WTS701EF

ENGLISH FEMALE VERSION TEXT-TO-SPEECH

USER'S MANUAL

Publication Release Date: June, 2003

Revision 1.00

TABLE OF CONTENTS

1. INTRODUCTION

The WTS701 is a high quality, fully integrated, single-chip Text-to-Speech solution that is ideal for use in applications such as automotive appliances, GPS/navigation systems, cellular phones and other portable products or accessories. The WTS701 product accepts ASCII (Unicode and Big5 for Mandarin) input via a SPI port and converts it to spoken audio via an analog output or digital CODEC output.

The WTS701 integrates a text processor, smoothing filter and multi-level memory storage array on a single-chip. Text-to-speech conversion is achieved by processing the incoming text into a phonetic representation that is then mapped to a corpus of naturally spoken word parts. The synthesis algorithm attempts to use the largest possible word unit in the appropriate context to maximize natural sounding speech quality. The speech units are stored uncompressed in a multi-level, non-volatile analog storage array to provide the highest sound quality to density trade-off. This unique, single-chip solution is made possible through Winbond's patented multilevel storage technology. Voice and audio signals are stored directly into solid-state memory in their natural, uncompressed form, providing superior quality voice reproduction.

The chip can be programmed through the SPI port, allowing downloading of different languages and speaker databases when made available by Winbond.

2. WINBOND TEXT-TO-SPEECH TECHNOLOGY OVERVIEW

2.1. OVERVIEW OF DEVICE INITIALIZATION SEQUENCE

Configuration

After power-on or a Reset command (RST) the WTS701 processor can be configured for operation. This involves initializing the internal configuration registers for the users requirements.

Table 1. Initialization Command Sequence

State	Command	Description
POWER DOWN	--------	State after power-on or RST command.
	SCLC	Set clock configuration.
	PWUP	Power up device.
IDLE	SCOM	Set up communication register to enable interrupts.
	SCOD	Set up CODEC configuration (if used).
	SAUD	Set up audio control register.
	SVOL	Set the initial volume level.
	SSPD	Set the initial speech output speed level.
	SPTC	Set the initial speech pitch level.

2.2. DESCRIPTION OF USER INTERFACE

As a real System-On-Chip solution, the WTS701 performs the overall control functions for host controller and text-to-speech processing.

The WTS701 system architecture consists of the following functions:

- Serial interface to monitor the SPI port and interpret commands and data
- Text normalization module to pre-process incoming text into pronounceable words
- Words to phoneme translator, which converts incoming text to phoneme codes
- Phoneme mapping module that maps incoming phonemes to words, sub-words, syllables or phonemes present in the MLS memory
- Volume and speed adjustments
- Digital and analog output blocks for off-chip usage

The WTS701 system performs text-to-speech synthesis based on concatenative samples. The units for concatenation can vary from whole words down to phoneme units. The convention is that the larger the sub-word unit used for synthesis the higher the quality of the speech output. A corpus of pre-recorded words is stored in Winbond's patented multilevel storage (MLS) memory and a mapping of the various sub-word parts is held in a lookup table. The speech creation is achieved by concatenation of these speech elements to produce words. The system process flow is shown in Figure 1.

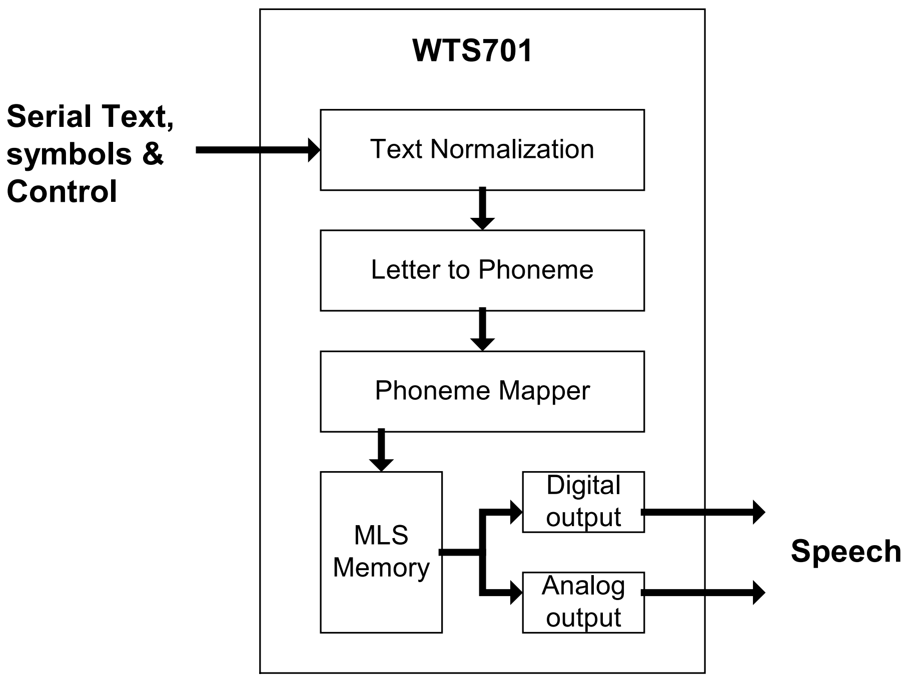

Figure 1. WTS701 System Process Flow.

2.3. DESCRIPTION OF WTS701 STATE MACHINE

2.3.1 Text-To-Speech Mechanism

The text to speech component of the system consists of three principal blocks:

- Text normalization
- Letter-to-phoneme conversion
- Phoneme mapping

2.3.2. Text Normalization

Text normalization involves the translation of incoming text into pronounceable words. It includes such functions as expanding abbreviations and translating numeric strings to spoken words. It involves a certain amount of context processing to determine correct spoken form.

In addition, the WTS701 looks into the abbreviation list stored in the device's internal memory and converts acronyms, abbreviations or special characters (such as Instant Messaging icons or emoticons) into the appropriate text representation.

The default abbreviation list supported by the WTS701 is a general one that cannot be modified by the user to match the domain that the text is being loaded from. But the default list can be overridden by the user abbreviation list. This enables a flexibility of adding abbreviation specifically for the text either by the developer or even the end user to best customize the product for its preferences. Instant Messaging or Short Messages Service (SMS) unique characters are supported through this functionality as well, defining the icon, ASCII/Unicode/Big5 text, and its replacement. The default abbreviation list supported is described in the specific language release letter.

2.3.3. Letter-to-Phoneme Conversion

Once the data stream has been translated to pronounceable words, the system next determines how to pronounce them. This function is obviously highly language dependent. For a language such as English it is impossible to break this task down to a set of definitive rules. The task is achieved by a combination of rule based processing together with exception processing.

2.3.4. Phoneme Mapping

This algorithm maps phoneme strings into the MLS phonetic inventory. This task falls into two portions. First, the word must be split into sub-word portions. This splitting must be done at appropriate phonetic boundaries to achieve high quality concatenation. Once a sub-word unit is determined, the inventory is searched to determine if a match is present. A matching weight is assigned to each match depending on how closely the phonetic context matches. Each sub-word has a left and right side context to match as well as the phoneme string itself. If no suitable match is found in the inventory, then the sub-word is further split in a tree like manner until a match is found. The splitting tree is processed from left to right and each time a successful match occurs the address and duration of the match in the corpus is placed in a queue of phonetic parts to be played out the audio interface.

2.4. SPI COMMANDS OVERVIEW

The WTS701 is controlled by a series of SPI transactions to send commands to the device. The general format of an SPI transaction is shown in Figure 2 A transaction is always started by sending a command word. The command word consists of a command byte followed by a command data byte. At the same time, the status register is shifted out on the MISO line. What follows depends on what command is sent. The general case is that following the command word, up to *n*-bytes of data can be sent to the device and *n*-bytes can be read from the device. An SPI transaction is finished when SS\ is returned to the HIGH condition.

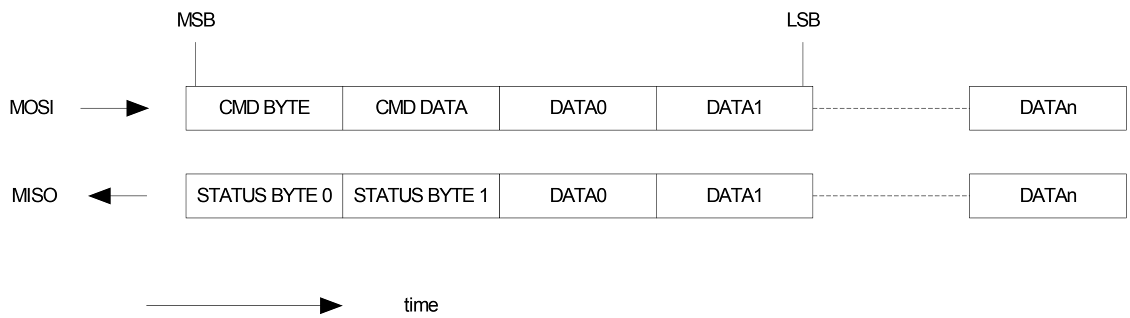

Figure 2. SPI Transaction Format.

2.4.1. Command Classes

The SPI transactions to the WTS701 fall into four classes. The four classes represent variations in how the command, and any associated data, is handled. The class of a command is defined by the two most significant bits of the command byte. A summary of the command classes is given below

CLASS 0 COMMANDS

These are commands that are executed irrespective of the state of the WTS701. That is, the command will execute even if the device is busy or powered down. These commands are executed internally by a hardware command interpreter. All commands not of class 0 require that the WTS701 be in a powered up state. Example of class 0 command is the Read Status (RDST) command.

CLASS 1 COMMANDS

Class 1 commands require interpretation by the internal firmware of the WTS701. Class 1 commands consist only of a command byte and command data byte. Any further data sent in a transaction is ignored. Class 1 commands are most often used for setting a configuration register in the device or sending commands that have no data such as the conversion pause (PAUS) command.

CLASS 2 COMMANDS

Class 2 commands have associated data. After the command word, any data bytes following are loaded into an internal FIFO buffer for processing. If this FIFO becomes full, the R/$\overline{B}$ signal is asserted (LOW) indicating that the host must pause data transfer. An alternative to monitoring the R/$\overline{B}$ line, the R/$\overline{B}$ bit of the status register can be monitored instead (see subsection 7.3.2) or via the RDST command.

CLASS 3 COMMANDS

Class 3 commands have data to return to the host. The R/$\overline{B}$ line will go to busy immediately following the command word indicating that the WTS701 is fetching the requested data. Data is put into the BCNT0 and BCNT1 (see subsection 7.3.4) registers and is read out in the two subsequent bytes after R/$\overline{B}$ is released. If more than two bytes are returned from the command, R/$\overline{B}$ will again be asserted until data is ready to read. The primary Class 3 commands are to read the contents of internal configuration registers such as RREG command.

2.5. DESCRIPTION OF PHONETIC ALPHABET FEATURE

As indicated in 2.3.3, the WTS701 coverts spelled-out pronounceable words into phonetic transcriptions, i.e. a string of phonemes providing an abstract representation of the target pronunciation. This feature allows the input to contain phonetic transcriptions instead of ordinary English text, explicitly indicating the desired pronunciation.

Phonetic transcriptions can be sent directly to the WTS701. This can be done by embedding phoneme strings in the text stream for conversion. To embed a phoneme string, the string must be preceded by a control-P (^P, ASCII 0x10) character and terminated by a space character. If there is following punctuation or word, a space must still intervene.

For example:

"The quick brown ^Pf1Aks jumped over the lazy ^Pd1cg ."

The following table lists the phoneme symbols acceptable by the WTS701 (English Female software version). As the acceptable phoneme symbols are language and version dependent, please refer to the specific language User's Guide for details regarding characters accepted and other development considerations.

Table 2. Acceptable Phoneme Symbols.

Vowels		Consonants	
Phoneme	**Example**	**Phoneme**	**Example**
i	beat	p	*p*et
I	bit	t	*t*en
e	bait	k	*k*it
E	bet	q	wri*tt*en
@	bat	b	*b*et
u	boot	d	debt
U	book	g	get
o	boat	f	fat
c	bought	T	*th*ing
a	Bob	s	sat
A	but	S	*sh*ut
R	bird	h	hat
O	bo*y*	v	vat
Y	bu*y*	D	*th*at
W	down	z	*z*oo
x	*a*bout	Z	azure
X	roses	y	*y*ou
		w	*w*it
		r	*r*ent
		P	eigh*ty*
		l	*l*et
		m	*m*et
		n	*n*et
		G	si*ng*
		C	*church*
		J	*judge*

Note that each phoneme is represented by exactly one character, and there must be an indication of stress before each vowel. The digit '1' is used to indicate primary stress, and each word has only one primary stress. All other vowels are marked with the digit '0'.

Examples:

Input	Phonetic transcription
hi	h1Y (phoneme /h/, followed by a 1-stress vowel phoneme Y)
cute	ky1ut
backpack	b1@kp0@k
encyclopedia	0Ins0Ykl0xp1id0i0x

Constraints on stress:
All vowels may bear 0 stress. However, neither x nor X can bear primary stress, marked with '1'. The best primary stress alternative for these two is 1A, but 1I may be more appropriate for X in some cases.

Dialect note:
The WTS701EF presents a dialect of Standard American English spoken in the San Francisco Bay area, and the transcriptions generated are consistent with this dialect. Transcriptions can be modified to emulate some properties of other dialects, including Standard British and Australian dialects, but a complete transformation cannot be successful, as many other dialect-dependent features cannot be modified with transcriptions.

Abstractness:
These phonetic transcriptions are abstract representations, and some transformations are performed on these transcriptions on the way to creating audio output. Three of phonemes in the preceding table are commonly products of these transformations.

-The vowel 0X is associated with both 0I and 0x. It can be used in input transcriptions, but the user can use 0I and 0x instead, as selecting among the two as appropriate. For example, 'roses' may also be transcribed as ^Pr1oz0Iz . The output will be identical.
-Both q and P are associated with t. 'Written' can be transcribed as ^Pr1It0xn as well as ^Pr1Iq0xn , and 'eighty' may be transcribed as ^P1et0i as well as ^P1eP0i . The output will be identical in both pairs. The user may simply use t except to force output with q or P.

Uses:
-Greater control
The phonetic alphabet feature may be used to specify particular pronunciations.

Examples:
"Do you say ^Pt0xm1eP0o or ^Pt0xm1at0o ?"
"People from Missouri say they're from ^Pm0Iz1ur0x ."
-Foreign and unusual words
The phonetic alphabet can also be used to specify the pronunciation of unusual words, outside the central core of English words covered by the letter-to-phoneme module. For example, the following transcriptions can be provided for French President Jacques Chirac's name: ^PZ1ak ^PS0ir1ak .

-Tweaking pronunciations

The phonetic alphabet can also be used to 'tweak' pronunciations. Many English words, even common words, have multiple valid transcriptions, but in some cases, the output for each is not equally as good. For example, the default transcription for 'current' for the WTS701EF is ^Pk1Rr0Int . The transcription ^Pk1R0Int is equally valid, but the output is much worse. Exploring variant phonetic transcriptions can solve many output problems.

Developers of TTS applications are often tempted to fix pronunciation problems by using abberant spellings, but they should use transcriptions instead, as phonetic transcriptions provide much greater control. Indeed, many issues can only be addressed by providing transcriptions.

2.6. DESCRIPTION OF ABBREVIATIONS FEATURE

The WTS701 has support for entering and using custom abbreviations in addition to the general abbreviation table supported internally by the WTS701. These supplement the default abbreviation support. (See sections 3.8 and 4.2) There are 2K bytes of flash memory reserved for this purpose. After the WTS701 internal software has been initially programmed, this entire area is free and available for custom abbreviations.

The commands associated with custom abbreviations are:

Command	Command Byte	Command Data Byte	
ABBR_ADD	0xaf	0x00 + abbreviation data.	Adds a new abbreviation to the abbreviation table in the WTS701 See below for the format of the abbreviation data.
ABBR_DEL	0x83	0x00+ abbreviation data.	Deletes an existing abbreviation from the abbreviation table in the WTS701. See below for the format of the abbreviation data.
ABBR_NUM	0xc8	0x00 + 0x00 + 0x00.	Returns the number of abbreviation currently active in the abbreviation table of the WTS701.
ABBR_MEM	0xc7	0x00 + 0x00 + 0x00.	Returns the number of free bytes in the abbreviation table of the WTS701.
ABBR_RD	0xc9	0x00 + 2048 0x00s.	Returns the abbreviation table contents from the WTS701. See below for the format of the abbreviation table data.
ENTER_RRSM	0x0c	0x00	Causes the xdata and code store memory to swap spaces. The WTS701 begins to execute code previously stored into xdata after this command.

For both ABBR_ADD and ABBR_DEL, the abbreviation data should be formatted as follows:

input_abbreviation + comma + output_string + semicolon.

For example, the following associates the abbreviation "Fr" with "Father": "Fr,Father;". If this data were entered with ABBR_ADD, "Fr. Miller" would be read "Father Miller". (See section 3.8 for constraints on abbreviation matching.) An abbreviation can also map onto multiple words, e.g. "BC,before Christ;" or "BC,British Columbia;". This feature can also be used to provide transcriptions, according to the conventions in section 2.5. For example, "NASA,^Pn1@s0x;" would provide the common pronunciation of 'NASA'. This can also be done for ordinary spelled out words, as a means for consistently providing user transcriptions. Thus, if "Chirac,^PS0ir1ak;" were added, the pronunciation for the input text 'Chirac' would consistently be ^PS0ir1ak .

The abbreviation table is provided in the following format when an ADDR_RD command is issued:

"abbreviation_number. input_abbreviation

output_string"

Abbreviations are listed and numbered in the order in which they are added. If an abbreviation is deleted, the text "(deleted)" appears after the input_abbreviation.

Example:

01. Fr

Father

02. BC (deleted)

before Christ

03. BC

British Columbia

04. NASA

^Pn1@s0x

05. Chirac

^PS0ir1ak

3. RULES TO BE APPLIED

3.1. TEXT

There is no pre-set character length limit for an input text.

3.2. SENTENCE

There is no pre-set character length limit for an input sentence.

3.3. WORD

Any space-bounded string is treated as a word. The maximum character length of a word is 53. Words 54 or more characters in length will be truncated after the 53rd character.

3.4. CHARACTER

A character must be encoded in ASCII. The set of ASCII characters defined by the system ranges from '0x00' to '0x7A' excluding '0x22', '0x3B', '0x3C', '0x3E' and '0x60' (please refer to **section 5.1** for more details). All the undefined characters will be deleted prior to letter-to-phoneme conversion.

3.5. CONTROL CHARACTERS

To allow users to manipulate the system functions via the input text, certain control characters are defined as command flags. Actual HEX values must be sent to the chip. These should not be confused with the sequence of '^' followed by a capital letter.

- Control-P ('0x10'): Phoneme control flag. This command flag indicates a phoneme string is to follow. A space character is required after the end of the phoneme string.

Example:
'^Ppr0Ez0Int1eS0In ' will be pronounced as 'presentation'.

- Control-Q ('0x11'): Pause control flag. Pauses with variant length can be added within a sentence by using the '^QX' flag. 'X' is an integer which indicates the pause duration in tenths of a second. A space character is required after the pause-duration digit.

Example:
'^Q10 ' will add in a one-second pause.

- Control-S ('0x13'): Speed control flag. This command flag controls the speed change.

'^S+': Increase the playback speed by 1 level. A space character is required after '+'.
Example:
'^S+ Hello, world'

'^S-': Decrease the playback speed by 1 level. A space character is required after '-'.
Example:
'^S- Hello, world'

'^SX': Set the playback speed to level X. X is an integer, and its valid range is from 0 to 4, from fast to slow. The default speed level is 2. Any number that is greater than 4 will be set to 4. A space character is required after the level number.
Example:
'^S0 Hello, world'

- Control-U ('0x15'): All-capital word control flag. This command flag tells the system whether a string comprised exclusively of capital letters should be spelled out or not.

'^U0': Force all short all-capital words (4 or less letters) to be spelled out, unless the word is an abbreviation established in the system. Abbreviations will be treated as indicated in the built-in abbreviation list (see 4.2) or by the user (see 2.6). Longer all-capital words (5 or more letters) will be pronounced if deemed to be a word pronounceable in English. This is the default setting. A space character is required after '0'.
Example:
'^U0 USA' will be pronounced as 'u s a'.
'^U0 CA' will be pronounced as 'California'. (A built-in abbreviation)
'^U0 ZZZZZZZ' will be spelled out because it is not pronounceable in English.

'^U1': Avoid spelling out words in all capitals. The system will pronounce every all-capital word (as a normal word) as long as the word is pronounceable. In this mode, no abbreviations will be detected. A space character is required after '1'.
Example:
'^U1 ROM' will be pronounced as 'rom' whereas '^U0 ROM' will be pronounced as 'r o m'.

'^U2': Force every word to be spelled out regardless of case or length. In this mode, no abbreviations will be detected. A space character is required after '2'.
Example:
'^U2 GOOD' will be pronounced as 'g o o d'.
'^U2 good' will be pronounced as 'g o o d'.

'^U3': Force every all-capitol word to be spelled out regardless of its string length, unless the word is an abbreviation established in the system. Abbreviations will be treated as indicated in the built-in abbreviation list (see 04.2) or by the user (see 02.6). It simlar to '^U0' except for the insensitivity to string length. A space character is required after '3'.
Example:
'^U3 HELLO' will be pronounced as 'h e l l o'.

- Control-V ('0x16'): Volume control flag. This command flag changes the volume.

'^V+': Increase the playback volume by 1 level. A space character is required after '+'.

Example:
'^V+ Hello, world'.

'^V-': Decrease the playback volume by 1 level. A space character is required after '-'.
Example:
'^V- Hello, world'.
'^VX': Set the playback volume to level X. X is an integer and its valid range is from 0 to 7 (0dB to –28dB). The default volume level is 3 (-12dB). Any number that is greater than 7 will be set to 0. A space character is required after the level number.

The following example shows how these control characters can be used together, handling not only capitalization and pronunciation issues, but also establishing emphasis and conveying some emotion.

^V3 Well, ^V- ^S4 to tell you the truth, ^V+ ^S2 it was ^Q2 ^V+ ^U1 JOHN ^U0 ^V- ^Q1 who ^V- ^S- wrote the report, ^S+ ^V+ not ^Q1 ^V+ ^Pg0iy1erm0o ^V-.

3.6. DASH

- When a dash appears between words with no space preceding or following, it is recognized as hyphen in a hyphenated word. There is no abbreviation support for the hyphenated words.

  Example:

  'three-year-old' will be pronounced as 'three year old'.

- When there is a space preceding or following the dash, it will be pronounced as 'dash'.

  Example:

  'three- year' will be pronounced as 'three dash year'.

  'three -year' will be pronounced as 'three dash year'.

- When used in front for a digit, it will be pronounced as 'minus'.

  Example:

  '-5' will be pronounced as 'minus five'.

3.7. SLASH

When a slash appears between words with no space preceding or following, it is treated as a space. Otherwise, it will be pronounced out as 'slash'. There is no abbreviation support in this case.

Example:

'boy/girl' will be pronounced as 'boy girl'.

'boy/ girl' will be pronounced as 'boy slash girl'.

3.8. DOT

- When '.' is used within a decimal number, it will be pronounced as 'point'.

  Example:

'2.6' will be pronounced as 'two point six'.

- When '.' is used at the end of a word or a sentence, it is treated as silence (please see **section 3.7** for more details).

- For all the other cases that are not described above, '.' will be pronounced out as 'dot'.

3.9. INTERNET/ E-MAIL ADDRESS

Any string which contains '@', 'www', '.com' or '.gov' as a substring will be treated as a web or e-mail address.

Example:

'winbond.com' will be pronounced as 'winbond dot com'.

'www.irs.gov' will be pronounced as 'w w w dot i r s dot gov'.

3.10. PUNCTUATION

Punctuation	Pause Duration
,	0.2 sec
.	0.6 sec
!	0.5 sec
?	0.5 sec
...	0.5 sec

Note: A space character is required at the end of ellipse (...).

While pauses are insterted for punctuation, punctuation also influences the pronunciation of neighboring words. Punctuation should not added in violation of standard English usage in order to insert or extend pauses; this should be done instead with the pause control flag control-Q ('0x11') (see section 3.5).

In accordance with standard English orthography, all of these punctuation markers should immediately follow the preceding word with no internening space. Violating this norm can adversely affect output quality. For example, there should be no space before the period in "Believe it or not." If "Believe it or not." is entered instead, the output for "not" will not be as appropriate to the context. The only exception is for command character sequences, e.g. "Good night ^V-, my sweet ^Ppr1Ins ." (see section 3.5).

3.11. ABBREVIATIONS

The system recognizes a default set of frequently used abbreviations (see **section 5.2**) and also allows users to add abbreviations of their own (see section 2.6). Matching is case sensitive, but both the presence and the absence of final periods are ignored in matching. For example, the abbreviation "Feb" is provided for "February". Both "Feb" and "Feb." would constitute matches, but both "feb" and "feb." would not. Note also that there is no abbreviation support in both ^U1 and ^U2 modes. (See section 3.5.)

3.12. NUMERATION

3.12.1 Numbers

- Integers:

Example:

'10' will be pronounced as 'ten'.

'10,000' will be pronounced as 'ten thousand'.

'94087' will be pronounced as 'nine four zero eight seven'

- Decimal numbers:

Example:

'1.23' will be pronounced as 'one point two three'.

- Ordinal numbers: Any number terminated by 1st, 2nd, 3rd, 4 (5, 6, 7, 8, 9, 0) th, is recognized as an ordinal number.

Example:

'21st' will be pronounced as 'twenty first'.

3.12.2. Time

The time representation format is '00:00AM' or '00:00PM' (one semicolon between 2 integers). 'AM' (am) or 'PM' (pm) is optional, but if needed it must be added right after the second integer.

Example:

'9:30pm' will be pronounced as 'nine thirty p m'.

3.12.3. Date

The support for dates goes from year 2000 to 2099 and the format is 'MM/DD/YY'.

For the dates before year 2000, please use 'MM/DD/YYYY'.

Example:

'1/15/03' will be pronounced as 'one fifteen two thousand three'.

3.12.4. Dollar sign:

The combination of a '$' and any integer or any decimal number is recognized as money. (No space is allowed in between)

Example:

'$15.99' will be pronounced as 'fifteen dollars and ninety nine cents'.

'$500 million' -> 'five hundred million dollars'.

3.12.5. Combination of digits and other characters:

This occasion happens mostly in telephone numbers.

Example:

'(408) 123-4567' will be pronounced as 'four zero eight one two three four five six seven'. A pause will be placed after 8 and another pause will be placed after 3.

'123-4567' will be pronounced as 'one two three four five six seven'.

3.12.6. Percent Sign %:

Example:

'12%' will be pronounced as 'twelve percent'.

4 APPENDICES

4.1. INPUT CHARACTER TABLE

0x0	Reserved	0x20	Space	0x40	@	0x60	Undefined
0x1	Reserved	0x21	!	0x41	A	0x61	a
0x2	Reserved	0x22	Undefined	0x42	B	0x62	b
0x3	Reserved	0x23	#	0x43	C	0x63	c
0x4	Reserved	0x24	$	0x44	D	0x64	d
0x5	Reserved	0x25	%	0x45	E	0x65	e
0x6	Reserved	0x26	&	0x46	F	0x66	f
0x7	Reserved	0x27	‘ (apostrophe)	0x47	G	0x67	g
0x8	Reserved	0x28	(	0x48	H	0x68	h
0x9	Space	0x29	)	0x49	I	0x69	I
0xa	Space	0x2a	*	0x4a	J	0x6a	j
0xb	Space	0x2b	+	0x4b	K	0x6b	k
0xc	Space	0x2c	, (comma)	0x4c	L	0x6c	l
0xd	Space	0x2d	- (dash)	0x4d	M	0x6d	m
0xe	Reserved	0x2e	. (period)	0x4e	N	0x6e	n
0xf	Reserved	0x2f	/ (slash)	0x4f	O	0x6f	o
0x10	^P	0x30	0	0x50	P	0x70	p
0x11	^Q	0x31	1	0x51	Q	0x71	q
0x12	Reserved	0x32	2	0x52	R	0x72	r
0x13	^S	0x33	3	0x53	S	0x73	s
0x14	Reserved	0x34	4	0x54	T	0x74	t
0x15	^U	0x35	5	0x55	U	0x75	u
0x16	^V	0x36	6	0x56	V	0x76	v
0x17	Reserved	0x37	7	0x57	W	0x77	w
0x18	Reserved	0x38	8	0x58	X	0x78	x
0x19	Reserved	0x39	9	0x59	Y	0x79	y
0x1a	EOT	0x3a	: (colon)	0x5a	Z	0x7a	z
0x1b	Reserved	0x3b	Undefined	0x5b	] (open bracket)	0x7b	Undefined
0x1c	Reserved	0x3c	Undefined	0x5c	\ (back slash)	0x7c	Undefined
0x1d	Reserved	0x3d	=	0x5d	[(close bracket)	0x7d	Undefined
0x1e	Reserved	0x3e	Undefined	0x5e	^	0x7e	Undefined
0x1f	Reserved	0x3f	?	0x5f	_ (under score)	0x7f	Undefined

Note 1: Characters ranges from '0x09' to '0x0D' (escape characters) will be treated as white space.

Note 2: The characters marked 'Reserved' above are the characters reserved for future system development. Unpredicted behavior may occur if these characters are used.

4.2. DEFAULT ABBREVIATION LIST

The list below is stored in WTS701 and used for all incoming text.

Note also that there is no abbreviation support in both ^U1 and ^U2 modes.

"Sen","Senator",
"Rep","Representative",
"Jan","January",
"Feb","February",
"Mar","March",
"Apr","April",
"Jun","June",
"Jul","July",
"Aug","August",
"Sep","September",
"Sept","September",
"Oct","October",
"Nov","November",
"Dec","December",
"Mon","monday",
"Tue","tuesday",
"Wed","wednesday",
"Thu","thursday",
"Fri","friday",
"Sat","saturday",
"Sun","sunday",
"A.M","A M",
"P.M","P M",
"a.m","A M",
"p.m","P M",

"pm","P M",
"PM","P M",
"PhD","P H D",
"Mr","Mister",
"Mrs","Missus",
"Ms","Miz",
"Dr","Doctor",
"Jr","Junior",
"Esq","Esquire",
"Pres","President",
"Prof","Professor",
"Sgt","Sargeant",
"Lt","Lieutenant",
"Lieut","Lieutenant",
"Maj","Major",
"MPH","mile per hour",
"Col","Colonel",
"Gen","General",
"Sr","Senior",
"St","Street",
"Av","Avenue",
"Ave","Avenue",
"Rd","Road",
"Rt","Route",
"Rte","Route",
"Blvd","Boulevard",
"Terr","Terrace",
"Exwy","Expressway",
"Expwy","Expressway",
"Drwy","Driveway",
"Dept","Department",
"Ct","Court",
"AL","Alabama",

"AK","Alaska",
"AZ","Arizona",
"AR","Arkansas",
"CA","California",
"Cal","California",
"Calif","California",
"Cal","California",
"DE","Delaware",
"FL","Florida",
"Fla","Florida",
"GA","Georgia",
"IEEE","I triple E",
"ID","Idaho",
"IL","Illinois",
"IN","Indiana",
"IA","Iowa",
"KS","Kansas",
"KY","Kentucky",
"IL","Illinois",
"MA","Massachusetts",
"MI","Michigan",
"MS","Mississippi",
"MN","Minnesota",
"MO","Missouri",
"MT","Montana",
"NV","Nevada",
"NH","New Hampshire",
"NM","New Mexico",
"NY","New York",
"NJ","New Jersey",
"NC","North Carolina",
"ND","North Dakota",
"OH","Ohio",

"PA","Pennsylvania",
"RI","Rhode Island",
"SC","South Carolina",
"SD","South Dakota",
"TN","Tennessee",
"TX","Texas",
"Tex","Texas",
"UT","Utah",
"VT","Vermont",
"VA","Virginia",
"WA","Washington",
"WV","West Virginia",
"WI","Wisconsin",
"WY","Wyoming",
"N","North",
"S","South",
"E","East",
"W","West",
"LA","Los Angeles",
"SF","San Francisco",
"SJ","San Jose",
"Ans","Answer",
"Q","Question",
"Asst","Assistant",
"Atty","Attorney",
"Bldg","Building",
"asap","ASAP",
"cc","CC",
"cm","centimeters",
"mm","millimeters",
"ft","feet",
"yd","yards",
"yds","yards",

"pt","pints",
"pts","pints",
"qt","quarts",
"qts","quarts",
"km","kilometers",
"mi","miles",
"deg","degrees",
"cu","cubic",
"diam","diameter",
"FAX","fax",
"Geog","geography",
"govt","government",
"min","minute",
"mins","minutes",
"hr","hour",
"hrs","hours",
"yr","year",
"yrs","years",
"doz","dozen",
"Inc","incorporated",
"KW","kilowatt",
"lb","pounds",
"lbs","pounds",
"mngr","manager",
"Nat","National",
"NASDAQ","nasdaq",
"ok","OK",
"oz","ounces",
"Pl","place",
"rcvd","received",
"rpm","RPM",
"tel","telephone",
"tv","TV",

"vs","versus",

"U.S","yoo ess",

"U.S.A","yoo ess ay",

"VISA","veeza",

"w","with",

"wrt","with respect to",

"wk","week",

"Xmas","Christmas",

"etc","etcetra",

"c/o","care of",

"w/o","without",

"III","the third",

"WYSIWYG","whisywig",

"THE","the"

5. VERSION HISTORY

VERSION	DATE	PAGE	DESCRIPTION
1.00	June 2003	All	Initial Version

Headquarters
No. 4, Creation Rd. III
Science-Based Industrial Park,
Hsinchu, Taiwan
TEL: 886-3-5770066
FAX: 886-3-5665577
http://www.winbond.com.tw/

Winbond Electronics Corporation America
2727 North First Street, San Jose,
CA 95134, U.S.A.
TEL: 1-408-9436666
FAX: 1-408-5441797
http://www.winbond-usa.com/

Winbond Electronics (Shanghai) Ltd.
27F, 299 Yan An W. Rd. Shanghai,
200336 China
TEL: 86-21-62365999
FAX: 86-21-62356998

Taipei Office
9F, No. 480, Pueiguang Rd.
Neihu District
Taipei, 114 Taiwan
TEL: 886-2-81777168
FAX: 886-2-87153579

Winbond Electronics Corporation Japan
7F Daini-ueno BLDG. 3-7-18
Shinyokohama Kohokuku,
Yokohama, 222-0033
TEL: 81-45-4781881
FAX: 81-45-4781800

Winbond Electronics (H.K.) Ltd.
Unit 9-15, 22F, Millennium City,
No. 378 Kwun Tong Rd.,
Kowloon, Hong Kong
TEL: 852-27513100
FAX: 852-27552064