Introduction to robotics

7th lab

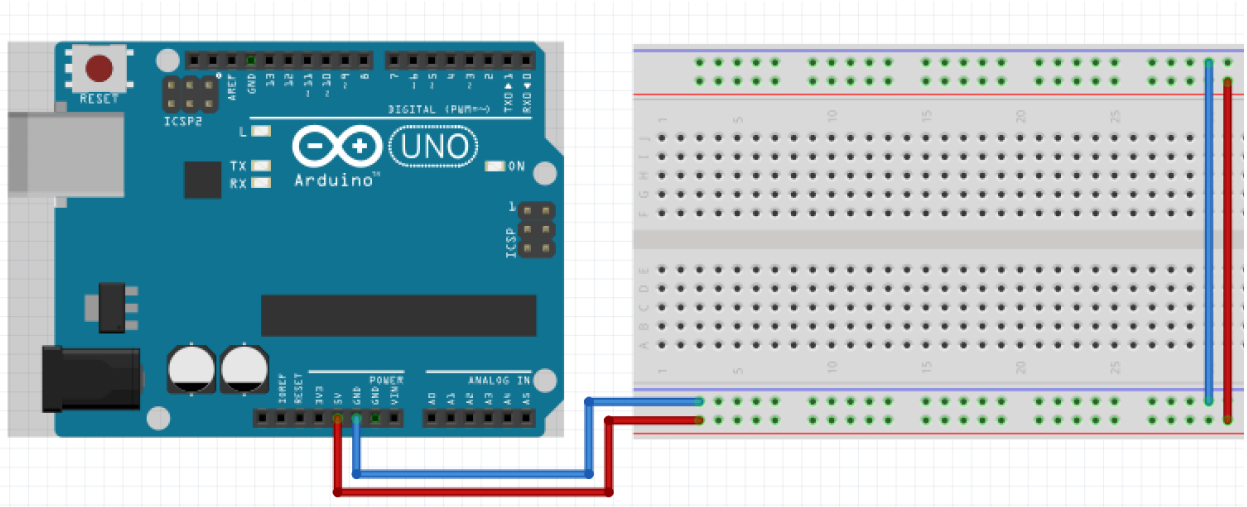
Remember, when possible, choose the wire color accordingly:

- **BLACK** (or dark colors) for **GND**
- **RED** (or colored) for **POWER (3.3V / 5V / VIN)**
- **Remember** than when you use digitalWrite or analogWrite, you actually send power over the PIN, so you can use the same color as for **POWER**
- **Bright Colored** for read signal
- We know it is not always possible to respect this due to lack of wires, but the first rule is **NOT USE BLACK FOR POWER OR RED FOR GND!**

Now, let's pick it up where we left off…

Pull out your Arduino and breadboard and connect them like in the schematic. This is to "power up" the breadboard so we can easily have access to **5V** and **GND**.

Attention! Remember how the breadboard works. Use correct wire colors.



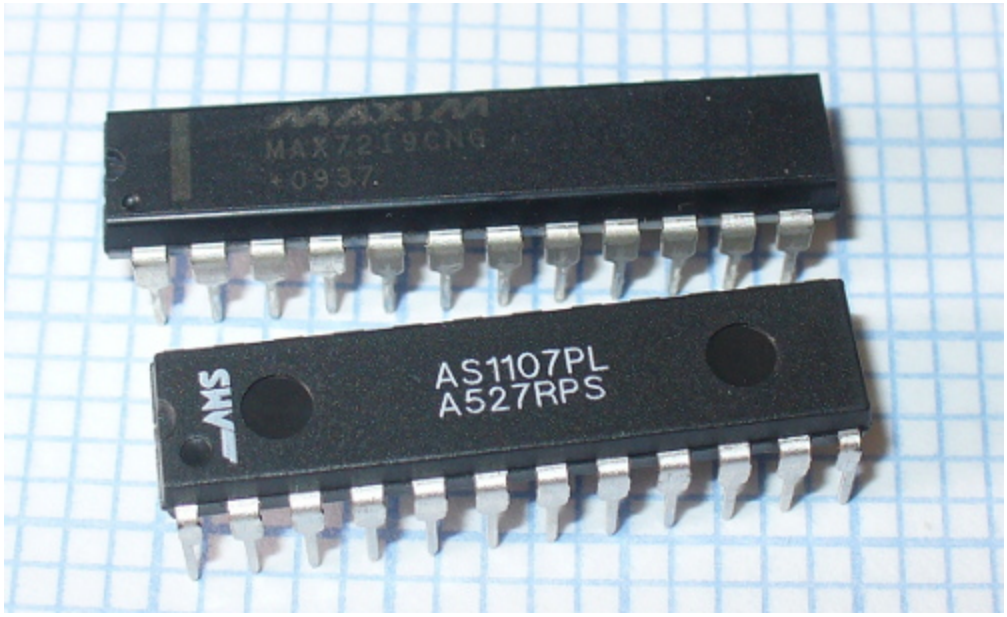
1. MAX7219 Driver

Datasheet: https://www.sparkfun.com/datasheets/Components/General/COM-09622-MAX7219-MAX7221.pdf

> The MAX7219/MAX7221 are compact, serial input/out-put common-cathode display drivers that interface microprocessors (µPs) to 7-segment numeric LED displays of up to 8 digits, bar-graph displays, or 64 indi-vidual LEDs.

(source: https://www.sparkfun.com/datasheets/Components/General/COM-09622-MAX7219-MAX7221.pdf)

Basically, it's a simple and somewhat inexpensive method of controlling 64 LEDs in either matrix or numeric display form. Furthermore they can be chained together to control two or more units for even more LEDs. Overall – they're a lot of fun and can also be quite useful.

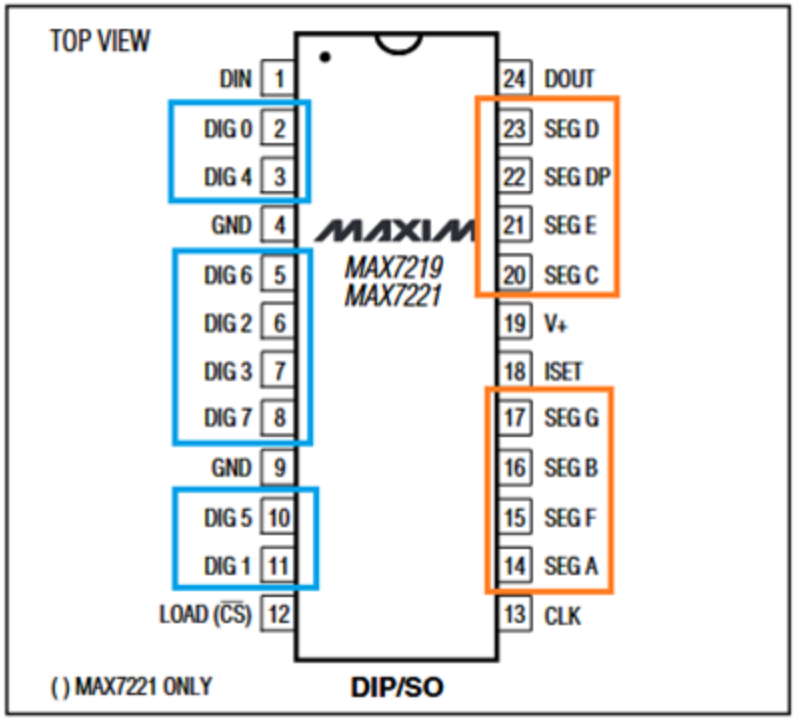


As mentioned earlier, the MAX7219 can completely control 64 individual LEDs – including maintaining equal brightness, and allowing you to adjust the brightness of the LEDs either with hardware or software (or both). It can refresh the LEDs at around 800 Hz, so no more flickering, uneven LED displays.

2. Controlling the LED matrix display with the MAX7219

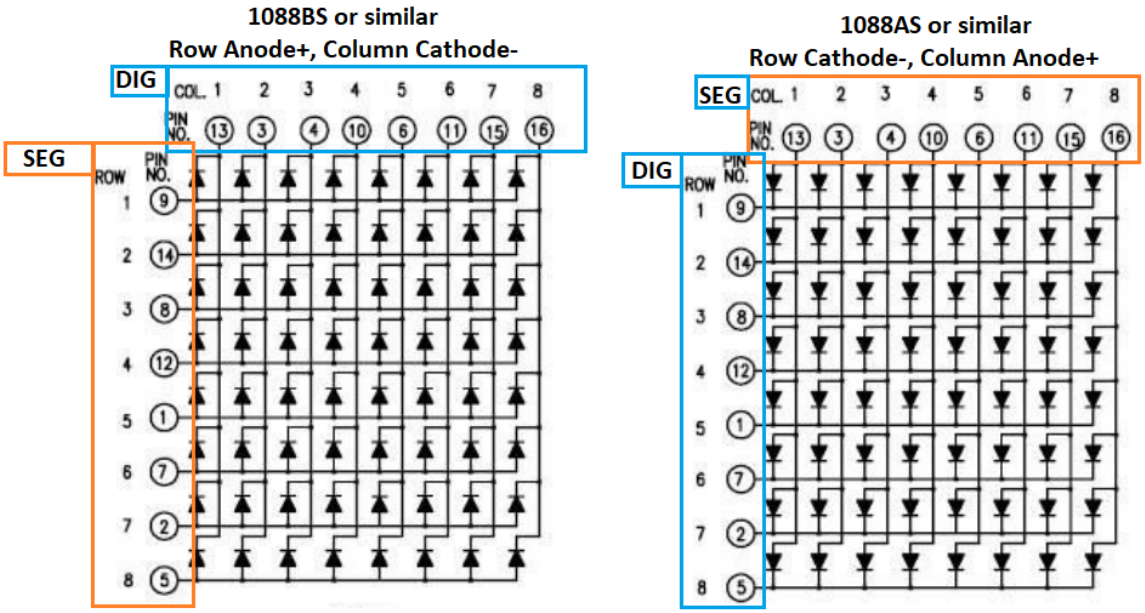
Max7219 Pinout:

Pin Configuration



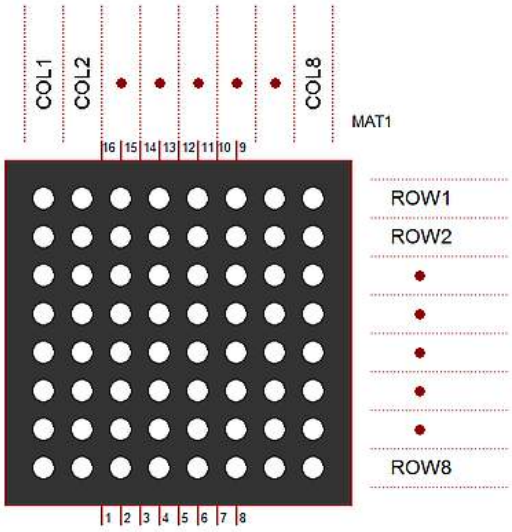
Pin Description

PIN	NAME	FUNCTION
1	DIN	Serial-Data Input. Data is loaded into the internal 16-bit shift register on CLK's rising edge.
2, 3, 5–8, 10, 11	DIG 0–DIG 7	Eight-Digit Drive Lines that sink current from the display common cathode. The MAX7219 pulls the digit outputs to V+ when turned off. The MAX7221's digit drivers are high-impedance when turned off.
4, 9	GND	Ground (both GND pins must be connected)
12	LOAD (MAX7219)	Load-Data Input. The last 16 bits of serial data are latched on LOAD's rising edge.
	$\overline{CS}$ (MAX7221)	Chip-Select Input. Serial data is loaded into the shift register while $\overline{CS}$ is low. The last 16 bits of serial data are latched on $\overline{CS}$'s rising edge.
13	CLK	Serial-Clock Input. 10MHz maximum rate. On CLK's rising edge, data is shifted into the internal shift register. On CLK's falling edge, data is clocked out of DOUT. On the MAX7221, the CLK input is active only while $\overline{CS}$ is low.
14–17, 20–23	SEG A–SEG G, DP	Seven Segment Drives and Decimal Point Drive that source current to the display. On the MAX7219, when a segment driver is turned off it is pulled to GND. The MAX7221 segment drivers are high-impedance when turned off.
18	ISET	Connect to V_{DD} through a resistor (R_{SET}) to set the peak segment current (Refer to *Selecting R_{SET} Resistor* section).
19	V+	Positive Supply Voltage. Connect to +5V.
24	DOUT	Serial-Data Output. The data into DIN is valid at DOUT 16.5 clock cycles later. This pin is used to daisy-chain several MAX7219/MAX7221's and is never high-impedance.



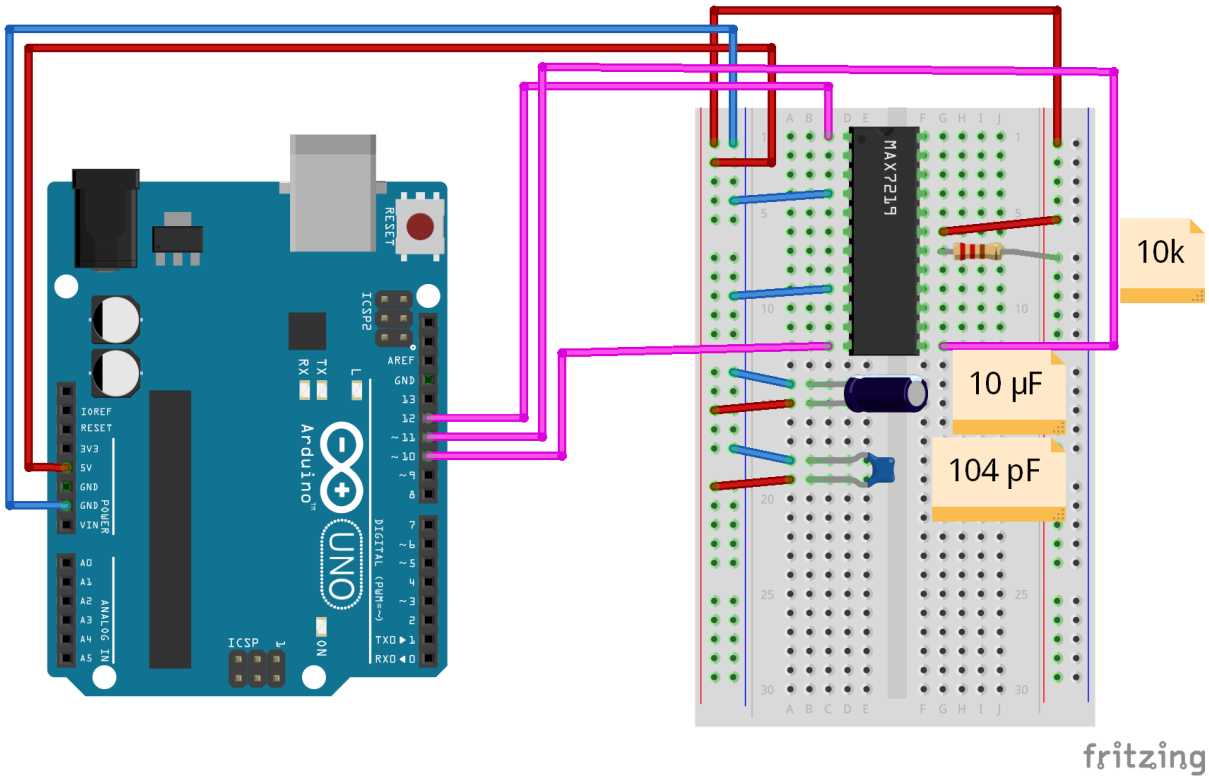
From the description it is clear that:

1. **DIG** pins are used for the cathodes of the LED matrix
2. **SEG** pins are used for the anodes of the LED matrix
3. Careful, first check which type of matrix you have



4. Connecting the driver to Arduino

Note: if the connection doesn't work, try a 100k resistor instead of 10k.



Connection list:

Max7219 Driver Pins	Arduino Pins
4 (GND)	GND
9 (GND)	GND
18 (ISET)	5V, **through a 10k or 100k resistor**
19 (V+)	5V
1 (DIN)	12
12 (LOAD/CS)	10
13 (CLK)	11

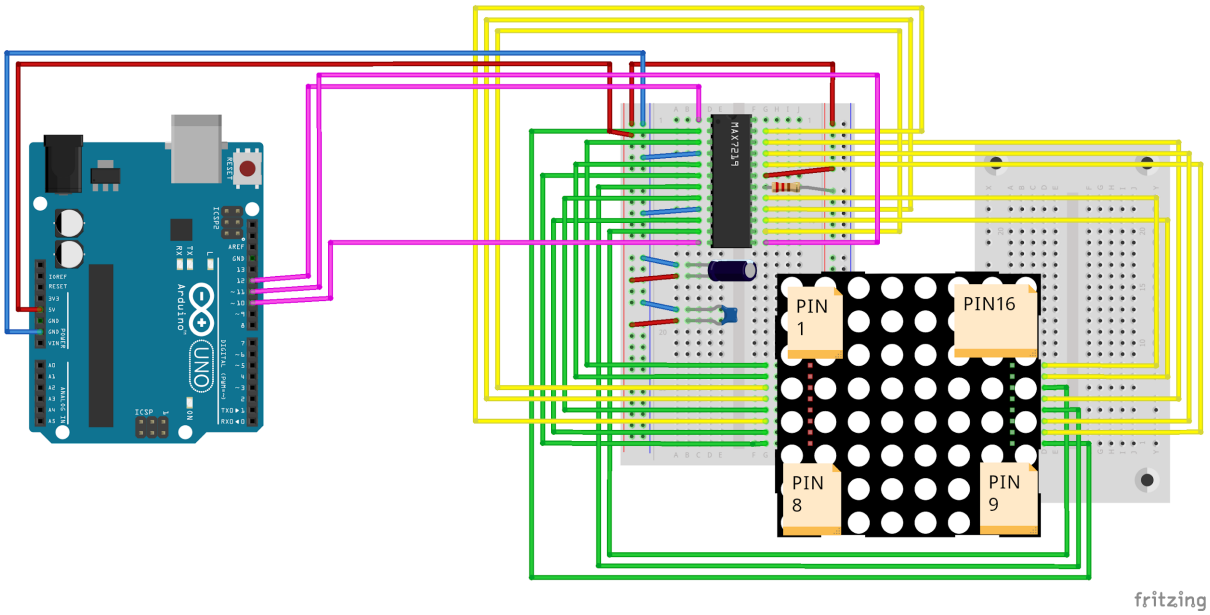
As you can see in the schematic, there are also 2 capacitors that are connected in parallel to the + and - of our circuit.

- 1 electrolytic capacitor of 10 µF
- 1 ceramic capacitor of 104 pF

!important If connected wrong, the electrolytic capacitor can blow up (remember the first lab, that's what we blew up).
We use the capacitors to filter circuit noise.

If you change the pinout order, remember to change it in the code as well.

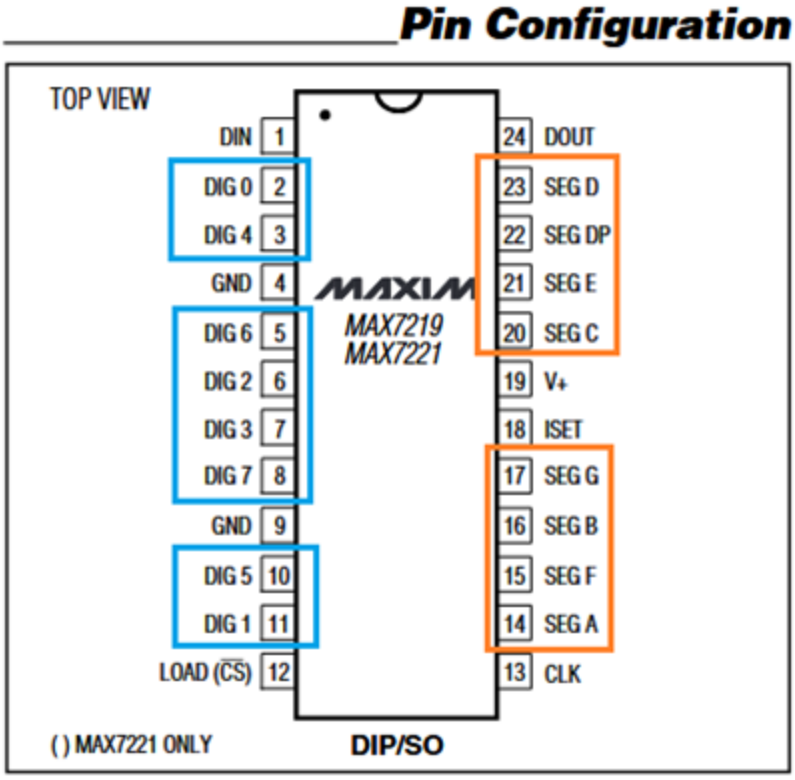
The schematic is for 1088AS, pay attention to what type of matrix you have!



1088AS Matrix		MAX7219 Driver	
Row	Matrix Pin	DIG Number	Pin Number
Row 5	1	DIG4	3
Row 7	2	DIG6	5
Col 2	3	SEG A	14
Col 3	4	SEG B	16
Row 8	5	DIG7	8
Col 5	6	SEG D	23
Row 6	7	DIG5	10
Row 3	8	DIG2	6
Row 1	9	DIG0	2
Col 4	10	SEG C	20
Col 6	11	SEG E	21

Row 4	12	DIG3	7
Col 1	13	SEG DP	22
Row 2	14	DIG1	11
Col 7	15	SEG F	15
Col 8	16	SEG G	17

1088BS Matrix		MAX7219 Driver	
Row / Column	Matrix Pin	DIG Number	Pin Number
Row 1	13	DIG0	2
Row 2	3	DIG1	11
Row 3	4	DIG2	6
Row 4	10	DIG3	7
Row 5	6	DIG4	3
Row 6	11	DIG5	10
Row 7	15	DIG6	5
Row 8	16	DIG7	8
Col 1	9	SEG DP	22
Col 2	14	SEG A	14
Col 3	8	SEG B	16
Col 4	12	SEG C	20
Col 5	1	SEG D	23
Col 6	7	SEG E	21
Col 7	2	SEG F	15
Col 8	5	SEG G	17



5. Code Examples

We will use the **LedControl library** in our code.

```
#include "LedControl.h" //  need the library
const int dinPin = 12;
const int clockPin = 11;
const int loadPin = 10;
const int rows = 8;
const int cols = 8;
LedControl lc = LedControl(dinPin, clockPin, loadPin, 1); //DIN, CLK, LOAD, No. DRIVER

// pin 12 is connected to the MAX7219 pin 1
// pin 11 is connected to the CLK pin 13
// pin 10 is connected to LOAD pin 12
// 1 as we are only using 1 MAX7219

void setup()
{
  // the zero refers to the MAX7219 number, it is zero for 1 chip
  lc.shutdown(0, false); // turn off power saving, enables display
  lc.setIntensity(0, 2); // sets brightness (0~15 possible values)
```

```
  lc.clearDisplay(0);// clear screen
}

void loop()
{
  for (int row = 0; row < rows; row++)
  {
    for (int col = 0; col < cols; col++)
    {
      lc.setLed(0, col, row, true); // turns on LED at col, row
      delay(25);
    }
  }
  for (int row = 0; row < rows; row++)
  {
    for (int col = 0; col < cols; col++)
    {
      lc.setLed(0, col, row, false); // turns off LED at col, row
      delay(25);
    }
  }
}
```