

LED display with MAX7219

Gunther Zielosko

1. Introduction

When you know the 4x20 alphanumeric LC display of the Plug & Play Lab, you will be more than satisfied with its possibilities. There are nevertheless applications with the BASIC-Tiger[™] which need other variations of displaying measured values. May it that

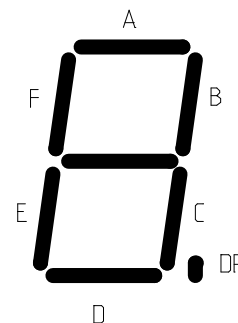
- Numbers have to be displayed in bad lighting conditions,
- Large digits are needed,
- In own developments the Plug & Play Lab isn't available,
- Simply money should be saved and the purchase of an additional LCD is not worth it,
- Other forms of display are wanted, like e.g. light bars or "british flag",

or other reasons for the usage of LED's in various forms.

Exponent of LED displays are the so-called 7-segment displays, which can represent all digits with 7 "light bars" (Segments). If you count in the almost always present comma (DP), you have 8 segments per digit. Pic 1 shows a LED display in action, Pic 2 the commonly used designations for the single segments.



Pic 1: typical 7-segment LED display



Pic 2: Segment designation

Until now the control of LED displays with micro controllers, to which the BASIC-Tiger[™] belongs, gave various problems. Some of these are looked at in the following. Let's start with a common 7-segment LED display (see above) with several digits.

At first, it is not that easy to convert digits resp. numbers in Tiger-Basic into the 7-segment form used by these LED displays, as Tiger-Basic doesn't support this form of presentation directly.

It doesn't look much better on hardware side, normally many connections (Ports) are needed to logically address the single segments. If you additionally need power (so the LED's light brightly), you won't get further without an amplifier for each segment. Somewhat more artful

is using the so-called multiplexing procedure, hereby the single digits are activated after each other and the 7-segment code belonging to that digit is passed to the single electrodes (Segments). While in “direct operation” for e.g. 8 digits 64 I/O lines would have to be addressed (8 digits x 7 segments + DP), it are only 8 (Segments) plus 8 (Digits), so 16 altogether, in multiplex operation. But the controlling effort is higher, because the control unit here is strongly occupied distributing the single signals. Furthermore this has to be done very fast to prevent flickering. Additionally also the Hardware is stretched more, as you have only 1/8 of the total time for each digit, what at same brightness mathematically demands 8 times the current at the segments compared to the “direct operation”.

A difficult task, wasn't it be for the MAX7219 from MAXIM, which manages all these tasks single-handed. It only needs three port lines, through which it gets all information serially from the BASIC-Tiger[™]. Control of LED displays of all kind, including 7-segment displays, but also exotic variations as bargraph or “british flag” displays is possible without any problem. How the MAX7219 works with the BASIC-Tiger[™] is dealt with in this application note.

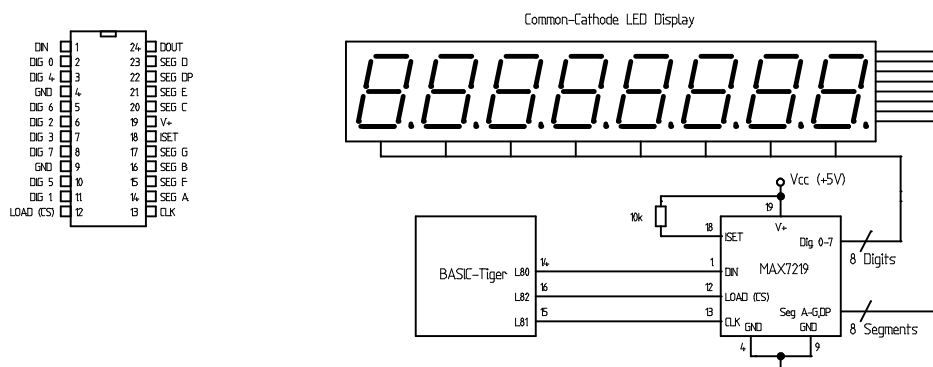
2. The MAX7219

2.1. Connection to the BASIC-Tiger[™]

The 24-pin DIL-IC contains all drivers for the LED's, does the complete timing for the multiplexed output and saves the serially received information until new info comes in. Modification of single digits is possible. Furthermore the brightness of the display can be altered with two methods, by resistor setting on ISET (e.g. with photo resistor) or by programming. Also different display modes can be set comfortable via registers, this is especially important if no 7-segment display should be controlled (e.g. bargraph displays etc.). A datasheet of the MAX7219 can loaded from:

<http://209.1.238.250/arpdf/1339.pdf>

The following drawing shows the pinout of the MAX7219 and the simple connection to the BASIC-Tiger[™]:



Pic. 3: Connection BASIC-Tiger[™] with MAX7219 (Digit 0 of displays is on the right!).

2.2. Programming the MAX7219

As we have seen, the hardware for the LED display is fairly simple. Because of the many possibilities the programming effort is somewhat higher. At first we take a look at the principle of programming the MAX7219 and later try out specific examples with the BASIC-Tiger™.

Each “record” for the MAX7219 contains of 16 data bits, which are arranged as follows:

- D 0 - D 7 Characters or control information to be displayed
- D 8 - D 11 Target address of data, either memory address for the digit or the address of the respective control register
- D 12 - D 15 not used (e.g. always set to 0)

Data transmission always starts with bit 15 (MSB), bear that in mind when providing the data in BASIC-Tiger™.

Next we deal with the target addresses (D 8 - D 11), which either determine the digit or certain control functions. Table 1 shows the complete register allocation:

Hexadecimal	Address				Function	Name
	D11	D10	D9	D8		
0h	0	0	0	0	No function	No-Op
1h	0	0	0	1	Direct addressing digit 0	Digit 0
2h	0	0	1	0	Direct addressing digit 1	Digit 1
3h	0	0	1	1	Direct addressing digit 2	Digit 2
4h	0	1	0	0	Direct addressing digit 3	Digit 3
5h	0	1	0	1	Direct addressing digit 4	Digit 4
6h	0	1	1	0	Direct addressing digit 5	Digit 5
7h	0	1	1	1	Direct addressing digit 6	Digit 6
8h	1	0	0	0	Direct addressing digit 7	Digit 7
9h	1		0	1	Decoding mode	Decode mode
Ah	1	0	1	0	Brightness	Intensity
Bh	1	0	1	1	Digit selection	Scan limit
Ch	1	1	0	0	Switch off display	Shutdown
Dh	1	1	0	1		
Eh	1	1	1	0		
Fh	1	1	1	1	Display test	Display test

Table 1 Data bits 8 - 11, the register addresses of MAX7219

Into these registers now data can be written, which have various effects depending on target address. Let's begin with the pure control data.

Register No-Op

This register is actually only needed when several MAX7219 are cascaded and only single ones are to be reprogrammed.

Register Decode-Mode

Here the single presentation alternatives on the display resp. the single LED arrangement can be chosen. In principle there is you each digit the possibility to program the 7-segment presentation or a not decoded display. A 0 in the data bit means that there is no decoding in the appropriate digit, a 1 means that the appropriate digit is addressed in 7-segment code. Samples are shown in table 2:

Hex	Address								Function
	D7	D6	D5	D4	D3	D2	D1	D0	
00h	0	0	0	0	0	0	0	0	No decoding* at all digits
01h	0	0	0	0	0	0	0	1	BCD-Code B** only for digit 0
0Fh	0	0	0	0	1	1	1	1	BCD-Code B** only for digits 0-3
FFh	1	1	1	1	1	1	1	1	BCD-Code B** for all digits

Table 2: Example for allocation of Decode-mode register

* no decoding means at logical 1

- D0 = Segment G lights up
- D1 = Segment F lights up
- D2 = Segment E lights up
- D3 = Segment D lights up
- D4 = Segment C lights up
- D5 = Segment B lights up
- D6 = Segment A lights up
- D7 = Segment DP lights up

** BCD-Code B is the normal 7-segment presentation of numbers 0-9, the following hexadecimal values A - F are displayed as -, E, H, L, P and blank. Bit 7 is set if the decimal point (comma) should light up.

Register Intensity

With an entry into this register the display brightness is set. For this only the data bits D0 to D3 are needed, the bits D4 to D7 can be 0 or 1. The minimum brightness (almost dark) you get with the value 0h, the maximum with Fh. The graduation is done in 1/32 of the total display time of the digit, with 0h = 1/32, 1h = 3/32, 2h = 5/32 etc. up to Fh = 31/32.

Scan Limit Register

This register also is easy to program. Here only bits D0 to D2 are used. For control word 0h only the least significant digit to addressed (one digit display), for 1h the digits 0 and 1 (two-digit operation), for 2h three-digit etc. So 7h means that all 8 positions are addressed.

If not all digits are addressed, the MAX7219 distributes the available display time between fewer digits, meaning the brightness (and with it the current per segment) increases. That can be compensated by hardware choosing a suited resistor on ISET or by software with the intensity register.

Shutdown Register

This register is for switching off the display, e.g. to save power or to let the display blink. Both possible states are controlled by bit D0, on 0 the display is switched off, on 1 it is working normal.

Display Test Register

With bit D0 = 1 all segments of all digits are driven with maximum brightness, meaning everything is lit up. On D0 = 0 the display is working normal again.

Digit Register 0 - 7

If the digit register is addressed, data to be displayed (Bits D0 - D3) is directly written into the display memory. Please bear in mind that bit D7 is responsible for the comma. If you want to set the comma at a certain position, there is – in addition to the number to be displayed – a “1” written into bit 7. As already mentioned, this can be the numbers 0-9 resp. some special characters or, directly addressed, single light bars (then all bits are accessible). As each single digit can be addressed, only modified data has to be transmitted. This is advantageous, when e.g. two measured values from different subroutines are entered, current on the left, voltage on the right. Each subroutine addresses only its half of the display. This makes programming significantly easier.

By the way, you can arrange the single digits of the display any way you like, as you can address each digit separately by software. If however the digits should appear on the display according to their valency, it makes sense to arrange digit 0 of the display at the very right. Doing so makes e.g. programming of a counter much easier.

That's it, having that information we should have a good grip on the MAX7219 with the BASIC-Tiger™.

3. Programming the BASIC-Tiger™

3.1. Basics

It's now necessary to control the three lines from BASIC-Tiger™ to MAX7219 in a way that the latter can process the data.

The input LOAD (CS) of the MAX7219 must be high in idle state (meaning no data is transmitted), the clock input CLK must be low. Before the beginning of data transmission LOAD (CS) is set to low. Data transmission is then done with the Tiger-BASIC™ instruction

SHIFT_OUT, where parallel to the data bits synchronous a clock is transmitted. When we now use that instruction as it is needed for communicating with the MAX7219, the main task is done. The complete syntax of the instruction is:

SHIFT_OUT PortAddr, DataPin, ClockPin, Variable, Number

Let's make an overview of the single components of this instruction with text from the Tiger-BASIC™ online help or the manual:

- **PortAddr** is a constant of type BYTE, WORD or LONG and specifies the logical internal port address. *We want to use port 8, so an 8 has to be entered here.*
- **Data pin** is a constant of type BYTE, WORD or LONG and determines at which pin the data bits are to be read in. *We are using pin L80 for data transfer, so a 0 (Bit 0 of port 8) is entered here.*
- **Clock pin** is a constant of type BYTE, WORD or LONG and specifies the bit number of the pin which is to be used as the clock pin. *In our case this is pin L81, so a 1 (Bit 1 of port 8) is entered.*
- **Variable** is a constant, variable or expression of data type BYTE, WORD, LONG or STRING and is output. *This is our 16 bit data word, meaning each output is done as WORD according to Tiger-BASIC™ rules. Here all control commands and the practical data are entered.*
- **Number** is a variable, constant or expression of type BYTE, WORD or LONG and specifies how many bits are to be written. With a positive number the lowest-order bit is written first, with a negative number the highest order bit. *We always want to write 16 bit, meaning here generally a 16 is entered. With the preceding sign you can comfortable determine with which bit the transmission should start. As the MAX7219 normally starts with the MSB, we have to enter -16.*

With this our basic instruction for writing data is always:

SHIFT_OUT 8, 0, 1, DATA*, -16

Data* is the respective content of the “message” to the MAX7219 and is build according to the task.

At last we have to determine what idle state the single control lines should have. We determine these:

L80 = DIN = Data pin of MAX7219 = High = 1
L81 = CLK = Clock pin of MAX7219 = Low = 0
L82 = LOAD (CS) = Activation of MAX7219 = High = 1

The rest is simple. At the end of this article now some sample programs, which show some possibilities of controlling the LED display:

LED1.TIG is a program, sequentially writes to a single digit of an 8-digit display the matching number from 1 to 8.
LED2.TIG controls the single segments of one digit sequentially.
LED3.TIG shows illustrated by a brightness control the possibilities of the MAX7219.

Have fun trying out other effects!