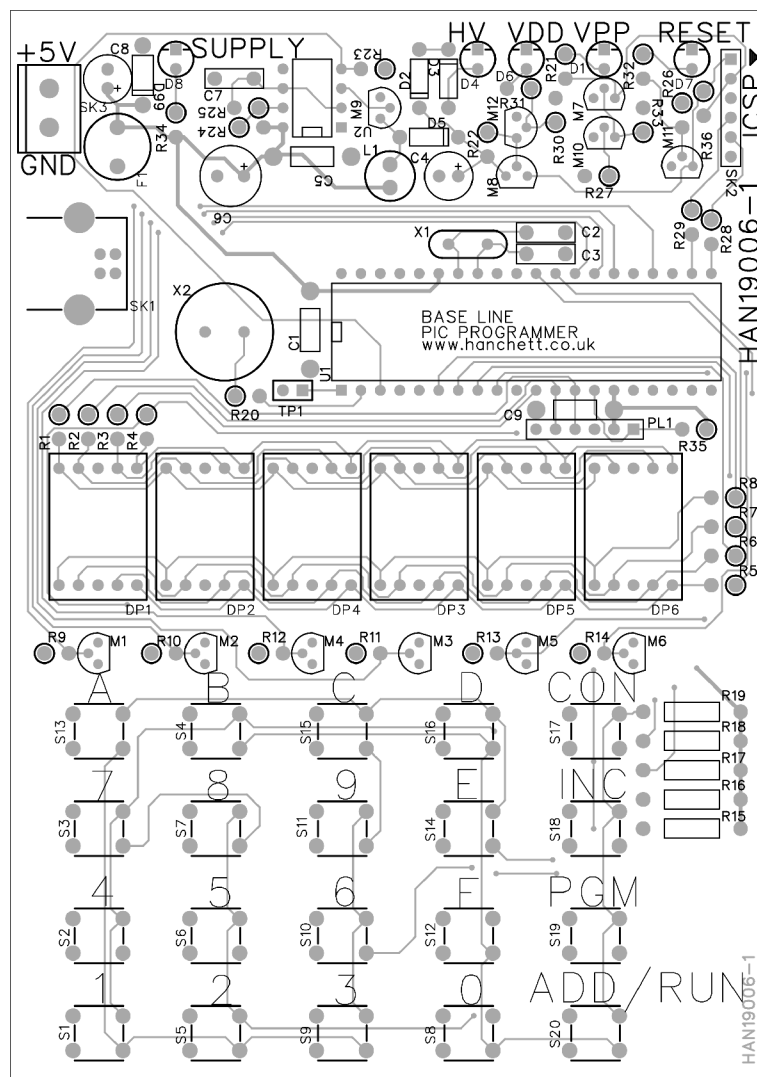


## PIC54 Programmer board quick start.

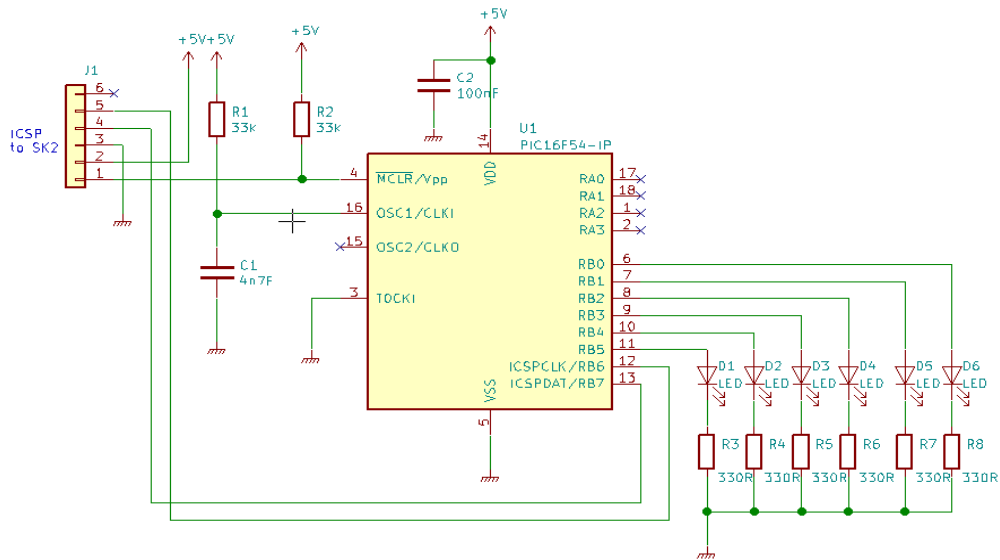
### Connecting Up.

Connect power via SK1 this is a USB B connector, the power supply can be any 5V USB source that can supply 100mA. USB chargers, USB battery packs. Alternatively SK3 provides screw terminals for a battery pack or power supply. Power can be from 3 reasonably new 1.5V Alkaline cells so long as the voltage is between 4.5V and 5.5V. Note the programmer board takes between 40mA and 70mA leaving at least 30mA for the target, be careful of loads that are driven by the target PIC pins, e.g. LEDs. The fuse F1 is 250mA and there is a reverse protection diode D99 (1N4001) to prevent damage if the power is connected in reverse, this will blow the fuse F1. LED D8 indicates that power is connected.



Connect the target to SK2 with the cable supplied.

The basic PIC16F54 tutorial circuit is shown below, this can be built on VERO board solderless breadboard etc.



### Turn On.

At power up the display will show 54 as below, indicating that the board is set up for the PIC16F54, this is the default mode.

000054

With a target connected, pressing the [CON]nect button, will result in the display showing:

3FF FFF

assuming an un-programmed chip is connected and the wiring is correct. If the display shows:

000000

Then there is probably a connection problem, one of the following conditions is likely:

- \* No device connected
- \* Device not being powered
- \* MCLR not connected/HV fault
- \* ICSPCLK/ICSPDAT swapped or not connected.

If the green HV led is brightly lit then the HV supply is healthy, if the green HV led is dim or off the there is either a short to ground on the MCLR pin, it is connected to the wrong pin or there is a fault with the HV supply. To check the HV supply remove the target and if the green LED remains off/dim the supply is faulty, this is very unlikely however.

If all was well and the display did show 3FF FFF press B on the keypad, the display will change to 3FF FFb, now press [PGM] to write the data to memory, the RST LED may flash and the display should continue displaying 3FF FFb. Press dis[CON]nect, and the display will show 000054. Now press [CON]nect again and the display should show 3FF FFb, this indicates that the device configuration word is now programmed, Code Protect and Watchdog disables and oscillator set for RC mode.

Press [INC]rement and the display will change to 000FFF this is the first address in program memory, which as it has not been programmed is set to FFF (the instruction XORLW 0xFF more on this later). The left hand three digits display the address, the right hand 3 display the contents of the location.

Now enter the following short program, for each instruction enter the 3 digit instruction, press [PGM] check that the display still shows what you enters and then press [INC]rement to go to the next address.

```
000040    clrw
001005    tris 0x05    ;set PORT A Tristate Register
002006    tris 0x06    ;set PORT B Tristate Register
003005    clrf 0x05   ;clear PORT A
004006    clrf 0x06   ;clear PORT B
```

Once you have entered this short program, press dis[CON]nect. The display will show 000054. Now press [ADD/RUN]. The green HV LED should go out, the power LED should be lit and the RST LED off, you should see that the LEDs attached to PORT B of the target come on.

Press [ADD/RUN] again to stop the program. Press [CON]nect to connect the programmer and [INC]rement until you get to address 005. Now enter the following.

```
005055    movlw 0x55
006026    movwf 0x06   ;write 0x55 to PORT B, binary 01010101
```

Dis[CON]nect and [RUN] the program. This time you should see the LEDs lit in the pattern of the bits.

Now we can change the program again to make the number in PORT B increment. Stop the program, Connect to the target at increment to address 005, we can and erase the instruction with out erasing all of memory, but we can over write it with 000 the No Operation (nop) instruction. Enter the following lines:

```
005000    nop
006000    nop
0072A6    incf 0x06,f ;increment PORTB and write the data back to PORT B
```

Dis[CON]nect and [RUN] the program. This time you should see the LEDs following a binary count, this is because the program is running to the top of memory, executing all the XORLW 0xFF instructions and the looping round to the beginning of the program. The time between LED changes is effectively 512 instruction cycles, because there are 512 memory location for the program counter to cycle through before it repeats.

Stop the program, Connect to the target and enter 000 press the [ADD]ress button, the display should show:

```
000FFF
```

now enter the following command and press [PRG]

```
00A00    goto 0x00
```

The display should show after programming:

100000

Dis[CON]ject and [RUN] the program. This time you should see the count running twice as fast because only half of memory is being executed before the GOTO instruction sends the program counter back to the increment instruction. Try moving the GOTO around memory, to speed up reduce the address, to slow down increase the address, remember to replace previous GOTOs with NOPs when you want to make the count slower.

Note, while a NOP instruction can be used to delete an instruction, leaving a pair of blank memory locations i.e. sequence of two XORLW 0xFF has no effect on the program, one XORLW 0xFF has the effect of inverting the Working register, thus two have no effect on the program apart from time and provides a convenient place to insert a GOTO or any other pair of instructions - just remember that W is inverted if you leave one of the XORLWs in place.