USB Universal CNC Controller User Guide 1.05 2023/03/26

Designed, built, and programmed in the USA.

The universal keyboard emulator puts full system control in your hand close to where you want to be.

The Universal CNC keyboard emulator offers 54 programmable commands in three groups of 18. Each command can support up to 16 keystrokes, allowing you to generate a very sophisticated command set with a single push of a button.

This device can be programmed with up to 54 direct access commands to be compatible with virtually any CNC control software that allows the use of the keyboard for most of its command control.

The device mimics a standard USB keyboard and can reproduce any key sequence that can be entered from a standard keyboard. Two 5-way joy-switches offer simplified motion control of the each of the CNC axis. The device also presents itself as a serial port (115200 BAUD) for programming customized keystrokes. The buttons can be programmed individually or in groups.

The keyboard emulator is preprogrammed with the most common Mach3 keyboard commands. All of the joy-switch manual motion commands are also compatible with Mach4. Mach4 also supports user macros activated by the keyboard.

Programming the universal keyboard emulator

The keyboard emulator can be easily programmed with macros to output any keyboard sequence as needed. The process isn't difficult but the rules are strict. The keyboard macros are stored in EEPROM and are retained and ready for use indefinitely. The keyboard sequences can be modified as many times as needed.

INSTALLATION:

The keyboard emulator presents itself as two different USB devices; Keyboard and Serial device:

1) It will appear as a generic keyboard. The computer will accept all input from it as though someone was typing at a standard keyboard. A PAUSE function permits pacing if it is required.
2) It will appear as a serial port device with a BAUD rate of 115200. The port number will be automatically assigned by the computer. It is through the serial port that the buttons are programmed as keyboard entries. The serial Port is accessed by using a terminal program such as PuTTY or the Monitor module in the Arduino IDE.

Both the keyboard and serial devices should register automatically when attached to the computer.

For programming purposes, the port number can be found by looking in the device manager window under "Ports (COM & LPT)". Chances are it is the only one. It will be labeled "Arduino Leonardo" (COM#). This is the port number you will use to program macros into the device. If it isn't the only device, you can unplug it and see which COM port goes away.

The easiest way to program it is to create a text file that contains the commands for all of the buttons you intend to use. Then copy the file to the COM port or into the terminal command line.

The command structure stars with the command "SB" followed by a button number and macro for that button

For example button #1, (which happens to be the left Joy-switch right position) could be programmed with the keys SHIFT Right Arrow. The sequence would be

```
SB :01,\KS,\KRA,\K@; // send a SHIFT Right Arrow. Or

SB :01,160,39,\K#; // send a SHIFT Right Arrow.
```

Either the mnemonic or the actual key value may be used

1) The **SB** initiates the programming/upload function.
2) The Colon is followed by the button number.
3) All entries must be followed by a comma with no spaces.
4) The \ backslash indicates the following will be a special character sequence.
5) The \KS mnemonic performs a SHIF key press and hold.
6) The \KRA is a mnemonic for keyboard Right Arrow. Again followed by a comma.

7) The \K@; indicates that this is the end of a macro line. All lines must be terminated by either the \K@; or the \K#;
8) Any trailing comment/text is ignored until the next colon (**:**) is found.
9) A trailing \K#; sequence is similar to the \K@ sequence in that it terminates the macro line. It also terminates the SB command and everything after it is ignored.
10) Everything following the semi-colon (**;**) is user comment and not processed.
11) The programmed macros are stored in EEPROM and can be reprogrammed as often as needed.

Text in a macro is done with double quotes, For example, the following:

```
SB :05,"I am Groot. ",\K#;
```

Will key out:

I am Groot.

Every time button 5 is pressed.

LAYOUT:

1) Two 5-way joy-switches: control for X++, X--, Y++, Y--, Z++, Z--, Feed++, Feed--. Emulates the arrow keys, Page UP, Page Down, A++ and A--. A++ is preprogrammed to 'F10' (Feed faster) And A-- is 'F11' Feed slower. The push switches are not programmed.

2) One Mode button: Provides Three states for the arrow keys; SHIFT, Normal, and CTRL. Used in conjunction with the 5-way switches the axis can be moved FAST, SLOW, or Step-Jog. The LEDs are sequenced with the Mode button. The LEDs clearly indicate the control characteristics of the Joy-switches.

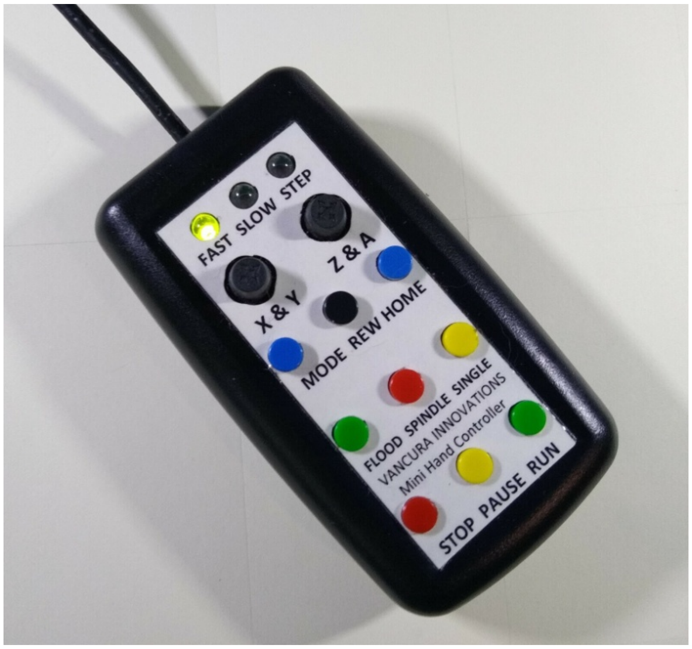
3) Eight Control buttons for the most frequently used Mach3 functions; REWIND, HOME, FLOOD Toggle, SPINDLE Toggle, SINGLE-STEP Toggle, STOP, PAUSE, and RUN.

4) Step-Jog function has both single step and auto-repeat feature (not found on any standard keyboard). This allows for moving the axis quickly and precisely to the desired position.

Note: Mach3 hand-wheel (TAB) must be set up to be Jog Mode = **Continuous**. The **Slow Jog Rate** should about 25%, and **Cycle Jog Step** needs to be set to about 0.025mm (0.001").

There are 60 macro buffers, each 16 character long. 54 of the buffers are directly accessible by one or two touch of the buttons. Nine spare buffers are accessible by concatenation allowing longer key sequences.

All of the non printable keys (SHIF, CTRL, ALT, etc.) are accessible allowing you to replicate any user key sequence



Button association:

Macro played (left, Center, Right LED ON)

```
Left joy switch (SW1)
     Down      =     1, 21, 41
     Up        =     2, 22, 42
     Right     =     3, 23, 43
     Left      =     4, 24, 44
     Push      =     5, 25, 45

Right joy switch (SW2)
     Down      =     6, 26, 46
     Up        =     7, 27, 47
     Right     =     8, 28, 48
     Left      =     9, 29, 49
     Push      =    10, 30, 50

     SW3       =    11, 31, 51
     SW4       =    12, 32, 52
     SW5       =    13, 33, 53
     SW6       =    14, 34, 54
     SW7       =    15, 35, 55
     SW8       =    16, 36, 56
     SW9       =    17, 37, 57
     SW10      =    18, 38, 58
     SW11      =    19, 39, 59
```

Physical layout

```
 ____________________
/                    \
| LED1  LED2  LED3   |
|                    |
| SW1         SW2    |
|                    |
|                    |
| SW3   SW4   SW5    |
|                    |
|                    |
| SW6   SW7   SW8    |
|                    |
|                    |
| SW9   SW10  SW11   |
\____________________/
```

Left LED = Macros 1-19 Selected
Center LED = Macros 20-39 Selected (must have key = \M2 or\M3 in each group)
Right LED = Macros 40-59 Selected (must have key = \M3 in each group)
Note: The default Mach3 program uses SW3 to toggle through the three LED macro modes.

Command Line Help:

After connecting to the keyboard emulator with a terminal program at 115200 BAUD, Send the command H for help and the following should appear:

```
 Command set:
  SD   = Set Auto Step Delay. (Range: 250ms to 5000ms)
  SL   = Set Scan Loop time. (Range: 10ms to 100ms)
  SR   = Set Auto step Rate. (Range: 10ms to 1000ms
  ST   = Set keyboard timeout (Range 1sec to 255sec).
  SB   = Set Buttons (Program with text string)
  D    = Set Default Values (Requires confirmation).
  F    = Programming Command Table
  G    = Sample Programming Text File
  H    = HELP (This listing).
  Q    = Quit / Exit.
  R    = Read Current and default Values.
  Z    = Debug Toggle.
  Command Format: XX=nnn<ret> or XX = nnn<ret> or XX nnn<ret>
```

Sample Programming File: Mach3 Keyboard Commands

Entering the command “G” should output the following sample file. It can be copied and pasted into a text document for easy editing. Once edited as desired, it can be copied and pasted into the terminal command buffer for programming the changes. The **SB** starts the upload process and the final **\K#** terminates it. Each macro buffer can be programmed individually or in groups or all at once. The macros need not be programmed in sequential order. Studying the sample file is probably the best way to understand the programming rules and procedure.

```
SB
:01,\KS,\RTA,\K@;    Shift RT Arrow
:02,\KS,\LTA,\K@;    Shift LT Arrow
:03,\KS,218,\K@;     Shift UP arrow
:04,\KS,217,\K@;     Shift Down arrow // SB
:05,"Vancura ",\KN10,\K@; // test
// Table 1, Row 2; Joy Switch Right
:06,204,\K@;         Shift A++    // Feed Faster
:07,203,\K@;         Shift A--    // Feed Slower
:08,\KS,211,\K@;     Shift Page UP
:09,\KS,214,\K@;     Shift Page Down
:10,"Innovations ",\KN20,\K@;  // Not used
//
// Table 1, row 3
:11,\M3,\K@;         MODE-3 SELECTER
:12,\KC,w,\Kc,\K@;   CTRL-W REWIND
:13,\KH,\K@;         <HOME>
:14,\KC,f,\Kc,\K@;   CTRL-F FLOOD ON/OFF
:15,198,\K@;         <F5> MOTOR ON/OFF
// Table 1, row 4
:16,\KA,n,\K@;       <ALT>N Single Step Toggle
:17,\KA,s,\K@;       <ALT>S STOP
:18,\ ,\K@;          SPACE BAR
:19,\KA,r,\Ka,\K@;   <ALT>R RUN
// === TABLE TWO ===
:20,\K@;             Not Used
//  Table 2
:21,\KU,215,\K@;     Right arrow      Sw 1 right
:22,\KU,216,\K@;     Left arrow       Sw 1 Left
:23,\KU,218,\K@;     UP Arrow         Sw 1 Up
:24,\KU,217,\K@;     DN arrow         Sw 1 Down
:25,\K@;             not used         Sw 1 Push
//
:26,\F11,\K@;        // Feed Faster   Sw 2 right
:27,\F12,\K@;        // Feed Slower   Sw 2 left
:28,211,\K@;         Page UP          Sw 2 Up
:29,214,\K@;         Page Down        Sw 2 Down
:30,\K@;             not used         Sw 2 Push
//
:31,\M3,\K@;         MODE-3 SELECT
:32,\KC,w,\Kc,\K@;   CTRL-W REWIND
:33,\KH,\K@;         <HOME>
:34,\KC,f,\Kc,\K@;   CTRL-F FLOOD ON/OFF
:35,198,\K@;          <F5> MOTOR ON/OFF
//
:36,\KA,n,\K@;       <ALT>N Single Step Toggle
:37,\KA,s,\K@;       <ALT>S STOP
```

```
:38,\ ,\K@;          SPACE BAR PAUSE
:39,\KA,r,\K@;       <ALT>R RUN
:40,\K@;             Not Used
//  === Table THREE ===
:41,\KC,215,\Kc,\Z,\K@; <CTRL> RT arrow
:42,\KC,216,\Kc,\Z,\K@; <CTRL> LT arrow
:43,\KC,218,\Kc,\Z,\K@; <CTRL> UP Arrow
:44,\KC,217,\Kc,\Z,\K@; <CTRL> DN arrow
:45,\K@;                Not used
//
:46,204,\K@;            // Feed Faster (SW2 RT)
:47,203,\K@;            // Feed Slower (SW2 LT)
:48,\KC,211,\Kc,\Z,\K@; <CTRL> Page UP (SW2 UP)
:49,\KC,214,\Kc,\Z,\K@; <CTRL> Page Down (SW2 DN)
:50,\K@; not used
//
:51,\M3,\K@;            MODE-3 SELECT
:52,\KC,w,\K@;          CTRL-W REWIND
:53,\KH,\K@;          <HOME>
:54,\KC,f,\Kc,\K@;    CTRL-F FLOOD ON/OFF
:55,198,\K@;          <F5>  MOTOR ON/OFF
//
:56,\KA,n,\K@;        <ALT>N Single Step Toggle
:57,\KA,s,\K@;        <ALT>S STOP
:58,\ ,\K@;           SPACE_BAR PAUSE
:59,\KA,r,\K@;        <ALT>R RUN
:60,\K#;              Not used
```

KEYBOARD MACRO COMMAND TABLE

Entering text: “ (double quote) Start & End an ASCII string
Includes: all Numbers, Letters, and all printable characters except “ (double quote)

Example: SB `:10,"I am Groot.",\K#;  // Text string`
Keys out: ***I am Groot.*** When button 10 is pressed.

===

Key Function	Code / Symbol	Definition
KEY_CTRL	128 \KC	// CTRL set
KEY_SHIFT	129 \KS:	// SHIFT set
KEY_ALT	130 \KA	// ALT set
KEY_GUI	131 \KG	// Graphical User Interface set
KEY_CTRL_X	224 \Kc	// CTRL release
KEY_SHIFT_X	225 \Ks	// SHIFT release
KEY_ALT_X	226 \Ka	// ALT release
KEY_GUI_X	227 \Kg	// Graphical User Interface release
KEY_ALL_UP	228 \Ku	// ALL keys release
//		
KEY_ESC	177 \ESC	
KEY_F1	194	
KEY_F2	195	
KEY_F3	196	
KEY_F4	197	
KEY_F5	198	
KEY_F6	199	
KEY_F7	200	
KEY_F8	201	
KEY_F9	202	
KEY_F10	203	

KEY_F11	204		
KEY_F12	205		
KEY_INSERT	209	\INS	
KEY_HOME	210	\KH	// Home Key
KEY_DELETE	212	\DLE	
KEY_PAGE_UP	211	\PGU	
KEY_PAGE_DOWN	214	\PGD	
KEY_RIGHT_ARROW	215	\RTA	
KEY_LEFT_ARROW	216	\LTA	
KEY_DOWN_ARROW	217	\DNA	
KEY_UP_ARROW	218	\UPA	
// Control codes that affect keyboard output			
KEY_PAUSE	230	\Pnnn	// pause 0.25sec/count nnn = 0-255, 0-63.75 sec (uses two bytes for command)
//			
KEY_MODE2	232	\M2	// Cycle through two banks of Macros
KEY_MODE3	233	\M3	// Cycle through three banks of Macros
//			
KEY_NEXT	252	\KNnn	// Append macro nn where nn = macro number (uses two bytes)
KEY_REPEAT	253	\Z	// Auto repeat when button held after delay
DOC_END	254	\K#	// End of programming flag
KEY_END	255	\K@	// End of Macro line flag
null	---	\0	// (must use \0, Not the same as "\0" which is text)
Control codes:	binary 0 through 31		// (use \nn where nn is control character value)

Notes:

1) The last character of the macro remains true as long as the button is held pressed.
2) The keyboard is completely cleared after the button has been released and the macro has finished (All keys are released).
3) Auto repeat will replay the macro, after the delay time, it will clear the macro, and repeat it at the programmed rate.
4) A macro buffer is 16 characters long. It can be extended by using KEY_NEXT to access additional macro buffers. KEY_NEXT consumes two bytes.
5) Macros 20, 40, & 60 are unreachable through a button and can be used to extend any other button macro.
6) Macros 5, 10, 25, 30, 45, & 50 are on the 5-way joy-switch as the push button. These are hard to use without also pressing one of the direction switches on the joy-switch, and are not programmed. These too can be used for macro extension.
7) When playing a quoted string, each character, in the string, is pressed and then released before the next character is played. Sticky keys like CTRL, SHIFT, and ALT are not affected, as they are not allowed to be in a string, and stay pressed until released by command. They are also released when the macro line ends.
8) All control characters (binary 0 to 31) are entered by using a leading backslash '\'. A null is entered as **\0** (not the same as **"\0"** which is text).
9) Spaces are NOT permitted in the programming macro line before or after a command.
10) Commas are required to separate all commands.
11) The mode functions KEY_MODE2 & KEY_MODE3 allow the MODE button to sequence through two or three levels of macros. When the MODE button is pressed it sequences to the next level and loops back to level 1. **\M2** sequences levels 1, 2, & back to 1. **\M3** sequences levels 1, 2, 3, & back to 1. If only one level is used, no MODE button is needed.

Mouse Compatible Commands

The Universal keyboard controller supports some mouse commands. Because of limited program memory the command must use the numbers only.

MOUSE_PRESS	240,nn	// Press and Hold Selected Mouse Button
MOUSE_CLICK_LT	241	// Mouse Left Button Click and Release
MOUSE_CLICK_RT	242	// Mouse Right Button Click and Release
MOUSE_CLICK_MDL	243	// Mouse Wheel Button Click and Release
MOUSE_MOV_RT	244,nn	// Move Mouse Right 1 to 127 Mickies
MOUSE_MOV_LT	245,nn	// Move Mouse Left 1 to 127 Mickies
MOUSE_MOV_UP	246,nn	// Move Mouse Up 1 to 127 Mickies
MOUSE_MOV_DN	247,nn	// Move Mouse Down 1 to 127 Mickies
MOUSE_HOM	248	// Move Mouse to Bottom Left Corner of Screen
MOUSE_REL	249,nn	// Release Selected Mouse Button
MOUSE_SCROLL_IN	250,nn	// Scroll wheel IN 1 to 127 Mickies
MOUSE_SCROLL_OUT	251,nn	// Scroll wheel OUT 1 to 127 Mickies

Parameters for nn:

Button Select: which mouse button to press.

MOUSE_LEFT	= 1
MOUSE_RIGHT	= 2
MOUSE_MIDDLE	= 3

Move Range:

Move range	= 1 to 127 Mickies

Note: All mouse movement is relative to the current mouse location. Only the screen borders block and limit the movement of the mouse.

Universal keyboard Emulator Specifications:

Dimensions: 100mm x 55mm x 25mm (4" x 2.2" x 1").

Cable: Standard USB2 Male, length 1.5M (5ft).

Weight: 125 grams (4oz).

Power: USB powered with 5 Volts at 40ma consuming about 200mw.

All specifications are approximate.

Note:

1) As with any electronic equipment attached to a CNC system, there is risk of injury and damage. Care must be taken to become familiar with it and train with it until you can use it safely. Vancura Innovations cannot guarantee that it will work, as designed, on your system.

2) Windows recognizes this product as a generic keyboard attached to your system, and does not differentiate which keyboard is issuing a command. Using this product at the same moment as keying on a keyboard may confuse Windows and/or Mach3. This device will interact with any program that uses the keyboard for input.

3) The buttons on this product are de-bounced for keying reliability. This timing can be user programmed.

4) The Auto-repeat feature timing can be user programmed.

Contact Information:

Vancura Innovations, Rock Island IL 61201, bill@vancura.biz, bill@vancura-innovations.com