 Turkish Exception Dictionary

For example the statement “arabalar nerede” is typed as it is in the text window if the text window is in text mode as shown in Figure 10. Whenever the text is synthesized by the play button, its phonetic representation appears in the phonetic output window as shown in Figure 11. On the other hand if text window is in the phonetic mode, the input text is supposed to be in the phonetic form. That operation is quietly useful for one to decide on the pronounciation rules empirically.

Figure 10 RCStudio Text Editor

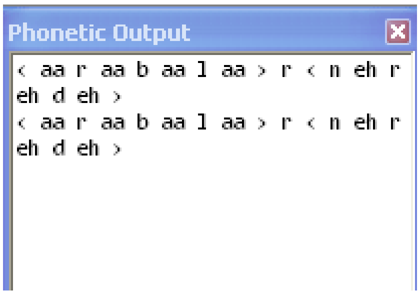

Figure 11 RCStudio Phonetic Output

RCStudio communicates with V-POD and V-STAMP via RS232 protocol. It has communication preferences as shown in Figure 12. Configuring the COM port number and baud rate (300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200) is possible here.

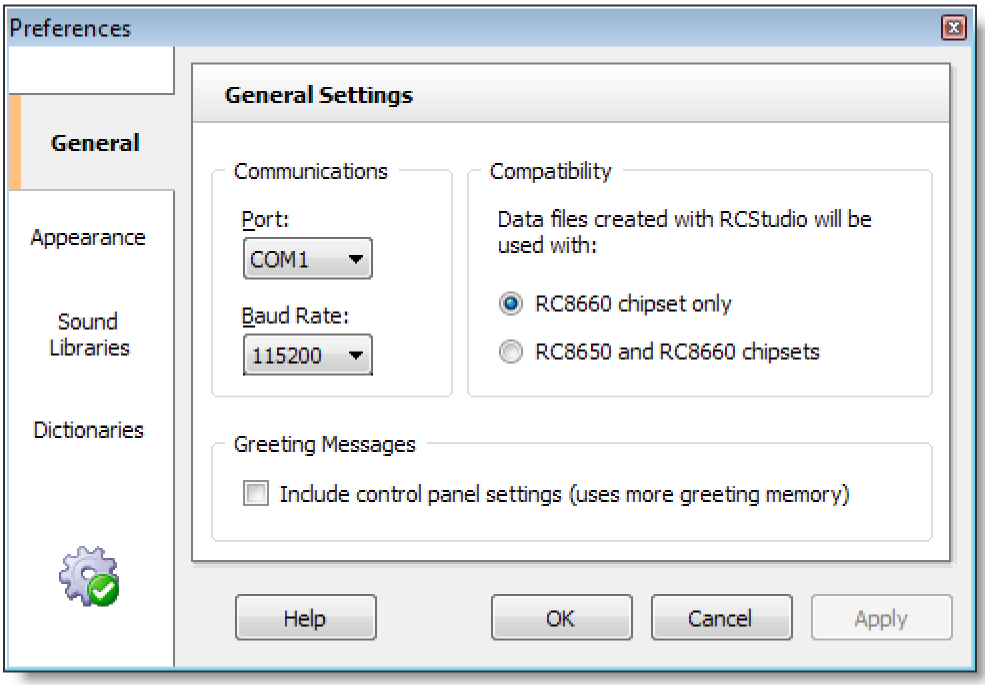

Figure 12 Serial Communication Sets

CHAPTER 4

ENGLISH TO TURKISH PHONEME MAPPING

There are two ways for developing a text to speech synthesis system namely: dictionary based (concatenative) and rule based (formant, articulitary). For this work, in order to make RC8660 to synthesize Turkish, an Exception Dictionary is created and rules for Turkish grammer are defined in this dictionary. Exception Dictionary allows the user to organize English to Turkish Phoneme Mapping and this is a powerful tool for the user. It is based on phoneme mapping from English to Turkish and the details of this mapping is illustrated in this chapter.

4.1 Dictionary Based Synthesis

This work is based on a dictionary, and RC system gives us the permition of using the phoneme sources for another language, and that is called “Exception Dictionary”. There is already an Exception Dictionary for Spanish language in the RCStudio examples directory, and this Exception Dictionary helped us in creating Turkish Exception Dictionary.

4.2 Exception Dictionary

Exception Dictionary is the way for RC8660 to translate any character or string which it receives. Exception Dictionary allows the users with full authorization to change mispronounced section in the sentences or words. User can redefine phonemes in the Exception Dictionary file and can make tests with the defined phonemes. For example we can set more than one phoneme for one letter, using the Exception Dictionary. The rules for proper pronunciation is very important for intelligible and natural synthesis.

RCStudio has a status bar at the bottom of the RCStudio GUI, and the user is able to see the loaded Exception Dictionary which the RC8660 will use. If RC8660 has a loaded Exception Dictionary from the previous work, user will see the dictionary named as: "Unknown". RC8660 holds the Exception Dictionaries in volatile memory.

4.3 How to Create Rules for Exception Dictionaries

In Table 3, the symbols and their definitions that can used in defining Exception Dictionary rules are given. How to define the rules for RC8660 will be illustrated with examples below.

Table 3 Syntax For RC8660 System

Symbol	Definition
#	A vowel: a, e, i, o, u, y
+	A front vowel: e, i, y
^	A consonant: b, c, d, f, g, h, j, k, l, m, n, p, q, r, s, t, v, w, x, z
*	One or more consonants
:	Zero or more consonants
?	A voiced consonant: b, d, g, j, l, m, n, r, v, w, z
@	One of: d, j, l, n, r, s, t, z, ch, sh, th
!	One of: b, c, d, f, g, p, t
%	A suffix: able(s), ably, e(s), ed(ly), er(s), ely, eless, ement(s), eness, ing(s), ingly (must also be followed by a non-alphabetic character)
&	A sibilant: c, g, j, s, x, z, ch, sh
$	Any non-alphabetic character (includes numbers, spaces, etc.)
~	One or more non-printing characters (spaces, controls, line breaks, etc.)
\	A digit (0-9)
|	One or more digits (commas are ignored)
`	Wildcard (matches any character)

Example 1: “ #^(a)$= ”

If there is a consonant and a vowel on the left hand side of “a”, and also if there is a space on the right hand side of “a”; then no sound output.

Another example of the "a" is "$(a)^=aa ".

If there is space on the left hand side of "a", and if there is a consonant on the right hand side of "a"; then pronounce "aa" .

Example 1 statements are summarized as follows:

 # ^ (a) $ =
A vowel-a consonant-(recorded a)-space = no sound

$(a)^=aa
Space-(recorded a)-a consonant = pronounce "aa"

Example 2: "(1)\\\=b ix n"

With this rule, we define how to read the numbers. If there are three more numbers after (1), then pronounce as: as a "b ix n" (bin (TR) / thousand (ENG)). For the "hundred", if there are two numbers coming after the (1) like: "(1)\\=yy uh z", pronounce as: "yy uh z" (yüz (TR) / hundred (ENG)).

Example 2 statements are summarized as follows:

(1)\\\ = b ix n
(1)-a number-a number-a number = pronounce "b ix n"

(1)\\=yy uh z
(1)-a number-a number = pronounce "yy uh z"

Example 3: "(.f.)=f eh"

With this rule, we define an abbreviation in the middle. If there is a "dot" on the left and on the right of the letter "f"; then pronounce as: "f eh".

Another example: "$(f.)=f eh"

This is an example of an abbreviation in the beginning. If "f" has space on the left side and dot on right side of "f"; then pronounce as: "f eh".

Example 3 statements are summarized as follows:

(.f.) =f eh

Point-(recorded f)-dot = f eh

(f.) =f eh

Space-(recorded f)-dot = f eh

The remaining rules for abbreviations and special charaacters are listed in Table 4, Table 5 and Table 6.

Table 4 Description of Abbreviations

$(TBMM)$	=	--t eh --b eh --m eh -- m eh
(THY)	=	-- t eh --h eh --y eh
$(SHGM)$	=	--s eh -h eh --g eh --m eh
(DOC)	=	d o ch eh n t

Table 5 Punctuation

$( )=	($)=d ow l aa r	(;)=,
( )=	(=)=eh sh eh t	(,)=,
;\(-)\=	(&)=v eh	(.)\=n ow k t aa
(-)\=ey k s ix	(*)=y ax l d ax z	(:)\=
(-).\=ey k s ix	(%)= yy uh z d eh	(^)=d uh z eh l t m eh ix m ix
(-)=	(,)=,	(_)=aa l t ch ix z g ix
(+)=aa r t ax	(#)=d ix y eh z	

Table 6 Special Characters

(()=aa ch p aa r aa n t eh z
())=k aa p aa p aa r aa n t eh z
(<=)=k uh ch uh k eh sh eh t
(>=)=b uh y uh k eh sh eh t

4.4 Syllables

While making tests on Turkish text to speech synthesis, differencies between the Turkish and English syllabic structures became very important for right pronounciation. Since the English grammer is different from Turkish grammer, we had to define the Turkish syllabic structure to the RC8660.

Turkish Syllabic Structure:

In Turkish language, there are seven different syllabic structures, which are given below:

- Vowel type.

  Sample: **a**

- Consonant - Vowel type.

  Sample: a-**ra-ba**, **bi-çi-mi-ne**

- Vowel - Consonant type.

  Sample: **An**-ka-ra, **el**-bi-se

- Consonant - Vowel - Consonant type.

  Sample: **sev-mek,** Is-**tan-bul**

- Vowel Consonant Consonant type.

  Sample: **alt**-lık, **ult**-ra

- Consonant - Vowel - Consonant - Consonant type.

  Sample: **Türk**-çe, **kork**-mak

- Consonant - Consonant - Vowel - Consonant type.

  Sample: **prog**-ram, **krip**-to (from western origin words)

As can be seen in the above examples, there is only one vowel in one syllable. In addition, in Turkish syllabic structure, the consonant before a vowel is always joined to that vowel. For example, the word “inek” (cow), is not syllabified as: “in-ek” but “i-nek”. Therefore, the consonant “n” is joined to the vowel “e” as indicated in this example. On the other hand the word “araba” must be divided as: “a-ra-ba”, but according to English syllabic structure it is syllabified as: “a-rab-a” and this results with a wrong pronounciation. To overcome this conflict, a rule is described in the Exception Dictionary to make RC8660 pronounce Turkish correctly.

4.5 Encountered Difficulties

There are some difficulties encountered throughout the dictionary defining process. Especially different syllabic structures of Turkish and English affected the pronounciations of RC8660 very much. Depending on the different syllabic structures of Turkish and English languages, we implemented new rules to the Exception Dictionary. Also special Turkish letters (ğ, ç, ö, ü, ı) had to be defined to the RC8660 since these letters do not exist in English.

4.5.1 Implementing Turkish Consonant-Vowel-Consonant Syllabic Structure in the RC8660

The main diffence between the Turkish and English syllabic structures is in the words consisting of Consonant-Vowel-Consonant type syllables.

We noticed this problem while synthesizing the word “araba” (car). At first we heard the synthesis incorrectly as: “a-rab-a”, but it should be pronounced as: “a-ra-ba”. Another incorrect sound occured for “otobüs” (bus). We again hear that as: “o-tob-üs”, and it should be pronounced as: “o-to-büs”. For solving this problem we defined rules as:

(a) = aa	(o) = ow
a(r)a = r aa	o(t)o = t ow
a(b)a = b aa	o(b)ü = b uh
	(s) = s

All the other consonant-vowel-consonant type rules can be found in Appendix A.

4.5.2 Implementing “ğ, ç, ö, ü, ı” in the RC8660

We ran into another problem with the letters specific to the Turkish alphabet, which are: “ç, ö, ü, ğ, ı”. Of course the reason is that these letters are not existing in English. We tried to find out the right sounds for them from 55 recorded phonemes in RC8660 voice synthesizer. Table 7 shows the recorded phonemes for “ç, ö, ü, ı”. But for “ğ” we did not assign and use any phoneme, since “ğ” in Turkish has the only effect of lengthening the vowel preceding it.

Table 7 Implement Turkish Special Letters

Letter	Place in Turkish	Place in English	Words
Ğ	Ğ	Previous Vowel Extended	--
Ç	Ç	ch	church
Ö	Ö	er	bird
Ü	Ü	yy uh / uh	cure / book
I	I	ax	bottom

For this work, one of the challenges was the letter “ü”. The words starting and ending with “ü” in Turkish are not pronounced good enough with the “ü” sound paired to the one which is in the middle of the words in English language. Therefore, we used two phonemes for the “ü” sound. “yy uh” from “cure” is for “ü” taking place in mid syllables in Turkish words. “uh” is for words starting or ending with “ü” .

4.6 C# Graphical User Interface

In the thesis, we also designed a Graphical User Interface (GUI) for customizing the text input and TTS controls to communicate with RC8660. This GUI in Figure 13 allows the user to send any text without using RCStudio controls. The command syntax in C#:

<command character>[<number string>]<ASCII character>

is used to send the values set by the slider bars on the GUI to RC8660 voice synthesizer. The following scripts are for setting “speed” values.

```
private void button1 _Click( object sender, EventArgs e)
{
byte[] data = new byte[] {0x01};
serialPort1 . Write( data, 0 , data.Length) ;
serialPort1 . Write("0s") ;
}
```

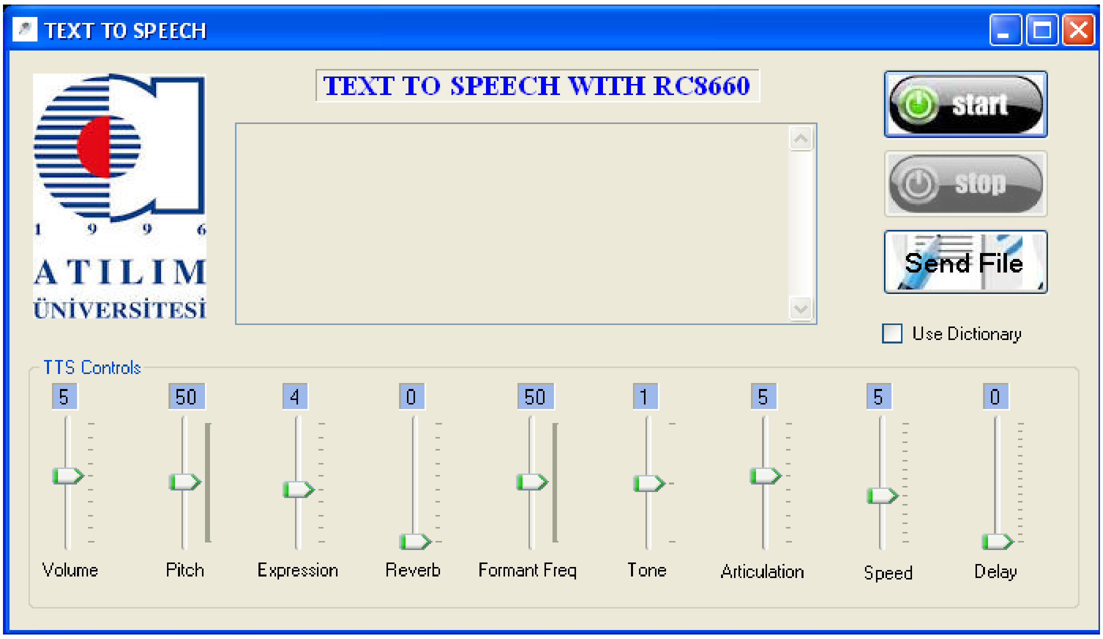

Figure 13 Text To Speech GUI With TTS Controls

4.7 RS232 Communication

For this thesis we used standart RS232 serial port communications with MAX3232 on V-POD. That is the way of communication between the computer and RC8660 directly via serial to USB converter.

RS232 is a popular way to transfer commands and data between PC and the microcontroller having UART interface. We have a typical example of sending string of numbers using this interface in Figure 14. In this example RS232 uses signals and hardware for communication between two devices. The character ‘A’ (ASCII:65 in decimal, 0x41 in Hex) is transferred from device A to Device B and appears on the line from TX to RX between devices.

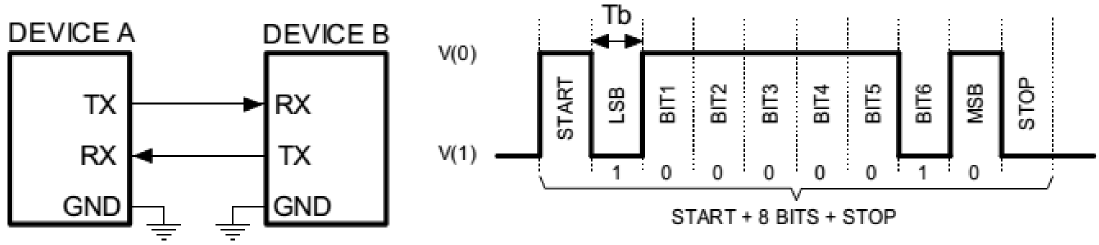

Figure 14 RS232 Communication Standart

CHAPTER 5

TESTS AND RESULTS

In this chapter, the speech synthesized by the RC8660 card is assessed by subjective tests. Effects of the prosodical adjustments are also evaluated. All the results of the experiments done with the card are presented herein.

In all the conducted tests, our TTS Controls were set as: Volume is 5, Speed is 2, Pitch is 40, Expression is 3, Reverb is 0, Formant Freq is 51, Tone is 1, Articulation is 5, and Delay is 0 as best settings.

5.1 Mean Opinion Score (MOS) and Sample Sentences

We use the Mean Opinion Score test to measure the performance of our TTS system. Mean Opinion Score test is a way for evaluating synthesized sentences subjectively. After choosing the sample test sentences, people listen to the sentences and they score the quality of uttered sentences between 1 and 5. The scores and their equivalents are given in Table 8. 1 is "very annoying" and 5 is "imperceptible".

Table 8 Mean Opinion Score (MOS)

MOS	Quality	Impairment
5	Excellent	Imperceptible
4	Good	Perceptible but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

The test statements and their MOS results are shown in Table 9. As seen, 21 utterances are listened to 20 people of age between 20-45. The average score of all the tests is 3.80 as seen in Table 10. To select the statements we especially paid attention to include all the special Turkish letters: "ü, ö, ğ, ç, ı".

Examining the individual scores in Table 9, the highest score is achieved at the statement: “Ramazan” which is 4.85, and the lowest score is achieved at the statement: “Anne sevgisi” as 1.55.

Table 9 Sample Sentences for Mean Opinion Score

No	Analyzed Sentences	MOS Score(1-5)
1	“Ali topu tutamadı.”	2.2
2	“Anne sevgisi.”	1.55
3	“Araba.”	4.2
4	“Asansör.”	4.5
5	“Bahçe kapısı.”	3.65
6	“Bu yemek çok tuzsuz”	3.35
7	“Bugün 15 Temmuz 2015”	4.3
8	“Bugün hava çok güzeldi”	4.65
9	“Çarşamba”	4.8
10	“Dokuz”	4.7
11	“Karadeniz yeşil bir bölgedir”	2.05
12	“Kitap okumayı severim”	3.95
13	“Meyveli yoğurt yedim”	3.2
14	“Nisan”	4.65
15	“Okulun bahçesi”	4.2
16	“Ramazan”	4.85
17	“Salı”	3.95
18	“Spor yapıyorlar mı?”	4.45
19	“Tatile gidemediniz mi?”	3.65
20	“Yedi”	2.75
21	“Yemek alabilir miyim?”	4.35

Table 10 Average of MOS

Number of Participants	Average MOS
20	3.8

According to the results of our tests, some of the words and sentences are unclear for audiences. Most of the testers did not understand “Ali Topu Tutamadı”, “Anne Sevgisi” and “Karadeniz Yeşil Bir Bölgedir” and they confused these statements with the ones given in Table 11.

Table 11 List of Confused Utterances

Test Sentences	Confused With
Anne Sevgisi	PC
Salı	Sağlık, sağla, sağa
Ali Topu Tutamadı	Ali
Yedi	Geri, Yeni, Deri
Karadeniz Yeşil Bir Bölgedir	Karadeniz
Yedim	Yerim, yedi

On the other hand, some of the statements got very high scores. For example, for the statement: “Bugün 15 Temmuz 2015”, listeners do not have any difficulty in understanding, since our TTS pronounced the numbers correctly.

5.2 The Effect of TTS Controls

Adjusting of the TTS Controls gives developers more power to make sounds much more clean and comprehensible for changing the prosody. In Figure 15 shown are the prosody control parameters of RCStudio. We conducted many tests to see the effects of those parameters on the quality of the pronounciation. Some of the resulting audio files after applying those effects are given below.

When TTS Controls values are default —> Bugun 14 Temmuz 2015

When Reverb Control changes —> Bugun 14 Temmuz 2015

When TTS Controls sets voice of Alvin —> Bugun 14 Temmuz 2015

When Delay Control changes —> Bugun 14 Temmuz 2015

When Speed Control changes —> Bugun 14 Temmuz 2015

Which is used for Our Study —> Bugun 14 Temmuz 2015

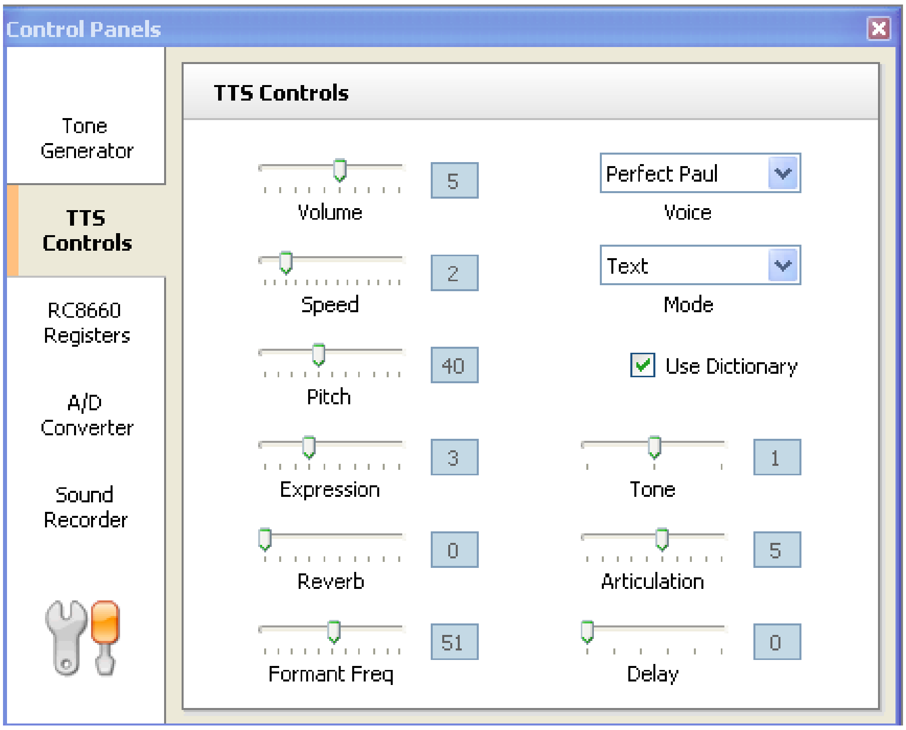

Figure 15 Settings of the TTS Controls

5.3 Prosodic Tests

The RC8660 has the features: Volume, Speed and Pitch that can manipulate the prosodic attributes of phonemes, as seen in Figure 16. The Pitch can be increased or decreased by m steps. Similarly, the Speed and Volume can also be increased or decreased by steps.

Symbol	Function	Equiv Cmd
nn	Set pitch to 'nn' (0-99)	*n*P
/	Increase pitch m steps *	+ *m*P
\	Decrease pitch m steps *	– *m*P
+	Increase speed 1 step	+1S
–	Decrease speed 1 step	–1S
>	Increase volume 1 step	+1V
<	Decrease volume 1 step	–1V

Figure 16 Phoneme Attribute Modifiers

While working on the Turkish Exception Dictionary, we realized that the phonemes alone corresponding to "p, ç, t, k" letters, especially in the middle and end of words, were not adequate for proper pronounciation. In order to fix this we increased the Volume of each phoneme.

In questions words ending with "….miyim?" the stress must be assigned to the question suffix. For example, in the statement: "Yemek alabilir miyim?", the stress at "miyim" is implemented by the command: "//m" which increases the pitch by 2 steps.

In Turkish, the stress of words is usually at the last syllable. Therefore we increased the pitch by 1 step at the last syllables of each word in "Bugün hava çok güzeldi".

CHAPTER 6
CONCLUSION

A Turkish Text to Speech Synthesis system is implemented by the RC8660 voice synthesizer in this thesis. We built an Exception Dictionary for Turkish language. We significantly exploited IPA to form an English to Turkish phoneme mapping. The letters specific to Turkish alphabet: “ç, ö, ü, ğ, ı” are also defined. The syllabic structure of RC8660 which is designed originally for English is adapted to Turkish syllabic structure by defining rules in the dictionary. The prosodic parameters of the RC8660 are extensively tested and some rules for Turkish prosody are defined. A promising performance can be said to be achieved by RC8660 considering the MOS results where the average is 3.80/5.00. To have a system independent from RCStudio, a GUI is designed in C# language.

As future work, this system can be integrated with other systems to realize many practical applications in Turkish such as: Robotics, e-book readers, ATM machines, talking pagers and PDAs, GPS navigation systems, remote diagnostic reporting, electronic test and measurement, aids for the orally or visually disabled.

REFERENCES

[1] İbrahim Baran USLU, “Konuşma İşleme ve Türkçenin Dilbilimsel Özelliklerini Kullanarak Metinden Doğal Konuşma Sentezleme”, Doktora Tezi, Ankara Üniversitesi Fen Bilimleri Enstitüsü, Mayıs 2012

[2] Thierry Dutoit, “An Introduction to Text-to-Speech Synthesis”, Kluwer Academic Publishers, visited on 1997

[3] http://en.wikipedia.org/wiki/English_phonology, visited on 7 March 2015

[4] http://en.wikipedia.org/wiki/Mean_opinion_score, visited on 7 March 2015

[5] http://www.phonicsontheweb.com/syllables.php, visited on 10 March 2015

[6] Çağlayan Erdem and Janez Stergar, “Adapting Prosody in a Text-to-Speech System”, Siemens Corporate Technology, University of Maribor, Faculty of Electrical Engineering and Computer Science, visited on 15 June 2015

[7] Eric Lewis and Mark Tatham, “Word and Syllable Concatenation in Text to Speech Synthesis”, University of Bristol and University of Essex, visited on 15 June 2015

[8] http://en.wikipedia.org/wiki/Help:IPA_for_English, visited on 10 March 2015

[9] http://en.wikipedia.org/wiki/Turkish_alphabet, visited on 7 March 2015

[10] http://tr.wikipedia.org/wiki/Diyakritik_i%C5%9Faretler, visited on 15 March 2015

[11] http://www.intea.hr/downloads/introductiontoserialcommunication.pdf, visited on 2015

[12] http://tr.wikipedia.org/wiki/Noktalama_i%C5%9Faretleri, visited on 15 March 2015

[13] http://www.phonicsontheweb.com/syllables.php, visited on 15 March 2015

[14] https://www.englishclub.com/pronunciation/word-stress-rules.htm, visited on 15 March 2015

[15] http://www.dyslexia-reading-well.com/44-phonemes-in-english.html, visited on 15 March 2015

[16] http://www.auburn.edu/academic/education/reading_genie/spellings.html, visited on 15 March 2015

[17] http://en.wikipedia.org/wiki/International_Phonetic_Alphabet_chart_for_English_dialects, visited on 5 March 2015

[18] http://www.ontrackreading.com/wordlists/multisyllable-words-by-vowel-sound, visited on 20 March 2015

[19] http://en.wikipedia.org/wiki/Pronunciation_respelling_for_English, visited on 20 March 2015

[20] http://www.really-learn-english.com/english-pronunciation-lesson-04-short-i-sound.html, visited on 20 March 2015

[21] http://en.wikipedia.org/wiki/Speech_Assessment_Methods_Phonetic_Alphabet_chart_for_English, visited on 15 March 2015

[22] http://en.wikipedia.org/wiki/International_Phonetic_Alphabet_chart_for_English_dialects, visited on 25 March 2015

[23] http://www.tdk.gov.tr/index.php?option=com_content&view=article&id=208:Hece -Yapisi-ve-Satir-Sonunda-Kelimelerin-Bolunmesi&catid=50:yazm-kurallar&Itemid=132, visited on 25 March 2015

[24] İlhami SEL, "Türkçe Metinler için Hece Tabanlı Metinden Konuşma Sentezleme Sistemi", Yüksek Lisans Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü, Temmuz 2013

[25] Ekrem GÜNER, "A Hybrid Statistical / Unit-Selection Text-To-Speech Synthesis System For Morphologically Rich Language", Master Thesis, Ozyegin University Graduate School of Sciences and Engineering, June 2013

[26] Abdulkadir KARACI, "Ses sentezleme ve tanıma teknolojilerini kullanarak Türkçenin ana dil olarak öğretimi için zeki öğretim sistemi geliştirilmesi", Doktora Tezi, Gazi Üniversitesi Bilişim Enstitüsü, 2013

[27] Amir MOHAMMADI, "Speaker Adaption with Minimal Data In Statistical Speech Synthesis Systems", Master Thesis, Ozyegin University Graduate School of Sciences and Engineering, September 2014

[28] http://retired.beyondlogic.org/serial/serial.pdf, visited on 5 March 2015

[29] http://www.rcsys.com/Downloads/rc8660.pdf, visited on 1 December 2014

[30] http://www.rcsys.com/Downloads/v-stamp.pdf, visited on 1 December 2014

[31] http://www.rcsys.com/Downloads/v-pod.pdf, visited on 1 December 2014

[32] www. http://www.rcsys.com, visited on 1 December 2014

[33] İclâl ERGENÇ, “Konuşma Dili ve Türkçenin Söyleyiş Sözlüğü”, Multiligual, 2002.

[34] Tuncay Şentürk and Eşref Adalı, “Türkçe Metin Seslendirme”, ITU Bilgisayar Mühendisliği Bölümü, 1 June 2015

APPENDIX A

TURKISH EXCEPTION DICTIONARY

A1: TURKISH SYLLABLE READING RULES

a(b)a=b aa	a(g)a=g aa	a(l)a=l aa	a(s)a=s aa
a(b)e=b eh	a(g)e=g eh	a(l)e=l eh	a(s)e=s eh
a(b)o=b ow	a(g)o=g ow	a(l)o=l ow	a(s)o=s ow
a(b)ö=b er	a(g)ö=g er	a(l)ö=l er	a(s)ö=s er
a(b)u=b uw	a(g)u=g uw	a(l)u=l uw	a(s)u=s uw
a(b)ü=b yy uh	a(g)ü=g yy uh	a(l)ü=l yy uh	a(s)ü=s yy uh
a(b)i=b ix	a(g)i=g ix	a(l)i=l ix	a(s)i=s ix
a(b)ı=b ax	a(g)ı=g ax	a(l)ı=l ax	a(s)ı=s ax
a(c)a=j aa	a(ğ)a= ----aa	a(m)a=m aa	a(ş)a=sh aa
a(c)e=j eh	a(ğ)e= ----eh	a(m)e=m eh	a(ş)e=sh eh
a(c)o=j ow	a(ğ)o= ----ow	a(m)o=m ow	a(ş)o=sh ow
a(c)ö=j er	a(ğ)ö= ----er	a(m)ö=m er	a(ş)ö=sh er
a(c)u=j uw	a(ğ)u= ----uw	a(m)u=m uw	a(ş)u=sh uw
a(c)ü=j yy uh	a(ğ)ü= ----yy uh	a(m)ü=m yy uh	a(ş)ü=sh yy uh
a(c)i=j ix	a(ğ)i= ----ix	a(m)i=m ix	a(ş)i=sh ix
a(c)ı=j ax	a(ğ)ı= ----ax	a(m)ı=m ax	a(ş)ı=sh ax
a(ç)a=ch aa	a(h)a=h aa	a(n)a=n aa	a(t)a=t aa
a(ç)e=ch eh	a(h)e=h eh	a(n)e=n eh	a(t)e=t eh
a(ç)o=ch ow	a(h)o=h ow	a(n)o=n ow	a(t)o=t ow
a(ç)ö=ch er	a(h)ö=h er	a(n)ö=n er	a(t)ö=t er
a(ç)u=ch uw	a(h)u=h uw	a(n)u=n uw	a(t)u=t uw
a(ç)ü=ch yy uh	a(h)ü=h yy uh	a(n)ü=n yy uh	a(t)ü=t yy yy uh
a(ç)i=ch ix	a(h)i=h ix	a(n)i=n ix	a(t)i=t ix
a(ç)ı=ch ax	a(h)ı=h ax	a(n)ı=n ax	a(t)ı=t ax
a(d)a=d aa	a(j)a=zh aa	a(p)a=p aa	a(v)a=v aa
a(d)e=d eh	a(j)e=zh eh	a(p)e=p eh	a(v)e=v eh
a(d)o=d ow	a(j)o=zh ow	a(p)o=p ow	a(v)o=v ow
a(d)ö=d er	a(j)ö=zh er	a(p)ö=p er	a(v)ö=v er
a(d)u=d uw	a(j)u=zh uw	a(p)u=p uw	a(v)u=v uw
a(d)ü=d yy uh	a(j)ü=zh yy uh	a(p)ü=p yy uh	a(v)ü=v yy uh
a(d)i=d ix	a(j)i=zh ix	a(p)i=p ix	a(v)i=v ix
a(d)ı=d ax	a(j)ı=zh ax	a(p)ı=p ax	a(v)ı=v ax
a(f)a=f aa	a(k)a=k aa	a(r)a=r aa	a(y)a=y
a(f)e=f eh	a(k)e=k eh	a(r)e=r eh	a(y)e=y
a(f)o=f ow	a(k)o=k ow	a(r)o=r ow	a(y)o=y
a(f)ö=f er	a(k)ö=k er	a(r)ö=r er	a(y)ö=y
a(f)u=f uw	a(k)u=k uw	a(r)u=r uw	a(y)u=y
a(f)ü=f yy uh	a(k)ü=k yy uh	a(r)ü=r yy uh	a(y)ü=y
a(f)i=f ix	a(k)i=k ix	a(r)i=r ix	a(y)i=y
a(f)ı=f ax	a(k)ı=k ax	a(r)ı=r ax	a(y)ı=y
a(z)a=z aa	a(z)a=z aa	a(z)a=z aa	a(z)a=z aa

a(z)e=z eh	a(z)e=z eh	a(z)e=z eh	a(z)e=z eh
e(b)a=b aa	e(g)a=g aa	e(l)a=l aa	e(s)a=s aa
e(b)e=b eh	e(g)e=g eh	e(l)e=l eh	e(s)e=s eh
e(b)o=b ow	e(g)o=g ow	e(l)o=l ow	e(s)o=s ow
e(b)ö=b er	e(g)ö=g er	e(l)ö=l er	e(s)ö=s er
e(b)u=b uw	e(g)u=g uw	e(l)u=l uw	e(s)u=s uw
e(b)ü=b yy uh	e(g)ü=g yy uh	e(l)ü=l yy uh	e(s)ü=s yy uh
e(b)i=b ix	e(g)i=g ix	e(l)i=l ix	e(s)i=s ix
e(b)ı=b ax	e(g)ı=g ax	e(l)ı=l ax	e(s)ı=s ax
e(c)a=j aa	e(ğ)a=----aa	e(m)a=m aa	e(ş)a=sh aa
e(c)e=j eh	e(ğ)e=----eh	e(m)e=m eh	e(ş)e=sh eh
e(c)o=j ow	e(ğ)o=----ow	e(m)o=m ow	e(ş)o=sh ow
e(c)ö=j er	e(ğ)ö=----er	e(m)ö=m er	e(ş)ö=sh er
e(c)u=j uw	e(ğ)u=----uw	e(m)u=m uw	e(ş)u=sh uw
e(c)ü=j yy uh	e(ğ)ü=----yy uh	e(m)ü=m yy uh	e(ş)ü=sh yy uh
e(c)i=j ix	e(ğ)i=----ix	e(m)i=m ix	e(ş)i=sh ix
e(c)ı=j ax	e(ğ)ı=----ax	e(m)ı=m ax	e(ş)ı=sh ax
e(ç)a=ch aa	e(h)a=h aa	e(n)a=n aa	e(t)a=t aa
e(ç)e=ch eh	e(h)e=h eh	e(n)e=n eh	e(t)e=t eh
e(ç)o=ch ow	e(h)o=h ow	e(n)o=n ow	e(t)o=t ow
e(ç)ö=ch er	e(h)ö=h er	e(n)ö=n er	e(t)ö=t er
e(ç)u=ch uw	e(h)u=h uw	e(n)u=n uw	e(t)u=t uw
e(ç)ü=ch yy uh	e(h)ü=h yy uh	e(n)ü=n yy uh	e(t)ü=t yy uh
e(ç)i=ch ix	e(h)i=h ix	e(n)i=n ix	e(t)i=t ix
e(ç)ı=ch ax	e(h)ı=h ax	e(n)ı=n ax	e(t)ı=t ax
e(d)a=d aa	e(j)a=zh aa	e(p)a=p aa	e(v)a=v aa
e(d)e=d eh	e(j)e=zh eh	e(p)e=p eh	e(v)e=v eh
e(d)o=d ow	e(j)o=zh ow	e(p)o=p ow	e(v)o=v ow
e(d)ö=d er	e(j)ö=zh er	e(p)ö=p er	e(v)ö=v er
e(d)u=d uw	e(j)u=zh uw	e(p)u=p uw	e(v)u=v uw
e(d)ü=d yy uh	e(j)ü=zh yy uh	e(p)ü=p yy uh	e(v)ü=v yy uh
e(d)i=d ix	e(j)i=zh ix	e(p)i=p ix	e(v)i=v ix
e(d)ı=d ax	e(j)ı=zh ax	e(p)ı=p ax	e(v)ı=v ax
e(f)a=f aa	e(k)a=k aa	e(r)a=r aa	e(y)a=y
e(f)e=f eh	e(k)e=k eh	e(r)e=r eh	e(y)e=y
e(f)o=f ow	e(k)o=k ow	e(r)o=r ow	e(y)o=y
e(f)ö=f er	e(k)ö=k er	e(r)ö=r er	e(y)ö=y
e(f)u=f uw	e(k)u=k uw	e(r)u=r uw	e(y)u=y
e(f)ü=f yy uh	e(k)ü=k yy uh	e(r)ü=r yy uh	e(y)ü=y
e(f)i=f ix	e(k)i=k ix	e(r)i=r ix	e(y)i=y
e(f)ı=f ax	e(k)ı=k ax	e(r)ı=r ax	e(y)ı=y
e(z)a=z aa	e(z)a=z aa	e(z)a=z aa	e(z)a=z aa
e(z)e=z eh	e(z)e=z eh	e(z)e=z eh	e(z)e=z eh

i(b)a=b aa	i(g)a=g aa	i(l)a=l aa	i(s)a=s aa
i(b)e=b eh	i(g)e=g eh	i(l)e=l eh	i(s)e=s eh
i(b)o=b ow	i(g)o=g ow	i(l)o=l ow	i(s)o=s ow
i(b)ö=b er	i(g)ö=g er	i(l)ö=l er	i(s)ö=s er
i(b)u=b uw	i(g)u=g uw	i(l)u=l uw	i(s)u=s uw
i(b)ü=b yy uh	i(g)ü=g yy uh	i(l)ü=l yy uh	i(s)ü=s yy uh
i(b)i=b ix	i(g)i=g ix	i(l)i=l ix	i(s)i=s ix
i(b)ı=b ax	i(g)ı=g ax	i(l)ı=l ax	i(s)ı=s ax
i(c)a=j aa	i(ğ)a=----aa	i(m)a=m aa	i(ş)a=sh aa
i(c)e=j eh	i(ğ)e=----eh	i(m)e=m eh	i(ş)e=sh eh
i(c)o=j ow	i(ğ)o=----ow	i(m)o=m ow	i(ş)o=sh ow
i(c)ö=j er	i(ğ)ö=----er	i(m)ö=m er	i(ş)ö=sh er
i(c)u=j uw	i(ğ)u=----uw	i(m)u=m uw	i(ş)u=sh uw
i(c)ü=j yy uh	i(ğ)ü=----yy uh	i(m)ü=m yy uh	i(ş)ü=sh yy uh
i(c)i=j ix	i(ğ)i=----ix	i(m)i=m ix	i(ş)i=sh ix
i(c)ı=j ax	i(ğ)ı=----ax	i(m)ı=m ax	i(ş)ı=sh ax
i(ç)a=ch aa	i(h)a=h aa	i(n)a=n aa	i(t)a=t aa
i(ç)e=ch eh	i(h)e=h eh	i(n)e=n eh	i(t)e=t eh
i(ç)o=ch ow	i(h)o=h ow	i(n)o=n ow	i(t)o=t ow
i(ç)ö=ch er	i(h)ö=h er	i(n)ö=n er	i(t)ö=t er
i(ç)u=ch uw	i(h)u=h uw	i(n)u=n uw	i(t)u=t uw
i(ç)ü=ch yy uh	i(h)ü=h yy uh	i(n)ü=n yy uh	i(t)ü=t yy uh
i(ç)i=ch ix	i(h)i=h ix	i(n)i=n ix	i(t)i=t ix
i(ç)ı=ch ax	i(h)ı=h ax	i(n)ı=n ax	i(t)ı=t ax
i(d)a=d aa	i(j)a=zh aa	i(p)a=p aa	i(v)a=v aa
i(d)e=d eh	i(j)e=zh eh	i(p)e=p eh	i(v)e=v eh
i(d)o=d ow	i(j)o=zh ow	i(p)o=p ow	i(v)o=v ow
i(d)ö=d er	i(j)ö=zh er	i(p)ö=p er	i(v)ö=v er
i(d)u=d uw	i(j)u=zh uw	i(p)u=p uw	i(v)u=v uw
i(d)ü=d yy uh	i(j)ü=zh yy uh	i(p)ü=p yy uh	i(v)ü=v yy uh
i(d)i=d ix	i(j)i=zh ix	i(p)i=p ix	i(v)i=v ix
i(d)ı=d ax	i(j)ı=zh ax	i(p)ı=p ax	i(v)ı=v ax
i(f)a=f aa	i(k)a=k aa	i(r)a=r aa	i(y)a=y
i(f)e=f eh	i(k)e=k eh	i(r)e=r eh	i(y)e=y
i(f)o=f ow	i(k)o=k ow	i(r)o=r ow	i(y)o=y
i(f)ö=f er	i(k)ö=k er	i(r)ö=r er	i(y)ö=y
i(f)u=f uw	i(k)u=k uw	i(r)u=r uw	i(y)u=y
i(f)ü=f yy uh	i(k)ü=k yy uh	i(r)ü=r yy uh	i(y)ü=y
i(f)i=f ix	i(k)i=k ix	i(r)i=r ix	i(y)i=y
i(f)ı=f ax	i(k)ı=k ax	i(r)ı=r ax	i(y)ı=y
i(z)a=z aa	i(z)a=z aa	i(z)a=z aa	i(z)a=z aa
i(z)e=z eh	i(z)e=z eh	i(z)e=z eh	i(z)e=z eh

ı(b)a=b aa	ı(g)a=g aa	ı(l)a=l aa	ı(s)a=s aa
ı(b)e=b eh	ı(g)e=g eh	ı(l)e=l eh	ı(s)e=s eh
ı(b)o=b ow	ı(g)o=g ow	ı(l)o=l ow	ı(s)o=s ow
ı(b)ö=b er	ı(g)ö=g er	ı(l)ö=l er	ı(s)ö=s er
ı(b)u=b uw	ı(g)u=g uw	ı(l)u=l uw	ı(s)u=s uw
ı(b)ü=b yy uh	ı(g)ü=g yy uh	ı(l)ü=l yy uh	ı(s)ü=s yy uh
ı(b)i=b ix	ı(g)i=g ix	ı(l)i=l ix	ı(s)i=s ix
ı(b)ı=b ax	ı(g)ı=g ax	ı(l)ı=l ax	ı(s)ı=s ax
ı(c)a=j aa	ı(ğ)a=----aa	ı(m)a=m aa	ı(ş)a=sh aa
ı(c)e=j eh	ı(ğ)e=----eh	ı(m)e=m eh	ı(ş)e=sh eh
ı(c)o=j ow	ı(ğ)o=----ow	ı(m)o=m ow	ı(ş)o=sh ow
ı(c)ö=j er	ı(ğ)ö=----er	ı(m)ö=m er	ı(ş)ö=sh er
ı(c)u=j uw	ı(ğ)u=----uw	ı(m)u=m uw	ı(ş)u=sh uw
ı(c)ü=j yy uh	ı(ğ)ü=----yy uh	ı(m)ü=m yy uh	ı(ş)ü=sh yy uh
ı(c)i=j ix	ı(ğ)i=----ix	ı(m)i=m ix	ı(ş)i=sh ix
ı(c)ı=j ax	ı(ğ)ı=----ax	ı(m)ı=m ax	ı(ş)ı=sh ax
ı(ç)a=ch aa	ı(h)a=h aa	ı(n)a=n aa	ı(t)a=t aa
ı(ç)e=ch eh	ı(h)e=h eh	ı(n)e=n eh	ı(t)e=t eh
ı(ç)o=ch ow	ı(h)o=h ow	ı(n)o=n ow	ı(t)o=t ow
ı(ç)ö=ch er	ı(h)ö=h er	ı(n)ö=n er	ı(t)ö=t er
ı(ç)u=ch uw	ı(h)u=h uw	ı(n)u=n uw	ı(t)u=t uw
ı(ç)ü=ch yy uh	ı(h)ü=h yy uh	ı(n)ü=n yy uh	ı(t)ü=t yy uh
ı(ç)i=ch ix	ı(h)i=h ix	ı(n)i=n ix	ı(t)i=t ix
ı(ç)ı=ch ax	ı(h)ı=h ax	ı(n)ı=n ax	ı(t)ı=t ax
ı(d)a=d aa	ı(j)a=zh aa	ı(p)a=p aa	ı(v)a=v aa
ı(d)e=d eh	ı(j)e=zh eh	ı(p)e=p eh	ı(v)e=v eh
ı(d)o=d ow	ı(j)o=zh ow	ı(p)o=p ow	ı(v)o=v ow
ı(d)ö=d er	ı(j)ö=zh er	ı(p)ö=p er	ı(v)ö=v er
ı(d)u=d uw	ı(j)u=zh uw	ı(p)u=p uw	ı(v)u=v uw
ı(d)ü=d yy uh	ı(j)ü=zh yy uh	ı(p)ü=p yy uh	ı(v)ü=v yy uh
ı(d)i=d ix	ı(j)i=zh ix	ı(p)i=p ix	ı(v)i=v ix
ı(d)ı=d ax	ı(j)ı=zh ax	ı(p)ı=p ax	ı(v)ı=v ax
ı(f)a=f aa	ı(k)a=k aa	ı(r)a=r aa	ı(y)a=y
ı(f)e=f eh	ı(k)e=k eh	ı(r)e=r eh	ı(y)e=y
ı(f)o=f ow	ı(k)o=k ow	ı(r)o=r ow	ı(y)o=y
ı(f)ö=f er	ı(k)ö=k er	ı(r)ö=r er	ı(y)ö=y
ı(f)u=f uw	ı(k)u=k uw	ı(r)u=r uw	ı(y)u=y
ı(f)ü=f yy uh	ı(k)ü=k yy uh	ı(r)ü=r yy uh	ı(y)ü=y
ı(f)i=f ix	ı(k)i=k ix	ı(r)i=r ix	ı(y)i=y
ı(f)ı=f ax	ı(k)ı=k ax	ı(r)ı=r ax	ı(y)ı=y
ı(z)a=z aa	ı(z)a=z aa	ı(z)a=z aa	ı(z)a=z aa
ı(z)e=z eh	ı(z)e=z eh	ı(z)e=z eh	ı(z)e=z eh

o(b)a=b aa	o(g)a=g aa	o(l)a=l aa	o(s)a=s aa
o(b)e=b eh	o(g)e=g eh	o(l)e=l eh	o(s)e=s eh
o(b)o=b ow	o(g)o=g ow	o(l)o=l ow	o(s)o=s ow
o(b)ö=b er	o(g)ö=g er	o(l)ö=l er	o(s)ö=s er
o(b)u=b uw	o(g)u=g uw	o(l)u=l uw	o(s)u=s uw
o(b)ü=b yy uh	o(g)ü=g yy uh	o(l)ü=l yy uh	o(s)ü=s yy uh
o(b)i=b ix	o(g)i=g ix	o(l)i=l ix	o(s)i=s ix
o(b)ı=b ax	o(g)ı=g ax	o(l)ı=l ax	o(s)ı=s ax
o(c)a=j aa	o(ğ)a=----aa	o(m)a=m aa	o(ş)a=sh aa
o(c)e=j eh	o(ğ)e=----eh	o(m)e=m eh	o(ş)e=sh eh
o(c)o=j ow	o(ğ)o=----ow	o(m)o=m ow	o(ş)o=sh ow
o(c)ö=j er	o(ğ)ö=----er	o(m)ö=m er	o(ş)ö=sh er
o(c)u=j uw	o(ğ)u=----uw	o(m)u=m uw	o(ş)u=sh uw
o(c)ü=j yy uh	o(ğ)ü=----yy uh	o(m)ü=m yy uh	o(ş)ü=sh yy uh
o(c)i=j ix	o(ğ)i=----ix	o(m)i=m ix	o(ş)i=sh ix
o(c)ı=j ax	o(ğ)ı=----ax	o(m)ı=m ax	o(ş)ı=sh ax
o(ç)a=ch aa	o(h)a=h aa	o(n)a=n aa	o(t)a=t aa
o(ç)e=ch eh	o(h)e=h eh	o(n)e=n eh	o(t)e=t eh
o(ç)o=ch ow	o(h)o=h ow	o(n)o=n ow	o(t)o=t ow
o(ç)ö=ch er	o(h)ö=h er	o(n)ö=n er	o(t)ö=t er
o(ç)u=ch uw	o(h)u=h uw	o(n)u=n uw	o(t)u=t uw
o(ç)ü=ch yy uh	o(h)ü=h yy uh	o(n)ü=n yy uh	o(t)ü=t yy uh
o(ç)i=ch ix	o(h)i=h ix	o(n)i=n ix	o(t)i=t ix
o(ç)ı=ch ax	o(h)ı=h ax	o(n)ı=n ax	o(t)ı=t ax
o(d)a=d aa	o(j)a=zh aa	o(p)a=p aa	o(v)a=v aa
o(d)e=d eh	o(j)e=zh eh	o(p)e=p eh	o(v)e=v eh
o(d)o=d ow	o(j)o=zh ow	o(p)o=p ow	o(v)o=v ow
o(d)ö=d er	o(j)ö=zh er	o(p)ö=p er	o(v)ö=v er
o(d)u=d uw	o(j)u=zh uw	o(p)u=p uw	o(v)u=v uw
o(d)ü=d yy uh	o(j)ü=zh yy uh	o(p)ü=p yy uh	o(v)ü=v yy uh
o(d)i=d ix	o(j)i=zh ix	o(p)i=p ix	o(v)i=v ix
o(d)ı=d ax	o(j)ı=zh ax	o(p)ı=p ax	o(v)ı=v ax
o(f)a=f aa	o(k)a=k aa	o(r)a=r aa	o(y)a=y
o(f)e=f eh	o(k)e=k eh	o(r)e=r eh	o(y)e=y
o(f)o=f ow	o(k)o=k ow	o(r)o=r ow	o(y)o=y
o(f)ö=f er	o(k)ö=k er	o(r)ö=r er	o(y)ö=y
o(f)u=f uw	o(k)u=k uw	o(r)u=r uw	o(y)u=y
o(f)ü=f yy uh	o(k)ü=k yy uh	o(r)ü=r yy uh	o(y)ü=y
o(f)i=f ix	o(k)i=k ix	o(r)i=r ix	o(y)i=y
o(f)ı=f ax	o(k)ı=k ax	o(r)ı=r ax	o(y)ı=y
o(z)a=z aa	o(z)a=z aa	o(z)a=z aa	o(z)a=z aa
o(z)e=z eh	o(z)e=z eh	o(z)e=z eh	o(z)e=z eh

ö(b)a=b aa	ö(g)a=g aa	ö(l)a=l aa	ö(s)a=s aa
ö(b)e=b eh	ö(g)e=g eh	ö(l)e=l eh	ö(s)e=s eh
ö(b)o=b ow	ö(g)o=g ow	ö(l)o=l ow	ö(s)o=s ow
ö(b)ö=b er	ö(g)ö=g er	ö(l)ö=l er	ö(s)ö=s er
ö(b)u=b uw	ö(g)u=g uw	ö(l)u=l uw	ö(s)u=s uw
ö(b)ü=b yy uh	ö(g)ü=g yy uh	ö(l)ü=l yy uh	ö(s)ü=s yy uh
ö(b)i=b ix	ö(g)i=g ix	ö(l)i=l ix	ö(s)i=s ix
ö(b)ı=b ax	ö(g)ı=g ax	ö(l)ı=l ax	ö(s)ı=s ax
ö(c)a=j aa	ö(ğ)a=----aa	ö(m)a=m aa	ö(ş)a=sh aa
ö(c)e=j eh	ö(ğ)e=----eh	ö(m)e=m eh	ö(ş)e=sh eh
ö(c)o=j ow	ö(ğ)o=----ow	ö(m)o=m ow	ö(ş)o=sh ow
ö(c)ö=j er	ö(ğ)ö=----er	ö(m)ö=m er	ö(ş)ö=sh er
ö(c)u=j uw	ö(ğ)u=----uw	ö(m)u=m uw	ö(ş)u=sh uw
ö(c)ü=j yy uh	ö(ğ)ü=----yy uh	ö(m)ü=m yy uh	ö(ş)ü=sh yy uh
ö(c)i=j ix	ö(ğ)i=----ix	ö(m)i=m ix	ö(ş)i=sh ix
ö(c)ı=j ax	ö(ğ)ı=----ax	ö(m)ı=m ax	ö(ş)ı=sh ax
ö(ç)a=ch aa	ö(h)a=h aa	ö(n)a=n aa	ö(t)a=t aa
ö(ç)e=ch eh	ö(h)e=h eh	ö(n)e=n eh	ö(t)e=t eh
ö(ç)o=ch ow	ö(h)o=h ow	ö(n)o=n ow	ö(t)o=t ow
ö(ç)ö=ch er	ö(h)ö=h er	ö(n)ö=n er	ö(t)ö=t er
ö(ç)u=ch uw	ö(h)u=h uw	ö(n)u=n uw	ö(t)u=t uw
ö(ç)ü=ch yy uh	ö(h)ü=h yy uh	ö(n)ü=n yy uh	ö(t)ü=t yy uh
ö(ç)i=ch ix	ö(h)i=h ix	ö(n)i=n ix	ö(t)i=t ix
ö(ç)ı=ch ax	ö(h)ı=h ax	ö(n)ı=n ax	ö(t)ı=t ax
ö(d)a=d aa	ö(j)a=zh aa	ö(p)a=p aa	ö(v)a=v aa
ö(d)e=d eh	ö(j)e=zh eh	ö(p)e=p eh	ö(v)e=v eh
ö(d)o=d ow	ö(j)o=zh ow	ö(p)o=p ow	ö(v)o=v ow
ö(d)ö=d er	ö(j)ö=zh er	ö(p)ö=p er	ö(v)ö=v er
ö(d)u=d uw	ö(j)u=zh uw	ö(p)u=p uw	ö(v)u=v uw
ö(d)ü=d yy uh	ö(j)ü=zh yy uh	ö(p)ü=p yy uh	ö(v)ü=v yy uh
ö(d)i=d ix	ö(j)i=zh ix	ö(p)i=p ix	ö(v)i=v ix
ö(d)ı=d ax	ö(j)ı=zh ax	ö(p)ı=p ax	ö(v)ı=v ax
ö(f)a=f aa	ö(k)a=k aa	ö(r)a=r aa	ö(y)a=y
ö(f)e=f eh	ö(k)e=k eh	ö(r)e=r eh	ö(y)e=y
ö(f)o=f ow	ö(k)o=k ow	ö(r)o=r ow	ö(y)o=y
ö(f)ö=f er	ö(k)ö=k er	ö(r)ö=r er	ö(y)ö=y
ö(f)u=f uw	ö(k)u=k uw	ö(r)u=r uw	ö(y)u=y
ö(f)ü=f yy uh	ö(k)ü=k yy uh	ö(r)ü=r yy uh	ö(y)ü=y
ö(f)i=f ix	ö(k)i=k ix	ö(r)i=r ix	ö(y)i=y
ö(f)ı=f ax	ö(k)ı=k ax	ö(r)ı=r ax	ö(y)ı=y
ö(z)a=z aa	ö(z)a=z aa	ö(z)a=z aa	ö(z)a=z aa
ö(z)e=z eh	ö(z)e=z eh	ö(z)e=z eh	ö(z)e=z eh

u(b)a=b aa	u(g)a=g aa	u(l)a=l aa	u(s)a=s aa
u(b)e=b eh	u(g)e=g eh	u(l)e=l eh	u(s)e=s eh
u(b)o=b ow	u(g)o=g ow	u(l)o=l ow	u(s)o=s ow
u(b)ö=b er	u(g)ö=g er	u(l)ö=l er	u(s)ö=s er
u(b)u=b uw	u(g)u=g uw	u(l)u=l uw	u(s)u=s uw
u(b)ü=b yy uh	u(g)ü=g yy uh	u(l)ü=l yy uh	u(s)ü=s yy uh
u(b)i=b ix	u(g)i=g ix	u(l)i=l ix	u(s)i=s ix
u(b)ı=b ax	u(g)ı=g ax	u(l)ı=l ax	u(s)ı=s ax
u(c)a=j aa	u(ğ)a=----aa	u(m)a=m aa	u(ş)a=sh aa
u(c)e=j eh	u(ğ)e=----eh	u(m)e=m eh	u(ş)e=sh eh
u(c)o=j ow	u(ğ)o=----ow	u(m)o=m ow	u(ş)o=sh ow
u(c)ö=j er	u(ğ)ö=----er	u(m)ö=m er	u(ş)ö=sh er
u(c)u=j uw	u(ğ)u=----uw	u(m)u=m uw	u(ş)u=sh uw
u(c)ü=j yy uh	u(ğ)ü=----yy uh	u(m)ü=m yy uh	u(ş)ü=sh yy uh
u(c)i=j ix	u(ğ)i=----ix	u(m)i=m ix	u(ş)i=sh ix
u(c)ı=j ax	u(ğ)ı=----ax	u(m)ı=m ax	u(ş)ı=sh ax
u(ç)a=ch aa	u(h)a=h aa	u(n)a=n aa	u(t)a=t aa
u(ç)e=ch eh	u(h)e=h eh	u(n)e=n eh	u(t)e=t eh
u(ç)o=ch ow	u(h)o=h ow	u(n)o=n ow	u(t)o=t ow
u(ç)ö=ch er	u(h)ö=h er	u(n)ö=n er	u(t)ö=t er
u(ç)u=ch uw	u(h)u=h uw	u(n)u=n uw	u(t)u=t uw
u(ç)ü=ch yy uh	u(h)ü=h yy uh	u(n)ü=n yy uh	u(t)ü=t yy uh
u(ç)i=ch ix	u(h)i=h ix	u(n)i=n ix	u(t)i=t ix
u(ç)ı=ch ax	u(h)ı=h ax	u(n)ı=n ax	u(t)ı=t ax
u(d)a=d aa	u(j)a=zh aa	u(p)a=p aa	u(v)a=v aa
u(d)e=d eh	u(j)e=zh eh	u(p)e=p eh	u(v)e=v eh
u(d)o=d ow	u(j)o=zh ow	u(p)o=p ow	u(v)o=v ow
u(d)ö=d er	u(j)ö=zh er	u(p)ö=p er	u(v)ö=v er
u(d)u=d uw	u(j)u=zh uw	u(p)u=p uw	u(v)u=v uw
u(d)ü=d yy uh	u(j)ü=zh yy uh	u(p)ü=p yy uh	u(v)ü=v yy uh
u(d)i=d ix	u(j)i=zh ix	u(p)i=p ix	u(v)i=v ix
u(d)ı=d ax	u(j)ı=zh ax	u(p)ı=p ax	u(v)ı=v ax
u(f)a=f aa	u(k)a=k aa	u(r)a=r aa	u(y)a=y
u(f)e=f eh	u(k)e=k eh	u(r)e=r eh	u(y)e=y
u(f)o=f ow	u(k)o=k ow	u(r)o=r ow	u(y)o=y
u(f)ö=f er	u(k)ö=k er	u(r)ö=r er	u(y)ö=y
u(f)u=f uw	u(k)u=k uw	u(r)u=r uw	u(y)u=y
u(f)ü=f yy uh	u(k)ü=k yy uh	u(r)ü=r yy uh	u(y)ü=y
u(f)i=f ix	u(k)i=k ix	u(r)i=r ix	u(y)i=y
u(f)ı=f ax	u(k)ı=k ax	u(r)ı=r ax	u(y)ı=y
u(z)a=z aa	u(z)a=z aa	u(z)a=z aa	u(z)a=z aa
u(z)e=z eh	u(z)e=z eh	u(z)e=z eh	u(z)e=z eh

ü(b)a=b aa	ü(g)a=g aa	ü(l)a=l aa	ü(s)a=s aa
ü(b)e=b eh	ü(g)e=g eh	ü(l)e=l eh	ü(s)e=s eh
ü(b)o=b ow	ü(g)o=g ow	ü(l)o=l ow	ü(s)o=s ow
ü(b)ö=b er	ü(g)ö=g er	ü(l)ö=l er	ü(s)ö=s er
ü(b)u=b uw	ü(g)u=g uw	ü(l)u=l uw	ü(s)u=s uw
ü(b)ü=b yy uh	ü(g)ü=g yy uh	ü(l)ü=l yy uh	ü(s)ü=s yy uh
ü(b)i=b ix	ü(g)i=g ix	ü(l)i=l ix	ü(s)i=s ix
ü(b)ı=b ax	ü(g)ı=g ax	ü(l)ı=l ax	ü(s)ı=s ax
ü(c)a=j aa	ü(ğ)a=----aa	ü(m)a=m aa	ü(ş)a=sh aa
ü(c)e=j eh	ü(ğ)e=----eh	ü(m)e=m eh	ü(ş)e=sh eh
ü(c)o=j ow	ü(ğ)o=----ow	ü(m)o=m ow	ü(ş)o=sh ow
ü(c)ö=j er	ü(ğ)ö=----er	ü(m)ö=m er	ü(ş)ö=sh er
ü(c)u=j uw	ü(ğ)u=----uw	ü(m)u=m uw	ü(ş)u=sh uw
ü(c)ü=j yy uh	ü(ğ)ü=----yy uh	ü(m)ü=m yy uh	ü(ş)ü=sh yy uh
ü(c)i=j ix	ü(ğ)i=----ix	ü(m)i=m ix	ü(ş)i=sh ix
ü(c)ı=j ax	ü(ğ)ı=----ax	ü(m)ı=m ax	ü(ş)ı=sh ax
ü(ç)a=ch aa	ü(h)a=h aa	ü(n)a=n aa	ü(t)a=t aa
ü(ç)e=ch eh	ü(h)e=h eh	ü(n)e=n eh	ü(t)e=t eh
ü(ç)o=ch ow	ü(h)o=h ow	ü(n)o=n ow	ü(t)o=t ow
ü(ç)ö=ch er	ü(h)ö=h er	ü(n)ö=n er	ü(t)ö=t er
ü(ç)u=ch uw	ü(h)u=h uw	ü(n)u=n uw	ü(t)u=t uw
ü(ç)ü=ch yy uh	ü(h)ü=h yy uh	ü(n)ü=n yy uh	ü(t)ü=t yy uh
ü(ç)i=ch ix	ü(h)i=h ix	ü(n)i=n ix	ü(t)i=t ix
ü(ç)ı=ch ax	ü(h)ı=h ax	ü(n)ı=n ax	ü(t)ı=t ax
ü(d)a=d aa	ü(j)a=zh aa	ü(p)a=p aa	ü(v)a=v aa
ü(d)e=d eh	ü(j)e=zh eh	ü(p)e=p eh	ü(v)e=v eh
ü(d)o=d ow	ü(j)o=zh ow	ü(p)o=p ow	ü(v)o=v ow
ü(d)ö=d er	ü(j)ö=zh er	ü(p)ö=p er	ü(v)ö=v er
ü(d)u=d uw	ü(j)u=zh uw	ü(p)u=p uw	ü(v)u=v uw
ü(d)ü=d yy uh	ü(j)ü=zh yy uh	ü(p)ü=p yy uh	ü(v)ü=v yy uh
ü(d)i=d ix	ü(j)i=zh ix	ü(p)i=p ix	ü(v)i=v ix
ü(d)ı=d ax	ü(j)ı=zh ax	ü(p)ı=p ax	ü(v)ı=v ax
ü(f)a=f aa	ü(k)a=k aa	ü(r)a=r aa	ü(y)a=y
ü(f)e=f eh	ü(k)e=k eh	ü(r)e=r eh	ü(y)e=y
ü(f)o=f ow	ü(k)o=k ow	ü(r)o=r ow	ü(y)o=y
ü(f)ö=f er	ü(k)ö=k er	ü(r)ö=r er	ü(y)ö=y
ü(f)u=f uw	ü(k)u=k uw	ü(r)u=r uw	ü(y)u=y
ü(f)ü=f yy uh	ü(k)ü=k yy uh	ü(r)ü=r yy uh	ü(y)ü=y
ü(f)i=f ix	ü(k)i=k ix	ü(r)i=r ix	ü(y)i=y
ü(f)ı=f ax	ü(k)ı=k ax	ü(r)ı=r ax	ü(y)ı=y
ü(z)a=z aa	ü(z)a=z aa	ü(z)a=z aa	ü(z)a=z aa
ü(z)e=z eh	ü(z)e=z eh	ü(z)e=z eh	ü(z)e=z eh

A2: ABBREVIATIONS

(mb)=m eh g ax b ay t
(KW)=K IH L AH W AA T
(DR)=D O K T O R
(TV)=T EH L AX V IH ZH IX N
(AV)=aa v uw k aa t
$(müh)$=m yy uh h eh n d ix s
(DOC)=d o ch eh n t
$(PROF)$=p r o f eh s er r
(DT)=d ix sh ch ix
(SOK)=s o k aa k
(MAH)=m aa h aa l l eh s ix
(CAD)=j aa d d eh s ix
(KG)=k ix l ow g r aa m
(GR)=g r aa m
(KM)=k ix l ow m eh t r --//eh
$(T.C.)$=t yy uh r k ix y --//eh   j uw m h uw r ix y eh t --//ix
(TC)=t yy uh r k ix y --//eh   j uw m h uw r ix y eh t --//ix
$(TBMM)$=--t eh --b eh --m eh -- m eh
(THY)=--t eh --h eh --y eh
$(SHGM)$=--s eh -h eh --g eh --m eh
(ATO)=--aa --t o
$(DHMİ)$=--d eh -- h eh -- m eh -- ix ih
(GSM)=--g ix --eh s -- eh m
(HKK)=--h eh --k aa --k aa
(DKK)=--d eh --k aa --k aa
(KKK)=--h aa --k aa --k aa
$(ASKİ)$=--aa s k ix
$(KKTC)$=--k aa --k aa --t eh --j eh
(MEB)=--m eh --eh -- b eh
$(NASA)$=--n aa --s aa
$(NATO)$=--n aa --t o
$(ÖTV)$=--er --t eh --v eh
(PTT)=--p eh --t eh --t eh
$(SHÖ)$=--s eh -- h eh --H er
$(TEDAŞ)$=--t eh --d aa sh
(TRT)=--t eh --r eh --t eh
(TSK)=--t eh --s eh --k aa
(ABD)=--aa --b eh --d eh
$(ALES)$=--aa --l eh s
(AVM)=--aa --v eh --m eh
$(DMİ)$=--d eh --m eh --ix ih
$(DSİ)$=--d eh --s eh --ix ih

A3: PUNCTUATIONS and STRESS RULES

a(ğ)^=--	(e)y=ey	ı(ğ)^=--	i(ğ)^=--
a(ğ)#=--	(e)h=eh	ı(ğ)#=--	i(ğ)#=--
y^(a)=aa	e(ğ)^=--	y^(ı)=ax	y^(i)=ix
#^(a)^#=	e(ğ)#=--	#^(ı)^#=	#^(i)^#=
#^(a)=	y^(e)=eh	#^(ı)=	#^(i)=
#^(a)$=	#^(e)^#=	#^(ı)$=	#^(i)$=
$(a)^=aa	#^(e)=	$(ı)^=ax	$(i)^=ix
(a)=aa	#^(e)$=	(ı)=ax	(i)y=iy
	^(e)^=eh		(i)h=ih
	$(e)^=eh		$(i)=ix
	$^(e)=ey		(i)=iy
	(e)=eh		

o(ğ)^=--	ö(ğ)^=--	u(ğ)^=--	ü(ğ)^=--
o(ğ)#=--	ö(ğ)#=--	u(ğ)#=--	ü(ğ)#=--
y^(o)=ow	y^(ö)=er	y^(u)=uw	y^(ü)=yy uh
#^(o)^#=	#^(ö)^#=	#^(u)^#=	#^(ü)^#=
#^(o)=	#^(ö)=	#^(u)=	#^(ü)=
#^(o)$=	#^(ö)$=	#^(u)$=	#^(ü)$=
$(o)^=ow	$(ö)^=er	$(u)^=uw	$(ü)^=yy uh
(o)y=oy	(ö)=er	(u)=uw	$(ü)=uh
(o)=o			(ü)=yy uh

$( )=	(|)=
( )=	(')=
;\(-)\=	(()=aa ch p aa r aa n t eh z
(-)\=ey k s ix	())=k aa p aa p aa r aa n t eh z
(-).\=ey k s ix	({)=aa ch p aa r aa n t eh z
(-)=	(})=k aa p aa p aa r aa n t eh z
(+)=aa r t ax	([)=aa ch p aa r aa n t eh z
(#)=d ix y eh z	(])=k aa p aa p aa r aa n t eh z
($)=d ow l aa r	(")=ch ix f t t ax r n aa k
(=)=eh sh eh t	(<)=k yy uh ch yy uh k t yy uh r
(&)=v eh	(>)=b yy uh y yy uh k t yy uh r
(*)=y ax l d ax z	(<=)=k yy uh ch yy uh k eh sh eh t
(:)=	(>=)=b yy uh y yy uh k eh sh eh t
(;)=,	(~)=t ix l d aa
(,)=,	(.)=
(.)\=n ow k t aa	(?)=
(:)\=	(!)=
(^)=d yy uh z eh l t m eh ix m ix	(...)=
(_)=aa l t ch ix z g ix	(%)= yy yy uh z d eh

A4: PHONEME MAPPINGS of CONSONANTS

(.b.)=b eh	(k)$= --\\k	(.s.)=s eh
$(b.)=b eh	(.k.)=k aa	$(s.)=s eh
(.b)$=b eh	$(k.)=k aa	(.s)$=s eh
(b)=b eh	(.k)$=k aa	(s)=s --eh
(.c.)=j eh	(k)=k aa	(s)=s
$(c.)=j eh	$(k)=k	$(ş)$=sh eh
(.c)$=j eh	#(k)^=k k	(ş)=sh
(c)=j ei	(k)=k	(.t.)=t eh
(c)=j	(.l.)=l eh	$(t.)=t eh
(ç)=ch	$(l.)=l eh	(.t)$=t eh
$(ç)$=ch eh	(.l)$=l eh	(t)=t eh
(.d.)=d eh	(l)=l eh	$(t)=t
$(d.)=d eh	$(l)=l	#(t)^=t t
(.d)$=d eh	#(l)^=l l	(t)=t
(d)=d eh	(l)=l	(.v.)=v eh
(d)=d	(.m.)=m eh	$(v.)=v eh
(.f.)=f eh	$(m.)=m eh	(.v)$=v eh
$(f.)=f eh	(.m)$=m eh	(v)=v eh
(.f)$=f eh	(m)=m eh	(v)=v
(f)=f eh	$(m)=m	(.y.)=y eh
(f)=f	#(m)^=m m	$(y.)=y eh
(.g.)=g eh	(m)=m	(.y)$=y eh
$(g.)=g eh	(.n.)=n eh	(y)=y eh
(.g)$=g eh	$(n.)=n eh	#(y)^=yy yy
(g)=g eh	(.n)$=n eh	^(y)#=yy yy
(g)=g	(n)=n eh	(y)=yy
(ğ)=ih	$(n)=n	(.z.)=z eh
(.h.)=h eh	#(n)^=n n	$(z.)=z eh
$(h.)=h eh	(n)=n	(.z)$=z eh
(.h)$=h eh	(.p.)=p eh	(z)=z eh
(h)=h eh	$(p.)=p eh	(z)=z
$(h)=h	(.p)$=p eh	(x)=ae k s
#(h)^=h h	(p)=p eh	(q)=k u
(h)=h	#(p)^=p p	(w)=d ah b l yy uw
(.j.)=zh eh	(p)=p	
$(j.)=zh eh	(.r.)=r eh	
(.j)$=zh eh	$(r.)=r eh	
(j)=zh eh	(.r)$=r eh	
;(j)=jh	(r)=r eh	
(j)=zh	(r)=r	

A5: NUMBERS

\(0)=	(5)\\\=b eh sh b ix n
(0)\=	(5)\\=b eh sh yy yy uh z
(0)=s ax f ax r	(50)=eh l l ix
(10000)=ow n b ix n	(5)\=eh l l ix
(1)\\\\=b ix r	(5.)=b ih sh ix n j ix
(1)\\\=b ix n	(5)=b eh sh
(1)\\=yy yy uh z	(6)\\\\=aa l t ax
(10.)=ow n uw n j uw	(6)\\\=aa l t ax b ix n
(10)=ow n	(6)\\=aa l t ax yy yy uh z
(1)\=ow n	(60)=aa t m ax sh
(1.)$=b ix r ix n j ix	(6)\=aa t m ax sh
(1)=b ix r	(6.)=aa l t ax n j ax
(2)\\\\=ix k ix	(6)=aa l t ax
(2)\\\=ix k ix b ix n	(7)\\\\=yy eh d ix
(2)\\=ix k ix yy yy uh z	(7)\\\=yy eh d ix b ix n
(20)=yy ix r m ix	(7)\\=yy eh d ix yy yy uh z
(2)\=yy ix r m ix	(70)=yy eh t m ix sh
(2.)$=ix k ix n j ix	(7)\=yy eh t m ix sh
(2)=ix k ix	(7.)=yy eh d ix n j ix
(3)\\\\=yy uh ch	(7)=yy eh d ix
(3)\\\=yy uh ch b ix n	(8)\\\\=s eh k ix z
(3)\\=yy uh ch yy yy uh z	(8)\\\=s eh k ix z b ix n
(30)=ow t u z	(8)\\=s eh k ix z yy yy uh z
(3)\=ow t u z	(80)=s eh k s eh n
(3.)=yy uh ch yy uh n j yy uh	(8)\=s eh k s eh n
(3)=yy uh ch	(8.)=s eh k ix z ix n j ix
(4)\\\\=d er r t	(8)=s eh k ix z
(4)\\\=d er r t b ix n	(9)\\\\=d ow k u z
(4)\\=d er r t yy yy uh z	(9)\\\=d ow k u z b ix n
(40)=k ax r k	(9)\\=d ow k u z yy yy uh z
(4)\=k ax r k	(90)=d ow k s aa n
(4.)=d er r d yy uh n j yy uh	(9)\=d ow k s aa n
(4)=d er r t	(9.)$=d o k uw z uw n j uw
(5)\\\\=b eh sh	(9)=d ow k u z

APPENDIX B

C# GRAPHICAL USER INTERFACE CODES

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace RC8660 Control Panel
{
   public partial class Form1 : Form

{
      string RxString;
      int speed;

      public Form1()
      {

              InitializeComponent();

      }

      private void buttonStart_Click(object sender, EventArgs e)
      {

         serialPort1.PortName = "COM2";
         serialPort1.BaudRate = 9600;

         serialPort1.Open();
         if (serialPort1.IsOpen)

            {

               buttonStart.Enabled = false;
            buttonStop.Enabled = true;
            textBox1.ReadOnly = false;

         }
      }

      private void buttonStop_Click(object sender, EventArgs e)
      {
         if (serialPort1.IsOpen)
```

```
    {

        serialPort1.Close();
        buttonStart.Enabled = true;
        buttonStop.Enabled =  false;
        textBox1.ReadOnly = true;

    }
}
private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{

       if (serialPort1.IsOpen) serialPort1.Close();

}

private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
{

       if (!serialPort1.IsOpen) return;
       char[] buff = new char[1];
       buff[0] = e.KeyChar;
  //    if (buff[0] == ('ş'))
  //    {

  //        byte[] bt = Encoding.Unicode.GetBytes("ş");
  //        string str = Encoding.UTF8.GetString(bt);
  //        serialPort1.Write(str);

  //    }

        serialPort1.Write(buff,0,1);

}

private void DisplayText(object sender, EventArgs e)
{

    textBox1.AppendText(RxString);

}

      private void serialPort1_DataReceived(object sender,
    System.IO.Ports.SerialDataReceivedEventArgs e)
{

    RxString = serialPort1.ReadExisting();
    this.Invoke(new EventHandler(DisplayText));
}
```

```
private void Form1_Load(object sender, EventArgs e)
{

    string st1;
    trackBar1.Minimum = 0;
    trackBar1.Maximum = 15;
    trackBar1.Value = 0;
    st1 = trackBar1.Value.ToString();
    label3.Text = st1;

    string st2;
    trackBar2.Minimum = 0;
    trackBar2.Maximum = 13;
    trackBar2.Value = 5;
    st2 = trackBar2.Value.ToString();
    label4.Text = st2;

    string st3;
    trackBar7.Minimum = 0;
    trackBar7.Maximum = 9;
    trackBar7.Value = 5;
    st3 = trackBar7.Value.ToString();
    label2.Text = st3;

    string st4;
    trackBar8.Minimum = 0;
    trackBar8.Maximum = 99;
    trackBar8.Value = 50;
    st4 = trackBar8.Value.ToString();
    label5.Text = st4;

    string st51;
    trackBar4.Minimum = 0;
    trackBar4.Maximum = 9;
    trackBar4.Value = 4;
    st51 = trackBar4.Value.ToString();
    label6.Text = st51;

    string st6;
    trackBar5.Minimum = 0;
    trackBar5.Maximum = 9;
    trackBar5.Value = 0;
    st6 = trackBar5.Value.ToString();
    label7.Text = st6;

    string st7;
    trackBar9.Minimum = 0;
    trackBar9.Maximum = 99;
    trackBar9.Value = 50;
```

```
        st7 = trackBar9.Value.ToString();
        label8.Text = st7;

        string st8;
        trackBar6.Minimum = 0;
        trackBar6.Maximum = 2;
        trackBar6.Value = 1;
        st8 = trackBar6.Value.ToString();
        label9.Text = st8;

        string st9;
        trackBar3.Minimum = 0;
        trackBar3.Maximum = 9;
        trackBar3.Value = 5;
        st9 = trackBar3.Value.ToString();
        label10.Text = st9;

        string st10;
        trackBar1.Minimum = 0;
        trackBar1.Maximum = 15;
        trackBar1.Value = 0;
        st10 = trackBar1.Value.ToString();
        label3.Text = st10;

}

    private void trackBar1_Scroll(object sender, EventArgs e)
    {

        string val;
        label3.Text = trackBar1.Value.ToString();
        val = Convert.ToString(label3.Text);
        byte[] data = new byte[] { 0x01 };
        serialPort1.Write(data, 0, data.Length);
        serialPort1.Write(val + "t");

    }

    private void timer1_Tick(object sender, EventArgs e)
    {

            int val;
            string value_1;

    }

    private void textBox2_TextChanged(object sender, EventArgs e)
    {
```

```
        int val;
        string value_1;

}

private void button1_Click(object sender, EventArgs e)
{

    byte[] data = new byte[] { 0x01};
    serialPort1.Write(data, 0, data.Length);
    serialPort1.Write("0s");

}

private void button2_Click(object sender, EventArgs e)
{

  byte[] data = new byte[] { 0x01 };
  serialPort1.Write(data, 0, data.Length);
  serialPort1.Write("9s");

}

private void trackBar2_Scroll(object sender, EventArgs e)
{

  string val;
  label4.Text = trackBar2.Value.ToString();
  val = Convert.ToString(label4.Text);
  byte[] data = new byte[] { 0x01 };
  serialPort1.Write(data, 0, data.Length);
  serialPort1.Write(val + "s");

}

private void trackBar7_Scroll(object sender, EventArgs e)
{

  string val;
  label2.Text = trackBar7.Value.ToString();
  val = Convert.ToString(label2.Text);
  byte[] data = new byte[] { 0x01 };
  serialPort1.Write(data, 0, data.Length);
  serialPort1.Write(val + "v");

}

private void trackBar1_ValueChanged(object sender, EventArgs e)
{
```

```
            int val;
            string value_1;

        }

        private void trackBar7_ValueChanged(object sender, EventArgs e)
        {

            int val;
            string value_1;

        }

        private void track_2()
        {

            string st2;
            trackBar2.Minimum = 0;
            trackBar2.Maximum = 13;
            trackBar2.Value = 5;
            st2 = trackBar2.Value.ToString();
            label4.Text = st2;

        }

        private void label5_Click(object sender, EventArgs e)
        {
            int val;
            string value_1;

        }

        private void trackBar8_Scroll(object sender, EventArgs e)
        {

            string val;
            label5.Text = trackBar8.Value.ToString();
            val = Convert.ToString(label5.Text);
            byte[] data = new byte[] { 0x01 };
            serialPort1.Write(data, 0, data.Length);
            serialPort1.Write(val + "p");

        }



        private void checkBox1_CheckedChanged(object sender, EventArgs e)
        {
```

```
    if(checkBox1.Checked)
    {

        byte[] data = new byte[] { 0x01 };
        serialPort1.Write(data, 0, data.Length);
        serialPort1.Write("u");

    }

    else if(!checkBox1.Checked)
    {

        byte[] data = new byte[] { 0x01 };
        serialPort1.Write(data, 0, data.Length);
        serialPort1.Write("T");

    }

}

private void trackBar4_Scroll(object sender, EventArgs e)
{

    string val;
    label6.Text = trackBar4.Value.ToString();
    val = Convert.ToString(label6.Text);
    byte[] data = new byte[] { 0x01 };
    serialPort1.Write(data, 0, data.Length);
    serialPort1.Write(val + "E");

}

private void trackBar5_Scroll(object sender, EventArgs e)
{

    string val;
    label7.Text = trackBar5.Value.ToString();
    val = Convert.ToString(label7.Text);
    byte[] data = new byte[] { 0x01 };
    serialPort1.Write(data, 0, data.Length);
    serialPort1.Write(val + "r");

}

private void trackBar9_Scroll(object sender, EventArgs e)
{

    string val;
    label8.Text = trackBar9.Value.ToString();
```

```
            val = Convert.ToString(label8.Text);
            byte[] data = new byte[] { 0x01 };
            serialPort1.Write(data, 0, data.Length);
            serialPort1.Write(val + "f");

        }



        private void trackBar6_Scroll(object sender, EventArgs e)
        {

            string val;
            label9.Text = trackBar6.Value.ToString();
            val = Convert.ToString(label9.Text);
            byte[] data = new byte[] { 0x01 };
            serialPort1.Write(data, 0, data.Length);
            serialPort1.Write(val + "x");

        }

        private void trackBar3_Scroll(object sender, EventArgs e)
        {

            string val;
            label10.Text = trackBar3.Value.ToString();
            val = Convert.ToString(label10.Text);
            byte[] data = new byte[] { 0x01 };
            serialPort1.Write(data, 0, data.Length);
            serialPort1.Write(val + "a");

        }

        private void button3_Click(object sender, EventArgs e)
        {

                          OpenFileDialog dialog = new OpenFileDialog();
                          dialog.RestoreDirectory = false;
                          dialog.Title = "Select a file";
                          if (dialog.ShowDialog() == DialogResult.OK)
                          {

                          String text = System.IO.File.ReadAllText(dialog.FileName);
                          serialPort1.Write(text);
                          }
        }

    }
}
```