

Introduction:

The TINYU-002 Keyboard Edition is a modified and reprogrammed TINYU-002 that works as a USB Keyboard with 8 key inputs and 8 LED outputs.

Features:

- 8 Active Low Key Inputs
- 8 Active High LED outputs
- USB Mini-B Connector
- DIP24 package
- Keyboard is Win XP, Linux and MAC OS/X compatible.
- Command line program to drive LED's.
(Windows XP only. Porting to Linux and/or MAC OS/X will be done on request)

Technology:

The TINYU-002 is based on RISC processor with inbuilt USB interface.

Pin Diagram:

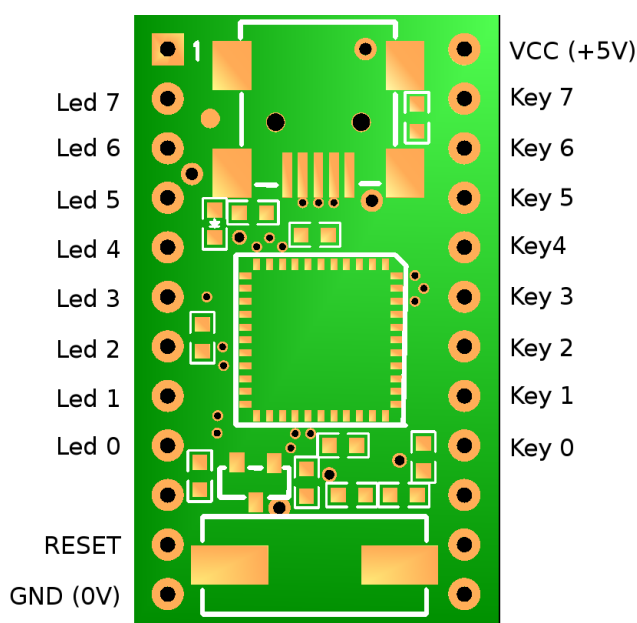


Figure 1. Module Pin Out. (USB to Top)

Connecting the Keys:

The TINYU-002 has internal pull up resistors on the inputs, so a switch from the key pin to ground is all that is required. If long lengths of wire are used the an additional pull up resistor of 10K ohms may be required from each key input pin to +5V. The wiring diagram is shown in Figure 2, including the optional RESET switch.

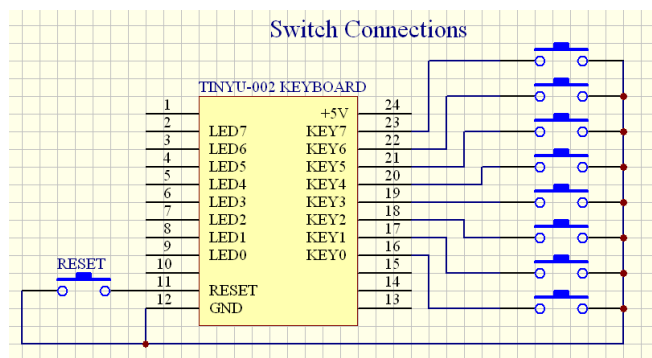


Figure 2.

Connecting the LED's:

The TINU-002 has a limit of 100mA total current draw so each LED should be current limited to 10mA. It is recommended that high efficiency or high brightness LED's be used to give maximum brightness at low power.

There are two options depending on the way you are using the LED's. Figure 3 is if you require all LED's on at once with maximum brightness where Figure 4 is ideal if you are only using one or two LED's at a time.

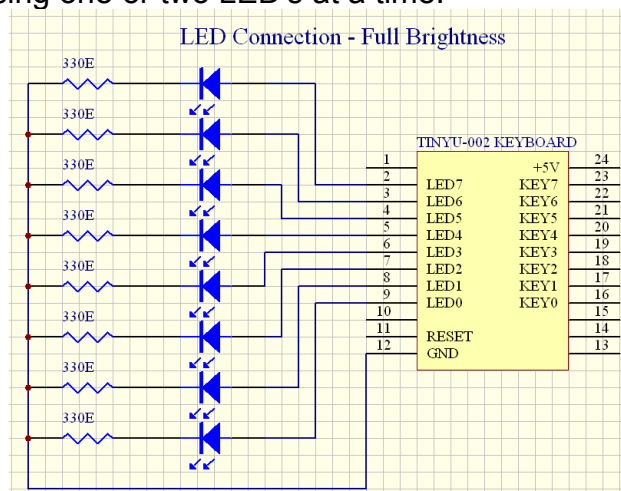


Figure 3. Full Brightness – 330 ohm Resistors

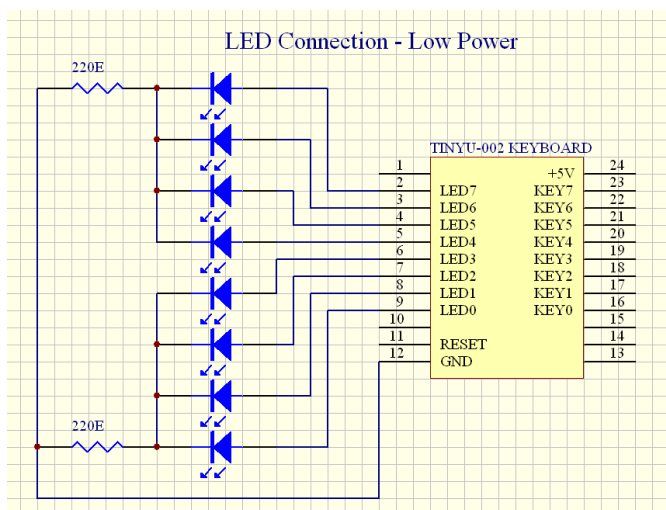


Figure 4. Low Power – 2 x 220ohm Resistors.

Using the circuit in figure 4, the brightness of each group of four LED's will reduce as more are turned on. Depending on the type of LED's you are using this effect may barely be noticeable.

Command Line Utility:

It is a simple command line program that you send a value of 0 to 255 to output the correct pattern of lights.

LED=VALUE

LED 0 = 1

LED 1 = 2

LED 2 = 4

LED 3 = 8

LED 4 = 16

LED 5 = 32

LED 6 = 64

LED 7 = 128

Simply add the numbers together to illuminate the correct LED's.

Command line:

kblight -d [Data]

For Example:

To turn All LED's ON

$1+2+4+8+16+32+64+128=255$

kblight -d 255

To turn LED 3 ON rest off

kblight -d 8

To turn LED's 0,4,7 ON

$1+16+128=145$

kblight -d 145

To Turn all LED's OFF

kblight -d 0

Altering the data transfer rate is done by the -k option. This is the clock rate in ms. The default setting is 100ms. To adjust the time the -k option must be used each time data is sent.

kblight -k [ms] -d [data]

IMPORTANT NOTICE:

kblight uses the Scroll Lock / Caps lock / Num Lock LEDs to transfer the data. If you have a standard keyboard plugged in, its LED's will flash and change state with the final state unknown.

This method was used to save the need for writing windows drivers.

kblight may be ported to Linux and MAC OS/X in the future.

More Information:

For up to date information, data sheets and downloads please visit:

<http://www.kiwacan.co.nz/>