USBtoUSB

USB Serial Port to USB Keyboard and Mouse Action

User Manual

Thank you for purchasing the model USBtoUSB Adapter.

HAGSTROM ELECTRONICS, INC.

is pleased that you have selected this product for your application. This unit is may be used a variety of ways in order to meet your specific requirements. Please take a few minutes to read this manual before using your USBtoUSB.

If you have any questions about the use of the USBtoUSB not covered in this manual, please contact us directly.

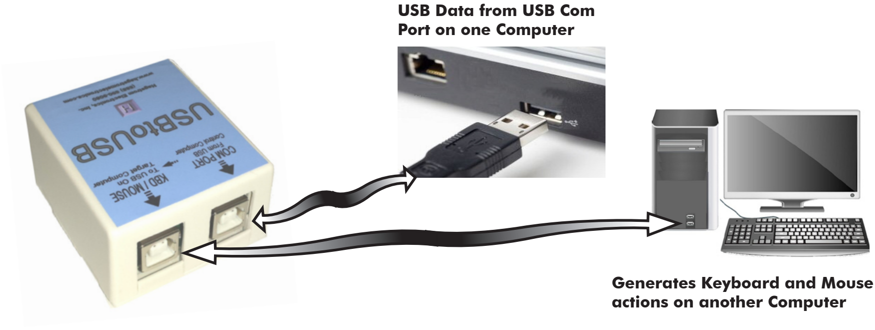
Please send an email to,

sales@hagstromelectronics.com

We respond to all email requests within one business day.

CONTENTS

Introduction to the USBtoUSB	3
Translation Modes of the USBtoUSB	3
ASCII Mode	3
Extended ASCII Mode	4
Key Number Mode	6
Additional Key Number Mode Commands	8
Key Number Mode Examples	9
Producing Mouse Action on the Target Computer	10
Mouse Packet Examples	12
Sending Serial Control Commands	13
USBtoUSB Configuration Program	14
USBtoUSB Connections & Specifications	17


USB Data from USB Com Port on one Computer
USBtoUSB
COM PORT From USB Control Computer
KBD / MOUSE To USB On Target Computer
Generates Keyboard and Mouse actions on another Computer

Questions or Comments?
Please send us an email!
www.hagstromelectronics.com
email: **sales@hagstromelectronics.com**

Introduction to the USBtoUSB

The USBtoUSB Keyboard/Mouse device is a product which emulates a USB Com Port on one computer and a USB keyboard and mouse on a target computer. Commands sent from the computer with the Com Port produce keyboard and mouse actions on the target computer.

The USB target port of this device can be connected to any USB port that supports a standard USB Keyboard and Mouse. No special drivers are required. The USBtoUSB Com Port also uses standard drivers.

The USBtoUSB is configurable for three different modes of serial data translation. This unit is shipped in ASCII mode, but can be changed to other modes using the supplied "USBtoUSB.EXE" program.

Translation Modes of the USBtoUSB

The USBtoUSB receives data from it's USB Com Port and translates that data into USB keystrokes and mouse action on another computer. USB Com Port data can be translated using one of three modes. ASCII Mode, Extended ASCII Mode, or Key Number Mode.

ASCII MODE (Default Mode)

The ASCII mode is the default factory setting for the method of data translation. In this mode, printable ASCII characters are sent to the USB Com Port as a one byte value, each in the range of 0x00 to 0x7F (0 to 127 decimal). Every character received in this mode generates it's corresponding USB keystroke on the computer where the USBtoUSB KBD/MOUSE port is connected. See the ASCII table on the next page for a list of the characters recognized and translated for ASCII mode.

Example: If a one byte value of 0x41 (decimal 65) is sent to the USB Com Port in this mode, a capital "A" character will be produced as a keystroke on the target computer's USB port.

ASCII characters sent to the Com Port which are out of the 0x00 to 0x7F range will be ignored in this mode.

ASCII Mode Translation Table

The ASCII mode table below lists the standard ASCII characters to be sent to the USB Com Port for producing a corresponding keystroke on the target computer when processed by the USBtoUSB unit.

ASCII Mode Translation Table

Value		Character	Value		Character	Value		Character	Value		Character
Dec	Hex		Dec	Hex		Dec	Hex		Dec	Hex	
00	00	none	32	20	Space	64	40	@	96	60	`
01	01	none	33	21	!	65	41	A	97	61	a
02	02	none	34	22	“	66	42	B	98	62	b
03	03	none	35	23	#	67	43	C	99	63	c
04	04	none	36	24	$	68	44	D	100	64	d
05	05	none	37	25	%	69	45	E	101	65	e
06	06	none	38	26	&	70	46	F	102	66	f
07	07	none	39	27	‘	71	47	G	103	67	g
08	08	Backspace	40	28	(	72	48	H	104	68	h
09	09	Tab	41	29	)	73	49	I	105	69	i
10	0A	none	42	2A	*	74	4A	J	106	6A	j
11	0B	none	43	2B	+	75	4B	K	107	6B	k
12	0C	none	44	2C	,	76	4C	L	108	6C	l
13	0D	Return	45	2D	–	77	4D	M	109	6D	m
14	0E	none	46	2E	.	78	4E	N	110	6E	n
15	0F	none	47	2F	/	79	4F	O	111	6F	o
16	10	none	48	30	0	80	50	P	112	70	p
17	11	none	49	31	1	81	51	Q	113	71	q
18	12	none	50	32	2	82	52	R	114	72	r
19	13	none	51	33	3	83	53	S	115	73	s
20	14	none	52	34	4	84	54	T	116	74	t
21	15	none	53	35	5	85	55	U	117	75	u
22	16	none	54	36	6	86	56	V	118	76	v
23	17	none	55	37	7	87	57	W	119	77	w
24	18	none	56	38	8	88	58	X	120	78	x
25	19	none	57	39	9	89	59	Y	121	79	y
26	1A	none	58	3A	:	90	5A	Z	122	7A	z
27	1B	Esc	59	3B	;	91	5B	[	123	7B	{
28	1C	none	60	3C	<	92	5C	\	124	7C	|
29	1D	none	61	3D	=	93	5D	]	125	7D	}
30	1E	none	62	3E	>	94	5E	^	126	7E	~
31	1F	none	63	3F	?	95	5F	-	127	7F	none

Extended ASCII Mode

The USBtoUSB Extended ASCII Mode of operation allows for the standard ASCII character set (values 0x00 to 0x7F), and an extended ASCII character set (values 0x80 to 0xFF). When a byte is sent to the USB Com Port in Extended ASCII mode, it produces a USB keystroke on the target PC which corresponds to the table on the next page.

Extended ASCII Mode Translation Table

Value		Character	Value		Character	Value		Character	Value		Character
Dec	Hex		Dec	Hex		Dec	Hex		Dec	Hex	
00	00	none	32	20	Space	64	40	@	96	60	`
01	01	none	33	21	!	65	41	A	97	61	a
02	02	none	34	22	"	66	42	B	98	62	b
03	03	none	35	23	#	67	43	C	99	63	c
04	04	none	36	24	$	68	44	D	100	64	d
05	05	none	37	25	%	69	45	E	101	65	e
06	06	none	38	26	&	70	46	F	102	66	f
07	07	none	39	27	'	71	47	G	103	67	g
08	08	Backspace	40	28	(	72	48	H	104	68	h
09	09	Tab	41	29	)	73	49	I	105	69	i
10	0A	none	42	2A	*	74	4A	J	106	6A	j
11	0B	none	43	2B	+	75	4B	K	107	6B	k
12	0C	none	44	2C	,	76	4C	L	108	6C	l
13	0D	Return	45	2D	-	77	4D	M	109	6D	m
14	0E	F11	46	2E	.	78	4E	N	110	6E	n
15	0F	F12	47	2F	/	79	4F	O	111	6F	o
16	10	none	48	30	0	80	50	P	112	70	p
17	11	F1	49	31	1	81	51	Q	113	71	q
18	12	F2	50	32	2	82	52	R	114	72	r
19	13	F3	51	33	3	83	53	S	115	73	s
20	14	F4	52	34	4	84	54	T	116	74	t
21	15	F5	53	35	5	85	55	U	117	75	u
22	16	F6	54	36	6	86	56	V	118	76	v
23	17	F7	55	37	7	87	57	W	119	77	w
24	18	F8	56	38	8	88	58	X	120	78	x
25	19	F9	57	39	9	89	59	Y	121	79	y
26	1A	F10	58	3A	:	90	5A	Z	122	7A	z
27	1B	Esc	59	3B	;	91	5B	[	123	7B	{
28	1C	none	60	3C	<	92	5C	\	124	7C	|
29	1D	none	61	3D	=	93	5D	]	125	7D	}
30	1E	none	62	3E	>	94	5E	^	126	7E	~
31	1F	none	63	3F	?	95	5F	-	127	7F	none

Value		Character	Value		Character	Value		Character	Value		Character
Dec	Hex		Dec	Hex		Dec	Hex		Dec	Hex	
128	80	€	160	A0	nb Space	192	C0	À	224	E0	à
129	81	none	161	A1	¡	193	C1	Á	225	E1	á
130	82	‚	162	A2	¢	194	C2	Â	226	E2	â
131	83	ƒ	163	A3	£	195	C3	Ã	227	E3	ã
132	84	„	164	A4	€	196	C4	Ä	228	E4	ä
133	85	…	165	A5	¥	197	C5	Å	229	E5	å
134	86	†	166	A6	¦	198	C6	Æ	230	E6	æ
135	87	‡	167	A7	§	199	C7	Ç	231	E7	ç
136	88	ˆ	168	A8	¨	200	C8	È	232	E8	è
137	89	‰	169	A9	©	201	C9	É	233	E9	é
138	8A	Š	170	AA	ª	202	CA	Ê	234	EA	ê
139	8B	‹	171	AB	«	203	CB	Ë	235	EB	ë
140	8C	Œ	172	AC	¬	204	CC	Ì	236	EC	ì
141	8D	none	173	AD	none	205	CD	Í	237	ED	í
142	8E	Ž	174	AE	®	206	CE	Î	238	EE	î
143	8F	none	175	AF	¯	207	CF	Ï	239	EF	ï
144	90	none	176	B0	°	208	D0	Ð	240	F0	ð
145	91	‘	177	B1	±	209	D1	Ñ	241	F1	ñ
146	92	’	178	B2	²	210	D2	Ò	242	F2	ò
147	93	“	179	B3	³	211	D3	Ó	243	F3	ó
148	94	”	180	B4	´	212	D4	Ô	244	F4	ô
149	95	•	181	B5	µ	213	D5	Õ	245	F5	õ
150	96	–	182	B6	¶	214	D6	Ö	246	F6	ö
151	97	—	183	B7	·	215	D7	×	247	F7	÷
152	98	˜	184	B8	¸	216	D8	Ø	248	F8	ø
153	99	™	185	B9	¹	217	D9	Ù	249	F9	ù
154	9A	š	186	BA	º	218	DA	Ú	250	FA	ú
155	9B	›	187	BB	»	219	DB	Û	251	FB	û
156	9C	œ	188	BC	¼	220	DC	Ü	252	FC	ü
157	9D	none	189	BD	½	221	DD	Ý	253	FD	ý
158	9E	ž	190	BE	¾	222	DE	ß	254	FE	þ
159	9F	Ÿ	191	BF	¿	223	DF	à	255	FF	ÿ

When using the Extended ASCII mode, characters sent to the USB Com Port are in the range of 0x00 to 0xFF (0 to 255 decimal). The corresponding keystrokes from the page 5 table will be generated.

Extended ASCII Mode allows for the ***ANSI/ISO Latin-1*** character group in the range of 0x80 to 0xFF (128 to 255 decimal) to be produced as well as the standard 0x00 through 0x7F ASCII codes. Additionally in this mode, the F1-F10 keys are supported for the character range of codes 0x11 to 0x1A values respectively. F11 and F12 keys are sent for Com Port characters of 0x0E and 0x0F in the Extended ASCII mode.

Example 1: If the value 0x41 (decimal 65) is received by the USB Com Port in this mode, a capital “A” character will be produced on the target computer’s USB port as a keystroke.

Example 2: If the value 0xA9 (decimal 169) is received by the USB Com Port in this mode, a “©” character will be produced on the target computer’s USB port as a keystroke.

Key Number Mode

The Key Number Mode provides complete control of the generation of the make (activation) and break (deactivation) of any standard keyboard key on the target computer. In this mode, a single byte sent to the USB Com Port will command the make or break of a specific keyboard key at the target computer. This mode allows for generation of any keystroke or combination of keystrokes on the target computer.

In general, to make a key (generate a press of a specific key), a one byte value between 0x00 and 0x7F is sent to the Com Port of the USBtoUSB. The key specified will be seen as held down on the target computer just as if someone was physically holding down that key on a keyboard.

Each time a make is done for a key, a corresponding break (release) of that key must be done at a later time to deactivate it. The break code for a key is the same value as the make code plus 0x80. The break code releases the key that was activated earlier by a make code. See the following table for Key Number Mode make and break codes.

US Key Number Table (Decimal Values)

An international key number table is available on the CD provided with the USBtoUSB.

Key	Make	Break
`~	01	129
1!	02	130
2@	03	131
3#	04	132
4$	05	133
5%	06	134
6^	07	135
7&	08	136
8*	09	137
9(	10	138
0)	11	139
-_	12	140
=+	13	141
BS	15	143
Tab	16	144
Q	17	145
W	18	146
E	19	147
R	20	148
T	21	149
Y	22	150
U	23	151
I	24	152
O	25	153
P	26	154
[{	27	155
]}	28	156
\\|	29	157
Caps	30	158
A	31	159
S	32	160
D	33	161
F	34	162
G	35	163
H	36	164

Key	Make	Break
J	37	165
K	38	166
L	39	167
;:	40	168
‘“	41	169
Enter	43	171
L Shift	44	172
Z	46	174
X	47	175
C	48	176
V	49	177
B	50	178
N	51	179
M	52	180
,<	53	181
.>	54	182
/?	55	183
R Shift	57	185
L Ctrl	58	186
L Alt	60	188
Space	61	189
R Alt	62	190
R Ctrl	64	192
L Win	70	198
R Win	71	199
Win APL	72	200
Insert	75	203
Delete	76	204
L Arrow	79	207
Home	80	208
End	81	209
Up Arrow	83	211
Dn Arrow	84	212
Page Up	85	213
Page Dn	86	214

Key	Make	Break
R Arrow	89	217
NumLock	90	218
7 (Num)	91	219
4 (Num)	92	220
1 (Num)	93	221
/ (Num)	95	223
8 (Num)	96	224
5 (Num)	97	225
2 (Num)	98	226
0 (Num)	99	227
* (Num)	100	228
9 (Num)	101	229
6 (Num)	102	230
3 (Num)	103	231
. (Num)	104	232
- (Num)	105	233
+ (Num)	106	234
Enter (Num)	108	236
Esc	110	238
F1	112	240
F2	113	241
F3	114	242
F4	115	243
F5	116	244
F6	117	245
F7	118	246
F8	119	247
F9	120	248
F10	121	249
F11	122	250
F12	123	251
Prt Scr	124	252
Scrl Lk	125	253
Pause/Break	126	254

When sending data to the Com Port on the USBtoUSB to generate keystrokes at the target computer in Key Number Mode, use the values shown above to produce the desired “make” and “break” actions for the corresponding key.

By using the make and break commands in Key Number Mode, any sequence that can by manually entered on a keyboard can be produced with the USBtoUSB. Use this Key Number Mode to emulate single keystrokes or combinations such as Ctrl+F1, or Shift+Alt+F5, etc.

Additional Key Number Mode Control Commands

The USBtoUSB features two additional commands for keyboard action. The first command provides a way to clear the keyboard buffer and is useful for ensuring that no keys are left in the “ON” state. The second command allows for the polling of the keyboard status LED states. This polling command is useful for checking the Shift Lock state, or for making sure of the Num Lock state before using Num Lock affected keys.

0x38 - USB Buffer Clear Command. Sending the byte 0x38 to the USBtoUSB Com Port port results in the device’s internal USB keyboard buffer being cleared. Use of this command ensures that all made keys currently in the keyboard buffer are released.

0x7F - Status LED Read Command. Sending a code 0x7F to the USBtoUSB Com Port results in the return of a character in the range of ASCII “0” - “7”. The USBtoUSB Com Port response character reflects the current state of the Scroll Lock, Caps Lock, and Num Lock LEDs on the target computer system as listed in the chart below.

ASCII Response	Scroll Lock Status	Caps Lock Status	Num Lock Status
“0”	Off	Off	Off
“1”	Off	Off	On
“2”	Off	On	Off
“3’	Off	On	On
“4”	On	Off	Off
“5”	On	Off	On
“6”	On	On	Off
“7”	On	On	On

Key Number Mode Examples

Generation of keystrokes on the target computer is done through the sending of special one byte codes to the USBtoUSB Com Port when in the Key Number Mode.

Each standard key of the computer keyboard is assigned a “make” code to emulate the press of a key, and a “break” code, which results in the release of the key. The Key Number Table on page 7 lists each of the supported keys and their corresponding make and break codes.

From the keycode table, the value of **2** (0x02) can be sent to the Com Port on the unit to generate the “make” or press and hold of the “1” key on the target computer. Sending a byte to the USBtoUSB Com Port with a value of **130** (0x82) will result in the release of the “1” key on the target computer.

Important: For any key that has been previously sent a “make” code, a “break” code of that key must be sent at a later time. Failure to send the corresponding break code will leave the key in the down state on the target computer, which may result in unintended keystrokes when new actions are sent. The last key left in the make state without a break will generally repeat until the break code is sent. Never command more than 60 keys in the “make” state at the same time as this will exceed the standard USB protocol keyboard buffer length for the USBtoUSB device.

The use of make and break codes allows the user to create virtually any keystroke combination. For example, if a sequence of CTRL+ALT+F1 was needed, the following decimal codes would be sent; **58** (Make L Ctrl), **60** (Make L Alt), **112** (Make F1), **240** (Break F1), **188** (Break L Alt), **186** (Break L Ctrl). Note that the values shown in the example are the decimal values of a single byte to be sent for each of the key actions for make and break.

Producing Mouse Action on the Target PC

The USBtoUSB is capable of controlling the mouse cursor and mouse button states of the target computer through a special six byte command sequence. This same command sequence sent to the Com Port of the USBtoUSB can be used in any of the operating modes.

The six byte mouse control sequence is construced as follows:

Byte#1	Byte#2	Byte#3	Byte#4	Byte#5	Byte#6
[Start]	[X PosMS]	[X PosLS]	[Y PosMS]	[Y PosLS]	[Scroll/Buttons]

Byte #1 - Always a 0x00 value, indicates start of packet.

Byte #2 & #3 - A two byte value which dictates the X axis position of the mouse cursor on the screen. A value of 0 will send the cursor to the leftmost screen position. Byte #2 is the Most Siginificant (MS) byte of the X position value, and Byte #3 is the Least Significant (LS) part of the X position value.

Byte #4 & #5 - A two byte value which dicatates the Y axis position of the mouse cursor on the screen. A value of 0 will send the cursor to the top of the screen. Byte #4 is the MS byte of the Y position value, and Byte #5 is the LS part of the Y position value.

Byte #6 - Byte for Scroll Wheel movement and mouse button control. The upper 4 bits of this byte contain a signed value for scroll wheel movement, while the lower 3 bits command the mouse Left, Right, and Middle button activation and deactivation.

This Byte #6 is constructed as shown,

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
[Ws]	[W2]	[W1]	[W0]	[1]	[MM]	[MR]	[ML]

[Ws] - The sign of the scroll wheel movement. A zero in this bit commands up movement on the scroll wheel, a one in this bit produces down scroll wheel movement. Movement magnitude based on the signed value in Ws through W0 values.

[W2] - The most significant bit of the scroll wheel movement magnitude.

[W1] - Bit 1 of the three bit value for scroll wheel movement.

[W0] - Least significant bit of the scroll wheel movement value.

[1] - Bit 3 of the #6 byte is always “1”. Sending “0” for this bit will cause the entire mouse control packet to be ignored.

[MM] - State of the middle mouse button. Set to 1 for middle mouse button on, 0 for the button to be off.

[MR] - State of the right mouse button. Set to 1 for right mouse button on, 0 for the button to be off.

[ML] - State of the left mouse button. Set to 1 for left mouse button on, 0 for the button to be off.

NOTES - When sending this six byte mouse command packet to the USBtoUSB Com Port, the five bytes following the first 0x00 byte should be sent with no more than 50msec between each byte. The starting value 0x00 must always be followed by five additional bytes to complete the mouse control packet and avoid any confusion between mouse control and keystroke commands.

The scroll wheel magnitude command is a 4 bit signed value located in bits Ws through W0, and in the range of 0x00 to 0x07 for up movement, 0x0F to 0x08 for down movement.

The three mouse button bits [MM], [MR], and [ML] command the states of the mouse buttons as seen by the target computer. A “1” in the respective bit indicates the mouse button is on, while a “0” indicates the button is off. Be sure to command the release of any mouse buttons that were in the on state at a later time once the mouse button task has completed.

When sending a mouse control packet for cursor movement only, be sure to send all “0” values for the scroll wheel and mouse button bits, provided those options are not being used for mouse actions at that time.

Mouse Packet Examples

The packet below is an example of a packet sent to the USBtoUSB Com Port to command the cursor to a screen postion of X = 100, Y = 520.

```
0x00, 0x00, 0x64, 0x02, 0x08, 0x08
Start   XMS   XLS    YMS   YLS    Scroll
```

Sending the values shown above will cause the cursor to move to the screen position 100, 520. The position of the mouse cursor for the values sent is an absolute screen position and will be repeatable for a specific computer and screen resolution.

A screen position of 0,0 represents the upper left of the screen. To send the cursor to specific coordinates on the screen, use the 2 byte values for both X and Y in the packet described above.

The cursor position that results from a command packet will depend on the particular computer’s display settings. Moving the cursor to a desired screen position may require working with a few values to get the exact value for a specific application.

The following packet can be used to send the scroll wheel command of up one position to the computer.

How far the scroll moves on the computer screen depends on the scroll system settings of the target computer.

```
0x00, 0x00, 0x00, 0x00, 0x00, 0x18
```

In the next example, four Com Port mouse packets are sent to emulate a double left mouse click on the target computer.

Packet 1 - 0x00, 0x00, 0x00, 0x00, 0x00, 0x09 (left mouse button on). Delay around 150 msec after sending this packet for the system to see the button.

Packet 2 - 0x00, 0x00, 0x00 0x00, 0x00, 0x08 (left mouse button off). Delay 150 msec again for the system to see the button release and create time between the release and activation of the left mouse button in step 3 below.

Packet 3 - 0x00, 0x00, 0x00, 0x00, 0x00, 0x09 (left mouse button on). Delay around 150 msec for the system to see the button.

Packet 4 - 0x00, 0x00, 0x00, 0x00, 0x00, 0x08 (left mouse button off). No delay required after this release.

Cursor position, scroll wheel movement and button actions may all be implemented within the same packet. The examples above focus only on a particular action at a time for clarity. For instance, a command that sends both X and Y cusor movement can also send button information and/or scroll wheel movement.

Note that X and Y coordinates require the proper X and Y screen resolution set by the USBtoUSB.exe to obtain a 1:1 ratio beween the commanded values and the resulting cursor position.

Sending Serial Control Commands

The CD included with the unit contains source code examples in various programming languages to help the user get started on their own Com Port control software. Use these examples as a reference point for writing programs on the computer attached to the USBtoUSB Com Port connection.

The user's software should always open the USBtoUSB Com Port using 19,200 BAUD, 8 Bit characters, No Parity, and One Stop Bit.

When the USBtoUSB Com Port is attached to the user's computer, it will be assigned a particular Com Port Number. In order to send communication to the USBtoUSB, the proper Com Port number must be opened to communicate with the USBtoUSB device.

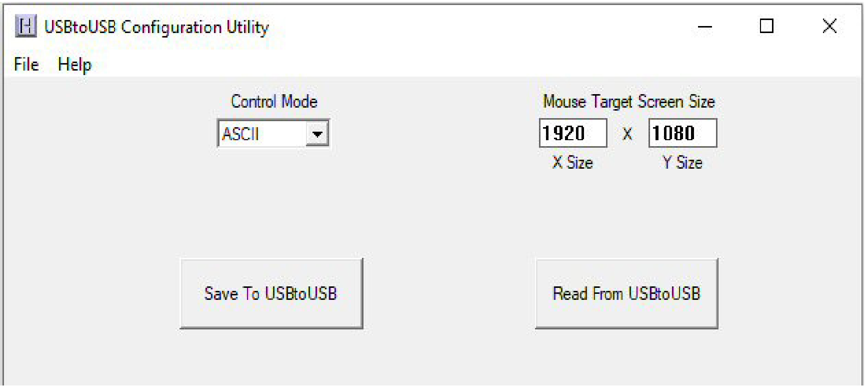
The included CD contains examples in various programming languages for sending commands to the USBtoUSB Com Port for the different modes of translation.

Use these programming examples as a starting point for the application software that will be used to control the keyboard and mouse actions through the USB com port to the target computer.

USBtoUSB Configuration Program

The unit features programmable settings for selecting the serial data translation mode and setting the target computer screen size information into the USBtoUSB unit for the user's application.

The included CD contains the program "USBtoUSB.exe". Copy the entire contents of the CD to a dedicated folder on the target computer's drive. Once copied, run "USBtoUSB.exe".


USBtoUSB Configuration Utility

Control Mode: ASCII

Mouse Target Screen Size: 1920 x 1080 (X Size, Y Size)

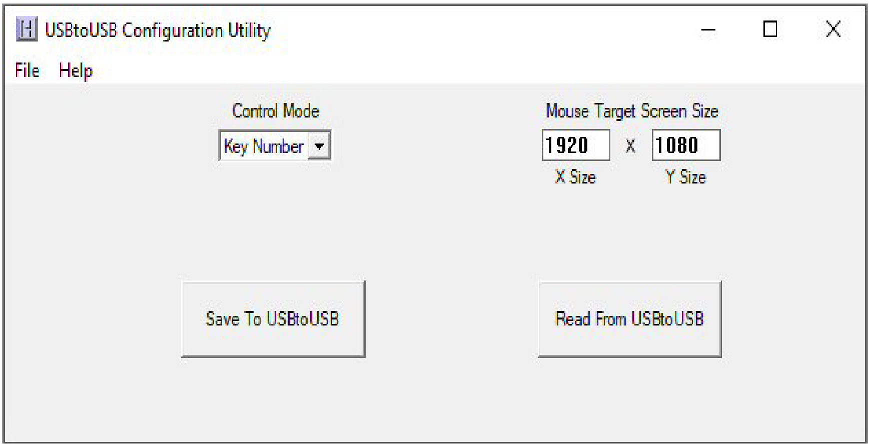
To configure the USBtoUSB, select the Control Mode from the dropdown box to choose the conversion mode. Select between ASCII (factory default), Extended ASCII, or Key Number Mode. See pages 3-9 for a description of each of the control modes.

Next, enter the screen X and Y size for the target computer’s display settings. The X and Y sizes are used to set the USBtoUSB internal storage such that the serial control for the mouse cursor will better match the target system. When the USBtoUSB.exe program is started, the X and Y values a are populated with the target computer’s current display settings. The user may change these if needed, but in general, the values selected by the program are the proper values to be used. (See note on page 16)

Saving the Configuration to the USBtoUSB

Once the desired parameters have been set on the USBtoUSB Configuration Utility Screen, it must be saved to the USBtoUSB unit to begin operation according to those settings.

Select the button “Save To USBtoUSB” to write the configuration to the unit. Once the configuration has been written, the USBtoUSB will restart and begin operating according to the selected mode.


USBtoUSB Configuration Utility

Control Mode: Key Number

Mouse Target Screen Size: 1920 X 1080

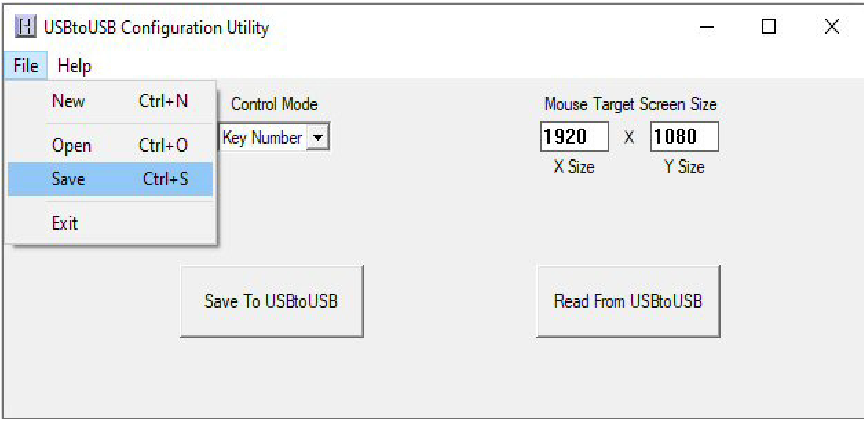
X Size: 1920

Y Size: 1080

The configuration may also be read from the USBtoUSB unit by selecting the button “Read From USBtoUSB”. The configuration in the USBtoUSB attached to the computer’s USB port will be read into the configuration parameters on the screen.

Once a configuration has been created, it is recommended that it be stored on the computer so that it may be recalled at a later time. Use the “File” menu to perform saving and opening of configurations.

Saving the configuration to a file on the computer provides an easy way to be able to recall the same configuration at a later time to save into additional USBtoUSB units.



USBtoUSB Configuration Utility

Control Mode: Key Number

Mouse Target Screen Size: 1920 x 1080 (X Size, Y Size)

Note about the X and Y size settings. *The information about the screen size is important for controlling the cursor with the mouse control packets. The program will show the current settings on the machine where the USBtoUSB.exe program is being run.*

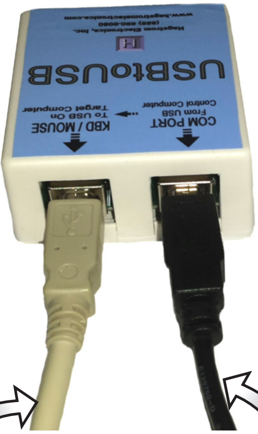
Assuming the program is being run on the target machine, no changes are required to the values the program populates. If the USBtoUSB will be run on a different computer, change the X and Y settings to match that system if they differ from the target machine’s screen resolution.

Custom USBtoUSB Options

We offer special modifications to our standard USBtoUSB unit to conform to your exact specifications. Customizations include different translation modes, different enclosure, etc. just to name a few options. Let us know if you have special requirements. Please email with your specific custom needs.

USBtoUSB Connections

Using the included USB cables, attach the Com Port USB connector to the computer that will be sending commands to the USBtoUSB. Attach the KBD/MOUSE connector to the target computer where the keyboard and mouse actions will be generated based on the commands.



This port is also to be plugged into the computer running the USBtoUSB.exe configuration program.

USB KBD/MOUSE Port, attach to the computer where the keyboard and mouse actions are to take place.

USB Com Port device, attach to the computer sending the serial commands.

USBtoUSB Specifications

Operating Voltage	**5 Volts DC +/- 5% (Powered from USB ports)**
Operating Current	**Less than 100 ma each port**
Operating Temp.	**0 to 70 Degrees C**
Computer Interface	**USB - Target Computer USB - Com Port Source**
Cable Length	**Maximum 15ft each port**
USB Com Port Protocol	**19200, 8 , N, 1**

Warranty

HAGSTROM ELECTRONICS, INC. warrants this product against defects in material or workmanship for a period of ONE YEAR from the original purchase date. We will repair or replace (at our option) the returned defective unit at no charge during this warranty period.

No responsibility is assumed for any special, incidental, or consequential damage resulting from the use of or inability to use this product. In no case is **HAGSTROM ELECTRONICS, INC.** to be liable for any amount which exceeds the purchase price of the unit, regardless of the claim.

No other warranty, written or verbal, is authorized. This warranty is applicable only to units sold in the United States. Units sold outside the United States are covered by a similar warranty.

Great care has been taken during the assembly, testing, and burn-in of your USBtoUSB to ensure its performance. If you have any questions, please send us an email.

customer service email*:* ***sales@hagstromelectronics.com***

NOTICE: The USBtoUSB product is designed to be used by technically oriented computer users.

HAGSTROM
ELECTRONICS, INC.

sales@hagstromelectronics.com
www.hagstromelectronics.com

V. 10.20