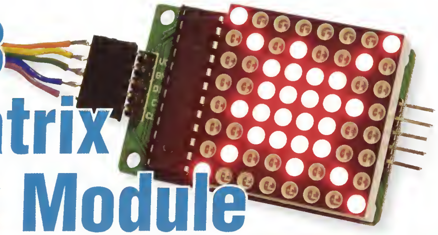
SPI 8x8 LED Matrix Display Module

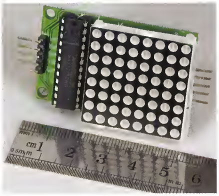


Using Cheap Asian Electronic Modules Part 7: by Jim Rowe

This low-cost module uses a Maxim MAX7219 serial LED display chip and comes complete with a plug-in 8x8 LED matrix display. But the MAX7219 is equally capable of driving an 8-digit 7-segment LED display and its SPI interface allows it to be driven by a micro using only three wires, meaning both the module and the chip are surprisingly flexible.

When I first noticed this type of 8x8 LED matrix display module being offered on eBay and AliExpress, I must confess that I didn't get overly excited. Sure, they were very cheap – but what could you actually use an 8x8 LED matrix display for? All I could think of was for displaying a few pretty patterns. Fun, perhaps, but not all that useful.

Despite this ho-hum first impression, I decided to order a couple of the modules just to see if they had any other uses. And when they arrived, I discovered that they did.



This very cheap module includes the MAX7219 IC and a plug-in 8x8 LED matrix display.

The data sheet for the MAX7219 controller chip is available from Maxim's website (https://datasheets.maximintegrated.com/en/ds/MAX7219-MAX7221.pdf) and indicates that it has primarily been designed to drive an 8-digit 7-segment LED display. In fact, the ability to drive an 8x8 LED matrix is in many ways just a bonus feature!

Inside the MAX7219

To understand the dual personality of the MAX7219, take a quick look at the block diagram, Fig.1. As you can see, there's more inside this modest-looking 24-pin DIP device than you might have expected.

Down at the bottom, you can see the 16-bit shift register where data and instructions are shifted into the chip from almost any micro, via a standard SPI (Serial Peripheral Interface) bus. Then above the eight least significant bits (D0-D7) is an eight-byte dual-port SRAM, where the display data is stored.

Four more bits, D8-D11, are decoded to determine whether the data in the lower eight bits of the shift register is to be loaded into one of the addresses in the display SRAM (either with or without further decoding), or into one of the control registers to set the chip's operating modes.

Five registers control shutdown, the mode, intensity, scan limit and display test.

Briefly, the purpose of the shutdown register is to blank the display when power is first applied or at a later time, to reduce the power consumption. It can also be used to flash the display on and off, for "alarm" situations. During normal operation, data bit D0 of this register is set to one.

The mode register is used to control whether the data in the SRAM registers for each digit is to be decoded (according to "CODE B") or used as-is. The interesting point here is that the mode register can be set for decoding all eight digits, none of them or virtually any combination in between.

So for driving an 8x8 LED matrix, for example, you wouldn't use the decoding features, while for driving an 8-digit 7-segment display you'd program it to decode all eight registers.

But you could also use it to drive a 6-digit 7-segment display by decoding just those six digits, with the remaining two digit positions either unused or used without

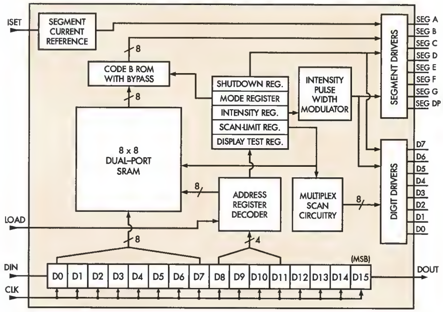


Fig.1: internal block diagram of the MAX7219 IC. The 8-byte dual-port SRAM is used to store the current LED state while the decoder block simplifies the software required to drive a 7-segment display. The segment drivers supply a fixed current determined by the current flow out of the I_{SET} pin and intensity is modulated by PWM applied by those same segment drivers.

decoding to drive other indicator LEDs. So it's quite flexible.

The intensity register provides programmable digital control over the brightness of the LEDs. As you can see from Fig.1, the chip has a segment current reference circuit (at upper left), controlled by the current fed in via the I_{SET} pin (pin 18).

The peak current sourced from the chip's segment driver outputs (upper right) is nominally 100 times the current entering the I_{SET} pin, which is normally connected to the +5V supply rail via a resistor of 9.53kΩ or more. The module shown in the pictures uses a 10kΩ resistor.

At the same time, the value stored in bits D0-D3 of the intensity control register determines the duty cycle of the chip's internal pulse-width modulator and hence the display brightness. The duty cycle is a 4-bit value, meaning that there are 16 different programmable duty cycle/brightness levels, from 1/32 (3%) to 31/32 (97%).

Then there's the scan limit control register, which is basically used to determine how many digits are scanned by the display multiplexing circuitry. This allows the chip to be programmed for any number of display digits between one and eight.

Note though that Maxim warns in the datasheet that if three or fewer digits are selected, the resistor connected to the chip's I_{SET} pin should be increased in value to reduce the power dissipation in the digit drivers.

Finally, there's the display test control register, which can be used to switch between normal operation and the test mode, where all segments are lit in order to test the display itself.

To help you put all of these functions of the MAX7219 into perspective, Fig.2 summarises the decoding of register address bits D8-D11 while Fig.3 shows the significance of data bits D0-D7 when segment decoding (ie, "CODE B") is enabled (A) or decoding is disabled (B).

Driving the 8x8 LED matrix

So that's a quick run-down on the MAX7219 device and its internal working. Fig.4 shows the full circuit for the module as it arrives and it has everything needed to drive the 8x8 LED matrix directly from a micro like an Arduino or a Micromite.

There's very little to the module apart from the MAX7219 (IC1), the 8x8 LED matrix and the two 8-pin connectors (CON2 and CON3) used to join them together.

There are two 5-pin SIL connectors; one used for the supply and serial bus inputs (CON1) and the other for the matching outputs (CON4) used for daisy-chaining further modules, plus the 10kΩ resistor connected to IC1's I_{SET} pin and a pair of bypass capacitors on the 5V supply line, one 100nF and one 10μF electrolytic.

Programming it to produce interesting patterns turns out to be fairly straightforward, as we'll see shortly. But before we do so, you'll recall that I mentioned earlier that the MAX7219 was originally intended for driving 7-segment LED displays of up to eight digits.

Driving 8-digit 7-segment displays

This configuration is shown in Fig.5, with a pair of 4x7-segment displays wired to CON2 and CON3 of the

DON'T CARE				REGISTER ADDRESS BITS				DATA BITS	
D15	D14	D13	D12	D11	D10	D9	DB	D7 D6 D5 D4 D3 D2 D1 D0	
X	X	X	X	0	0	0	0	= NO OP	(X0 hex)
X	X	X	X	0	0	0	1	= DIGIT 0	(X1 hex)
X	X	X	X	0	0	1	0	= DIGIT 1	(X2 hex)
X	X	X	X	0	0	1	1	= DIGIT 2	(X3 hex)
X	X	X	X	0	1	0	0	= DIGIT 3	(X4 hex)
X	X	X	X	0	1	0	1	= DIGIT 4	(X5 hex)
X	X	X	X	0	1	1	0	= DIGIT 5	(X6 hex)
X	X	X	X	0	1	1	1	= DIGIT 6	(X7 hex)
X	X	X	X	1	0	0	0	= DIGIT 7	(X8 hex)
X	X	X	X	1	0	0	1	= DECODE MODE	(X9 hex)
X	X	X	X	1	0	1	0	= INTENSITY	(XA hex)
X	X	X	X	1	0	1	1	= SCAN LIMIT	(XB hex)
X	X	X	X	1	1	0	0	= SHUTDOWN	(XC hex)
X	X	X	X	1	1	1	1	= DISPLAY TEST	(XF hex)

Fig.2: data is sent to the MAX7219 over a serial bus, 16 bits at a time. This table shows how bits 8-11 determine which register is written to, while bits 0-7 contain the new data for that register. With bits 8-11 set to a value between 1 and 8, one of the entries in the dual-port SRAM is updated while values of between 9 and 12 or 15 are used to write to one of the five control registers.

(A) WHEN DATA BIT 'CODE B DECODE' MODE IS SELECTED

DATA BITS

D7	D6	D5	D4	D3	D2	D1	D0		DECODED 7-SEGMENT CHARACTER
(DP)	X	X	X	0	0	0	0	(X0 hex) →	0
(DP)	X	X	X	0	0	0	1	(X1 hex) →	1
(DP)	X	X	X	0	0	1	0	(X2 hex) →	2
(DP)	X	X	X	0	0	1	1	(X3 hex) →	3
(DP)	X	X	X	0	1	0	0	(X4 hex) →	4
(DP)	X	X	X	0	1	0	1	(X5 hex) →	5
(DP)	X	X	X	0	1	1	0	(X6 hex) →	6
(DP)	X	X	X	0	1	1	1	(X7 hex) →	7
(DP)	X	X	X	1	0	0	0	(X8 hex) →	8
(DP)	X	X	X	1	0	0	1	(X9 hex) →	9
(DP)	X	X	X	1	0	1	0	(XA hex) →	–
(DP)	X	X	X	1	0	1	1	(XB hex) →	E
(DP)	X	X	X	1	1	0	0	(XC hex) →	H
(DP)	X	X	X	1	1	0	1	(XD hex) →	L
(DP)	X	X	X	1	1	1	0	(XE hex) →	P
(DP)	X	X	X	1	1	1	1	(XF hex) →	(blank)

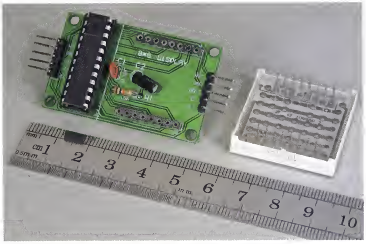
(B) WHEN DATA BIT 'NO-DECODE' MODE IS SELECTED

DATA BITS

D7	D6	D5	D4	D3	D2	D1	D0
(DP)	A	B	C	D	E	F	G

CORRESPONDING SEGMENTS WITH NO DECODING

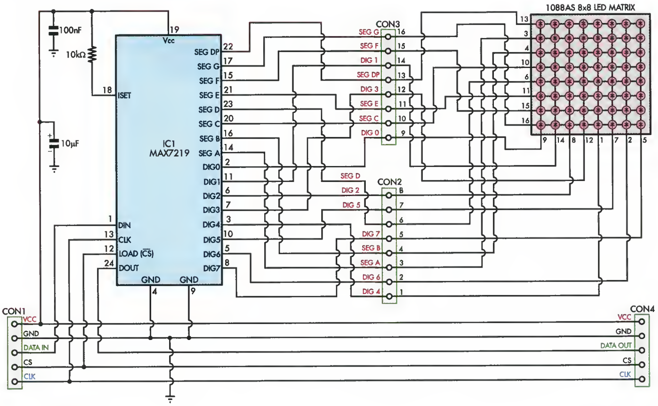
NOTE: DECODE/NO-DECODE MODE CAN BE SELECTED INDEPENDENTLY FOR EACH OF THE EIGHT DIGITS, USING THE DECODE REGISTER



Above is the layout of the module without the 7-segment display and below in Fig.4 is the matching circuit diagram.

Fig.3 (left): when "Code B" decoding is active for a segment, the lower four bits of the value for that segment forms a lookup table for one of 16 possible 7-segment display configurations, as shown at right. The top bit determines whether the decimal point is lit. Compare this to (B) at bottom, where decoding is not active and the eight bits in SRAM control the segment drivers directly.

Fig.4 (below): the circuit of a typical pre-built 8x8 LED matrix module with MAX7219 driver. A photo of this type of module is shown above. There's virtually nothing to it, just the LED matrix display module, the MAX7219 IC and some connectors to join them together and to provide connections to the microcontroller and optionally, more daisy-chained LED displays.



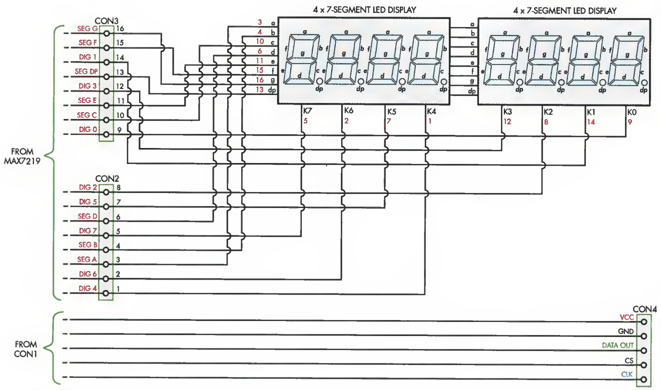


Fig.5: the circuit of a typical pre-built 8-digit 7-segment common cathode LED display using a MAX7219. Pre-built modules for this configuration are also available. We haven't shown the IC itself, as its configuration is identical to that of Fig.4 – all that's changed is that in place of the 8x8 LED matrix are two 4-digit 7-segment displays with the anodes wired in parallel.

module instead of the 8x8 LED matrix. Note that for space reasons, we haven't shown the MAX7219 chip or the rest of the module circuitry to the left of CON2 and CON3 in Fig.5, but these are all exactly the same as in Fig.4.

In fact, the only changes needed to drive a pair of 4x7-segment displays instead of an 8x8 LED matrix with the MAX7219 module are in terms of software rather than hardware.

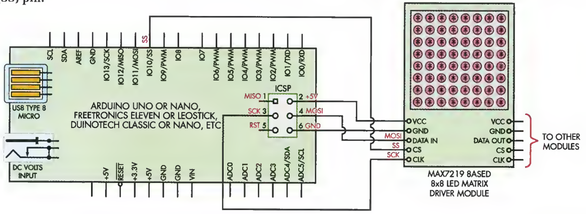
Specifically, it's just a matter of enabling decoding for all eight digits, instead of disabling it, as required for driving the 8x8 LED matrix. Which leads us on to hooking the MAX7219 module up to popular micros and programming it to display what you want.

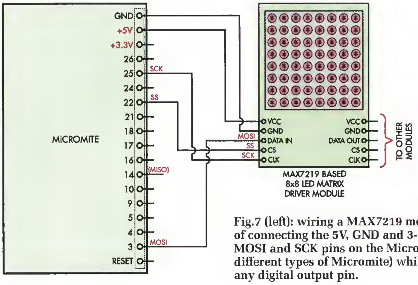
In fact, not only is it possible to drive an 8-digit display using a MAX7219, pre-built modules are available on eBay and AliExpress, etc. These incorporate a PCB with an SMD MAX7219 on the back and two 4-digit 7-segment displays plugged into header sockets on the front. Like the 8x8 matrix displays, they have 6-pin connectors at each end to wire up to your micro and also allow daisy chaining.

Driving them from an Arduino

As shown in Fig.6, it's quite easy to

Fig.6: connecting either type of MAX7219-based module to an Arduino is easy. Simply wire up the SPI pins and power supply to the ICSP header on the Arduino and the CS pin to a free GPIO – ideally IO10 which is the hardware slave select (SS) pin.





Directly below you can see the underside of the module is sparse, only having the markings for the connections to and from the module.

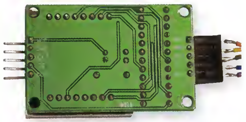


Fig.7 (left): wiring a MAX7219 module to a Micromite is simply a matter of connecting the 5V, GND and 3-wire SPI bus between the two units. The MOSI and SCK pins on the Micromite are fixed (and may vary between different types of Micromite) while slave select (SS) can go to pretty much any digital output pin.

connect these modules up to almost any Arduino or Arduino clone, by taking advantage of the fact that most of the connections needed for interfacing to an SPI peripheral are made available on the 6-pin ICSP header fitted to most Arduino variants.

The connections to the ICSP header are fairly consistent over just about all Arduino variants, including the Uno, Leonardo and Nano, the Freetronics Eleven and LeoStick, and the Duinotech Classic or Nano.

In fact the only connection that's not available via the ICSP header is the one for SS/CS/LOAD, which needs to be connected to the IO10/SS pin of an Arduino Uno, Freetronics Eleven or Duinotech Classic as shown in Fig.6.

With other variants you should be able to find the corresponding pin without too much trouble and even if you can't, the pin reference can be changed in your software sketch to match the pin you do elect to use.

Driving them from a Micromite

It's also quite easy to drive these modules from a Micromite, using the connections shown in Fig.7.

By connecting the MOSI, SCK and SS/LOAD lines to Micromite pins 3, 25 and 22 as shown, MMBasic's built in SPI protocol commands will have no trouble in communicating with the module.

So that's the basic story regarding the hardware side of the MAX7219 based module which can drive either an 8x8 LED matrix array or eight 7-segment LED displays.

Before we finish, a few words are in order regarding the software side, ie, how to write programs to get the module to display what you want.

Writing the software

The basic idea here is that when your program starts up, it needs to carry out a number of set-up tasks. These are:

- Declare the micro's pins that are going to be used by the SPI interface and set them to their idle state (normally high).
- Start up the SPI interface, with its settings configured for a clock rate of say 5MHz, the data to be sent MSB (most significant bit) first and using clock/data timing mode 0. If possible, it should also be set for the data to be exchanged in 16-bit words rather than bytes.
- Send the initialisation commands to the MAX7219 to set up its five control registers: shutdown, decode mode, intensity, scan limit and display test.

After these tasks have been done, you should be able to send out the actual display data for each of the 8-digit display addresses in the MAX7219's SRAM. And if the display is to be a dynamic one, you can send out revised data at the appropriate times.

To help you understand what's involved in writing your own programs for the module, I have written a couple of simple example programs which repeatedly display an expanding star pattern on the 8x8 LED matrix array (you can see the fully expanded star on the lead photo).

One of these programs is written for Arduino and is called "sketch2_for_Testing_MAX7219.ino". The other is written for the Micromite, and is called "MAX7219 LED array Star.bas". Both of these programs are available for download from the SILICON CHIP website (www.siliconchip.com.au).

Both programs simply blank the display for a second or so, then cause a small square pattern to appear first in the centre of the array and then expand out fairly quickly to form a star, with its tips at the four corners of the array. The expanded star remains visible for about three seconds before the array is blanked again and the sequence repeats.

It's all quite simple, but either program should give you a reasonably clear guide regarding how to use the MAX7219 display driver module in your own projects.

I've tried to provide a lot of explanatory comments in both programs, to help in this regard. SC