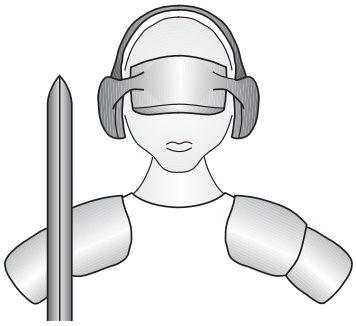
AN10: Driving a LED matrix

Applicable for IOW56 and KW100



Code Mercenaries

1. Overview

The ability to drive a LED matrix was orginally introduced with the now discontinued IO-Warrior24, IO-Warrior40, and KeyWarrior8LED. Currently the function is available in KeyWarrior100 and IO-Warrior56.

2. Theory of operation

Driving LEDs in matrixes saves a lot of effort in wiring and significantly reduces the number of driver circuits needed compared to a discrete direct control. Driveing 256 LEDs in a 8 x 32 matrix requires only 40 connections to the LEDs and 40 drivers, compared to 257 connections and 256 drivers for discrete driving. Still a LED matrix allows all possible combinations of lit and dark LEDs just like a discrete arrangement would.
A LED matrix is driven in multiplex mode. That means not all LEDs are on at the same time but just one row. This is done by the row drivers feeding current to just one row at a time. At the same time for all LEDs that are on in this row the corresponding column drivers are active so a current flows through that LED. After a short time the next row is driven and the corresponding column drivers are enabled.
This is continuously repeated at a speed high enough to make the human eye see only a continuous light from the active LEDs. The IO-Warrior56 and KeyWarrior100 drive the rows at 1000Hz, resulting in each individual row to be driven active every 8msec = 125Hz which is sufficient to not produce the impression of continous light.

2.1 LED matrix

The choice of having a 8x64 matrix may seem odd at first glance, something like 16x16 may sound more natural. Though there are good reasons for this particular layout: Driver requirements, multiplexing ratio, and scalability.
A column driver is relatively simple to build as it will always only have to care for the current that flows through a single LED. The row drivers on the other hand have to source enough current for all LEDs in a row as in worst case all of them are on at the same time. So it makes sense to keep the number of the powerful drivers low.
The multiplexing ratio is directly affecting the maximum brightness that can be obtained from a LED. Since a LED has a maximum current that can be passed through it without damaging it, increasing the current is a limited option to compensate high multiplexing ratios. Keeping to a moderate 1:8 ratio allows high brightness.
Also not all applications will require the maximum number of LEDs. By not implementing all column drivers the actual matrix size can be easily reduced.

2.2 LED current

Like with discrete driven LEDs resistors are used to set the proper current through the LED in a matrix. Though the matrix does not need a resistor for every LED but just one for all eight LEDs in a column. Since all LEDs in a column share the same resistor only LEDs of the same type or of identical technical data should be used in a column.
Calculating the resistor for a LED in a matrix is a bit more complicated than in a discrete driven arrangement. In any case the data sheet of the LED is helpful, at least two parameters need to be known: The forward voltage V_f and the maximum forward current I_f of the LED.
Forward voltage is the minimum voltage that will

be required to drive any current through the LED. Like any diode a LED acts like a kind of barrier that "cuts off" a certain voltage. The forward current is the maximum current you can pass through the diode without causing a degradation of lifetime and/or optical properties. Unfortunately most data sheets specify just a DC forward current and not a pulsed forward current, at best they may specify a maximum single peak forward current which does not help much. This does not apply to the operating conditions in a multiplexed matrix. To keep on the safe side for the lifetime and optical properties of your LEDs the current through the LEDs should not exceed the specified DC I_f.
Once the desired I_f is established the forward voltages of the drivers and the supply voltage for the whole circuit need to be known as well. The formula is quite simple:
$R = (V_{sup}-V_f-V_{fdriver})/I_f$
If $V_{sup}-V_f-V_{fdriver}$ should result in a negative value this is an indicator that your supply voltage is too low. Any LED matrix with higher brightness or more than 8x8 size (unless low power LEDs are used) should not be driven from the USB power.
For calculating the power requirements just multiply your I_f with the number of LEDs in a row. Since in worst case all LEDs in a row can be on at the same time the power source must be able to supply this current without any problems.

2.3 Driver design
IO-Warrior and KeyWarrior control the LED drivers by a serial data stream designed to connect to serial-to-parallel shift registers. Appropriate shift registers are available in the TTL/HighSpeed CMOS logic family like the 74HC/HCT595, which require external current drivers. Shift registers with integrated drivers are available too but they vary in significant details. So a thouro study of their data sheets is recommended before using them.
Depending on the required current for the LEDs it may be necessary to do a more or less complicated design for the row drivers.
In a simple case of just driving 8x8 LEDs with moderate brightness and 30mA I_f it is possible to supply them from USB power (set IO-Warrior/ KeyWarrior to high power!).
Fig. 1 shows the basic setup for a 8 x 8 matrix. The first shift register in the chain drives the rows. Discrete transistors are used to supply the current for the rows.
Additional shift registers are added for the column drivers. ULN2803 or similar inverting low side drivers may be used as column drivers to reduce the component count.
If fewer than the maximum 8 x 64 LEDs are required, only the necessary number of colmn drivers need to be connected. Fig. 2 shows the basic circuit for adding 8 x 8 additional LEDs.

3. Signals from IO-Warrior/KeyWarrior

IO-Warrior and KeyWarrior provides four signals to control the external LED registers/drivers. /OE is pulled low to enable the drivers. All drivers should be inactive while /OE is high, otherwise random states of the registers may cause LEDs to get a DC current that may be too high for permanent use. After enabling the LED function / OE stays high initially. It goes low after the data has been latched into the registers for the first time.
Data to the registers is provided via the Data pin. Each bit is clocked into the external shift registers with the rising edge of Clk. Clk idles in low state between shifting operations.
After shifting out all bits and pulling Clk low again IO-Warrior pulls Strb high for about 1µsec to signal to the registers to store the data.

3.1 Data format

IO-Warrior/KeyWarrior shifts a block of data every millisecond. Each block starts with 64 bits for the column drivers, the first bit shifted out corresponds to the highest numbered column driver. This allows columns that are not required to be left out of the circuit
Each "1" in that data corresponds to an active driver, i.e. if a column driver gets a 1 it should drive its output to ground.
Following the 32 column bits are the 8 row bits, also highest bit first. The highest bit corresponds to row 7.
The programming examples assume row 0 to be the topmost.
Column drivers must also go active if they get a 1, i.e. the driver must then source current into the matrix.
After all 72 bits have been shifted out IO-Warrior/ KeyWarrior issues a Strb pulse and then reasserts / OE.
Do not make assumptions beyond this information, future versions of IO-Warrior/KeyWarrior may extend this data format to allow driving larger matrixes.

3.2 Driving 7-Segment Displays

Since the physical arrangement of the LEDs is open it is possible to use the LED matrix function to drive 7-segment displays or other LED displays. Common anode displays are easiest to handle with IO-Warrior/KeyWarrior as this allows to connect all seven segments and the decimal point to consecutive column lines, which results in all segments of a single display being controlled by one byte of the data format.

Fig. 1: Basic driver for 8x8 matrix

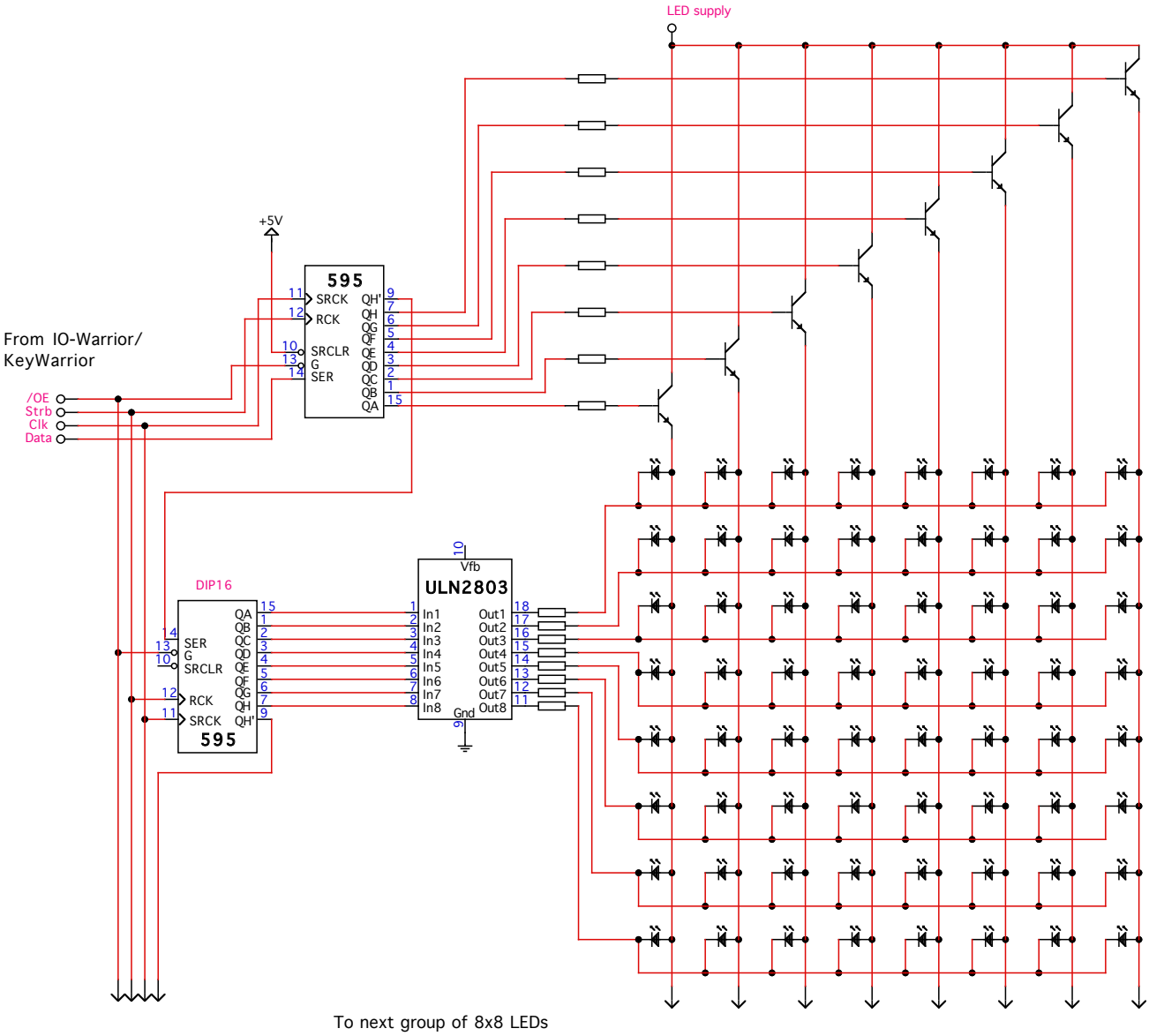
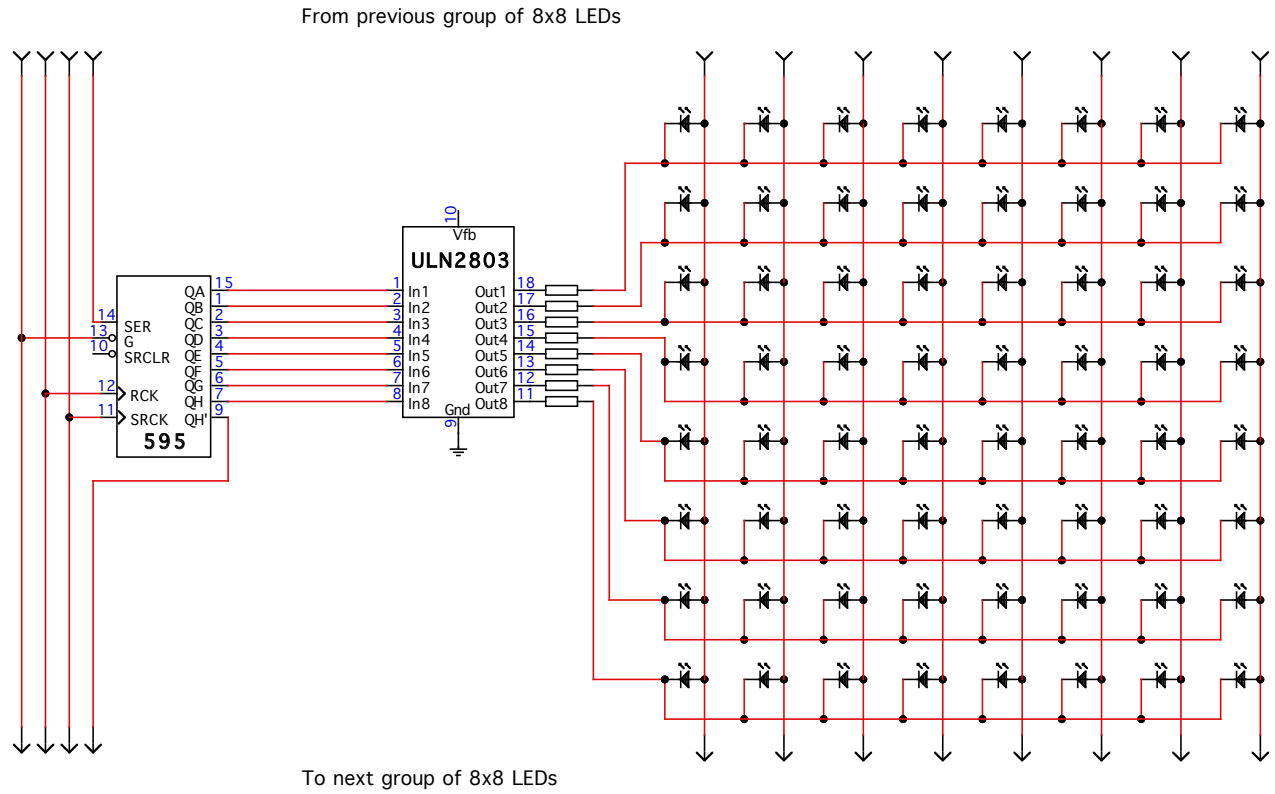


Fig.2: Extending matrix by 8x8 more LEDs



Legal Stuff

Code Mercenaries
Hard- und Software GmbH
Karl-Marx-Str. 147a
12529 Schönefeld OT Grossziethen
Germany
Tel: x49-3379-20509-20
Mail: support@codemercs.com
Web: www.codemercs.com

HRB 9868 CB
Geschäftsführer: Guido Körber, Christian Lucht