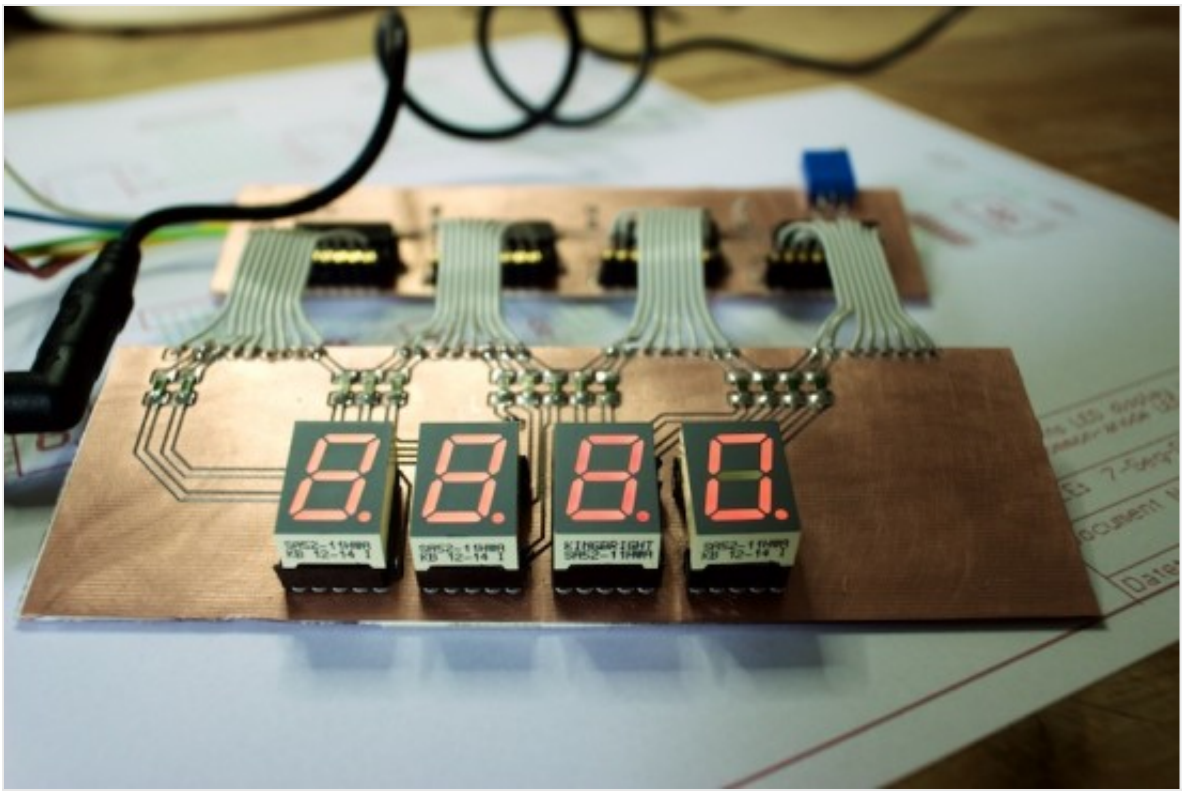
Multiple Shift-out Registers on Arduino – part 2



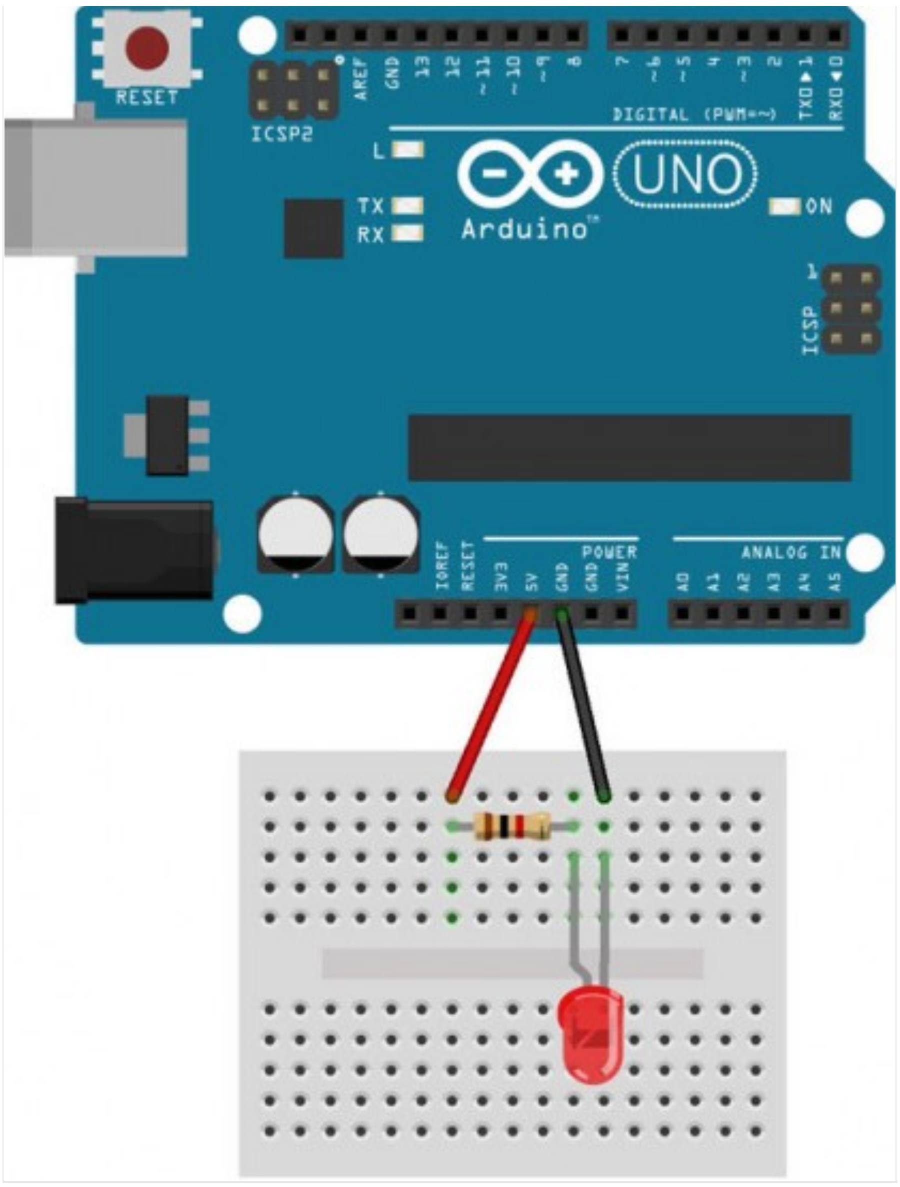
In this second article we will see a practical application of the daisy-chained shift registers.

We want to use four 7-Segments LEDs controlled by the Arduino board. As you can imagine, that is impossible without a series of shift-registers as we need 32 output pins. As we can do in many other applications of the daisy-chained shift registers, in this case we don't need a specific software or library: it is sufficient what we already know about the shift registers.

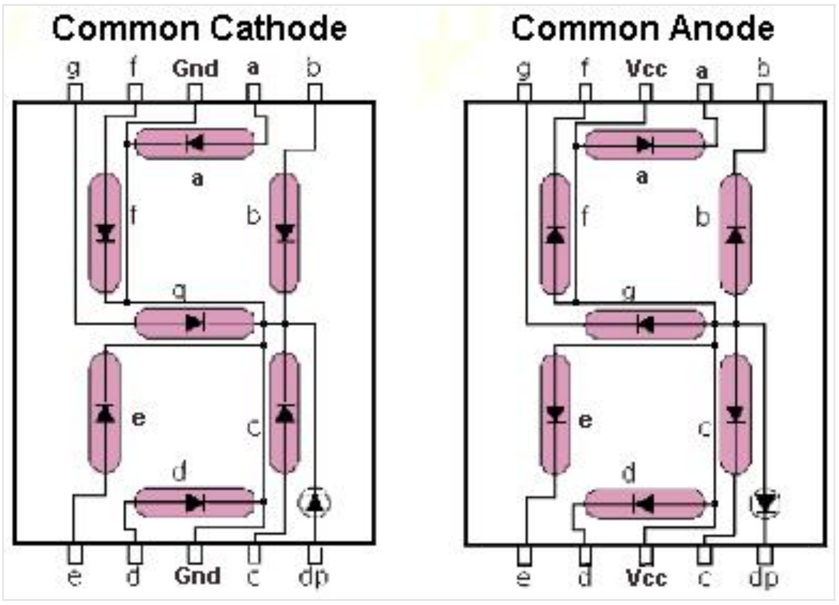
As in the **previous article** I wrote that the board is a general purpose tool, this is one of the hundreds possible applications that only need hardware. The only thing we must know is how the 7-segments LED devices work.

The 7-segments LED devices

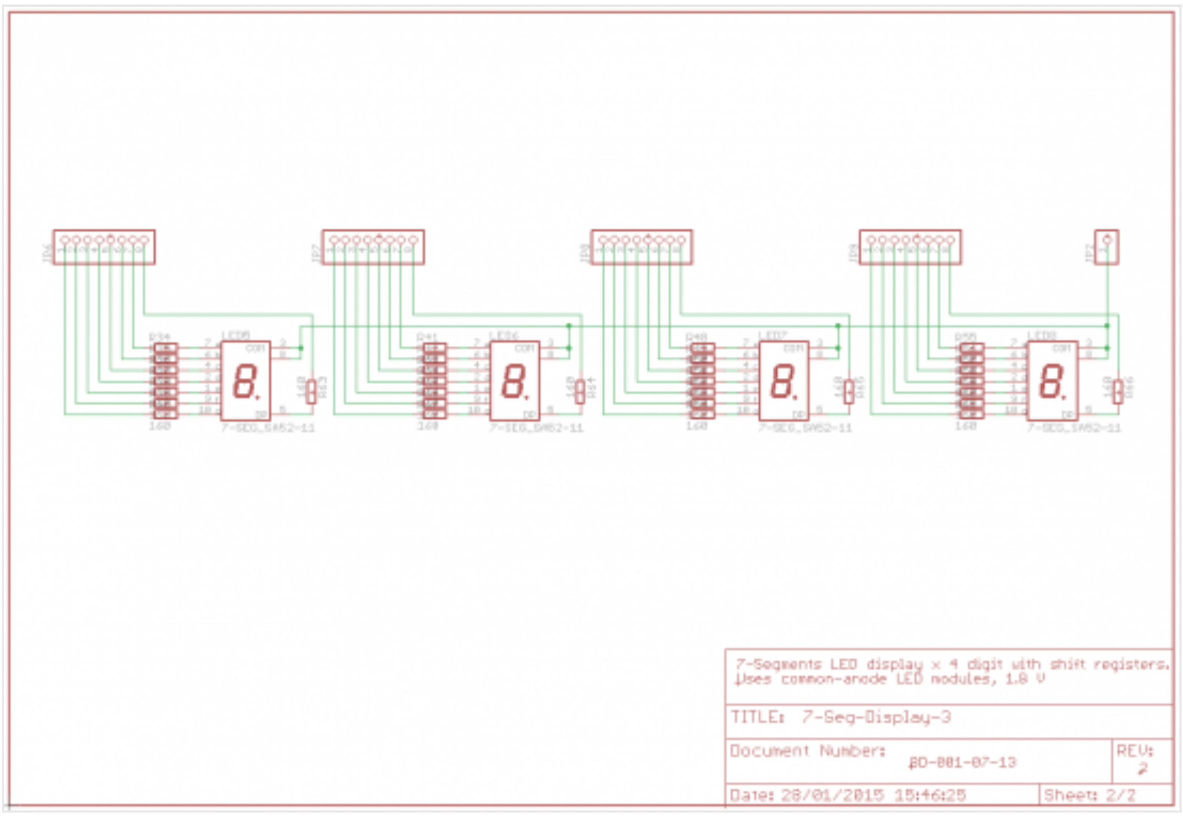
The general principle behind this kind of devices is the same that makes the common LEDs working. To get a LED lighting we should connect its ground pin to the circuit GND and the positive pin to an output pin on the microcontroller. Everytime the microcontroller pin goes high the LED is powered.



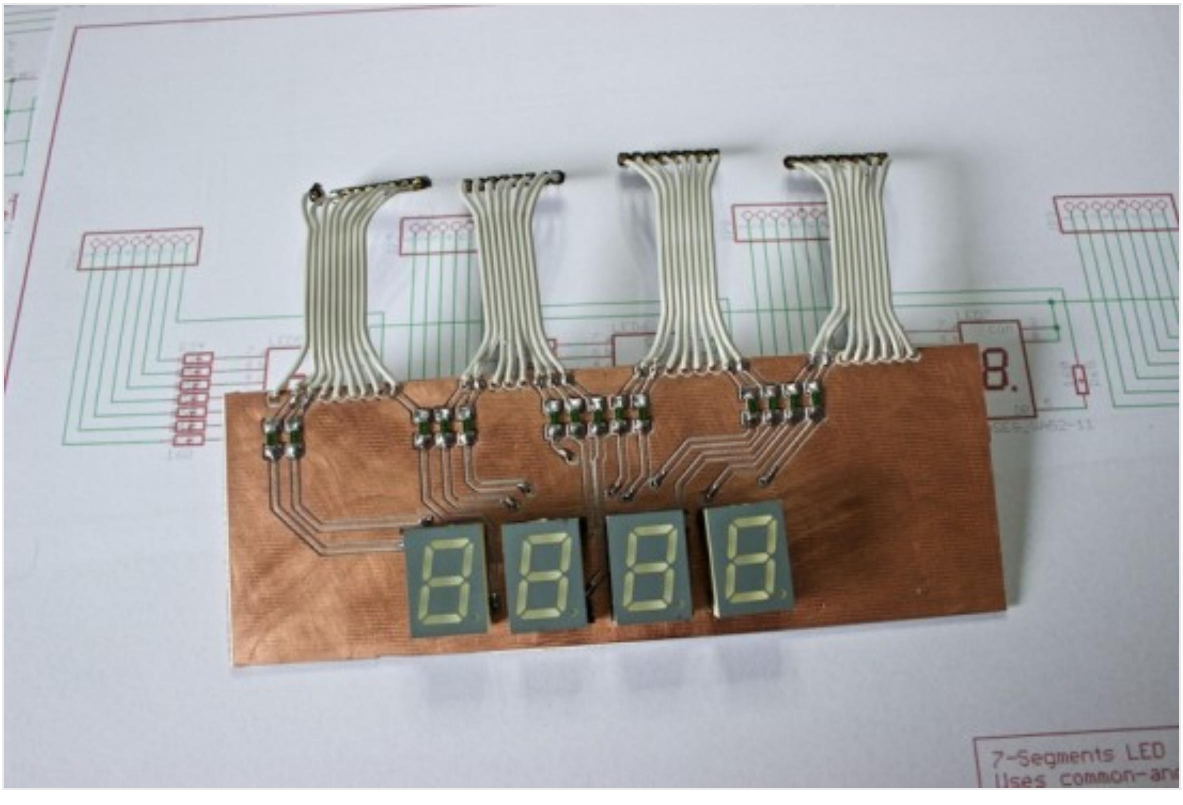
It's important to know that there are two different types of 7-segments LED displays: **common cathode** and **common anode**. As shown in the scheme below the *common cathode* display shares the GND signal while the *common anode* display shares the VCC signal.



By the point of view of the circuit schematics there are no particular differences and we have adopted common anode devices because of some advantages compared to the opposite common cathode.


7-Segments LED display x 4 digit with shift registers.
Uses common-anode LED modules, 1.8 V
TITLE: 7-Seg-Display-3
Document Number: BD-001-07-13
REV: 2
Date: 28/01/2015 15:46:25
Sheet: 2/2
7-SEG_SA52-11

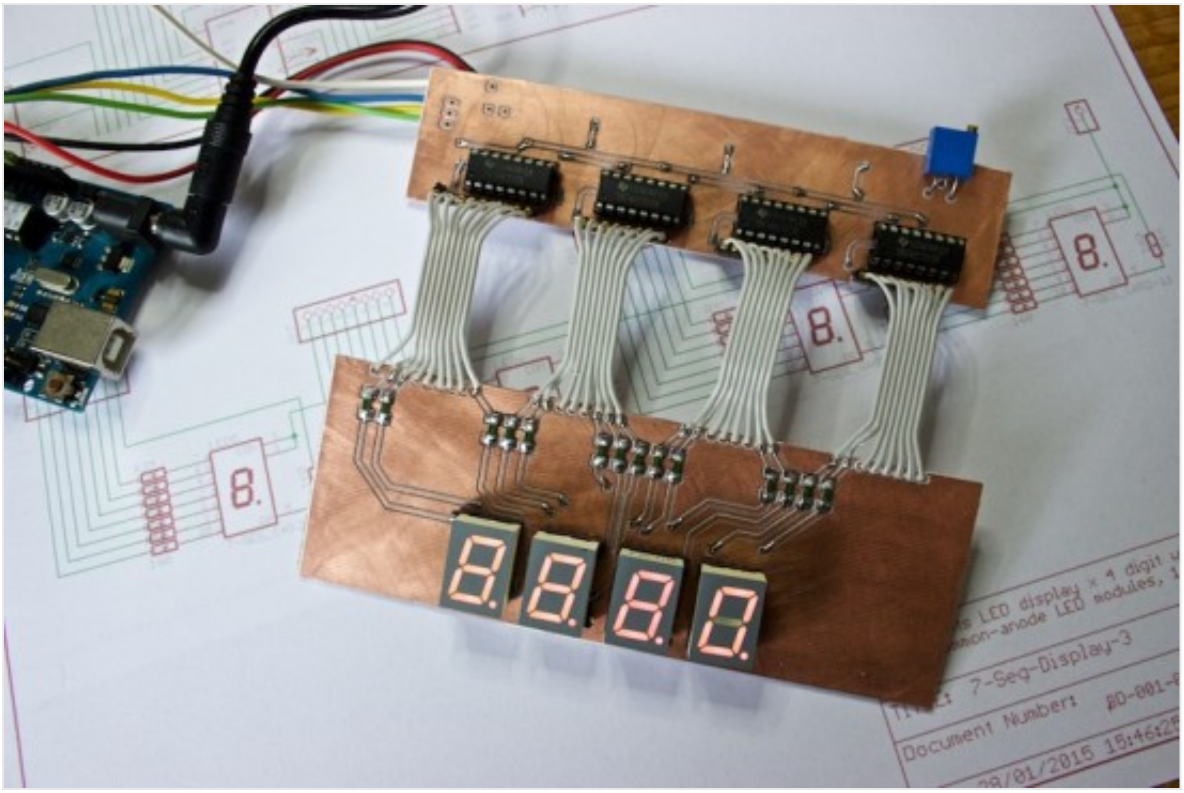
The only different thing is our sketch; We should consider that with this kind of configuration a certain LED segment will light when the corresponding digital pin of the shift register is set to LOW and vice-versa.



As the 7-segments display components have eight independent LEDs (seven for the bars and one extra for the point), every segment corresponds to one of the 8 signals/pins of the corresponding shift register so the four display board consumes four shift registers.

Why the choice of the Common Anode Display

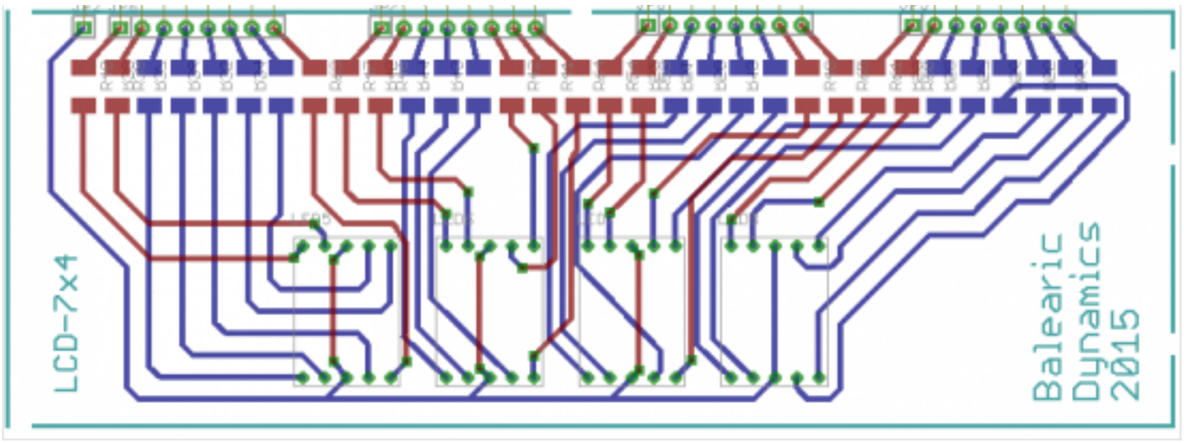
The board has a set of connectors showing 33 pins: four sets of eight pins plus the common pin connected to the anodes of the four devices. This 33th pin can be connected to the shift register board – explained in the previous article – that shares the VCC signal on its connectors: simply connect one-to-one the two boards and it will work.



But what happens when you connect the VCC common anode of the board to one of the free digital pins of the Arduino? Setting the common anode digital pin to HIGH, we power on the entire display board while setting it to LOW the display board will not work. One more suggestion: if we use an Arduino PWM pin as the common anode pin, we can power on/off the display and change its brightness.

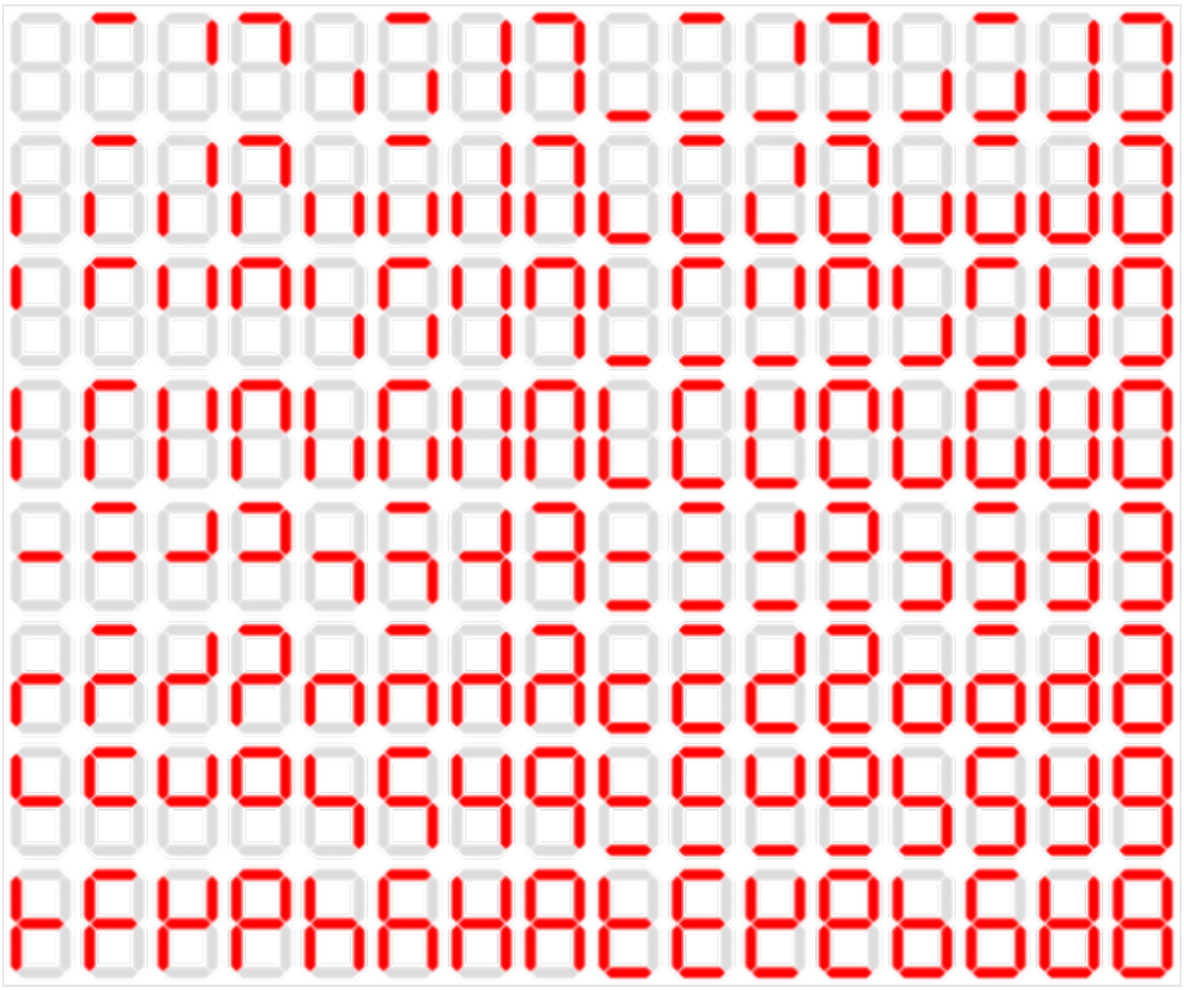
What if using Common Cathode Display

If we put in the sockets four 7-segments LED common cathode LCD display devices, we should connect the pin 33 of the board (the common pin) to the Arduino GND signal. At this point, all works fine again, the only difference being that every display LED segment is enabled when the corresponding shift register signal is ON and is disabled when the signal is OFF.



7-segments Display mapping

As every segment is associated to one of the corresponding bits (with an extra eight bit for the separator point) there are 128 different statuses of the devices. With the board and the daisy-chained shift register we can set any of these configurations but the most common are the characters **0-9** and **A-F**. For more information on how the 7-segments LED displays work, read this almost exhaustive description on **wikipedia**.



Sketches suggestions and samples

What now? There are many helpful suggestions to manage the board in your Arduino sketches. The ideal hardware configuration will be the four daisy-chained shift registers described in the first part of this article and the 7-segments board discussed above.

Don’t forget to try using an Arduino digital pin instead of the VCC pin to control the common anode signal of the LEDs.

```
#include <ShiftOutX.h>
#include <ShiftPinNo.h>

// Digital pins connected to 74HC595
int latchPin = 2;   // pin  12 on the 74hc595
int dataPin = 3;   // pin 14 on the 74hc595
int clockPin = 4;  // pin 11 on the 74hc595

// Digital pin to enable/disable the display common Anode
int vccControlPin = 7;

//shiftOutX(_latchPin, _dataPin, _clockPin, _bitOrder, _NofRegisters);
shiftOutX shiftDisplay(latchPin, dataPin, clockPin, MSBFIRST, 4);

void setup() {

  // Set the ShiftRegister PINs
  pinMode(latchPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
  pinMode(dataPin, OUTPUT);

  // Set the common Anode control pin
  pinMode(vccControlPin, OUTPUT);

  // Disable (power off) the display disabling the common Anode pin
   digitalWrite(vccControlPin, LOW);

  // Initialize the serial (for testing purposes)
  Serial.begin(9600);

  // Startup message
  Serial.println("/*** Display 7 Segments Test Started ***/");

}

void loop() {

  // Enable the display
   digitalWrite(vccControlPin, HIGH);
   Serial.println("- Display on");
   delay(2000);  // wait 2 sec

   shiftDisplay.allOn();
   Serial.println("- shiftDisplay.allOn()");
   delay(2000);  // wait 2 sec

   shiftDisplay.allOff();
   Serial.println("- shiftDisplay.allOff()");
   delay(2000);  // wait 2 sec

    // Disable (power off) the display disabling the common Anode pin
    digitalWrite(vccControlPin, LOW);
    Serial.println("- Display Off");
    delay(2000);  // wait 2 sec
}
```

You can find a pre-assembled daisy-chained four shift registers board or the PCB only at **BalearicDynamics**. Don't forget to use the discount code **ELEKTRO1008** to get 20% off discount, the special price for Electroschematics.com users.