4 Digit 7-Segment LED Display Board - “Rutherford”


You’ll find the latest versions of the instructions and example code at the address above.

Thank you for purchasing the Rutherford board! I hope you have lots of fun with it!

If you have any suggestions for future boards then please let me know. I’ll send you some free boards if I like your idea!

This kit can be powered by many microprocessors and development boards. I have been using it with the Arduino and PICAXE systems.
**Kit Contents**

This kit was put together by a team of highly skilled octopuses; it should contain the following items:

1) The main PCB  
2) A common-anode 4 digit 7-segment LED display  
3) 9x 330Ω resistors  
4) 1x 1N4001 diode  
5) 2x 16 pin DIP sockets  
6) 2x 74HC595 shift registers  
7) 2x 0.1µF capacitors  
8) A 6 pin header

**Important Things to Remember**

This kit is designed to run from 5V. **Do not exceed 5V** or the ICs and LED components will be damaged. It will work fine from 4x 1.2V rechargeable batteries.

The diode, 7-segment display and IC sockets (and corresponding ICs) must be placed on the board in the correct orientation or Rutherford will not work.

The two shift register ICs are sensitive to static shocks so handle them with care and avoid touching the legs (pins).

Before soldering any components check and then double-check that they are correctly oriented.

Don’t rush, and have lots of fun!

If you need a good tutorial on soldering then SparkFun Electronics has a good one (https://www.sparkfun.com/tutorials/106) as does Adafruit Industries (http://www.ladyada.net/learn/soldering/thm.html).
Building Instructions

You will need a soldering iron, solder, an octopus (or a set of handy helper things) and wire cutters. To keep some of the large components in place while soldering you might find Blue-Tac useful to stop them moving around or falling off, but be careful not to put it on anything that will get hot (resistor leads, LED leads etc.) or you’ll end up with a hot squidygy mess on your components.

Take your time and check the placement of every component before soldering them in place! All of the components sit on the top of the board, solder them to the underside.

STEP 1 – Resistors

Solder each of the nine resistors on to the board first. These protect the LEDs in the display from excessive current. It doesn’t matter which way around they go (but I like to make them all face the same way!).

They are labelled on the PCB as R1 to R9.

STEP 2 – DIP sockets

These sockets will eventually hold the two 74HC595 shift registers. Don’t put chips in until the end.

Place them on the board so that the “notch” is on the right hand side.

STEP 3 – The diode

The 1N4001 diode sits on the 5v line and protects the circuit from reverse polarity – that is, it will stop the sensitive components on the board from being damaged if the GND and 5v connections are connected the wrong way around.

The diode MUST be placed so that the end with the white marking is at the bottom.

STEP 4 – Headers

These will be used to connect power and data to the board.

Split the 6-pin header in to a 4-pin and 2-pin header. The 4-pin header goes in the space marked “Control” on the board. The 2-pin header goes above it, in the space marked “Power”. Place them so the pins are pointing outwards.

The control pins are labelled:

- **DS** Data (serial data input)
- **ST_CP** Latch (storage register clock input)
- **SH_CP** Clock (shift register clock input)
- **Colon** Connected directly to the colon LEDs of the 7-segment display. Like the rest of the display this is active low – ground the pin to light the LED. If you’re using this board with a realtime clock IC, such as the DS1307, then connect this pin to its **square wave** output to flash the colon at 1Hz (once a second).
STEP 5 – The LED display

This must be placed with the decimal points at the bottom (the same side as the IC sockets).

STEP 6 – The 0.1uF Capacitors

These are soldered on the underside of the board to pins 13 and 16 of the shift registers. It’s probably worth bending them slightly so their leads don’t touch any of the vias (the small silver circles that connect the top to the bottom).

Notice that pin one has a square solder pad.

STEP 7 – ICs in to the sockets

Now all of the components have been soldered to the board it’s time to place the two 74HC595 chips in to their sockets. They should be socketed so the notch on the chip is on the same side as the notch on the socket – facing to the right.

Before you try to carefully slot the chips in to sockets you’ll notice that the legs are spread out too much to fit. They’ll need to be bent inwards before they’ll fit. DO NOT FORCE THEM.

The way I like to do this is by very carefully pressing each side of the chip on my desk until both sets of pins are parallel. Once the legs are parallel it will take a bit of force to push the chips in to the sockets, be gentle and patient.

That’s all the hard work finished. You should now have something that looks like the board on the left.

Connect the board up to your microcontroller of choice and start having fun. There are plenty of example programs to get you going with code for Arduino, PICAXE and Raspberry Pi.
Output Format

To output to the 7-segment display send 16 bits to the shift register:

Set the ST_CP (latch) low

Do this section 16 times, once for each bit:

{
    Set the DS pin (Data) to low or high (depending on data).
    Set the CLOCK to low.
    Set the CLOCK to high.
}

Set the ST_CP (latch) high

Byte 1       Byte 2
EEE43210 xgfedcba

The first byte is the control byte. 0 means does nothing. 1, sets the output to digit 1, 2 to digit 2 etc. The E bits are the external outputs, these map to the “External” connector on the PCB. They might be useful, they might not, but they’re there if you need them. If you’re running low of microcontroller outputs then they could be very useful!

a-g represent the 7-segment segments. x is the decimal point.

It’s important to send the bits from LEFT to RIGHT. The MSB of byte 1 first; the LSB of byte 2 last.

Have a look at the example programs for more information on how to talk to the shift registers.
Example source code for the following platforms can be found here:

- Arduino (and compatible boards) (arduino.cc)
- Espruino (and compatible boards) (espruino.com)
- PICAXE microcontrollers (picaxe.com)
- Raspberry Pi (all models) (raspberrypi.org)

If you would like some example code for a different platform then please get in touch.

If you have any comments or suggestions for this kit then please let us know.

For more information, updates and details of new kits check out the following links:

- Website: www.MaximumOctopus.com
- Twitter: http://www.twitter.com/maximumoctopus
- Blog: http://maximumoctopus.wordpress.com
- Online store: http://store.MaximumOctopus.com
- YouTube: https://www.youtube.com/user/freshneyorg

This kit was designed and manufactured in the UK.

(c) Maximum Octopus Limited 2015
Last update: March 17th 2015