# Debug Strip

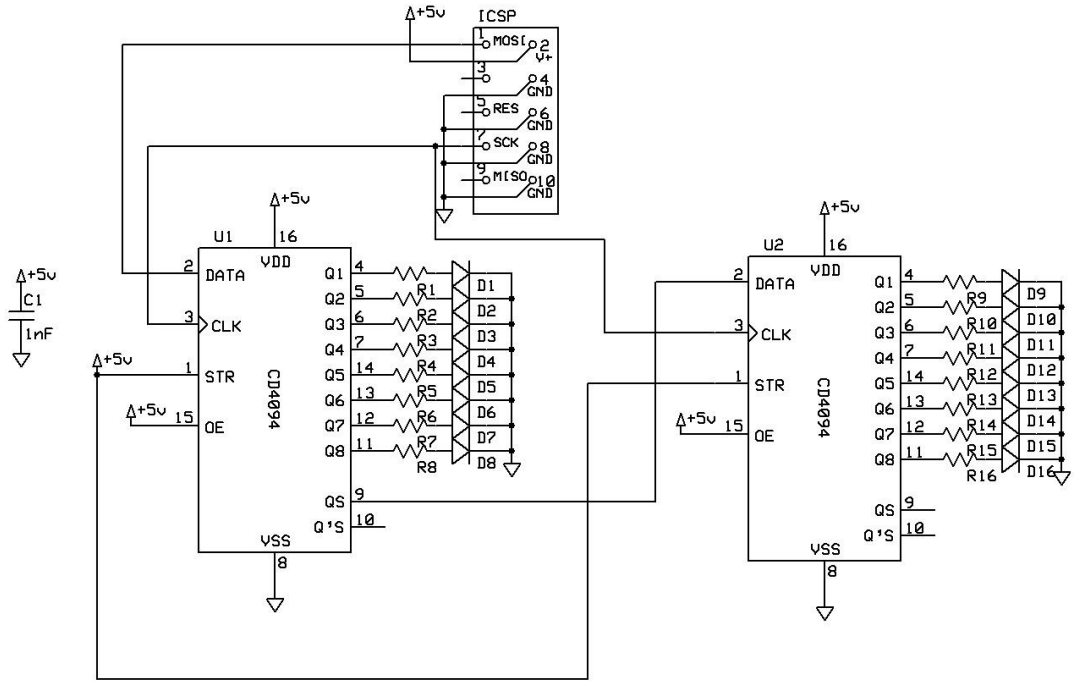
Designed by RM

The debug strip is a simple debugging tool for use in AVR systems with 10-pin ICSP programming headers. It could likely be modified for use with 6-pin programming headers or for use with PIC based systems using in-circuit programming headers.

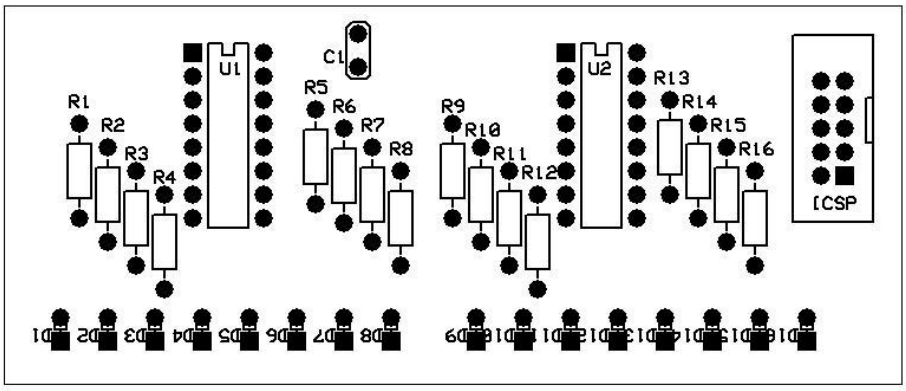
This document contains the layouts for making the device, as well as code for using it in both Assembly and C programming languages.

## *Contents*

## *Schematic, PCB and Layout*

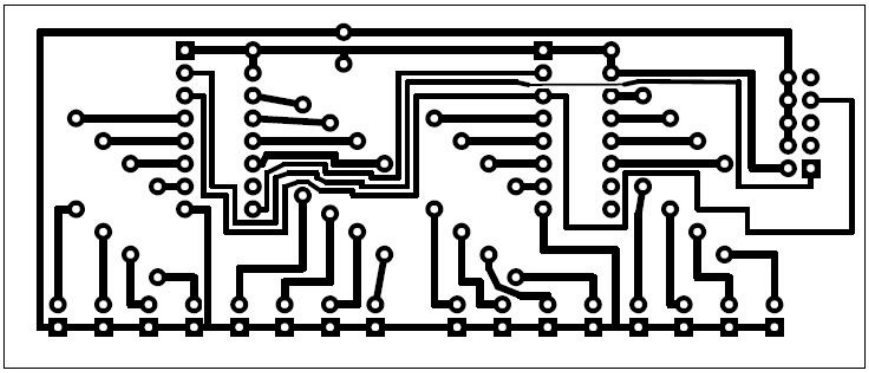



**Debug Strip**




Parts:

| | |
|---|---|
| C1 | 1nF |
| D1-D16 | 3mm Red LED |
| ICSP | 10-pin Header |
| R1-R16 | 3.3K |
| U1 | CD4094 |
| U2 | CD4094 |



## *Assembly and C code*

When using in assembly programs, a call must be made to the debug procedure below (i.e. “rcall debug”). The debug procedure will output the byte stored in the ‘data’ register. A second call of the debug procedure will shift the byte to the second half of the strip, and place the new byte in the first half.

The ‘data’ register must be defined in your register definitions, as must the ‘temp’ register. The stack pointer must also be set. Note that the procedure leaves both registers unchanged. The procedure assumes that the in-circuit programming pins MOSI and SCK are on PORTB at pins PB0 and PB2 respectively.

```
debug:
push        temp
push        data
in          temp, DDRB
sbr         temp, (1<<PB2) | (1<<PB0) ;make MOSI and SCK outputs
out         DDRB, temp
ldi         temp, 8
loop1:
sbrc        data, 0
sbi         PORTB, PB0
sbrs        data, 0
cbi         PORTB, PB0
sbi         PORTB, PB2
cbi         PORTB, PB2
lsr         data
dec         temp
brne        loop1
pop         data
pop         temp
ret
```

The two functions below are for use with C programs.

```
void debug_int(int int1)
{
        DDRB |= (1<<PB0) | (1<<PB2);
        for (int i=1; i<=16;i++)
        {
                PORTB = int1 & 1;
                PORTB = (int1 & 1) + (1<<PB2);
                PORTB = int1 & 1;
                int1 >>= 1;
        }
}

void debug_chars(char char1, char char2)
{
        debug_int(((int)char1 << 8) + (int)char2);
}
```

## *How it works*

The debug strip is basically a 16-stage shift register with the outputs going to LEDs. The stages are clocked by toggling the SCK pin, and data is entered from the MOSI pin. The resistors are high in value as CMOS ICs do not allow large output currents. High brightness 3mm LEDs are recommended if you find the light level too low.