

THE COMPUKIT[®] UK101

How to build your own micro UK101

(serial only - no keyboard/display)

...on-line since 1998 !

(You can also create a complete UK101 (with display etc.) on FPGA -
click [HERE](#))



by Grant Searle

For news and updates, follow me on Twitter:

Last update: 2nd January 2014

Please note that you are NOT allowed to reproduce any of this page elsewhere on the Web without my permission.

CONTENTS

- [Introduction](#)
- [Memory Map](#)
- [Circuit Diagram](#)
- [Construction](#)
- [Parts List](#)
- [Wiring diagrams](#)
- [ROM images](#)
- [Pictures of the finished UK101](#)
- [Using the micro UK101](#)
- [Loading and saving](#)
- [Picture of the original UK101](#)
- [Links to some of my other pages](#)

INTRODUCTION

The micro UK101 described here is effectively the stripped-down "core" of the original UK101, and is no way intended to be the full UK101 implementation. I have removed the display and the keyboard circuitry and, instead, used a terminal emulation program (eg HyperTerm for Windows) as the keyboard/display. This has resulted in a very simple circuit which is suitable for use embedded-control applications as well as a fully-functioning computer. I have tried to make as few functional changes as possible to allow old software to work, wherever possible. As a result the BASIC ROM is totally untouched.

Although not the very simplest that can be designed, it is still straightforward. It is slightly more complicated due to the following:

1. The CPU is to run at 1MHz to match the original UK101. As a result, suitable clock division was needed to allow suitable serial speeds to be produced.
2. The ROM enabling in the original UK101 is not contiguous (in one block). This circuit is designed to use the original software on the UK101 (patched as needed) so the ROM needed to be enabled at two separate places in the memory map. The ACIA and RAM also needed suitable address decoding.

Some [patches](#) have been applied to the CEGMON ROM to ensure ALL I/O is via the serial port only, not to the screen. The differences are illustrated in the table below:

Original UK101	The MicroUK101
3K RAM, expandable on-board to 8K	32K RAM
300 baud serial/cassette interface	19200 baud serial interface. No cassette port needed as the PC can store the prorgams.
16x32 display	All display via the terminal on the PC
TV output and serial I/O	Only serial I/O
1MHz 6502 CPU	1MHz (although up to 8MHz CPU clock speed is possible)
Keyboard input (32 keys soldered directly onto the board)	Input via serial connection
12 inch x 15 inch board	3 inch x 3 inch board
1.5A supply current	200mA supply current, probably less if CMOS CPU used.
Open-board construction (uncased)	Up to you!
Microsoft BASIC	Microsoft BASIC
Various monitors were available; CEGMON being a popular choice	CEGMON monitor. Patched to ensure all I/O is via the serial port only.

This is an ideal project for anyone wanting to build their first computer as there aren't many components and it can be built using wire-wrap or on a breadboard (as I originally did with this circuit) in a couple of evenings. You will, however, need to blow your own EPROM image. A logic probe or an oscilloscope will probably be a distinct advantage in case it doesn't work.

[Back to contents](#)

MEMORY MAP

The aim of this design is to match the ORIGINAL UK101 address space, so resulted in slightly more complicated decoding logic than would be needed for a contiguous ROM space.

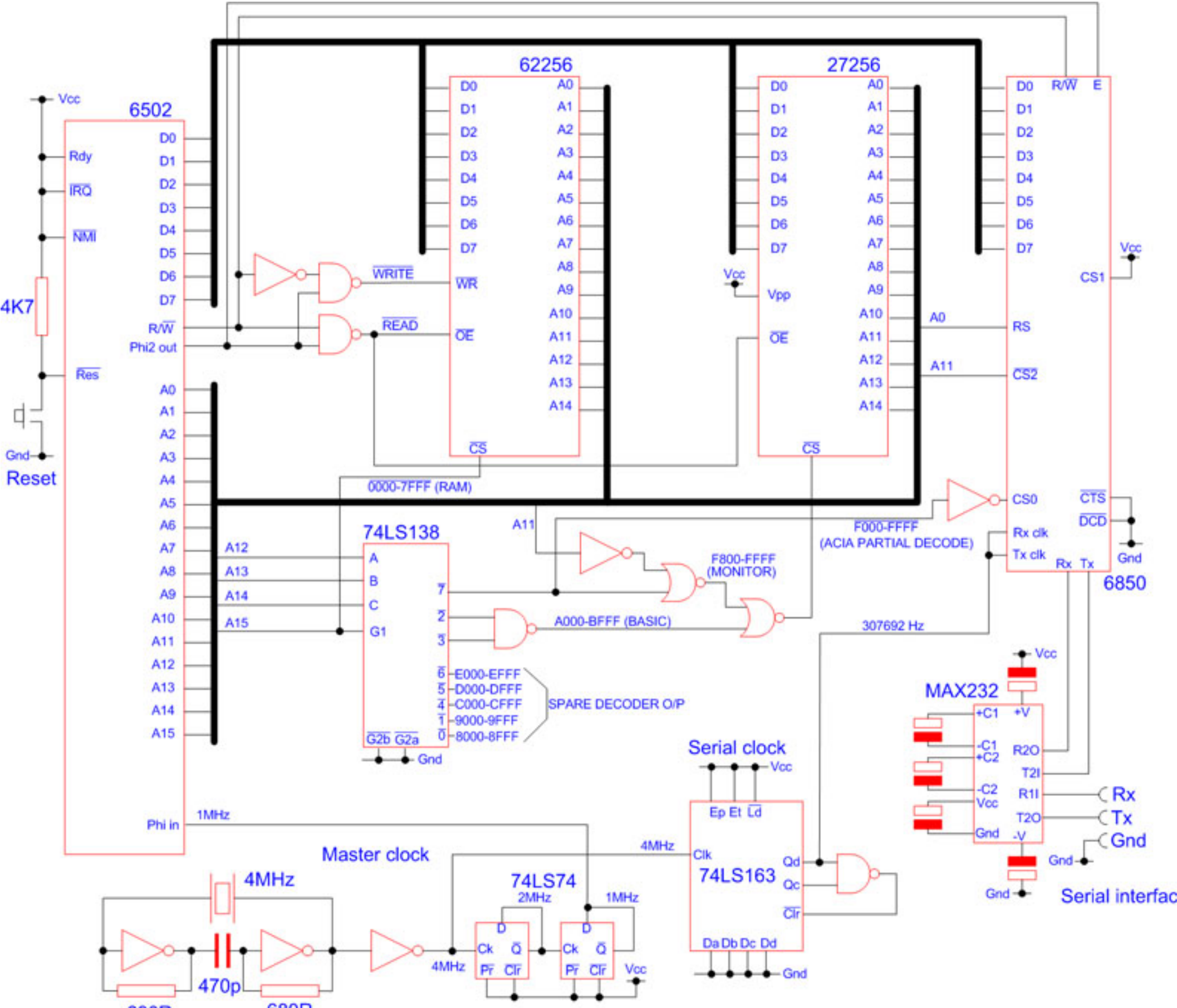
- 0000-7FFF - RAM
- 8000-9FFF - Free space
- A000-BFFF - BASIC ROM
- C000-EFFF - Free space
- F000-F7FF - ACIA
- F800-FFFF - MONITOR ROM

[Back to contents](#)

CIRCUIT DIAGRAM

Here is the complete circuit of the micro UK101.

The circuit is slightly more complicated than a fresh design due to the clock requirements for the CPU and serial interface, and also the memory map that was needed for the RAM, ROM and ACIA (see [above](#)).



Some notes on the circuit...

The CPU clock is 1MHz to match the original UK101. The serial clock then needed to use this.

The 19200 baud interface clock is derived as follows.

The 74LS74 divides the 4MHz clock by 2 then the 2nd stage divides by 2 again for the 1MHz CPU clock.

The 74LS163 runs from 0000 to 1100 (13 cycles) then is reset to 0000 on next clock (synchronous reset), because Qc and Qd is set to one, so divides the 4MHz clock by 13.

So, $4000000/13=307692$ (approx)

The 6850 is set for div 16, so the final baud rate = $307692/16=19231$ (approx) which is close enough to the required 19200.

Different crystal frequencies *can* be used, and if a suitable one is chosen that also produces the appropriate serial clock with simpler division then the circuit would be simplified.

Clock doubling: I have tried with success, doubling the clock frequency of the 6502. To do this, Phi in would connect to the 2MHz connection on the 74LS74 instead of the 1MHz connection.

[Back to contents](#)

CONSTRUCTION

Construction could be done in many ways, including the following:

- Solderless breadboard
I originally built this on 4 pieces of 30-strip breadboard
- Wire-wrap
This is an ideal project for beginning wire-wrapping
- Point-to-point
This is what I actually did for my final design - see [below](#) for details.
- PCB
I didn't design a PCB but is possible.

Assuming you are going to follow my layout, please see the [wiring diagrams](#) section for full details of the component layout and the point-to-point wiring needed. My version was built on a 3 inch square piece of tri-pad Veroboard (ie. the copper tracks on the reverse have breaks regularly resulting in groups of three holes being connected together. See the [reverse view](#) for further details). I used wire-wrap wire which requires soldering but, instead of wire wrapping, I used it for point-to-point wiring instead. All components are soldered in first then the wire laid across the soldered joint and heated with the soldering iron until the insulation melts completing the joint. Using this wire for point-to-point wiring is very quick and the whole board was soldered-up this way in a few hours.

Begin construction by positioning the IC sockets (see my pictures below for guidance). Next I recommend wiring the data and address. As each wire is soldered it, mark it off on the wiring diagram. This helps avoid any errors resulting from missing connections.

Wire the glue-logic wiring next, again marking each connection on the wiring diagram as you proceed.

Finally, wire up the power supply connections to all of the chips. For clarity, these are not shown on the wiring diagrams.

Once all components are soldered and without any ICs inserted into the sockets, connect a continuity tester between the power supply pins. If a short exists then check the underside of the board for any solder bridges. If all is well connect a power supply (current regulated to 500mA if possible). Check the +5V and 0V connections on each IC socket. Turn off the supply and insert the ICs. Turn on the power supply. If the circuit is working then expect a power consumption of around 210mA. Connect to a computer running a terminal emulation program set at 19200 baud, 8 bit, no parity, 1 stop bit, no handshake. Press the **Reset** button. If you see the microUK101 prompt then well done!

[Back to contents](#)

PARTS LIST

This is a full list of discrete components required to build the UK101 excluding any hardware such as power supplies, nuts and bolts etc.

Component No	Type	Qty
IC1	6502	1
IC2	62256	1
IC3	27256	1
IC4	MAX232*	1
IC5	6850	1
IC6	74LS04	1
IC7	74LS163	1
IC8	74LS74	1
IC9	74LS138	1
IC10	74LS00	1
IC11	74LS02	1
X1	4MHz crystal	1

* A MAX202 could be used instead, in which case C1-C5 should be 0.1uF

Tri-pad Veroboard

```
WireWrap wire
```

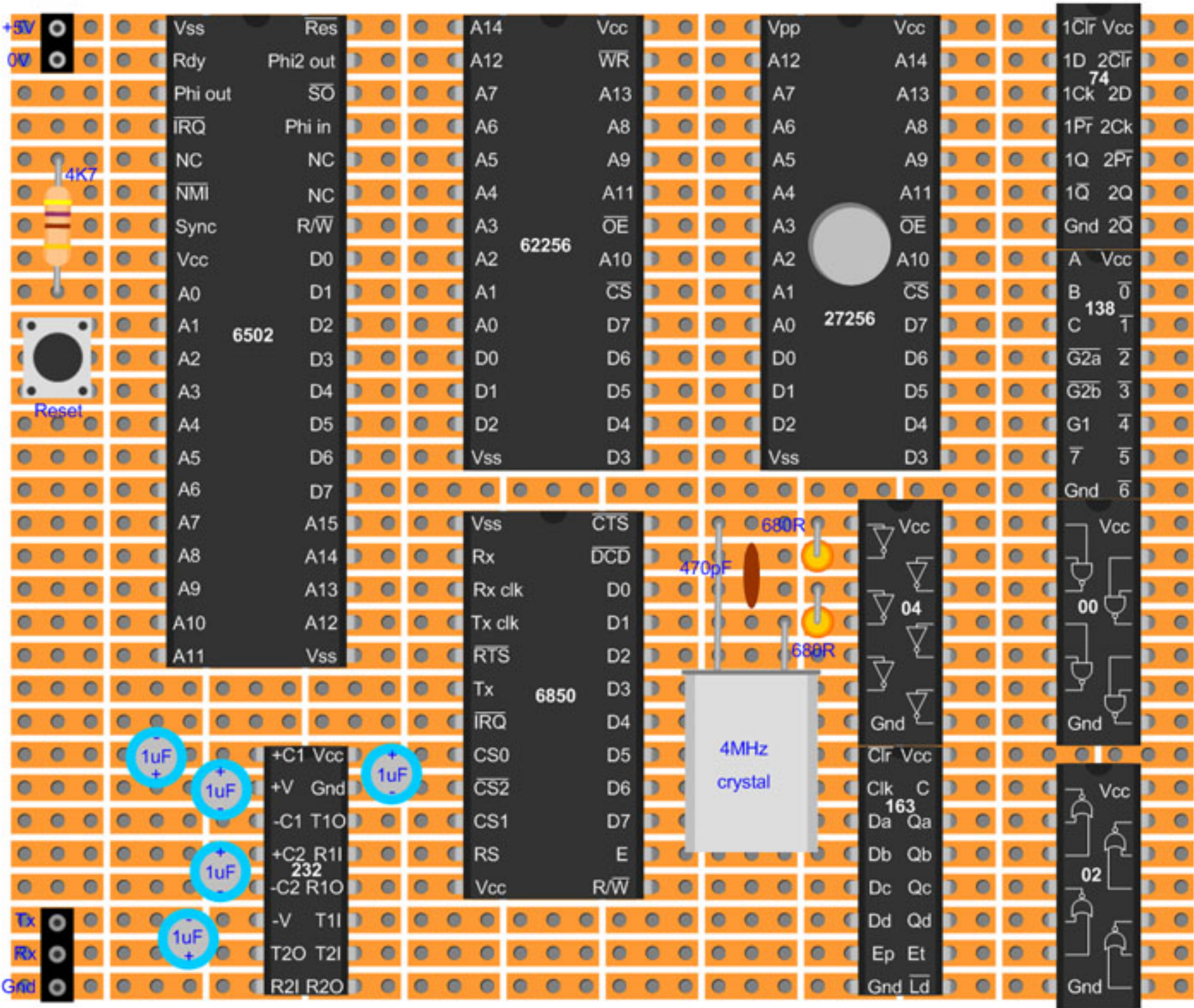
14 pin DIL socket	4
16 pin DIL socket	3
24 pin DIL socket	1
28 pin DIL socket	2
40 pin DIL socket	1
Connectors for power and serial I/O	

[Back to contents](#)

LAYOUT AND WIRING DIAGRAMS

Layout

Component layout of the prototype is as shown below. Please note the positioning of the components in relation to the tri-pad tracks.



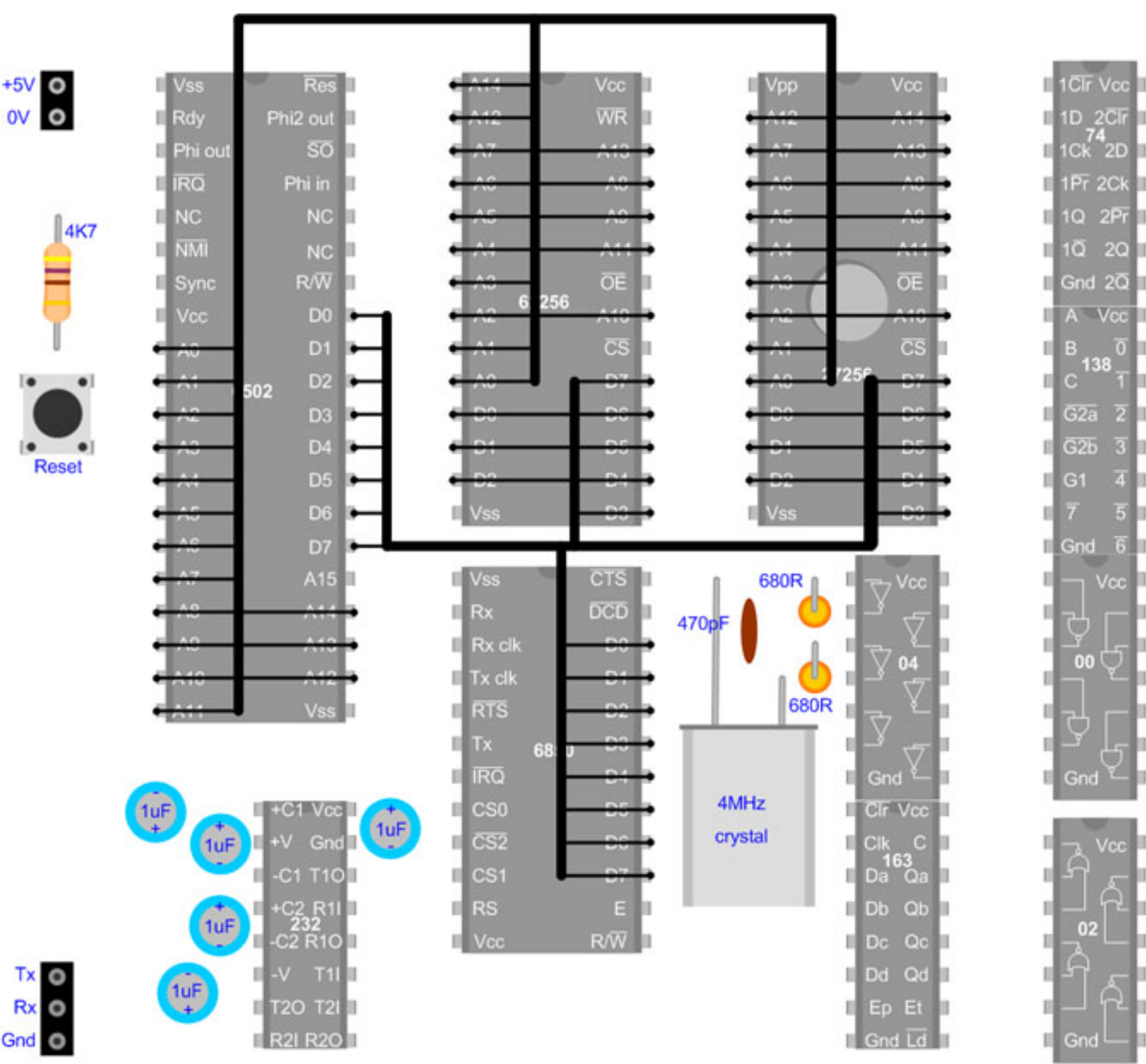
Wiring

The following diagrams show the data, address and signal connections. I have not shown the power connections, for clarity, but ensure that the appropriate pins of every IC are connected to Vcc (+5V) and Gnd

(0V) as needed.

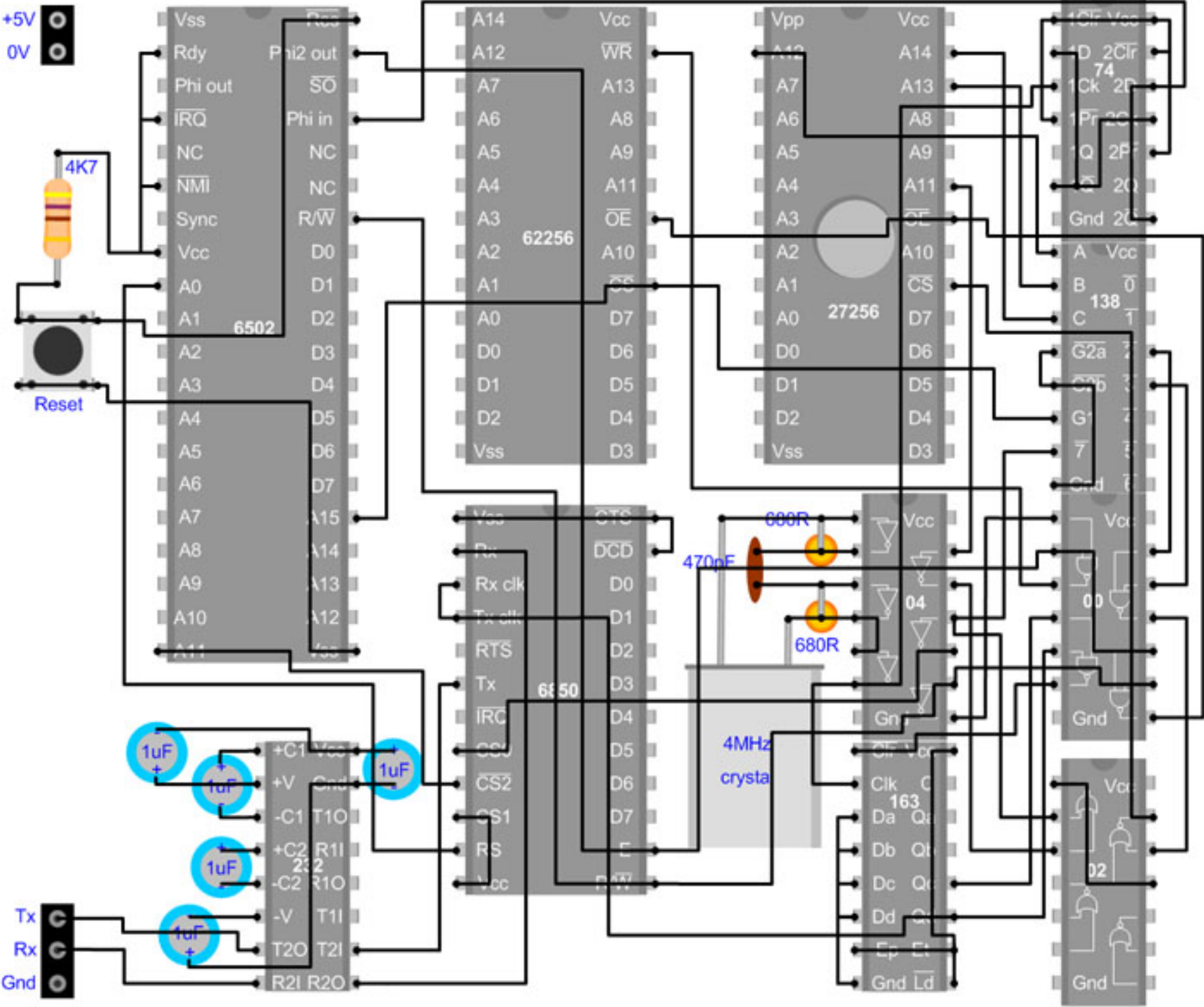
Wiring required for the data and address buses

The A0...A14 and D0...D7 on each chip are connected to the corresponding pin on other chips as identified below...



Wiring of the decoding, clock and serial interface

The remaining connections are shown here.



Power connections

Ensure all Vcc and Vpp pins are connected to +5V and all Vss/Gnd pins are connected to 0V. Additionally, I would recommend adding a couple of 0.1uF capacitors at various points between the two power rails. I have found that this decoupling has not been necessary, but it is recommended.

[Back to contents](#)

ROM IMAGES

The full ROM here is 32K long. Only part of it is actually used and all gaps are filed with FF which means that most EPROM programmers will skip these unused parts very quickly.

[Click here to download](#) (zip file)

This ROM consists of BASIC occupying the first 8K and CEGMON occupying the last 2K.

The original BASIC is untouched - it is identical to the original found in my UK101.

The original CEGMON ROM has been patched to allow it to work without needing a dedicated keyboard/screen. Details of the patches applied are shown [here](#).

[Back to contents](#)

PICTURES OF MY EARLIER PROTOTYPE VERSION

[Full component side view](#)

HOW TO USE IT

A scan of the original CEGMON manual is available [here](#). Please note that the commands relating to text positioning or windows are not applicable as this version is a pure serial interface.

BASIC usage

Connect the microUK101 to a terminal running at 19200 baud, 8 bit, no parity, no handshake.
Ensure CAPS-LOCK is ON on the terminal. Press RESET on the board and the following message will appear:

Micro UK101 C/W/M?

Press "C". The following message will appear:

MEMORY SIZE?

Press ENTER to allow the full memory to be used by BASIC. After a few seconds the following message will appear:

TERMINAL WIDTH?

Press ENTER. The following messages will be displayed:

31999 BYTES FREE

C O M P U K I T U K 1 0 1

Personal Computer

**8K Basic Copyright1979
OK**

The computer is now ready for programming in BASIC!

[Back to contents](#)

LOADING AND SAVING

Save

Type LIST but don't press enter.
In your terminal window on the PC, or whatever you are using to operate this board, enable the capture to the text file.
Press ENTER.
The listing will then be spooled to the text file on your PC.
Once the listing is complete, end the capture. The contents of the program will then be on the PC for later use.

Load

Most terminal programs have a method to pass a text file to the destination.
Ensure the microUK101 is ready to receive characters and then transfer the text file from the PC to the microUK101 using the terminal.
When transferring, you MUST have a delay after each new line, otherwise the BASIC interpreter will not be ready to accept the next line, so characters will be lost. You may possibly need a small delay after each character sent as well. This can be done in "HyperTerm" (for Windows) and may be available in other terminal emulators.

[Back to contents](#)

THE ORIGINAL UK101

[Picture of my UK101](#)

LINKS TO SOME OF MY OTHER PAGES

- [Build your own ZX80](#) - my page showing you how to build this old micro
- | [ZX80 to ZX81 conversion](#) - build the NMI generator needed to convert the ZX80 circuit into a ZX81
- | [ZX80 software](#) - Type in a Space Invaders game into the ZX80

- [Build your own Jupiter Ace](#) - my page showing you how to build this old micro
- [Build your own UK101](#) - my page showing you how to build a greatly simplified version of this old micro
- [Pong](#) - Pictures of my build of the Atari classic arcade game
- [My Machines](#) - My collection of classic 80's micros

I hope this page has been useful.

Grant.

To contact me, my current eMail address can be found [here](#). Please note that this address may change to avoid spam.

Note: All information shown here is supplied "as is" with no warranty whatsoever, however, please let me know if there are any errors. All copyrights recognised.