VG CONTROLS, Inc.
11 Butternut Dr, Vernon, NJ 07462
Tel (800) 524-6994 Fax (973) 764-6603
www.vgcontrols.com

USECON - the configuration utility for the USE150 and USE104 encoders, ver 1.1.4

USECON (**USE**150 **CON**figuration utility) is a program for setting or changing any of the encoder parameters, such as interface type, Baud rate, keypad pinout, etc. It also allows users to save or recall all parameters to or from a disk file. Before running this program the encoder should be powered up and connected to either a COM1 or COM2 port or plugged between keyboard and computer using extension cables.

NOTE. **USECON is a DOS program, it cannot run in WIN95 DOS window; you need either DOS or WIN95 command mode.** To get to Win95 command prompt mode select Start|Shutdown, in the shutdown dialogue select ”Restart the computer in MS-DOS mode”.

COMMAND LINE PARAMETERS.

```
Usage: USECON [options]
Options   :
-?, -h     : print help
-v         : print version
1          : use COM1 port; default is programming through keyboard
2          : use COM2 port; default is programming through keyboard
-r         : do not read the Encoder on start; default is to read
-f<file>   : read parameters from the <file>
-p<file>   : program the Encoder with parameters from the <file>
 -l        : keep log of communication between encoder and PC
```

No parameter means programming through keyboard port (for AT only).
For example, to specify that the port COM1 and program the encoder from file switch1.prm, use the following call:
C:\>**USECON -pswitch1.prm 1 <CR>**

KEY-CODES AND SCANCODES.

All numbers you ever see or enter are **HEX**es.

DIALOGUES.

USECON is organized as a system of menus, options and dialogues. A dialogue is a box of choices which will pop up as a result of you choosing particular options in submenus. Most of the actual configuration procedures are done from within dialogues. To validate your choices made in a particular dialogue, simply click your mouse on “OK” button or press “O” (for OK) key on your keyboard; otherwise press “ESC.” You can also press “C” (for Cancel) key or click mouse on “Cancel” button if one exists in the dialogue.

STARTUP.

Once invoked, USECON tries to establish communication with the encoder. If serial port was chosen it will try different Baud rates for specified COM port. Then USECON displays “Encoder Configuration” dialogue. If USECON is unable to communicate with encoder, it will display a message “encoder not responding”. If this is the case, please make sure that the encoder is properly connected, its power supply is “ON” and the COM/keyboard port you specified is the one connected to the encoder; then try to run the program again. It is possible to run USECON without the encoder (by specifying serial port on the command line) to create or modify parameter files for later use. In this case, USECON will signal an error every time the program tries to read/ write the encoder, but will still allow to read parameter files, modify parameters and save them as files.

MAIN MENU.

After communication is established between the encoder and the computer, you can access the Main Menu. By clicking the mouse on submenu name (or using Alt + highlighted letter of submenu name combination) you can pull down respective submenues. To select an option from a submenu click the mouse on the respective option or press highlighted letter from the option name on your keyboard.

The submenues and the options are arranged in the following groups:
File: Includes all options to work with files.
Once you program the encoder you can save its parameters as file on your disk, you can read a parameter files at any time and use them to program the encoder, and you can view text files by the viewer in this menu.
Encoder: Includes all options to configure/program the encoder.
Here you can: configure encoder parameters, program key-codes, choose function keys, compress the key-codes in the encoder if you run out of space, read encoder parameters program and erase encoder.
Tests: Here we can invoke a terminal mode to check communication with encoder, verify data in memory, compare encoder parameters to the memory and check for””phantom” keys.

They are described after "How to ..." section

To return to the Main Menu from any submenu, press "Esc." To exit USECON press Alt + X.

HOW TO. . .

How to decide which port to use

The programming through the keyboard port is easier, since all you need is a DIN5 male-to-male extension cable. It is, however, slower and a bit more prone to communication errors. **Also, you cannot use it with XT keyboards.**

Programming through the COM port is faster, but it requires some wiring to connect the RS232 Receiver, Transmitter and Common to the COM port and Common and +5V to a +5V power supply. A complete wiring kit with all the connectors and power supply, the Programming Adapter, is available for $80.00.

How to start up

Hook up the encoder. If you are programming through the COM port, check that the +5 V power is on.

For programming through the keyboard port :

USECON<CR>

For programming through the COM1 port :

USECON 1<CR>

Or through the COM2 port:

USECON 2<CR>

How to configure a brand-new encoder

- Start USECON with -r parameter (see page 1)
- Once in the program, select Encoder|Erase

Almost immediately you will see the **"Encoder Configuration"** dialogue box, described in the "Configure" chapter, page 3.

- **Start from configuring your input lines** (the J1 connector for your keypad). Consult the documentation that came with the keypad to determine how rows and columns are positioned in the keypad's connector.

Also note, that if you position the encoder facing components side and pointing J1 down, line 1 will be on your right. However, in the dialogue box line 1 is on the left. Please match the line numbers, not the positions. The correspondence between the pins of the connector J1 and the line numbers is shown in Encoder manual, Fig. 2.

Note1. Lines 1 through 4 can be only either "Vertical" or "None" type.

Note2. Rows can be "Vertical" and columns "Horizontal" or vice versa, as explained in the Encoder manual.

- **Choose the interface type** (AT, XT, or RS232).

Important! For those programming through the keyboard port: once you program XT interface into the encoder, you will loose communication with it and won't be able to program through keyboard port anymore. So for XT interface program through the COM port.

- If you need RS232 interface, choose the **Baud Rate**, otherwise it is irrelevant
- Choose the **Typematic Rate, Typematic Delay and Rollover.**

This is it. Press "Return" or click "OK."

How to assign key-codes for your keys

After you chose an interface and configure input lines, you can start programming key-codes.

Press "F7" on your keyboard to go to "Programming Mode." If you are in the programming mode, message at the bottom of the screen says "F7 Quit Programming Mode."

Now, if you press any key on your keypad, "Programming key-codes" dialogue box will pop up.

Using the rules described in the "Programming Key-Codes Interactively" (page 4) enter your Hex or Printable key-codes. A number of key-codes per one key (macro) is limited to 255. The chapter also describes how to enter codes in case if you want to have "Function" (or true shift) keys.

Press "Return" or click "OK" and the key-codes for the key you had pressed are programmed. If you've programmed a long macro code, do not press any new key on the keypad until you get message "Press a key on a keypad"

How to choose function keys

Function keys are the keys that will change other keys' key-codes. You can have up to two of them. To choose a key to be the function key:

Go to **Assign Function Keys** (page 3). Find the boldfaced words "**To declare a function key(s)**." Follow the instructions to the end.

How to handle pre-programmed encoder

Important: If you change your input lines' configuration, all the key-codes in the encoder will be erased. Save the encoder's contents to a file before you do it.

If your encoder has valuable data, after starting up the USECON do the following:

Start USECON, the encoder contents will be read in the memory. Then go to the "File|Save" menu . Assign some name to the file-to-be and save it. Use "View" to make sure your file's name appears in the list of files. If everything is OK, you can start your modifications.

How to save contents to a file and use it in the future

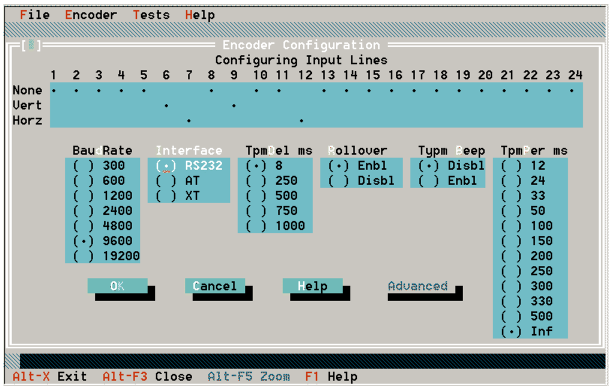
Make sure you encoder's contents are read into the computer's memory. Then go to the "File|Save" menu . Assign a name and save your file. Use it in the future to program other encoders. For that go to the “File|Read” menu, read your file and program the new encoders.

ENCODER MENU.

The Encoder Menu consists of the following options described further:

Configure
Assign Function Keys
Assign KeyCodes
Read Encoder
Program
Erase
Compress

1. CONFIGURE.



```
                         Configuring Input Lines
      1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
None  •  •  •  •  •        •     •  •     •  •  •  •  •  •  •  •  •  •  •  •
Vert                 •        •
Horz                    •              •

   BaudRate    Interface   TpmDel ms  Rollover    Typm Beep   TpmPer ms
   ( ) 300     (•) RS232   (•) 8      (•) Enbl    (•) Disbl   ( ) 12
   ( ) 600     ( ) AT      ( ) 250    ( ) Disbl   ( ) Enbl    ( ) 24
   ( ) 1200    ( ) XT      ( ) 500                            ( ) 33
   ( ) 2400                ( ) 750                            ( ) 50
   ( ) 4800                ( ) 1000                           ( ) 100
   (•) 9600                                                   ( ) 150
   ( ) 19200                                                  ( ) 200
                                                              ( ) 250
                                                              ( ) 300
                                                              ( ) 330
                                                              ( ) 500
                                                              (•) Inf
```

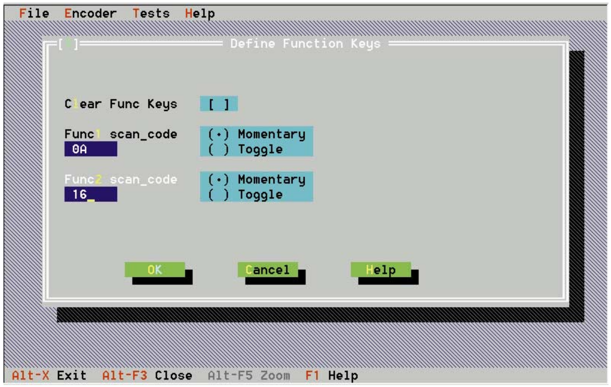
All the choices in this dialogue are “radio” buttons. “Left” and “Right” arrows or “Tab” key are used to select a parameter, and “Up” and “Down” arrows to change the value of a parameter. Or simply click the mouse on a particular field. Click on “OK” to validate your choices or on “Cancel” to discard them and escape to the Main Menu.

Configuring Input Lines changes input configuration to match your keypad’s matrix requirements. Note that the first 4 lines can be declared either Vertical or None (not used), the rest can be declared Horizontal, Vertical or None. Also note that the terms “vertical” and “horizontal” are purely conditional and have no physical meaning, it can be called “Raws” and “Columns instead”.
Interface type (RS232, IBM PC/XT or IBM PC/AT)
Baud Rate for the RS232 interface
TpmDel – typematic delay – the initial delay before sending repeating codes
TpmPer – typematic period – time between repeating codes in msec
Typm Beep – typematic beep – defines whether the optional buzzer beeps just once when you press a key or every time during typematic when a code is sent out.
Rollover - refers to a function in which the encoder transmits the “make”-code for a second key pressed while the first key hasn’t been released. If a typematic action is in progress when the second key is pressed, the encoder stops typematic action for the first key, sends a “make”-code for the second key, and starts sending keycodes for the second key if it is held down for longer than the “typematic delay.” “Rollover” action is not limited to two keys. In fact, any number of keys can be “rolled over.” If the rollover is disabled and you press a second key without releasing the first one, the encoder will disregard the second pressing. If the typematic action was in effect at that moment, the “make”-code will continue to be repeatedly sent for the first pressed key. It will consider the first key released only when no key is pressed anymore.

Note: If you change your input lines configuration and choose to press “OK” the old key-codes in the encoder will be erased!!!

Here are couple of tips:
1. If you wish to disable typematic action, select typematic period “Inf” (infinite).
2. Always declare unused lines as “NONE.” to reduce noise pick up.

2. ASSIGN FUNCTION KEYS.



```
Clear Func Keys      [ ]

Func1 scan_code      (•) Momentary
 0A                  ( ) Toggle

Func2 scan_code      (•) Momentary
 16_                 ( ) Toggle
```

These are the keys that sometimes are called “Shift” keys, (we call them “function”, please don’t confuse them with F1–F12 keys on a regular keyboard) since they affect the other keys’ codes. To make a key send different code(s) a function key(s) is pressed prior to pressing the key. Up to two keys could be declared as “Function.” A key-code sent by a key depends on the combination of function keys pressed along.

Function keys could be:
Momentary (active while pressed, somewhat like regular “Shift” on a keyboard).

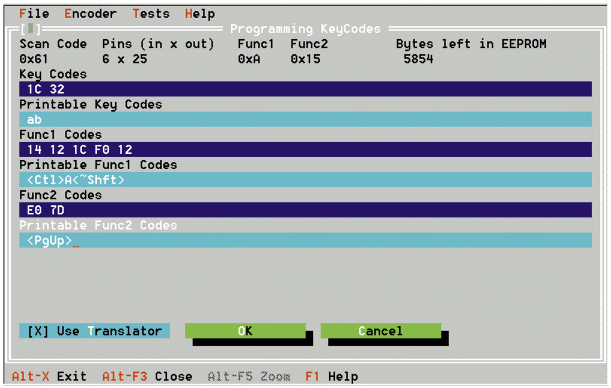
Toggle (activated when pressed first time, deactivated by being pressed second time, like "CAPS LOCK" on a keyboard).

To declare a function key(s):

1. Choose "Assign Func Keys" from "Encoder" menu
2. Using mouse or "Tab" key position your cursor within "Func1 scan_code" field
3. Press a function-key-to-be on your keypad. Watch respective scan-code appear in the field.
4. Using "Tab" key, arrow keys or mouse, choose your function key #1 to be "Momentary" or "Toggle" type.
5. Repeat steps from 2 to 4 for the second function key if you need one.

To **choose no function keys** or to get rid of existing ones click in the "Clear Func Keys" check box. A Hex code FF entered in a Scan-code field will also clear a respective function key.

3.PROGRAMMING KEY-CODES INTERACTIVELY.



```
Scan Code   Pins (in x out)   Func1  Func2          Bytes left in EEPROM
0x61        6 x 25            0xA    0x15            5854
Key Codes
 1C 32
Printable Key Codes
 ab
Func1 Codes
 14 12 1C F0 12
Printable Func1 Codes
 <Ctl>A<~Shft>
Func2 Codes
 E0 7D
Printable Func2 Codes
 <PgUp>_
```

This option is not part of any submenu. First you need to enter the **Programming Mode** by selecting "AssignKey-Codes" or pressing **F7**. The "Programming key-codes" dialogue is invoked automatically every time a key is pressed on your keypad while being in the **Programming Mode**. A Scan Code of a pressed key is displayed in the left upper corner of the dialogue along with pins connected to the key. Next to there are scan codes for "function" keys. The **number of bytes left in EEPROM** is displayed in upper right corner. Move between the fields of this dialogue using "TAB" key or mouse. To quit Programming Mode press F7 again.

You can program your key-codes as:**HEXADECIMALS** (i.e. "1C" for "a" or "5A" for Return) or **PRINTABLE** (i.e. A,B,C..., <SHIFT>, etc.).

PROGRAMMING KEY-CODES AS HEXADECIMALS NUMBERS.

To program a key (whether macro or not):

1. Move to "Key Codes" field.
2. Enter Key-code(s) as a Hex number(s) (like "1C").
3. If you intend to have a function key move to "Func1 Codes" field and enter Hex Key-code(s) that your key will produce with the function key pressed.
4. If you need a second function key, do the same for the "Func2 Codes" field.

If you leave any field empty a code of Hex 29 (AT Kbd Space) will be assigned to the key from this field.
Click on "OK" to validate a choice.
Note 1: Key-codes can be separated by spaces. If you do not separate them, then the program will assume each two non-separated digits to be one key-code. Example: "1 23" will become "1 23," but "123" will become "12 3."
Note 2: As you continue inputting codes you will see simultaneous translation into printables/mnemonics in a respective "Printable..." input line.
Note 3: If a code you are trying to enter is not a valid Hex Number you will hear a beep and the code will not appear in the input line.

HEX MACRO PROGRAMMING TIPS.

1. Break codes for all codes are generated automatically.
2. Break code for E0 prefix situation is also handled automatically.

To program macro with some uppercase characters like "MAma"

In AT mode:

- Put the "SHIFT" (like Hex 12) code before the code(s) you want upcased.
- After the last to-be-upcased code put F0 and the "SHIFT" code again; this is the AT-keyboard-type break code for the shift. There are two different "SHIFT" codes, so you'll need to use the same both times.

In XT mode:

- This step is the same as for AT. (Let's make "SHIFT" code a Hex 2A).
- After the last upcased code put your "SHIFT" code with bit #7 set to 1 (here 2A becomes AA. Setting MSB to 1 is the way for XT keyboard to send a break code).

The same technique can be used for "ALT" and "CTRL" prefixes.
EXAMPLE: Programming macro "**Mama**" for AT kbd.
The "Key Codes" line will be: "**12 3A F0 12 1C 3A 1C**."
The "Printable Key Codes" line will be: "**Mama**."

PROGRAMMING KEY-CODES AS PRINTABLE CHARACTERS/MNEMONICS.

At present this feature works only in AT mode, it allows you to program most of the codes without having to look up their hexadecimal values.
To program a key (whether macro or not):

1. Move to "Printable Key Codes" field.
2. Enter printable character(s)/mnemonic(s) as explained in **"*USING PRINTABLES/MNEMONICS*"** below.
3. If you intend to have a function key, move to "Printable Func1 Codes" field and enter the printable character(s)/mnemonic(s) that you want the key to send with the function key pressed.
4. If you need a second function key, do the same for the "Printable Func2 Codes" field.

5. You can switch back to programming in hexes any time without losing your work in printables. Just move back to the respective input line that handles hexes and continue working. The same is true for programming in hexes – you can switch to the respective line that handles printables and continue. The translation mechanism will convert printable and hex codes both ways.

Note: As you continue inputting codes you will see simultaneous translation into hex codes in a respective input line that handles hexes.

USING PRINTABLES/MNEMONICS.

At present time this feature works only in AT mode.

All the mnemonics are listed at the end of this manual in the "Mnemonics Index."

Even though it is useful to read the rules listed further, most of the programming in printables is transparent. If you are not using any extremely fancy macros, you just type something like "Bring Me a Bagel and some COFFEE!!!<return>" and most of the time you don't have to worry about the key-codes anymore.

Note: Translation is "real-time", so you will see the characters translated into hex codes while you are typing. You will also see a **PoundSign** in the end of your "Printable..." input line. This simply stands for the end of the line.

While typing in any "Printable..." input line (all examples are for AT key-codes):

1. A lower case character will produce basic key-code. Example: "a" will make it "1C," or "=" will make it "55."

2. An upper case will produce macro <Left Shift +basic key-code>;
Example: "A" will produce "12 1C"; the "+" sign will make "12 55."

3. An uppercase following another uppercase will produce basic code;
Example: "AB" will produce "12 1C 32."

4. An uppercase following <RightShift> will produce basic code.

5. A lower case following an uppercase will produce macro <Break Existing Shift+Basic key-code>.
"Existing" means Right or Left Shift, whichever was used to produce particular uppercase.
Example 1: "Ab" will produce "12 1C F0 12 32."
Example 2: "<RightShift>Ab" will produce "59 1C F0 59 32."
You will also see for both these examples that if you leave the "Printable..." line and then return back to it the translator will insert <~Shift>/<~Rightshift> mnemonic respectively before letter "b."

6. If a key is not printable, such as "RETURN," use the mnemonics listed in the end of this manual in the "Mnemonics Index." All the mnemonics are typed in the angle brackets. Some keys can have several mnemonics (synonyms), like: <Shift>,< shft>, <sft>, or <LeftShift> are the same.

Note1: Even the "Print Screen" has a mnemonic.
Note2: Mnemonics are not case-sensitive.

Example: the "RETURN" key will look like <RETURN>. "RIGHT SHIFT" is <rightshift>or any of other synonyms.

7. **Attention, a special case**: Since the Angle Brackets are used to delimit mnemonics, to type them in "make it double": type "<<" or ">>."

8. If you do need (and some need it will be) to enter a Hexadecimal from "Printable..." line , you can do it by putting it inside the angle brackets and preceding it by a dollar sign. It looks like this:
<$1C 32> for "ab."

9. A **"tilde"** (~) sign.
If typed inside the angle brackets the "~" will negate the next mnemonic (in other words, will produce a break code for it). Mostly used with "Left Shift" or "Right Shift."
Example: <~RIGHTSHIFT> means "F0 59."

10. **"Alt" and "Control."**
If you are using <Alt> or <Control> mnemonics similar to the <Shift> cases described above, all the letters after them become affected (Alted or Controlled). To release <Alt> or <Control>again, like in the <Shift> case, put the respective break code, like <~Alt> or <~Control>. (For <Right ...> mnemonics you put <~Right ...> break codes.)
Example: "<Ctrl>c<~Ctrl>a" will give you "14 2E F0 14 1C."

11. If what you typed cannot be translated, you will see a question mark in the input line with translated hexadecimals. If you try to leave the dialog by pressing "OK" with the error still present, you will get an error message and after clicking on "OK" in the error-message box the non-translatable mnemonic will be highlighted.

THE MNEMONICS INDEX.

It contains all the keys' names and mnemonics. The left column has names; the right has mnemonics (all the synonyms separated by spaces) for the lower cases in the first pair of angle brackets and for upper-cases in the second pair. The regular printable keys, like "A," "B," or "[" and their codes can be found in our Aplication Note1, IBM keyboard

4. ENCODER|READ.

This option will read all configuration parameters from the encoder into computer memory.

5. ENCODER|PROGRAM.

Will program whatever parameters are in the computer memory at the moment into the encoder. If, for example, you need parameters from a file to be programmed, just read it using **File/Read** and then run **Program**.

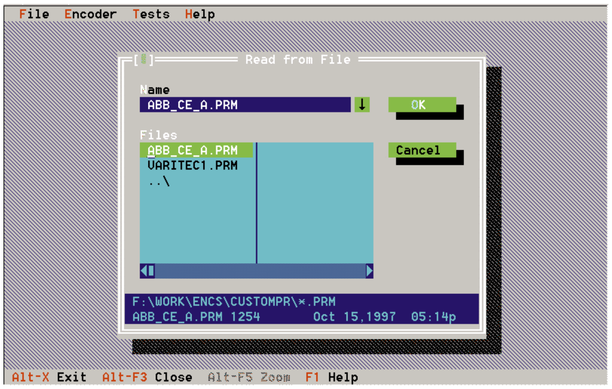
6. ENCODER|ERASE.

This option will erase all parameters from the encoder

7. ENCODER|COMPRESS.

This will defragment encoder's EEPROM and reclaims lost bytes. **This option works correctly only with the encoder hooked up**.

FILE MENU.



```
Read from File
Name
ABB_CE_A.PRM
Files
ABB_CE_A.PRM
VARITEC1.PRM
..\
F:\WORK\ENCS\CUSTOMPR\*.PRM
ABB_CE_A.PRM 1254         Oct 15,1997   05:14p
```

View is for viewing parameter files . The first 18 bytes are the parameters, the rest are the scan tables and key-codes.

Save allows to save whatever configuration is in the computer memory at the time to a file on your disk. It is wise to run Save every time you finish programming a new configuration so you can recall it in the future. Save has a safety feature which will warn you if you are trying to overwrite an existing file.

Read will read a file with parameters stored on disk into computer memory.

The names of the files are supplied through using the File Dialogue Box (picture above), which is automatically invoked by all three options.

TEST MENU

1. TERMINAL.

It is valid only if encoder is connected to computer via serial port. It allows you to check communication between the computer and the encoder and to verify the encoder programming, such as key-codes, typematic parameters, etc. The encoder echoes back all characters it receives from the computer. If you type something on the computer keyboard and these characters appear on the screen, it means that the encoder received them and sent back to the computer.

If you press a key on a keypad connected to the encoder, it will send a key-code to the computer. This key-code will be displayed differently depending upon the interface type. In RS232 mode the code itself will be displayed as a printable character, like “abc.” In IBM PC/XT or PC/AT modes the code will be shown in its hexadecimal form preceded by a “0x” (to show that it is a Hex). For example, the key-code for AT Keyboard letter “A” will be “0x1C.”

For example, if the encoder programmed for an IBM PC/AT and has macro with 3 codes:1, 2 and 3 programmed for key, then upon pressing the key in **Terminal** mode will show make codes “0x1 0x2 0x3” and then break codes ”0xF0 0x1 0xF0 0x2 0xF0 0x3.”

2. VERIFY

This option will check integrity of the programmed data in computer (macros, key codes, parameters)

3. COMPARE

This option compares data between encoder and computer memory

4. PHANTOM KEY

This option checks the possibility of “phantom codes” where pressing of three key simultaneously creates a closure for the fourth key. If the phantom key happens to be a function key this will create a very strange behavior.

5. INFORMATION

This option shows the versions of both USECON and USE encoder.

THE MNEMONICS INDEX.

KEY	MNEMONIC (many keys have few alternate mnemonics)
“Alt”	<Alt>
“Back space”	<BackSpace>
“Caps Lock”	<CapsLock>
“Control”	<Ctrl> < Ctl> < Control>
“Delete”	<Del Delete>
“Down Arrow”	<Down Dn> < ArrowDown> < ArrowDn>
“End”	<End>
“Enter”	<Return>
“Esc”	<Esc>
“F1"	<F1>
”F2"	<F2>
“F3"	<F3>
”F4"	<F4>
“F5"	<F5>
”F6"	<F6>
“F7"	<F7>
”F8"	<F8>
“F9"	<F9>
”F10"	<F10>
“F11"	<F11>
”F12"	<F12>
“Home”	<Home>
“Insert”	<Ins>
“Keypad *”	<Keypad-*> < Kpd-*>
“Keypad –”	<Keypad-Minus> < Kpd-Minus>
“Keypad +”	<Keypad-Plus> < Kpd-Plus>
“Keypad 7"	<Keypad-Home> < Kpd-Home> <Keypad-7> < Kpd-7>
”Keypad 8"	<Keypad-Up> < Kpd-Up> <Keypad-8> < Kpd-8>
“Keypad 9"	<Keypad-PgUp> < Kpd-PgUp> <Keypad-9> < Kpd-9>
”Keypad 4"	<Keypad-Left> < Kpd-Left> <Keypad-4> < Kpd-4>
“Keypad 5"	<Center> <Keypad-5> < Kpd-5>
”Keypad 6"	<Keypad-Right> < Kpd-Right> <Keypad-6 Kpd-6>
“Keypad 1"	<Keypad-End> < Kpd-End> <Keypad-1 Kpd-1>
”Keypad 2"	<Keypad-Dn> < Kpd-Dn> <Keypad-2 Kpd-2>
“Keypad 3"	<Keypad-PgDn> < Kpd-PgDn> <Keypad-3 Kpd-3>
”Keypad 0"	<Keypad-Ins Kpd-Ins> <Keypad-0 Kpd-0>
“Keypad Del”	<Keypad-Del> < Kpd-Del> <Keypad-. Kpd-.>
“Left Arrow”	<Left> < ArrowLeft>
“Left Shift”	<Shft> < LeftShift> < Shift Sft>
“Num Lock”	<NmLk> < NumLock>
“Page Up”	<PgUp> < PageUp>
”Page Down”	<PgDn> < PageDown>
”Pause”	<Pause> < Break>
“Print Screen”	<PrtScr> < PrintScreen> < SysRq>
“Right Arrow”	<Right> < ArrowRight>
“Scroll Lock”	<ScLck> < ScrollLock>
“Right Shift”	<RightShft> < RightShif> < tRightSft>
“Space”	<Space> < SpaceBar> (also can just type space without brackets)
“Tab”	<Tab>
“Up Arrow”	<Up> < ArrowUp>