

Overview

The Scriptel ScripTouch-STN protocol was designed to transfer signature data with a minimal number of “keystrokes” to overcome the lengthy polling intervals associated with Human Interface Devices. There are four key components to the protocol:

1. A radix-23 encoding scheme is used to convert coordinate values from 0 to 499 into two “digits₂₃”.
2. The most significant “digit₂₃” and the least significant “digit₂₃” are converted to key codes using different encoding tables to ensure that both “digits₂₃” can be transmitted simultaneously. The encoding tables were carefully designed to avoid key codes which may be consumed and acted upon by the operating system.
3. The X coordinate key codes are transmitted with the “Shift” indicator set TRUE while the Y coordinates are transmitted with the “Shift” indicator set FALSE. This virtually eliminates the need for “no key pressed” reports while providing the receiver a simple mechanism to detect lost key codes i.e. a coordinate point will always consist of two “shifted” key codes followed by two “un-shifted” key codes.
4. Three additional key codes are used to indicate “start”, “pen up”, and “complete” conditions.

When enumerated, the hardware advertises to the operating system that the only language supported is US English (Language ID 0x0409). While the key codes produced may work with other languages, this document and the character mappings referenced in this document assume the key codes have been mapped by the operating system to US English.

Supported Devices

The ScripTouch-STN protocol is supported by the following devices:

- **ST1401-STN**
- **ST1501-STN**
- **ST1526-STN**
- **ST1551-STN**

Signature Preamble

When a signature transmission is started, an identifying sequence is sent to the host before the first coordinate data. This sequence consists of:

1. a “start” character
2. the string “STSIGN”
3. a “pen up” character
4. a variable length string containing the protocol version – currently “A”
5. a “pen up” character
6. a variable length string containing the device model – e.g. “ST1501-STN”
7. a “pen up” character
8. a variable length string containing the firmware version – e.g. “02.00.00”
9. a “pen up” character

~STSIGN<space>A<space>ST1501-STN<space>02.00.00<space>

Signature Coordinates

As mentioned in the overview, coordinates are encoded using a radix-23 number system. Decoding the coordinate value is simply a matter of reversing the process. As an example, a single coordinate may appear to the application as

%_o[

Using the tables at the end of this document, we can decode these characters as follows:

% == 15 _ == 18 o == 7 [== 19

which produces a coordinate of

$X = (15 * 23) + 18 = 363$ $Y = (7 * 23) + 19 = 180$

To indicate the end of a line segment, a “pen up” character is inserted into the coordinate stream:

%_o[<space> %_o8

Signature Postscript

The signature is defined as complete when the user selects and releases the OK button on the signature pad. This causes a special “complete” character to be reported to the host. There may or may not be a “pen up” reported prior to “complete”.

`%_o[<space> %_o8 <space>`` or `%_o[<space> %_o8``

Signature Example

Below is an example of a VERY short signature. The special characters are black, the preamble fields are blue, the X coordinates are green, and the Y coordinates are red.

`~STSIGN<space>A<space>ST1501-STN<space>02.00.00<space>ON3nOR3jOV3fO$1\
O:14QN1vQ$1nQ_1lQ>1j<space>``

Complication 1

The delivery order of simultaneously pressed keys is not guaranteed by the USB HID specification. This means the coordinate from the previous example could have been presented to the application as:

`%_o[`
or `%_[o`
or `_%[o`
or `_%[3`

The application must simply ensure that it has two “shifted” key codes followed by two “un-shifted” key codes before decoding the coordinate value.

Complication 2

Some operating systems, including at least some versions of Windows and Linux, track the state of the Caps Lock key as a global setting. When the Caps Lock is active, “normal” alphabetic characters are capitalized, “shifted” alphabetic characters are lower-case, and all other characters are unaffected. The simplest way for the application to detect the Caps Lock state is to examine the “STSIGN” field in the signature preamble. This field is sent with “Shift” active so if the Caps Lock state is also active the application will see “stsign” instead of the expected “STSIGN”. The following tables list the decoding requirements for both the Caps Lock on and Caps Lock off states.

Most Significant Digits

		US English Key Code Translation							
		Shift On (X Values)				Shift Off (Y Values)			
		CapsLock Off		CapsLock On		CapsLock Off		CapsLock On	
Value (decimal)	Key Code (hex)	hex	Char	hex	Char	hex	Char	hex	Char
0	04	41	A	61	a	61	a	41	A
1	06	43	C	63	c	63	c	43	C
2	08	45	E	65	e	65	e	45	E
3	0a	47	G	67	g	67	g	47	G
4	0c	49	I	69	i	69	i	49	I
5	0e	4b	K	6b	k	6b	k	4b	K
6	10	4d	M	6d	m	6d	m	4d	M
7	12	4f	O	6f	o	6f	o	4f	O
8	14	51	Q	71	q	71	q	51	Q
9	16	53	S	73	s	73	s	53	S
10	18	55	U	75	u	75	u	55	U
11	1a	57	W	77	w	77	w	57	W
12	1c	59	Y	79	y	79	y	59	Y
13	1e	21	!	21	!	31	1	31	1
14	20	23	#	23	#	33	3	33	3
15	22	25	%	25	%	35	5	35	5
16	24	26	&	26	&	37	7	37	7
17	26	28	(28	(39	9	39	9
18	2e	2b	+	2b	+	3d	=	3d	=
19	30	7d	}	7d	}	5d]	5d]
20	34	22	“	22	“	27	’	27	’
21	36	3c	<	3c	<	2c	,	2c	,

Least Significant Digits

		US English Key Code Translation							
		Shift On (X Values)				Shift Off (Y Values)			
		CapsLock Off		CapsLock On		CapsLock Off		CapsLock On	
Value (decimal)	Key Code (hex)	hex	Char	hex	Char	hex	Char	hex	Char
0	05	42	B	62	b	62	b	42	B
1	07	44	D	64	d	64	d	44	D
2	09	46	F	66	f	66	f	46	F
3	0b	48	H	68	h	68	h	48	H
4	0d	4a	J	6a	j	6a	j	4a	J
5	0f	4c	L	6c	l	6c	l	4c	L
6	11	4e	N	6e	n	6e	n	4e	N
7	13	50	P	70	p	70	p	50	P
8	15	52	R	72	r	72	r	52	R
9	17	54	T	74	t	74	t	54	T
10	19	56	V	76	v	76	v	56	V
11	1b	58	X	78	x	78	x	58	X
12	1d	5a	Z	7a	z	7a	z	5a	Z
13	1f	40	@	40	@	32	2	32	2
14	21	24	\$	24	\$	34	4	34	4
15	23	5e	^	5e	^	36	6	36	6
16	25	2a	*	2a	*	38	8	38	8
17	27	29)	29)	30	0	30	0
18	2d	5f	_	5f	_	2d	-	2d	-
19	2f	7b	{	7b	{	5b	[5b	[
20	31	7c		7c		5c	\	5c	\
21	33	3a	:	3a	:	3b	;	3b	;
22	37	3e	>	3e	>	2e	.	2e	.

Special Functions

		US English Key Code Translation			
		CapsLock Off		CapsLock On	
Function	Key Code (hex)	hex	Char	hex	Char
Pen up	2c	20	<space>	20	<space>
Start	35	7e	~	7e	~
Complete	35	60	`	60	`

Magnetic Card Reader

On models equipped with a magnetic card reader, card data is returned immediately after a successful card read. The track data is converted to ASCII in its entirety, wrapped with a preamble and termination character, and transmitted to the host as a series of keycodes. It is the user's responsibility to determine the presence and/or validity of the track data.

The format of the magnetic card data transfer is as follows:

1. the string "!STCARD"
2. a <space> character (0x20)
3. a variable length string containing the protocol version – currently "A"
4. a <space> character (0x20)
5. Track 1 data if present and read successfully
6. Track 2 data if present and read successfully
7. Track 3 data if present and read successfully
8. a <return> character (0x0D)

!STCARD<space>A<space><TRACK1><TRACK2><TRACK3><return>

!STCARD<space>A<space><TRACK2><return>

!STCARD<space>A<space><TRACK1><TRACK2><return>