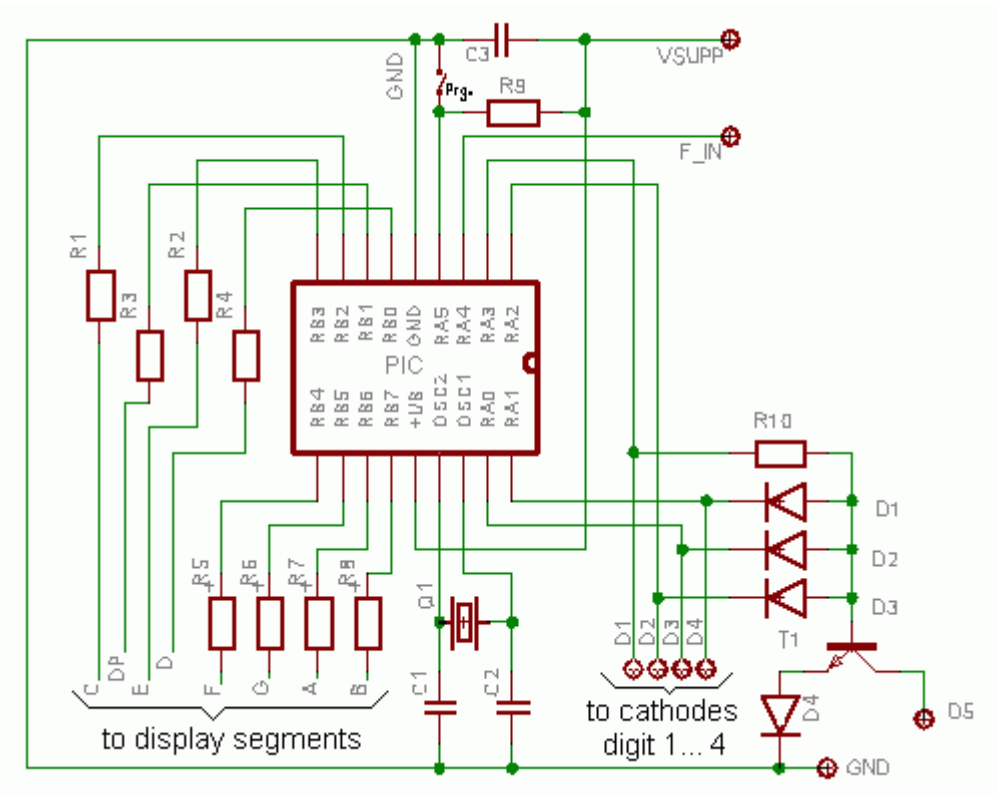
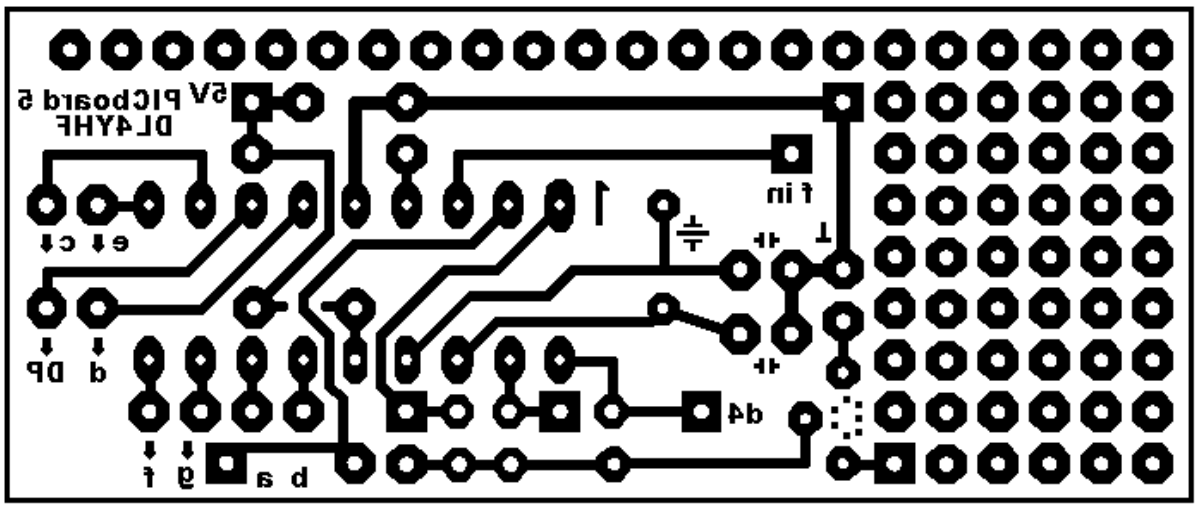
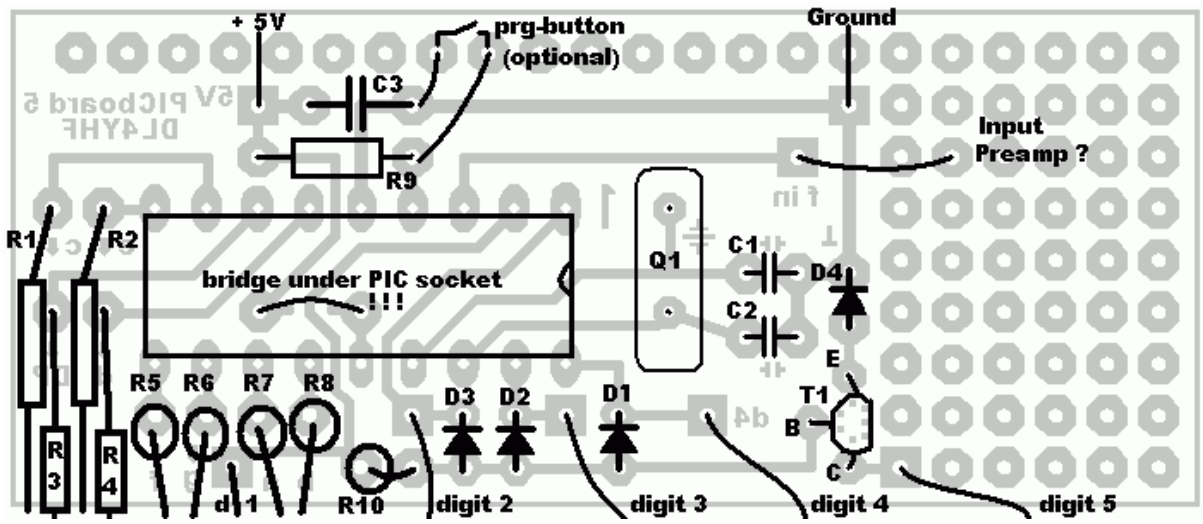
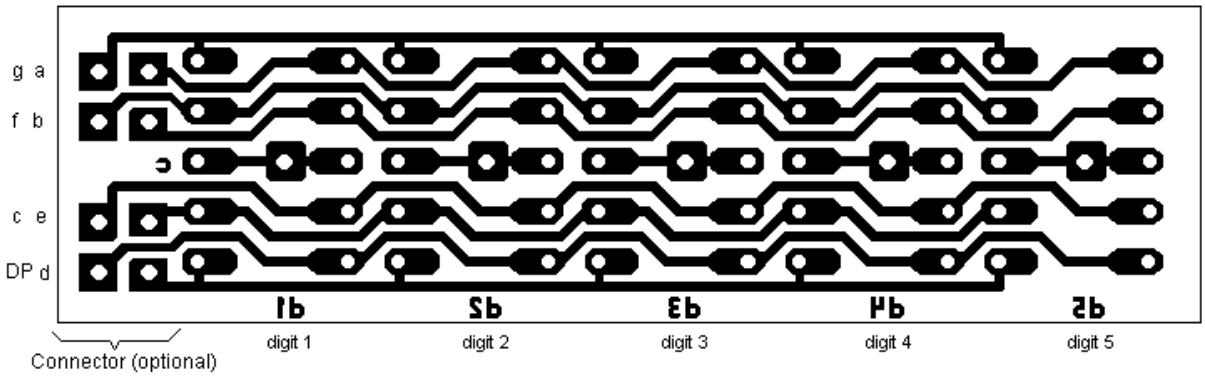
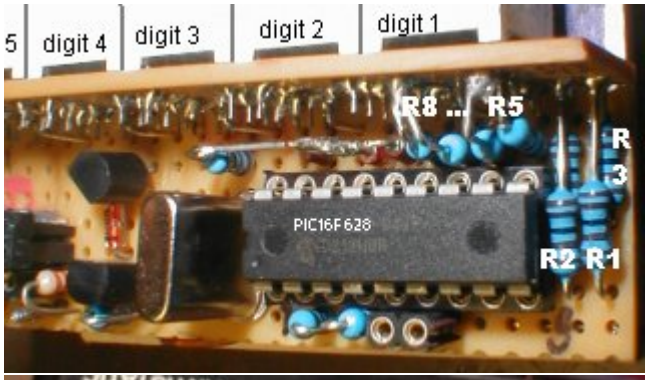

GND
C3
VSUPP
Prg.
R9
F_IN
R1
R2
R3
R4
PIC
R10
D1
D2
D3
R5
R6
R7
R8
Q1
T1
C1
C2
D4
D5
GND
to display segments
to cathodes
digit 1... 4

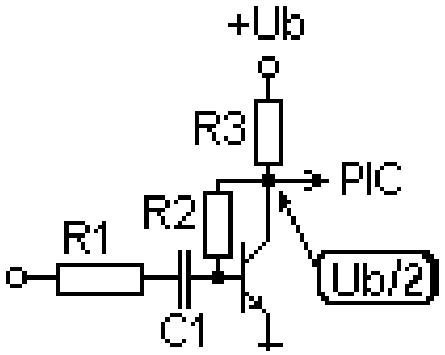




7-segment display, 5 digits SC 39-11 for DL4YHF's PIC frequency counter. Bottom layer, X-ray-view from TOP SIDE (not mirrored !)
Print with exactly 300 dpi .







With a BF199 in the preamp, R2 is 27 kOhm (+/-), and R3 must be as low as 560 Ohm to achieve the necessary bandwidth. The DC voltage at the collector and the voltage across R3 should almost equal - if not, adjust R2.

To save area on the breadboard, you can leave away the preamplifier. If you want to feed the counter with a TTL signal, leave the preamplifier away, which saves another 4 mA. If the maximum frequency in your circuit is below 10 MHz, you may increase the value of R3 and R2 by the same factor (say R3=1.2 k, R2=56k) to save some current when using the counter in a battery-powered device. R1 sets the input impedance and also the sensitivity. With R1=330 Ohm, the prototype required an input voltage of 600 mVpp (peak-to-peak) at 40 MHz and 150 mVpp at 15 MHz. If you need a higher input resistance, add a FET buffer before the bipolar transistor. Or use a fast integrated comparator as input stage if you find one in your junk box.

Frequency range	Display	Gate time	Decimal point
0 ... 9.999 kHz	x.xxx	1 second	flashing (which means "kHz")
10 ... 99.99 kHz	xx.xx(x)	1/2 second	flashing
100 ... 999.9 kHz	xxx.x(X)	1/4 second	flashing
1 ... 9.999 MHz	x.xxx(X)	1/4 second	steady (which means "MHz")
10 ... 50.00 MHz	xx.xx(X)	1/4 second	Steady

The frequency offset values are saved as a 32-bit integer numbers in the PIC's data EEPROM (at the EEPROM's first four memory locations, high-byte first, low-byte last). If you have no signal generator to produce the offset frequency for programming, or cannot tap the BFO frequency of your homebrew shortware receiver, you can enter the offset value with a suitable PIC programmer (like DL4YHF's WinPic). Use a scientific pocket calculator to convert the frequency (in Hertz, positive or negative) into a hexadecimal number, and enter this value in the PIC programmer's EEPROM DATA memory window. If you use WinPic, enable the HEX editor before typing the values into the memory window. Some examples:

```
4.194304 MHz : Add= 00 40 00 00  Subtract= FF C0 00 00  (yes, so simple)
4.433619 MHz : Add= 00 43 A6 D3  Subtract= FF BC 59 2D
0.455000 MHz : Add= 00 06 F1 58  Subtract= FF F9 0E A8
10.70000 MHz : Add= 00 A3 44 E0  Subtract= FF 5C BB 20
```

If the subtracted offset is higher than the counter's input frequency, the result of the subtraction is negative. The frequency counter makes the result positive before displaying it. This way, you can use the counter also in receivers where f_IF = f_RX + f_LO, or f_RX = f_IF - f_LO which means increasing LO frequency means decreasing RX frequency (the counter will seem to "run backwards" but that's no mistake).
Example for DL2YEO's 30 meter band QRP transceiver: f_RX = f_LO - f_IF = 14.314 MHz - 4.194 MHz = 10.120 MHz, which is the calculation inside the counter (f_LO=measured input, f_RX=display value, f_IF=programmed offset). If you don't need the 10-MHz-digit on the display, set the offset to -14.194 MHz instead of -4.194 MHz. This will give better display resolution, so you only need 4 digits (f_RX=10.120 MHz will be displayed as 120.0 kHz, which is sufficient because the receiver's tuning range is only 20 kHz anyway).

Some commonly used IF frequencies can be recalled from the "Table" menu, so you don't have to measure or enter them yourself. In many cases, there is a BFO for the last mixer (at the output of the IF amplifier) which produces a frequency close enough to the desired value.