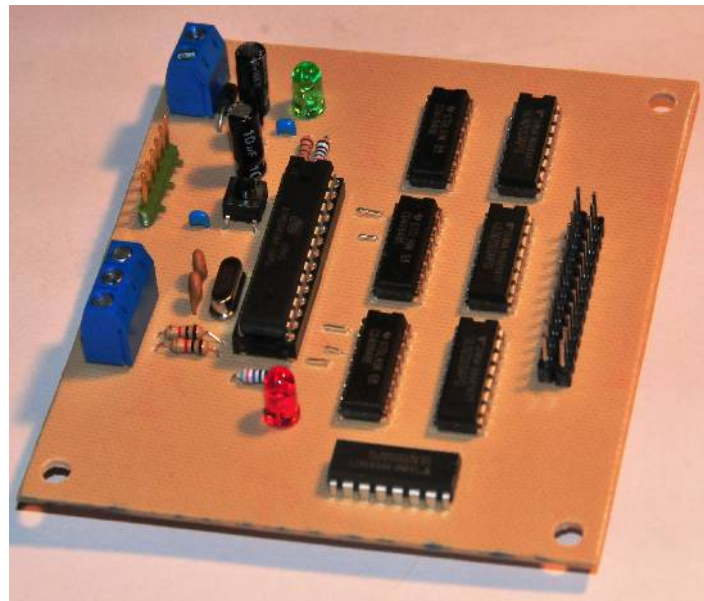


# Word Clock

## Arduino Controller PCB Construction Notes

A fun clock to build



Doug Jackson  
VK1ZDJ

March 2010

## Licence

The Word Clock Design, PCB layout, Manual, and Firmware is Copyright 2010, by Douglas Jackson, VK1ZDJ.

This Design, PCB layout, Manual and Firmware is licenced under the TAPR Open Hardware Licence. [www.tapr.org/OHL](http://www.tapr.org/OHL).

Permission is granted for anybody to:

- Modify the documentation and make products based upon it.
- Use the products for any legal purpose without limitation.
- Distribute unmodified documentation, but you must include the complete package as you received it.
- Distribute products you make to third parties, if you either include the documentation on which the product is based, or make it available without charge for at least three years to anyone who requests it.
- Distribute modified documentation or products based on it, if you license your modifications under the OHL.

If you do use my design to manufacture a kit, I would appreciate it if you would attempt to send the modified documentation by email to [doug@doughq.com](mailto:doug@doughq.com). This is a good faith obligation -- if the email fails, you need do nothing more and may go on with your distribution.

## Welcome

Thanks for deciding to make this great little clock project. You will find that this simple clock will be a centre piece for many discussions into the future, as well as providing a great way to tell the time.

Because there are so many ways to construct the clock, I have broken assembly down into various documents – this document details the construction of the Arduino Controller board.

You will find that construction of this clock is very simple. If you are methodical with your construction practices, and careful with you soldering, you will find that the clock almost assembles itself.

## Parts List

The first thing you need to do is to verify that you have all of the necessary components required to assemble your clock. Here is the complete parts list. Feel free to check off each component as you verify it is present.

### Capacitors

|       |   |
|-------|---|
| 10uF  | 1 |
| 100uF | 1 |

|       |   |
|-------|---|
| 18pF  | 2 |
| 0.1uF | 2 |

### Resistors

|      |   |
|------|---|
| 270R | 2 |
| 10k  | 1 |

|    |   |
|----|---|
| 1k | 2 |
|    |   |

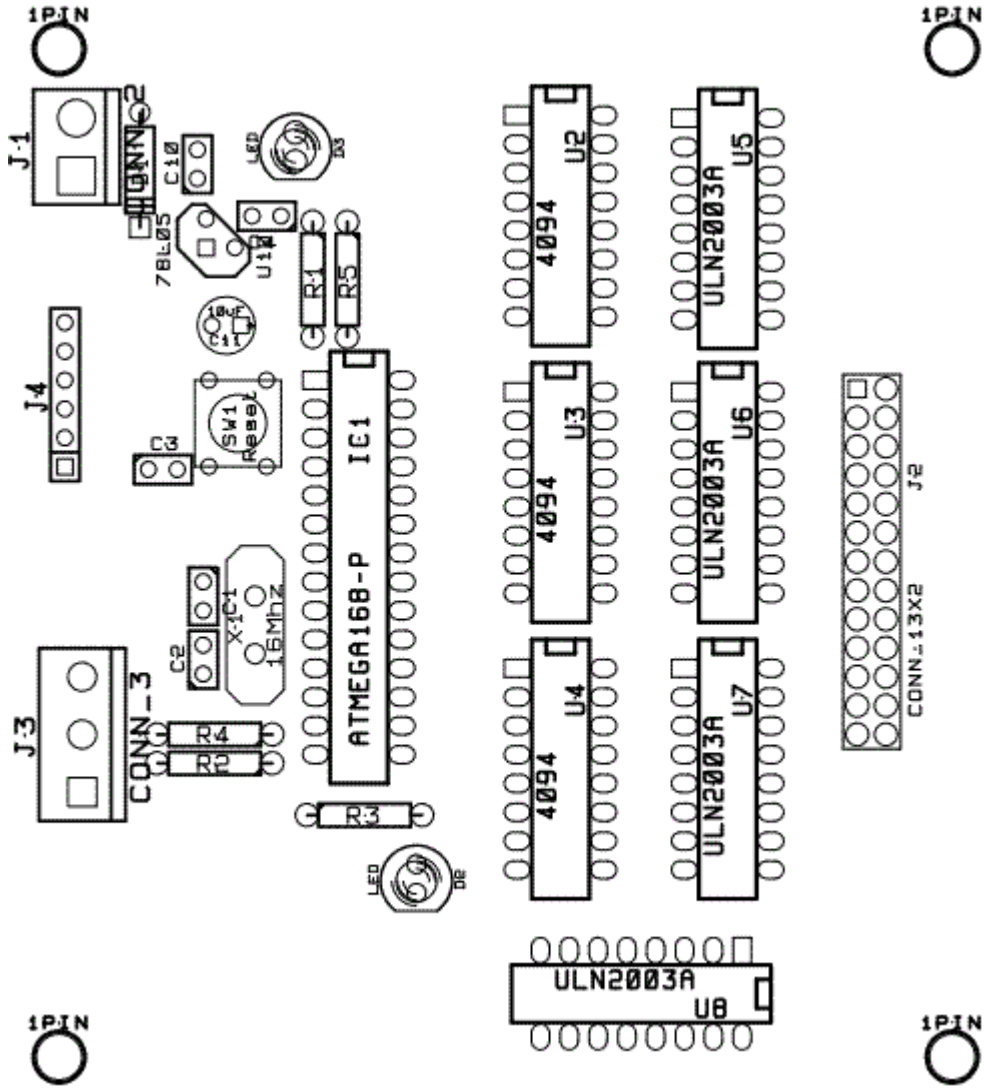
### Semiconductors

|           |   |
|-----------|---|
| AtMega168 | 1 |
| CD4094    | 3 |
| RED LED   | 1 |
| 16 Mhz    | 1 |

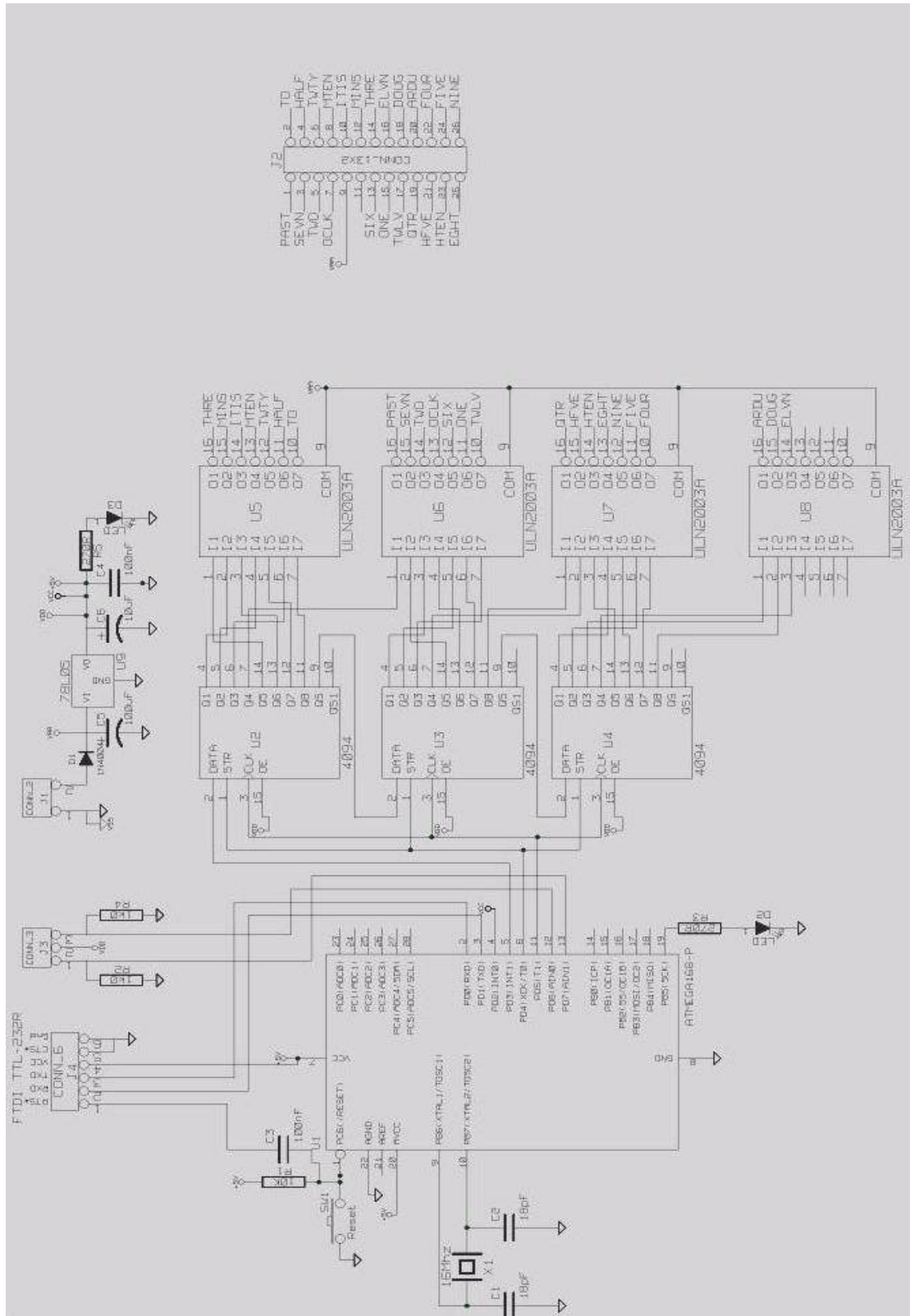
|           |   |
|-----------|---|
| ULN2003A  | 4 |
| 1N4001    | 1 |
| GREEN LED | 1 |
| 78L05     | 1 |

# Component Layout

Use the following diagram to assist you in locating components on the Arduino Controller PCB:



# Schematic Diagram

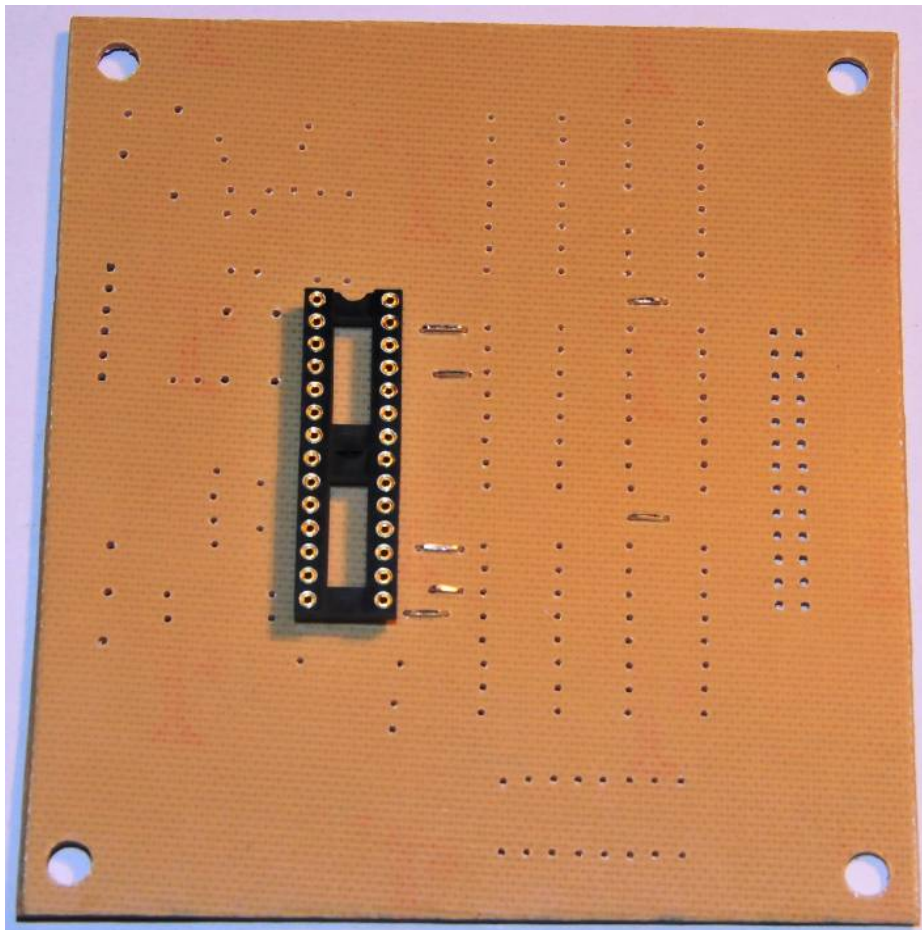


## Construction

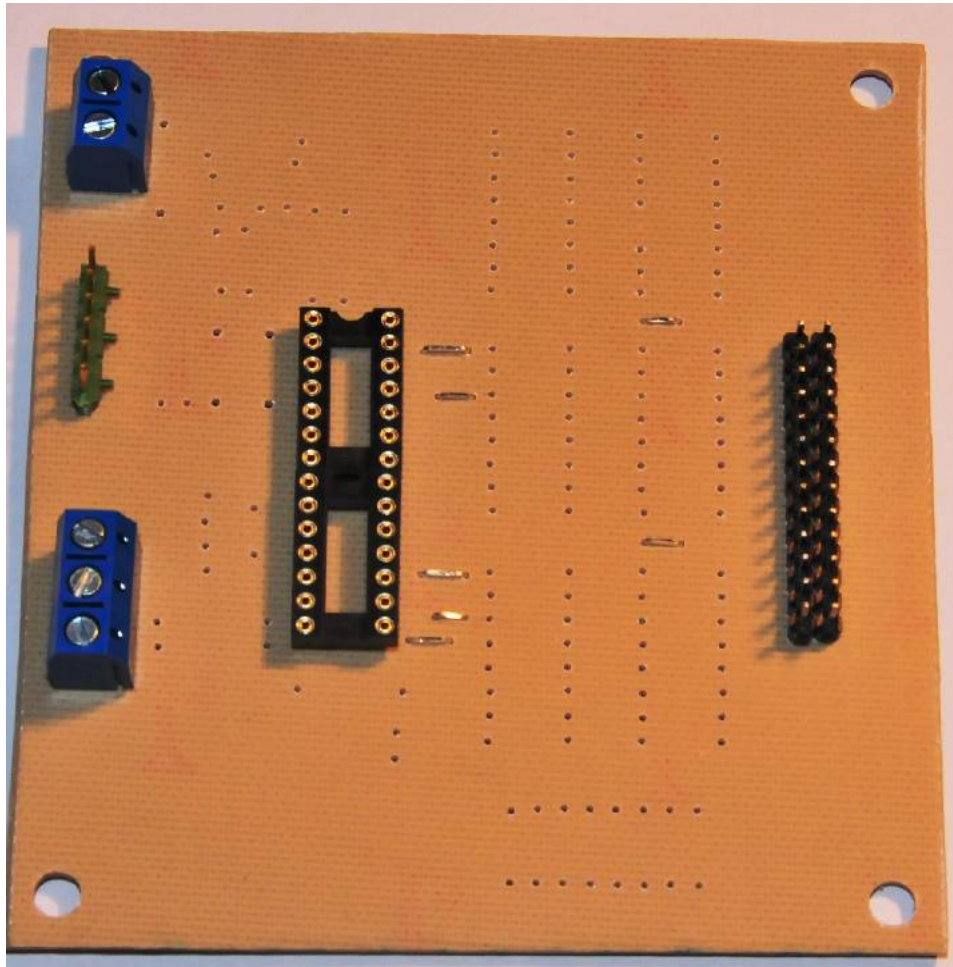
Assembling the controller board is fairly simple. Start by inserting the links on the component side of the PCB, and the IC socket for the CPU. Care must be taken to ensure that no solder bridges are produced during soldering, linking adjacent tracks on the PCB.

You may find that the protective coating that has been applied to the hand made PCB (to prevent oxidation) requires additional time to allow a proper solder bond. This is normal.

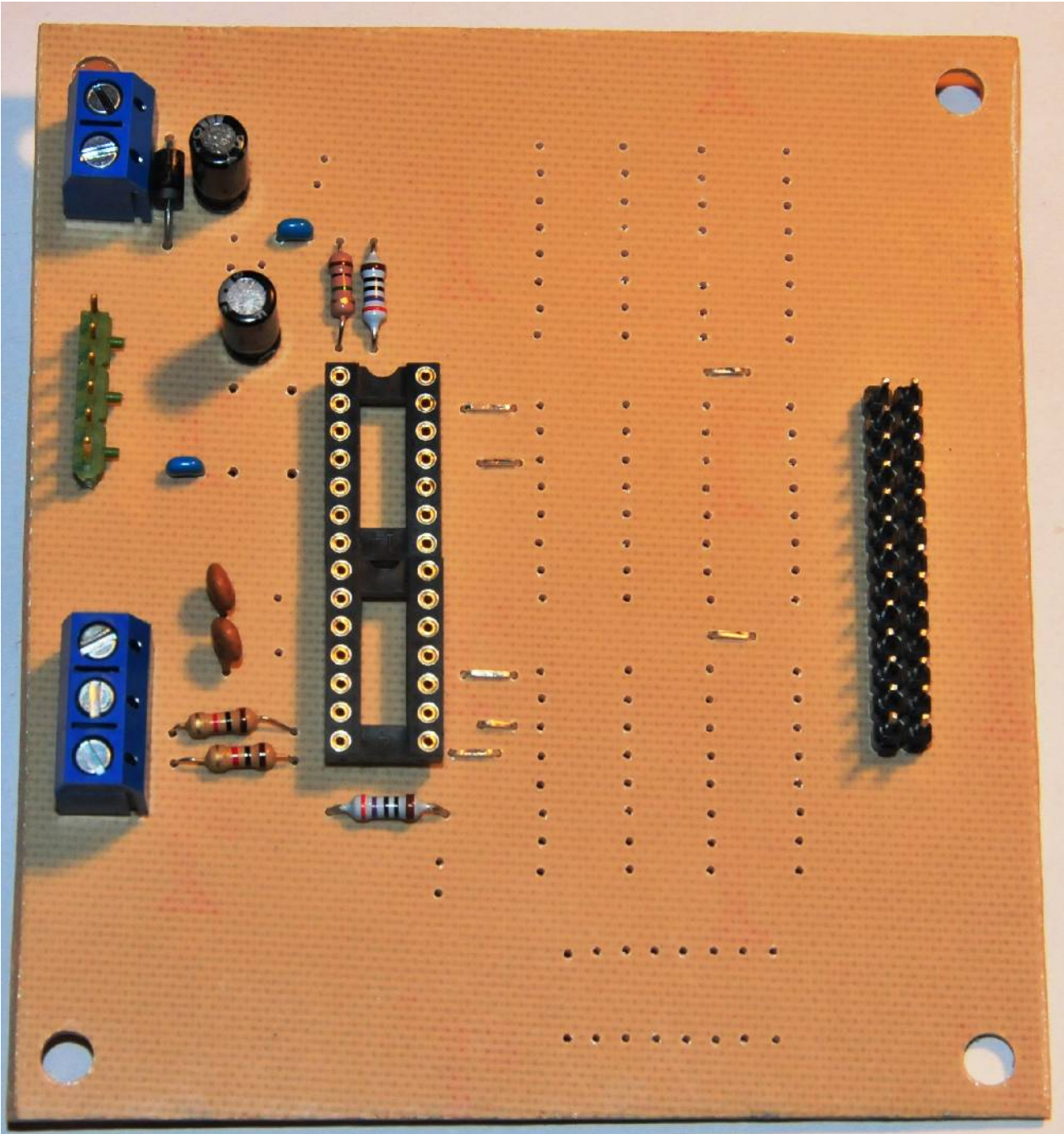
The below picture shows the location of the jumper wires. I use offcuts of component leads which have been bent using a pair of needle nose pliers for jumpers. There is no need to use insulated wire. In some cases, two 14 pin machined IC sockets are provided in place of the 28 pin socket. Simply place them one on top of each other, paying careful attention to pin 1 (the square pin on the PCB).



Continue assembly by inserting the 0.1 inch connector pins, and the wire insertion blocks. Be careful to orient the wire blocks (blue here) so that the wires can enter from the edge of the PCB.

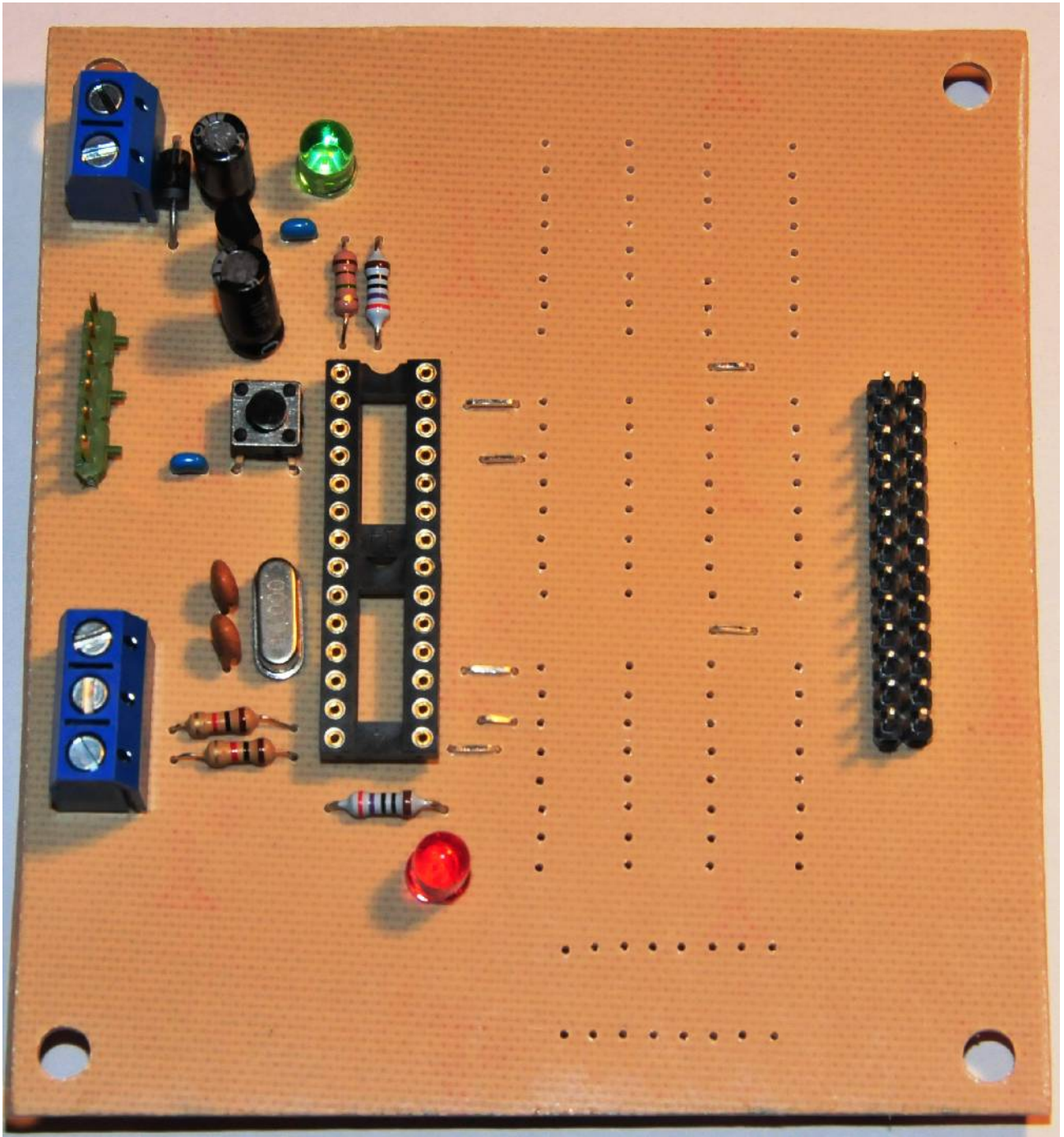


The next step is to insert and solder the resistors, capacitors and diode. Be careful that the correct orientation is used for the electrolytic capacitors and diode. The following diagram will help.



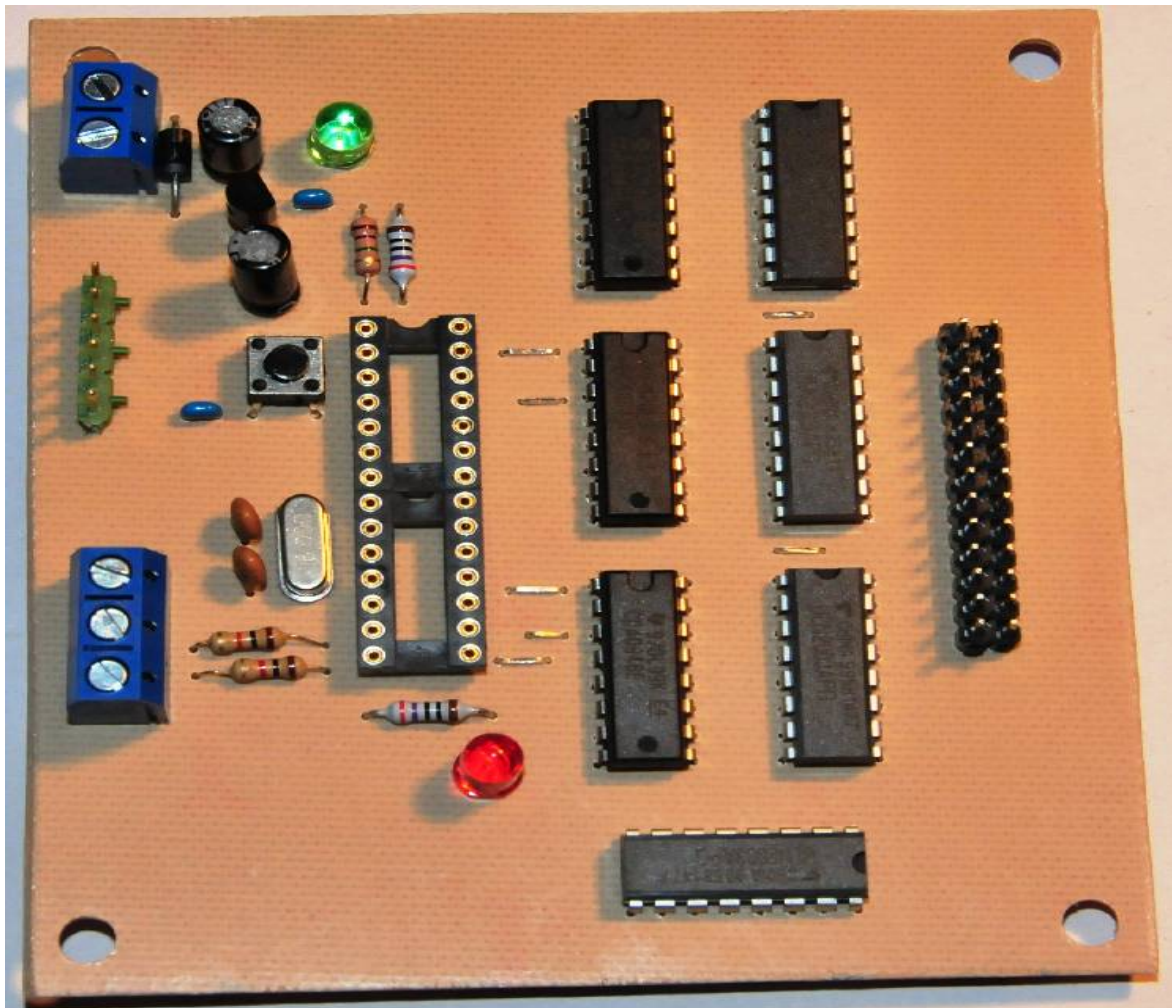


Continue assembly by installing the 78L05 Voltage regulator, switch, LEDs and crystal onto the PCB. The LEDs must be installed so that the short lead (cathode) is located at the bottom of the board (in the square hole)

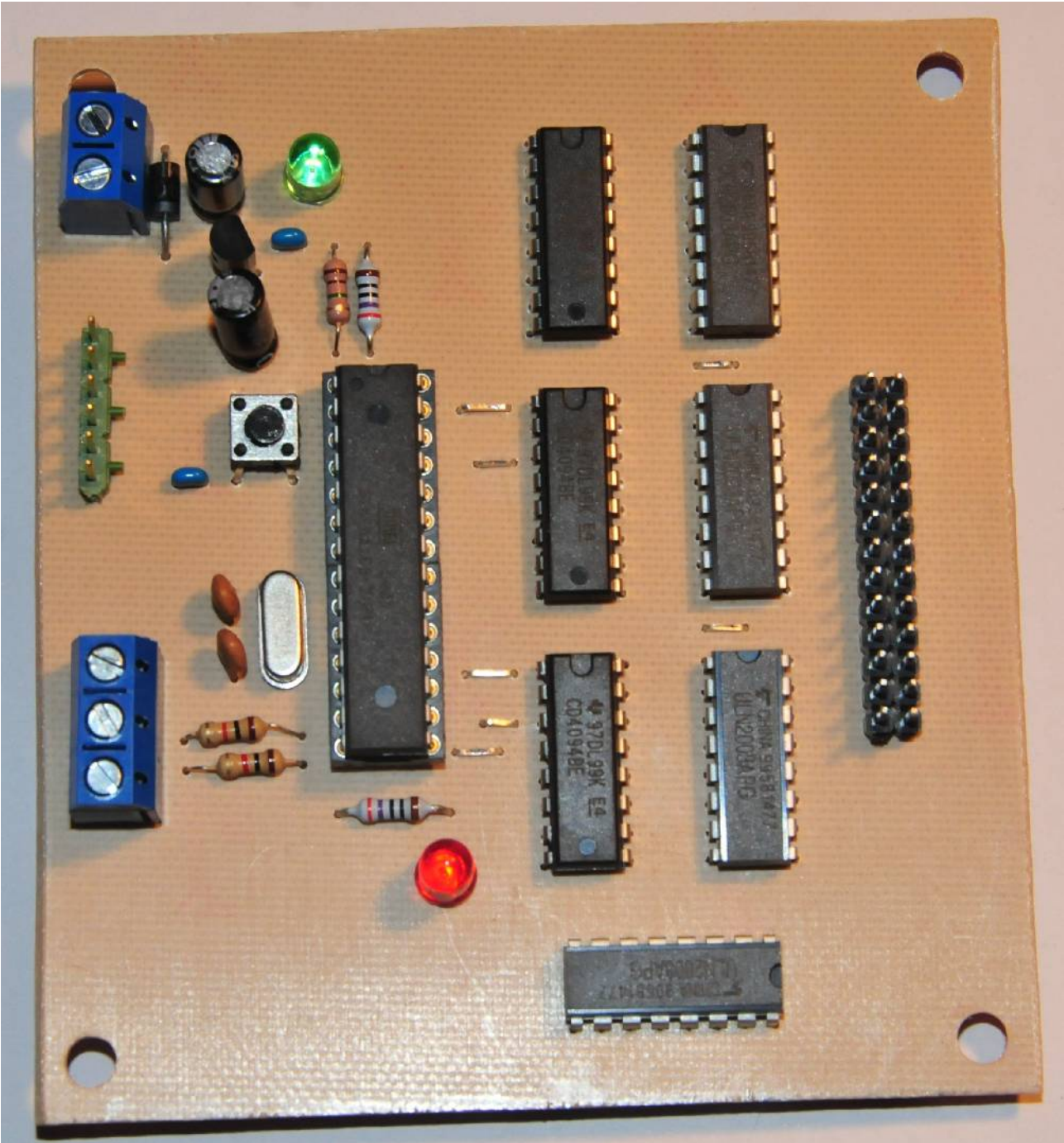


Next, insert the IC packages. Be very careful that the ICs are oriented so that pin one is in the correct location (matching the square hole on the PCB).

Depending on your preferences, you may like to install IC sockets for these components. They are not provided in the kit (as I don't normally use them for production parts that are not replaced regularly).



Complete assembly by inserting the microcontroller into the IC socket. Again, ensure that pin 1 is located at the top left of the board.



That completes assembly!

## Testing

The board is able to be tested by connecting a 9V DC supply to the power input (Top Left 2 pin wire header).

As soon as power is applied, the power LED should illuminate. If it does not, verify that the supply polarity is correct, and that the 78L05 has been installed correctly. There should be +9V on the input of the 78L05, and +5V on the output.

After the bootloader stops, and the self test is completed (within 15 seconds), the Pulse LED should pulse at a rate of one pulse per second.

You can now remove power, and connect your completed LED display PCB to the controller using the 26 way ribbon cable, and your clock should start operating.

## Time set input

The 3 pin wire connector on the board is designed to be connected to two normally open push buttons for setting the time. The centre pin is common, and the outside pins are to be connected to each time set pushbutton. One time set button increments the time, and the other decrements the time.

## Programming

The Arduino compatible controller board is designed to be programmed using a FTDI USB-TTL programming cable (available from [evilmadscience.com](http://evilmadscience.com))

<http://evilmadscience.com/partsmenu/130-usbttl>

The board is fully arduino compatible, and is able to be programmed from the Arduino IDE.

Source code is available from the [instructables.com](http://instructables.com) project website.

Have fun, and I hope that you enjoy your clock.

Doug Jackson