Shift-o-Tron – dual shift register breakout board


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

Thank you for purchasing the Shift-o-Tron 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) 2x 16 pin DIP sockets
3) 2x 74HC595 shift registers
4) 2x 0.1µF capacitors
5) 2x 8 pin headers
6) A 5 pin right-angled header

Important Things to Remember

This kit is designed to run from 5V. Do not exceed 5V or the ICs will be damaged. It will work fine from 4x 1.2V rechargeable batteries or 3x 1.5V alkaline batteries.

The IC sockets (and corresponding ICs) must be placed on the board in the correct orientation or the shift-o-tron 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. Don’t put it on anything that might get hot as you’re soldering as you’ll end up with a blue sticky mess around it!

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 – 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, the same side as the capacitors C1 and C2.

**STEP 2 – The capacitors**

The de-coupling capacitors sit close to the ICs, they are labelled C1 and C2. They are not polarised, don’t worry about which way around the go. Once they’ve been soldered in to place trim the legs.

**STEP 3 – Headers**

Split the 16 pin header in to two 8 pin headers. Split the right-angled header in a 2 pin a 3 pin header.

The two 8 pin headers connect to the outputs of the shift registers.
The 3-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)

**STEP 4 – 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 have something that looks like this:

Connect the board up to your microcontroller of choice and start having fun.
Output Format

Outputting to the dual shift registers is easy; send 16 bits to the shift registers:

```c
// the first 8 bits go to the leftmost shift register
// the second 8 bits to the rightmost shift register
Set the ST_CP (latch) low
Do this bit 16 times:
{
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

87654321 87654321

A 16 bit value, sent MSB first, will see the MSB at the left-most output of the left-most 74HC595. The LSB will be the first output of the right-most 74HC595.

A 16 bit value, sent LSB first, will see the LSB at the left-most output of the left-most 74HC595. The MSB will be the first output of the right-most 74HC595.

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) (raspberry.pi.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