PIC Digital Thermometer and Clock

INTERMEDIATE SKILL LEVEL

Features:

- Firmware written with the FREE Myamicus PIC compiler from Crownhill.
- All source-code supplied.
- Supports 2" LED seven segment displays but can be customized for larger digits if needed.
- Attach up to two temperature sensors.
- Each sensor can measure temperature from −55 oC to +125 oC
- Optional on-board Real-Time-Clock with battery backup.
- Runs from a single 12v AC/DC supply rated at around 500ma
- User can customize the firmware for their own requirements using the free compiler

Issue 6– 18th Apr 2013

A digital thermometer and clock that can drive large seven segment LED displays.

This project describes the construction of a digital thermometer and clock that can be used to drive large LED displays. The project can also be fully customised if required using free software development tools.

The project consists of two PCBs (Printed Circuit Boards). One is the main display PCB and the other the logic board that contains the PIC, segment driver circuits and real-time-clock.

The display board has four, 2" common anode seven segment displays with decimal points that are used to display the temperature or time. There are two LED's mounted in the centre of the board that act as hour and minute separators, and these flash every second when in time display mode. On the left of the board are two additional 5mm LED's. These are used to indicate which temperature sensor is currently being accessed.

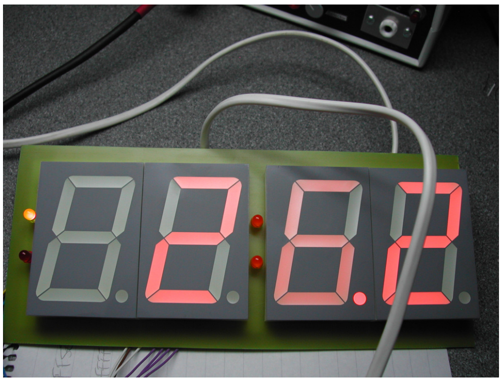


Fig 1. Completed display board

There is also provision to mount some type of light sensor LED100**** if later the constructor decides to add an automatic display dimming function.

The display board is separate from the main logic board in case the constructor wishes to drive different size or type LED displays, making it simpler to design just a new display board to suit, whilst leaving the logic board as is. The logic board contains a 18F25K22 PIC, Dallas 1302 battery-backed real-time-clock chip, two buffer chips and four transistors for the high-voltage LED driver, and a power supply.

MULTI-LED SEGMENT DISPLAYS

Usually, LEDs have a minimum forward voltage to make them illuminate of around 2v. Fig. 2 shows the internal arrangement of the seven segment LED displays used in this project, and it can be seen that each segment (except for the d.p.) is actually made of four individual LEDs connected in series. This means that a minimum of around 8v is required to illuminate an individual segment (4v for the d.p.), which is beyond the 5v the PIC can supply.

So, for the PIC to drive these displays with sufficient voltage to make them illuminate,

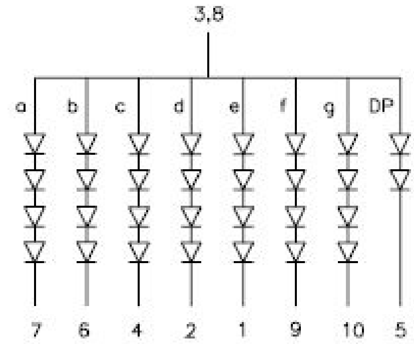


Fig 2. SA23-12EWA Seven segment LED display

some method of converting the 5v to a higher voltage must be employed.

Figures 10 and 12 show the complete circuit diagrams for the logic and display board.

CIRCUIT DESCRIPTION

Power is applied to SK1 on the logic board and this passes through bridge rectifier BR1 and into IC1 the 5v, voltage regulator. Because of the presence of BR1, either AC or DC (any polarity) can be connected to the board. If a voltage greater than around 12v is applied, a heat-sink will be required for IC1. Do **NOT** exceed around 30v on the input as this could damage the regulator. Check the data sheet for the manufacturer of your particular IC1 and make sure that the voltage rating of C1 is at least 25% over your expected maximum input voltage.

C1 provides DC smoothing and C2 & C3 provide decoupling and stability for the voltage regulator. LED1 with R22 are provided to provide a visual indication when IC1 is operating and the power is on. IC2 is the 18F25K22 PIC microprocessor. Make sure it's the K22 and **NOT** the K20 version, as the K22 is rated for up to 5.5v; the K20 is only rated to 3.3v. The external clock signal for the PIC is generated by X1, C4 & C5. C6 provides some additional de-coupling for the PIC. C4 & C5 can be in the range 15pf to 30pf.

IC3 and IC4 are the main segment driver...

..ICs. These contain NPN Darlington transistors in a handy 18-pin package (and are simple to replace incase of accidents). The outputs from IC3 are fed to the seven segment cathodes via resistors R1 to R8 and these provide current limiting for the display segments. R8 is a higher value resistance than the others as it limits the current to the d.p. segment, and this comprises only two LEDs and not four as used by the other segments. You could reduce the value of these resistors to increase the current flow for different LED display types, but you would probably have to use up-rated PNP transistors as they are already running a little warm.

With an input voltage of around 14v, after the voltage drop caused by the bridge rectifier, around 12v is present at the emitters of the transistors.

IC4 drives the remaining four indicator LEDs and also the four PNP transistors, Q1 to Q4 that in turn switch the high voltage DC to the display anodes, provided via jumper J1. Using PNP transistors to switch isn't as simple as using NPN transistors, and they need to be first driven by an NPN transistor, or in this case, four of the outputs from the IC4.

The PIC will drives each of these transistors (via IC4) in turn; this is called multiplexing and means that reality, only one of the four digits is ever on at any one time. However, the PIC drives the digits so fast, they appear almost static to the human eye. Multiplexed displays significantly reduce the number of connections and the amount of driver hardware required, at the cost of a slight display flicker, and the CPU code having to constantly refresh the display. They also reduce the power requirements slightly as the human eye adds a certain amount of "persistence" to the image it sees, meaning that the segments don't have to be illuminated for as long.

Options

This project is quite flexible in it's hardware and there are several options that the constructor may, or may not wish to include. The PIC firmware will scan the completed project on power-up to decide which options are present.

Real-Time-Clock *

BT1, J2, IC5, X2 and C7 form the real-time-clock module and if you do not require the RTC, these components can be omitted.

J2 is used to isolate the battery from the RTC chip if the project is left unconnected for extended periods of

COMPONENT LIST

Resistors

R1 to R7 (7 off)	47R (see text)
R8	100R
R9 to R12 (4 off)	1K (see text)
R13 to R16 (4 off)	10K
R17**	100R
R18	1.5K
R19 to R21 (3 off)	4.7K
R22	470R

Capacitors

C1	220uf / 35v (see text)
C2, C3, C6, C7 (4 off)	100nf
C4, C5 (2 off)	15pf to 30pf
C8**, C9** (2 off)	100pf

Semiconductors

IC1	7805 1A positive voltage regulator
IC2	PIC18F25K22
IC3, IC4 (2 off)	ULN2803A
IC5*	DS1302 RTC
IC6, IC7***	DS18B20 Temp.
D1	1N4148 or small signal diode
D2**	BAT85 or small schottky diode
BR1	Rectifier W005
Q1 to Q4 (4 off)	2N3906 PNP
LED1 to LED4 (4 off)	SA23-12EWA (CA) LED displays
LED5, LED6, LED8, LED9 (4 off)	5mm Red LED
LED7	5mm Green LED

Miscellaneous

X1	20 MHz XTAL
X2*	32.768 KHz (must be 6pF load capacitance)
BT1*	3.6v 80ma NiMh battery
SK1, SK8**** (2 off)	2 pin Molex
SK2**	6 pin Molex
SK3, SK9 (2 off)	12 pin Molex
SK4, SK10 (2 off)	7 pin Molex
SK5, SK6, SK7 (3 off)	3 pin Molex
J1,J2*	2 pin jumpers
	Push button for time setting.

You will also require sockets and crimps for the Molex connectors, connecting wire, IC sockets and the PCBs.

All resistors are 1/4 watt unless otherwise specified. When using IC sockets, make sure you choose ones of good quality.

time and will help prevent the battery from over-discharging.

The PIC firmware configures the RTC when the project is powered up, and forces the RTC to use 24 hour format and specifies the type of backup battery connected; in this case an HiMh. **Do NOT** substitute a different battery type without updating the PIC firmware. Read the datasheet for the DS1302 first. Only the time portion (hours and minutes) are output on the display but the RTC supports full date and time, and the firmware could be changed if required to use these.

When in time mode, the seconds passing are indicated by the flashing centre LEDs. A simple push-to-make push button can be connected to SK6 to set the time.

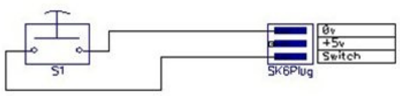


Fig 3. Time setting push-button cable

Holding the push button advances the time by one minute in quick succession. If the RTC chip is missing when the project powers up, the time function will be automatically disabled.

ICSP ** *(In-circuit software programming)*

Assuming you wish to program the PIC on-board, you will need to add the items from the parts-list marked with **. R17, C8 & C9 help with skew on the program and data lines with longish cables from the PIC programmer. D2 is required to prevent the PIC programmer's +5v from attempting to power the full project board; most programmers couldn't handle the load.

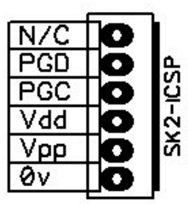


Fig 4. ICSP Connections to Logic board.

A schottky diode is used as the voltage drop across a standard silicone diode is a little too high. (You can try a standard diode if that's all that's available). SK2 connects to your PIC programmer.

Temperature sensors ***

Provision has been made for up to two temperature sensors (Maxim DS18B20) to be connected to the board via SK5 and SK7. One could be positioned indoors and the other outdoors for example.

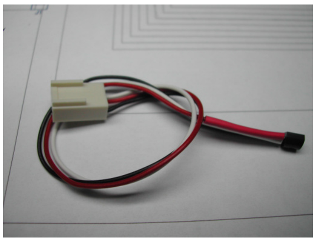


Fig 5. Temperature sensor and cable with Molex plug

If only one temperature sensor is required, SK7 should be left unconnected. On power-up, the PIC will initially scan for both sensors and you will see "-0.0" displayed if the second sensor is missing. After this initial scan, the firmware won't attempt to check for this sensor again, or display a value for it until the board is reset.

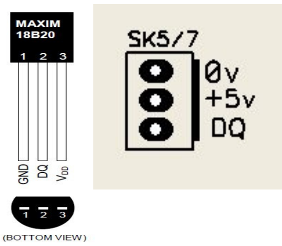


Fig 6. Temperature sensor and SK5/7 connections

When connecting the DS18B20 to SK5 or SK7, you have to make a "twist" for the DQ and Vdd(+5v) pins.

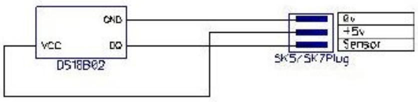


Fig 7. Temperature sensor leads

The DS18B20 actually allows for many devices to be connected to the same signal wire, and even for power to be applied along this signal wire (parasitic mode), but to maintain flexibility in the hardware neither of these features have been used. You are of course, free to change the design as you wish.

Light Sensor ****

J1 on the logic board allows for the input voltage to be disconnected from the LED drivers. This is a good place to "insert" an additional circuit if the constructor wishes to construct some form of automatic dimming control. To help with this, there is provision on the display board for a sensor (LED100) and SK8; a 2-pin Molex connector.

Construction

Construction should be straight forward with no real problems expected.

Starting with the logic board first, solder the IC sockets and then the Molex connectors (watch for correct orientation), followed by the two, two-pin jumpers. Next, the resistors, capacitors and then the semiconductors. Make sure you insert the bridge rectifier, LED and diode the correct way around. Also check the polarity of C1. There is one link wire to solder on the PCB underside.

When soldering the transistors, make sure you insert them correctly. If your not using the ones specified in the parts list, check the datasheet to make the sure pin-outs are correct. You may have to "cross" two of the transistor leads for the correct orientation. In this case, slip a small piece of insulation over one of the crossed legs to prevent short-circuits.

Solder in the crystals next. If X2 is being fitted, a drop of nail varnish or glue will secure it to the PCB surface.

Finally solder in BT1 if required.

First Tests

Once construction is complete there are a couple of basic tests to perform before you insert any of the ICs in their sockets.

Connect a PSU supplying around 12v (AC or DC) @ 250mA to SK1 (any polarity) and using a volt meter, check the voltage on pin 1 (input) of the 5v regulator. Touch the negative lead of the meter to the metal tab of the regulator for a negative connection. The voltage should be around 2v less than the input voltage at SK1 because of the voltage drop across the bridge rectifier. Next, check pin 3 (output) of the 5v regulator. This should read around 5v (it may be +/- 0.5v). LED1 should also be illuminated. If the voltage is present but the LED doesn't illuminate, check it's polarity. Next, check the voltage on the PIC power pins. Place the meters positive probe in pin 20, and the negative in pin 19 in the IC socket. Again, you should see a voltage of around 5v. If not, check that you have soldered in the jumper wire on the PCB underside correctly.

Switch off the power, and carefully insert all the ICs, making sure that they are oriented correctly and none of the pins bend under the IC body.

Next, assemble the display board. There are four wire links that **MUST** be soldered in before the LED displays. Once the links are in place, solder in SK9 and SK10. These solder from the reverse of the board and this is best done by carefully aligning each connecter over its holes or pads, and "tacking" one pin just to hold it in place. Solder a pin at the other end of the connector and then once its aligned correctly, solder the remaining pins. Finally solder in the four LEDs making sure their polarity is correct.

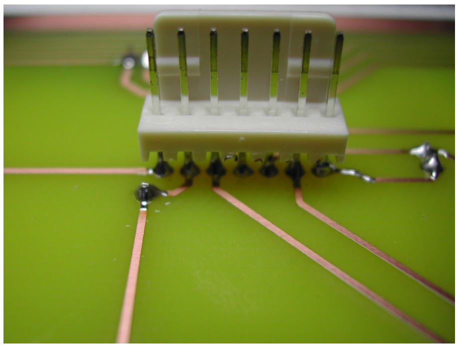


Fig 8. SK10 soldered to the underside of the display board.

Interconnect Cables

Two interconnect cables are required to link the display board with the logic board. These are made from two, eleven way, and two seven way Molex plugs.

The interconnect cables should be wired as follows:

Pin 1	Pin 11	Pin 1	Pin 7
Pin 2	Pin 10	Pin 2	Pin 6
Pin 3	Pin 9	Pin 3	Pin 5
Pin 4	Pin 8	Pin 4	Pin 4
Pin 5	Pin 7	Pin 5	Pin 3
Pin 6	Pin 6	Pin 6	Pin 2
Pin 7	Pin 5	Pin 7	Pin 1
Pin 8	Pin 4		
Pin 9	Pin 3		
Pin 10	Pin 2		
Pin 11	Pin 1		

Fig 9. Interconnect cable connections.

Final Tasks

After a final visual inspection of all soldered joints, attach the display and logic boards together using the two cables you've just assembled, and insert either a pre-programmed PIC, or attach a suitable PIC programmer and program a blank PIC.

PCB Holes

If you decide to make your own PCB's, drill all component holes with a 0.8mm bit except for IC1, BR1, all Molex connectors and the three BT1 holes. These should be drilled using a 1mm bit.

The holes for the seven segment displays depend on the display types used, but it's suggested to again drill with a 1mm bit.

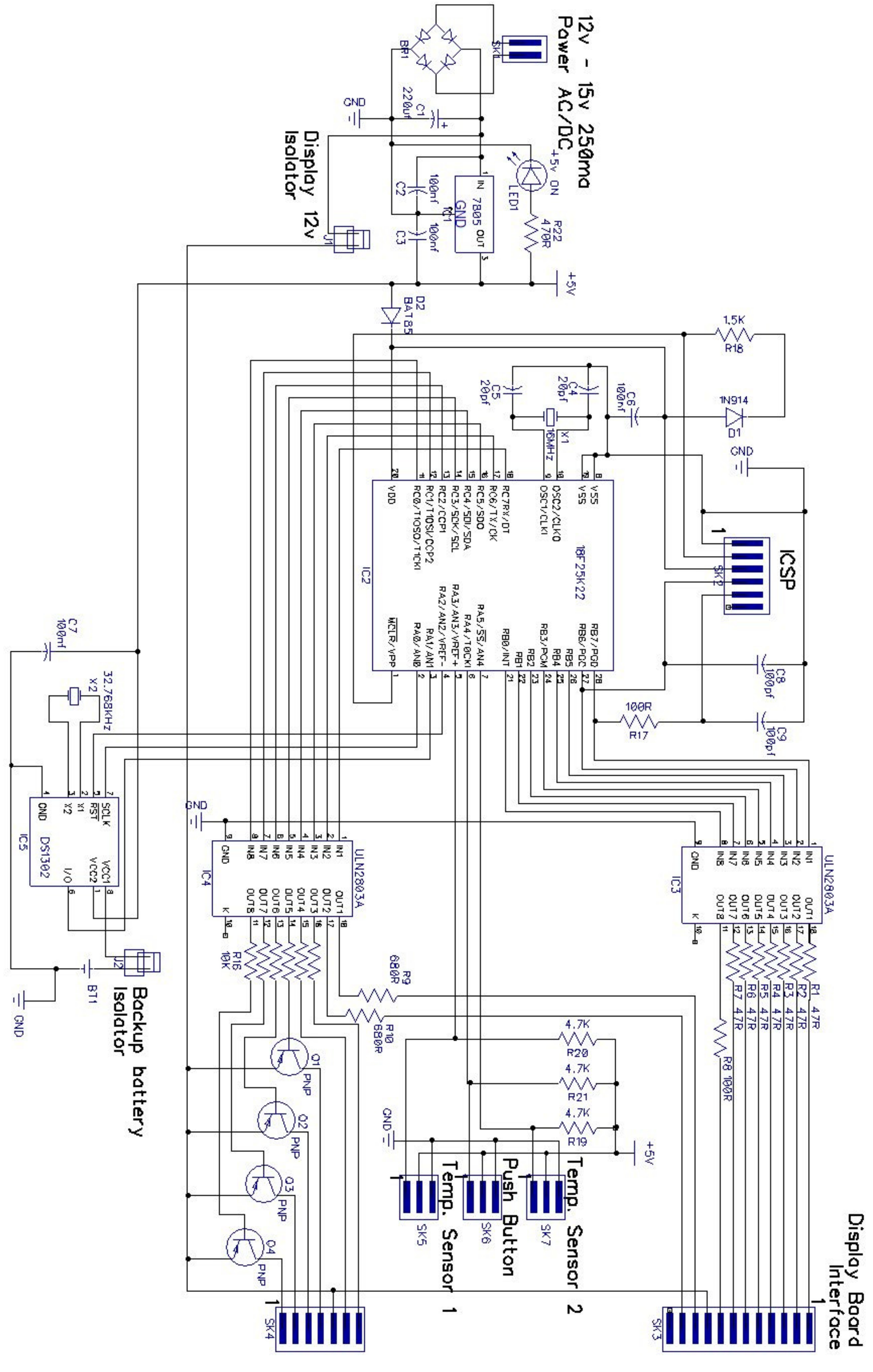


Fig 10. Logic Board Circuit Diagram

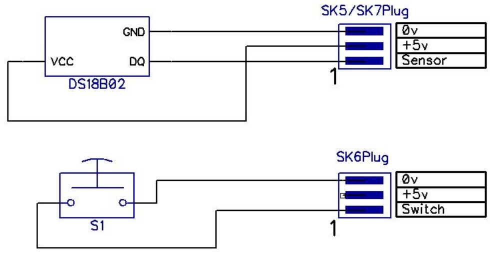


Fig 11. Push button and sensor cable assemblies.

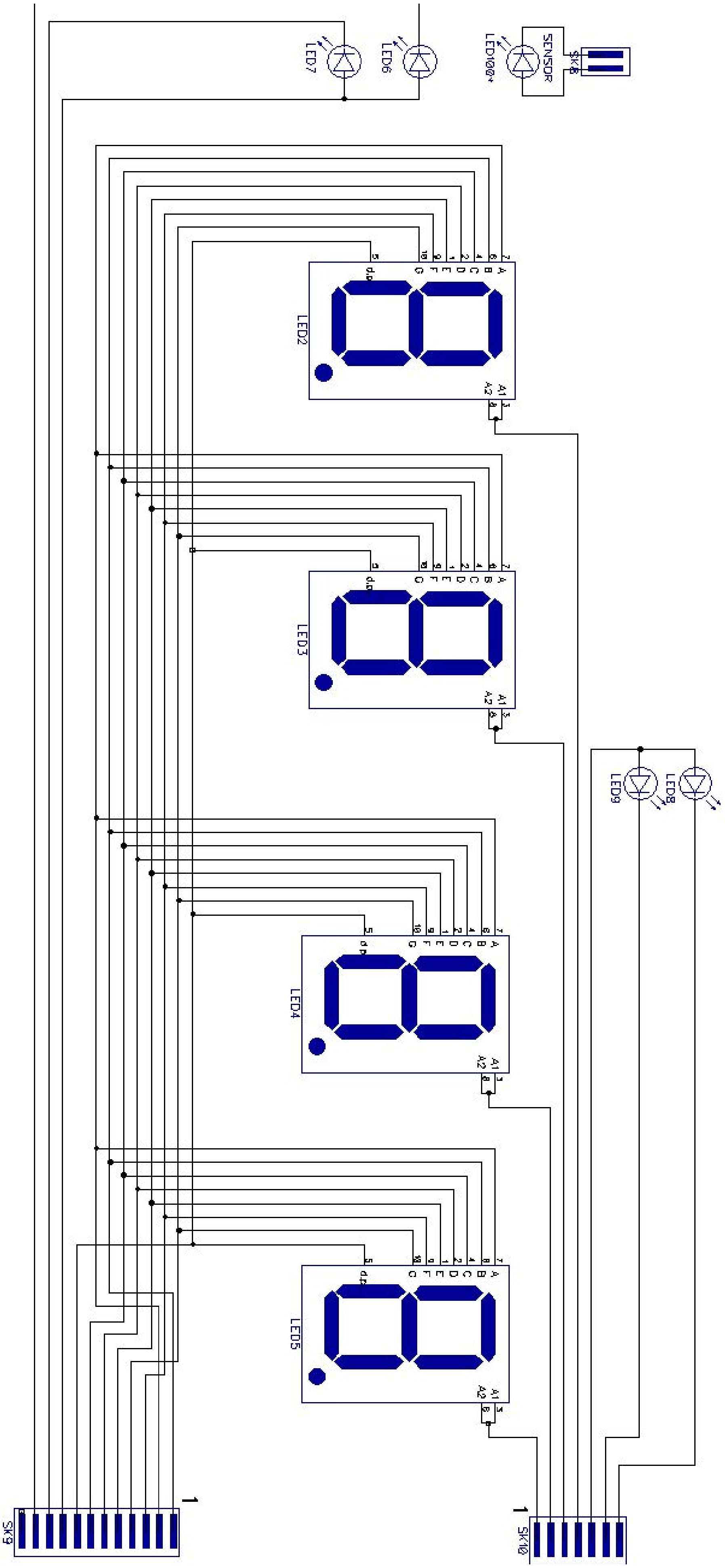
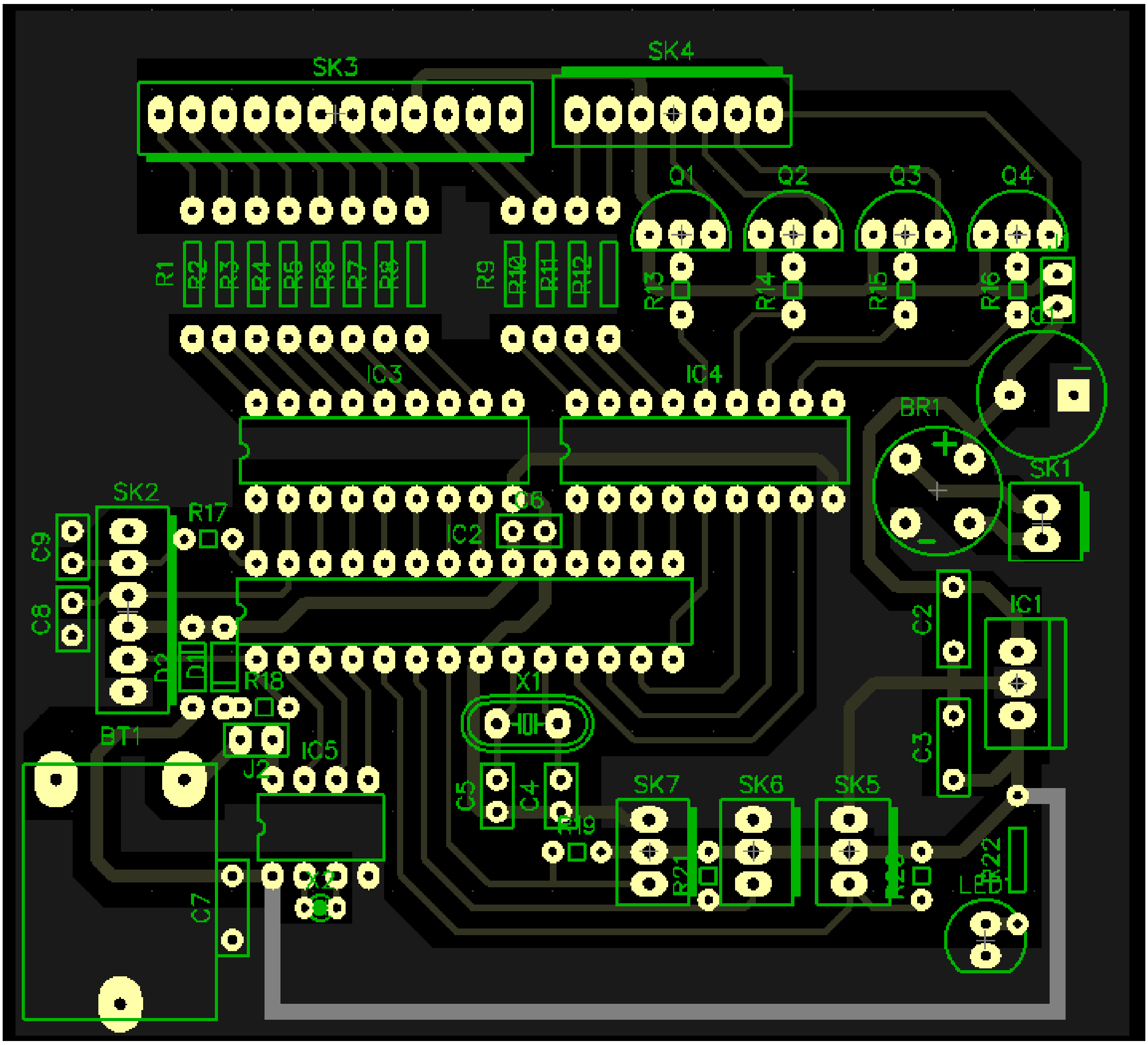
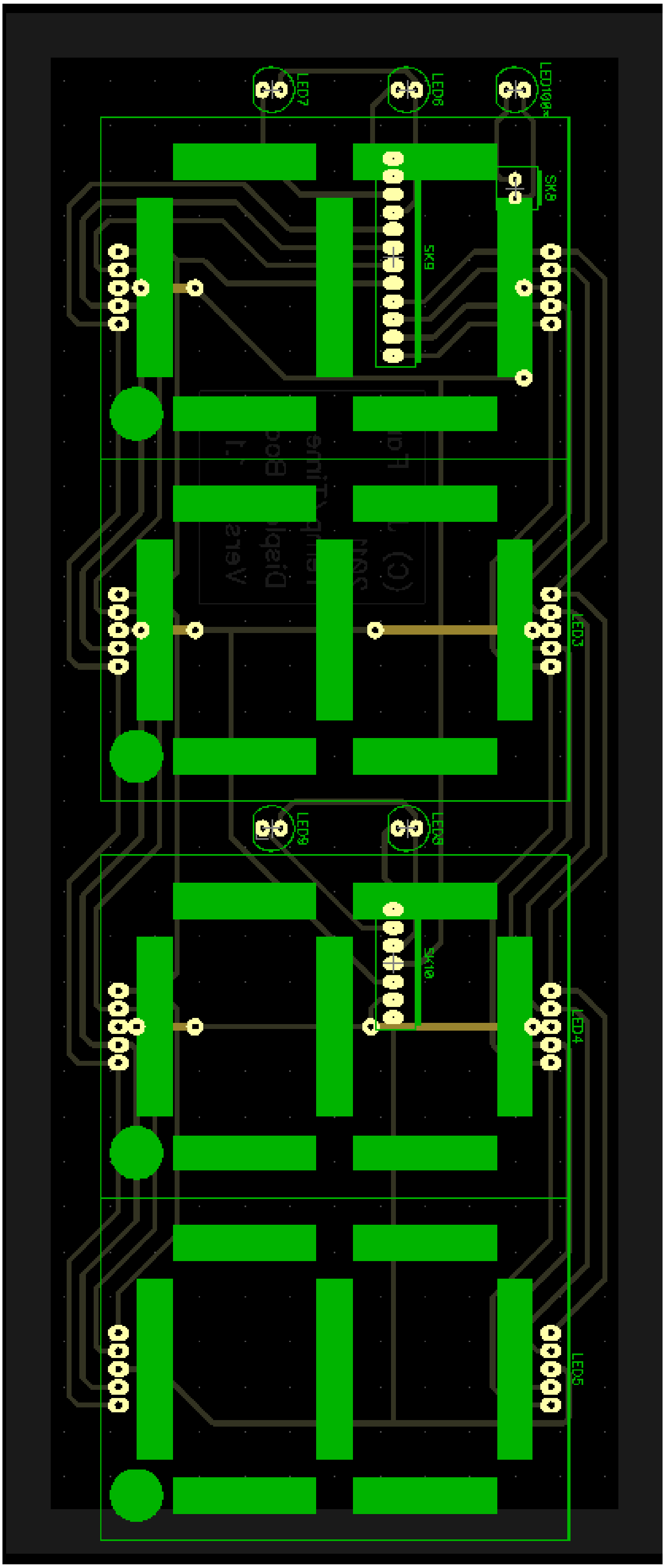


Fig 12. Display Board circuit diagram



The grey trace is the wire jumper you need to solder to the PCB underside !

Fig 13. Logic Board PCB Layout (Not to scale)



The four jumpers are shown by the yellow lines need to be on the top-side

Fig 14. Display Board PCB Layout (Not to scale)

Once programming is complete, attach at least one temperature sensor, (time setting cable if required), connect the power and switch on.

Instantly a value should be seen on the display. It's common for the first temperature readings to be garbage (often 85 oC). After the display has cycled through both temperature readings and one clock display, the PIC will have decided which physical hardware options are installed and from the second pass onwards, will only display data from those options that are installed.

PIC Firmware

The firmware for this project is available for download as a .HEX file that can be directly programmed into the PIC via a suitable PIC programmer, but the constructor may wish to make changes to the original program. The firmware was written in Proton BASIC from Crownhill, however, recently Crownhill made a special version of their BASIC compiler freely available called AMICUS18. This is designed to be used for a PIC version of the Arduino system. Arduino usually only runs with an Atmel AVR CPU, but not any more. AMICUS18 supports the PIC18F25K20 and PIC18F25K22 devices (the only difference being that the K20 is a 3.3v chip, and the K22 is a 5.5v version). The firmware was moved to the Amicus compiler and is fully compatible.

The AMICUS18 compiler is freely available from www.myamicus.co.uk. Check back often for the latest version as they often update it.

The source code has been fully documented and should be fairly easy to follow.

You should make sure that the four

Finding Parts & Substitutions

PNP transistors are suitable for your needs. You can also replace the Molex connectors with vero-pins and solder the connections but this isn't as flexible a solution. You may be able to use other connectors if Molex (or their equivalents) aren't available.

Most of the parts for this project were sourced from Rapid Electronics in the UK. They don't unfortunately have the DS18B02 thermometers or the PIC. When buying the Molex connectors, they sell several brands; some are MUCH cheaper than others !!! You need to buy crimp sockets for the plugs as

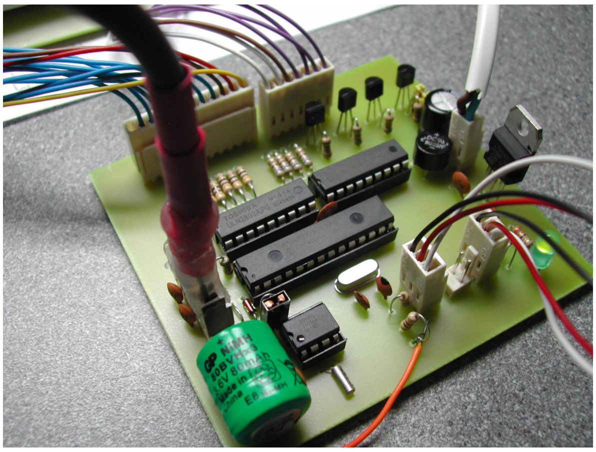


Fig 15. Original prototype version of the main Logic board. Notice the 2-pin Molex connector used for the push-button switch has now been replaced with a 3-pin connector.

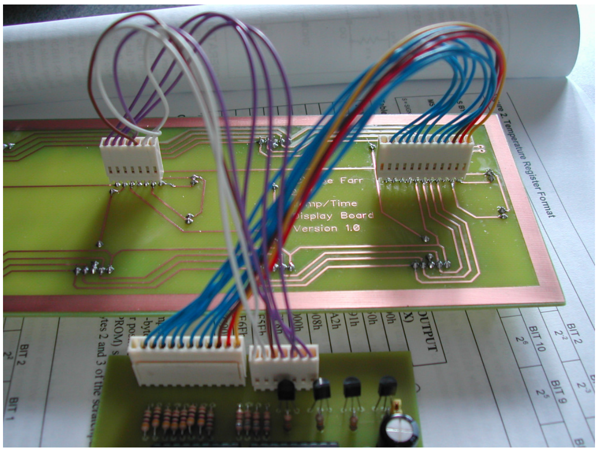


Fig 16. Notice the polarity of the board interconnect sockets. Check the article text for correct wiring. (You can also see that I used 12way instead of 11way Molex connectors)

well. I solder these.

All files including the source code, .HEX file, PCB foils can be found on my website.

Useful Websites

PIC Compiler	www.myamicus.co.uk
LED display datasheets	www.datasheetdir.com/SC23-12EWA+download
DS18B02 digital thermometer	www.maxim-ic.com/datasheet/index.mvp/id/2812
DS1302 Real-time-clock	datasheets.maxim-ic.com/en/ds/DS1302.pdf
Free version of DIP trace for reviewing and printing the PCB and circuit diagrams	www.diptrace.com
Main hobby electronics web site	http://hobbyelectronics.weebly.com

If you've enjoyed this article, found any mistakes, have any comments, or just thought it was fantastic and want to drop me a line, please Email me at: hobbyelectronics@kcsl.uk.com