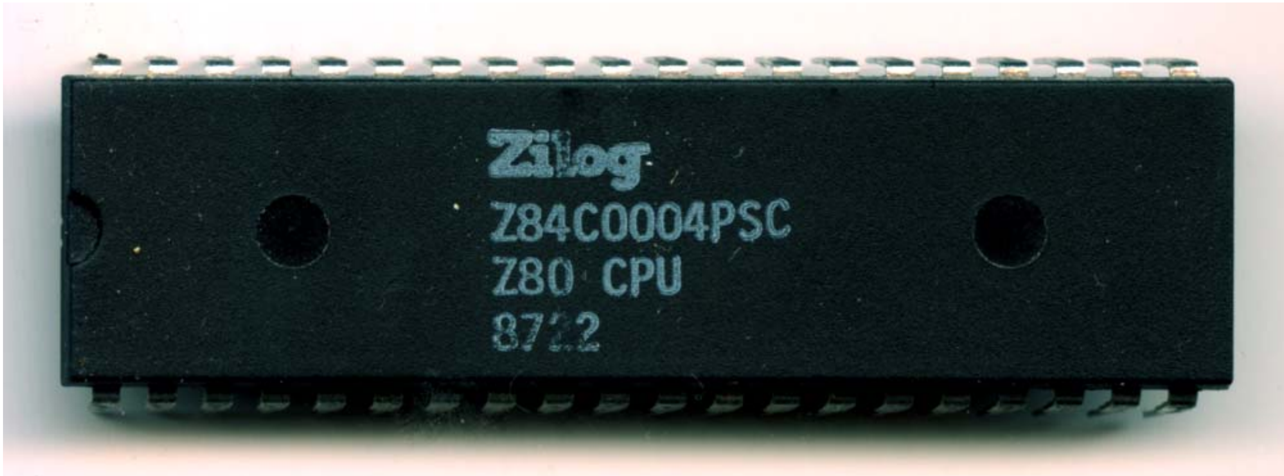
# Z80 Board

Designing a Z80 Based Microcomputer




How to design a minimal high performance Z80 based microcomputer with high quality graphics.

Using less than 15 ICs !

# Part 1

**Preparation.**
**Board Layout and components selection:**

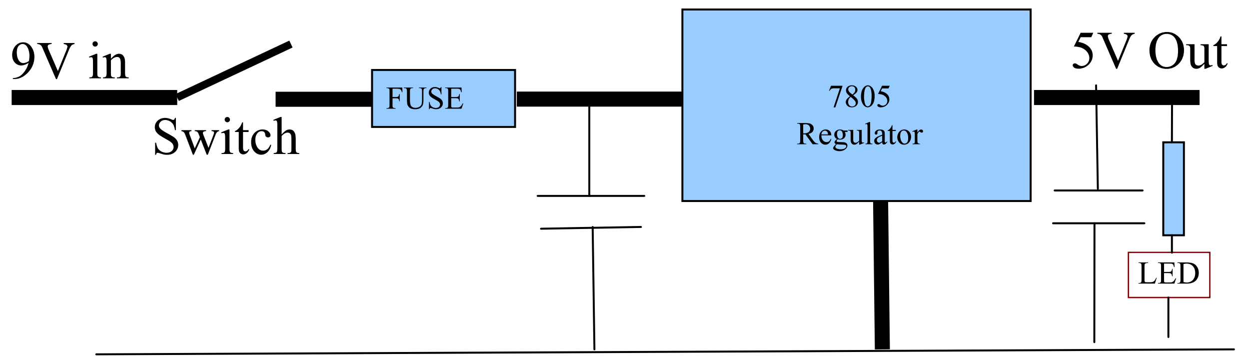
I've decided to use a board layout of 220 x 160 mm to give plenty of space for each component. I've also decided to use two EPM7128 CPLD. One as a memory decoder and one as an I/O decoder/interface.

You don't have to lay your board out the same way or use two EPM7128s. I'm using FreePCB to do the board layout and am not going to bother with a schematic as there's only a dozen ICs or so. But if you want to do a schematic I'd recommend using Eagle as it's cheap (Free) and easy to use.

So lets start:

Drag and drop your components from the parts library onto your virtual board. This should take you a quite a few minutes.

Now lets do the power supply. Connect the 9V DC input to the 7805, voltage regulator you could pop a fuse in series with this if you want as a safeguard, also popping in some smoothing capacitors between 0V and 9V input will help. I will also be putting in a LED to show that the board is ON or OFF.




Now we have 5 Volts and ground (0V) ! These need to go to every IC, Z80, EPM7128s, V9958, SRAM, EEPROM, FLASH, MAX232 , 41464 DRAMs. They'll also have to be connected to very connector, printer DB25 (or Centronics ), PS2 , Com2,Com2 DB9, Scart (TV) etc... This will take some time as you'll have to look at every connectors datasheet/pinouts to see which pins are ground and VCC 5v ! This is a time consuming part of the build. Use your logbook to make notes and jot things down. I tend to cut-out the pin-outs and glue them into my logbook to save drawing them! A bit like Blue Peter. It's very important to note things down and keep an accurate log otherwise you'll get in a muddle later !

We can also connect a RESET button placed at the front of the board, this is a SPST momentary switch. It simply momentarily pulls the CPU Z80 Bus RESET to zero . The exact wiring for this can be found in the Z80 Data-sheet with pull-up resistor values etc.

Now you know why you printed out all those datasheet IC pinout diagrams !! By now ( several hours later) your ready for a cup of tea. Save all your work and we'll resume next week !!

# Part 2
# Connector Z80 Bus , & I/O Bus.

Now lets connect the Z80 Bus to the EPM7128 (lower one) this will be used to decode RAM/EEROM/FLASH and could extra logic for producing Video (like on the ZX Spectrum or Amstrad CPC) . This video could theoretically be Fed into the V9958s external video in socket !

The Z80 BUS consists of address line A0..A15, Data lines D0..D7, and control signals, Halt, Reset, R/W , MREQ,IOREQ etc 36 lines in total 40- Crystal,Vcc,Gnd.

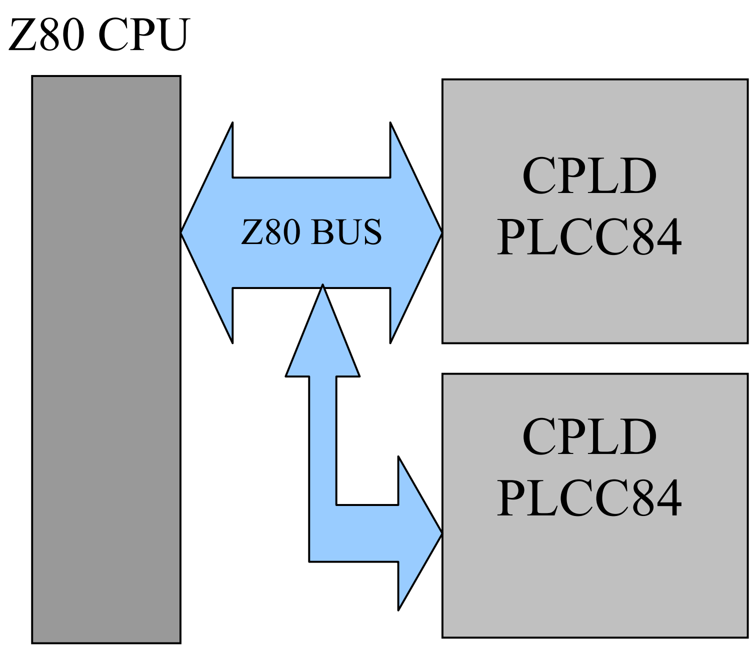
Make a note of which I/O lines on the EPM7128 you connected these too!! As you'll need this information later.

The Z80 I/O Bus consists of D0..D7, A0..A7, R/W, I/OREQ this will go into the upper EPM7128 CPLD , again noting which pins go to which CPLD I/O lines!

So now we should have the Z80 BUS connected to one CPLD EPM7128 and the I/O BUS connected to the other CPLD.

This all sounds quite easy, but you should of discovered that's not. It'll of taken you several hours to do this properly making notes as you go in your logbook.

Now we can connect the 20MHz Crystal to the Z84C0020 CPU.. this is straight from the datasheet, one crystal plus two capacitors!. Then the 21.4772 Crystal on the V9958 in much the same way !




Now you can go and have a cup of tea …....you've earned it !

* Don't forget to save your work *

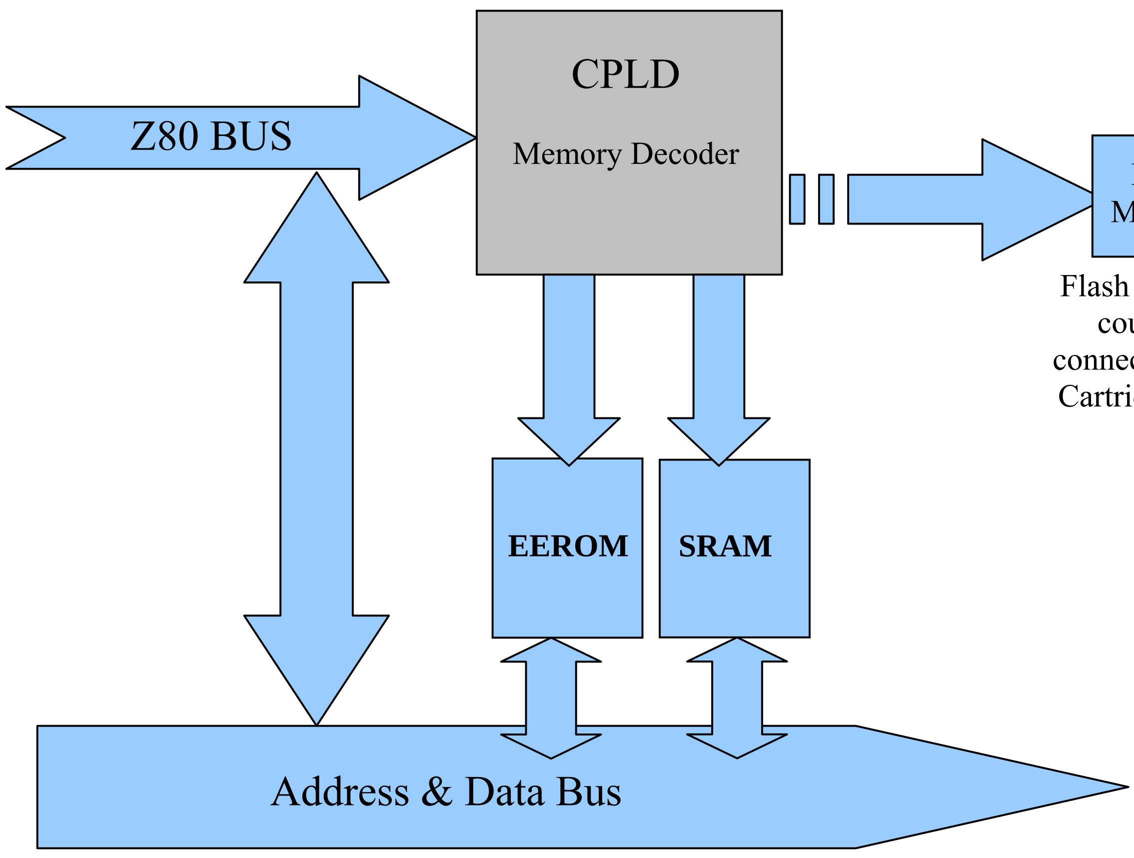
# Part 3
# Connecting SRAM, EEROM, & Flash

Now lets connect the SRAM, obviously data-lines D0..D7 go to the corresponding lines on the Z80 bus, as do address lines A0..A13 giving 16K bytes blocks , so RAM is put into 16k blocks. The control lines R/W go the Z80 Bus too. CS chip-select goes to the CPLD as this determines when the chip is ON or OFF and address lines A14,A15,A14 go to the CPLD as this selects which block is pages in.

EEPROM is connect in a very similar way to SRAM, D0..D7 to Z80 bus, A0..A13 , plus R/W to Z80 BUS again giving 16K blocks. However only A14 & A15 are connected to the CPLD , chip-select goes to the CPLD.

Flash memory can be missed off if you want, if connected then it's connected just like the SRAM again giving 16K blocks which can be switched in or out.. It could also be connected via a DIL/SIL connector like a ROM cartridge allowing it to be removed. So you could have lots of cartridges to plug in each 512K bytes big.... just like popping in SD cards !!

For exact details it's best to look at each ICs pinout from the datasheets !



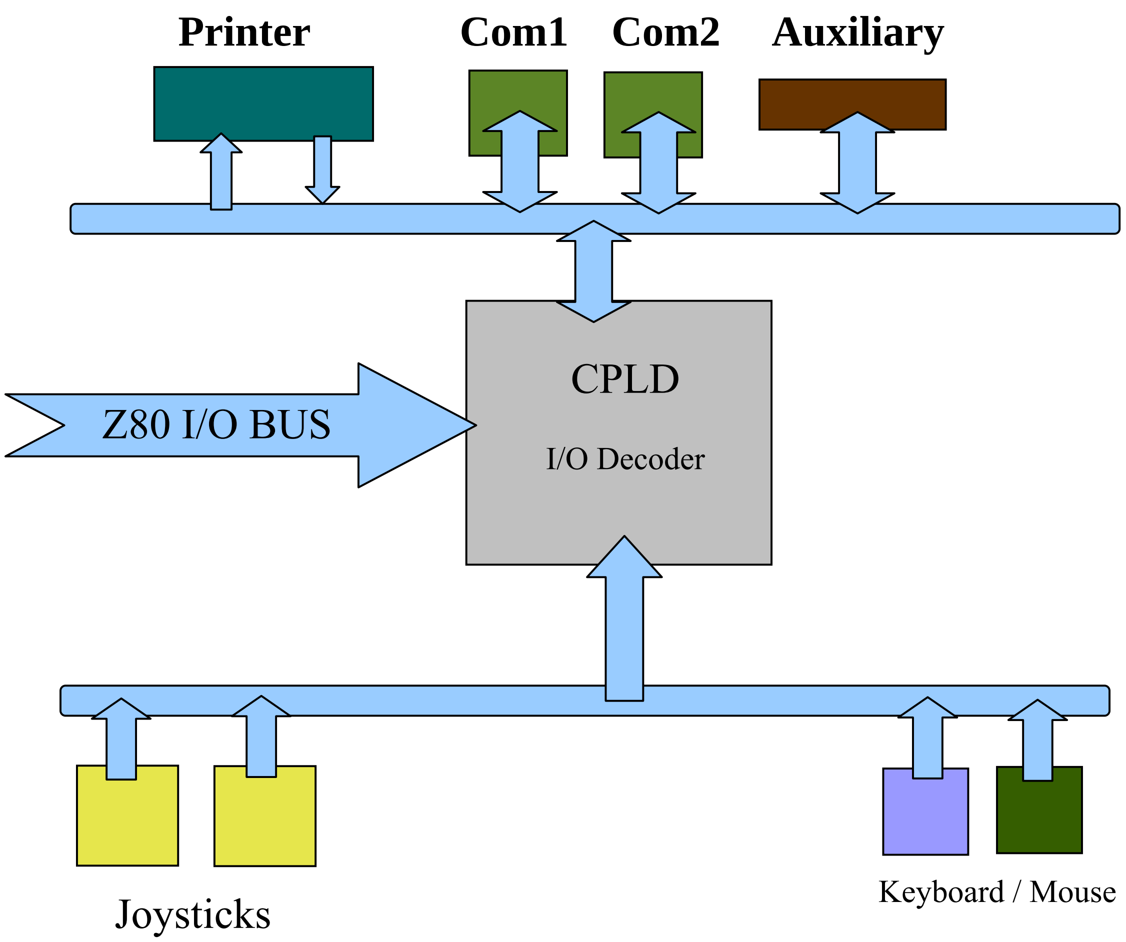
# Part4
## Connecting Printer,Expansion port,Keyboard & Joysticks

Now lets connect the printer, Connect it to spare I/O lines on the CPLD for I/O . I believe 13 I/O lines are needed these include D0...D7 plus control signals for handshaking and stuff.. 5V and Gnd should already be connected. You can either use a DB25 and configure it like a PC LPT port or a Centronics ports as in the original Einstein. The pinouts for an LPT1 port can be found on Wikipedia and a Centronics from the Einstein manual. Remember to make notes of what goes where in your logbook!

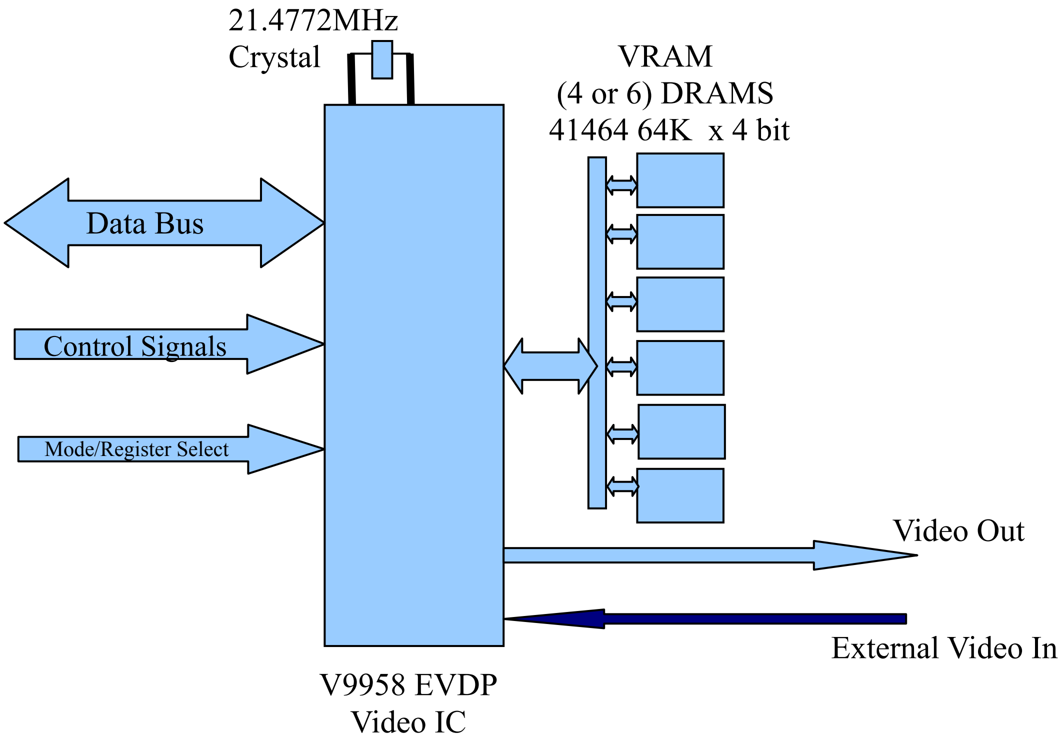
The Expansion port is normally just the Z80 BUS, but sometimes it is the Z80 BUS plus extra control signals.. it's up to you how you wire it. I wouldn't worry about the original Einstein's connectors and expansion ports.. as most of the stuff it was connected to is completely obsolete! Think of how it can be connected to modern equipment not old !

The PS/2 keyboard and mouse.. they have DATA and CLK that need to be connected to available I/O lines. It might be worth just connecting these to the nearest CPLD for convenience. Again making notes in your logbook!

The two joystick ports can be connected like Atari ones, I believe that gives 5 data-lines each , so giving 10 I/O lines to the CPLD. Of course you could miss them off. Or if your short of I/O lines use another CPLD like a ATF1502 or micro-controller to connect them.



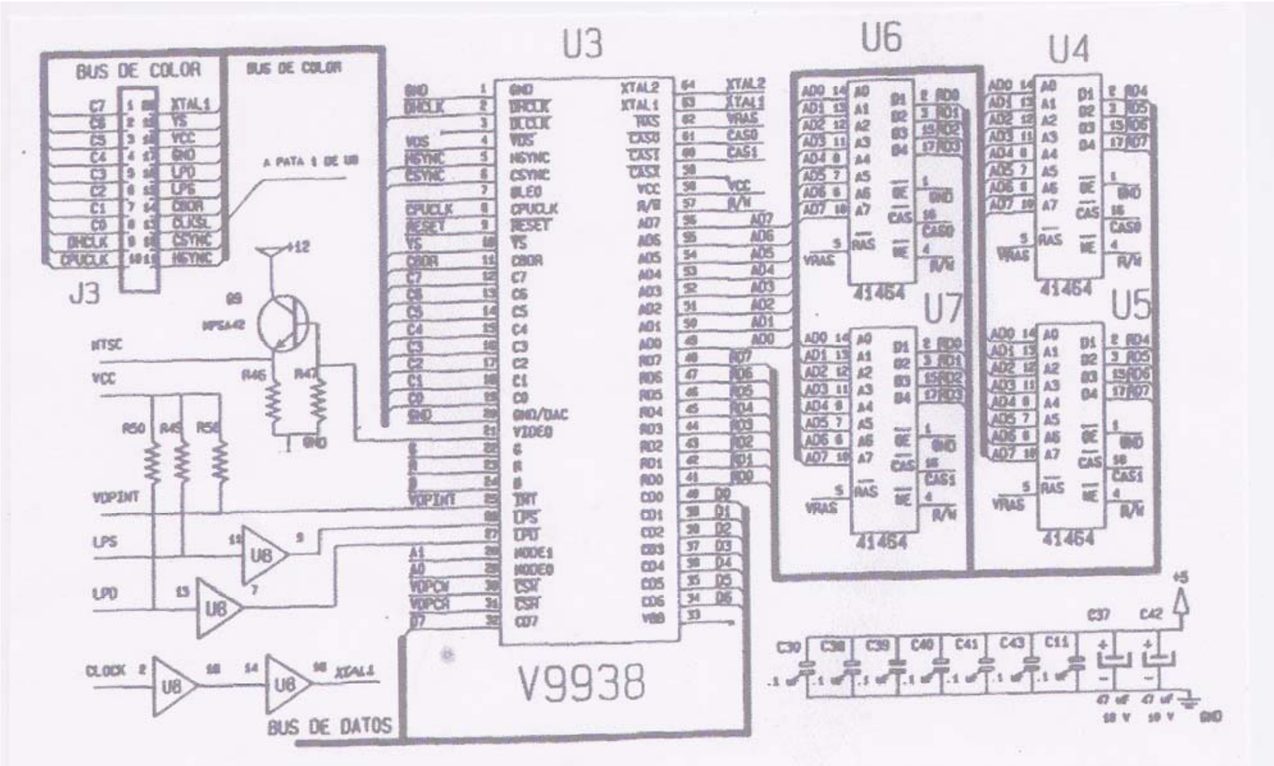
# Part 5
# Video V9958 Circuit.

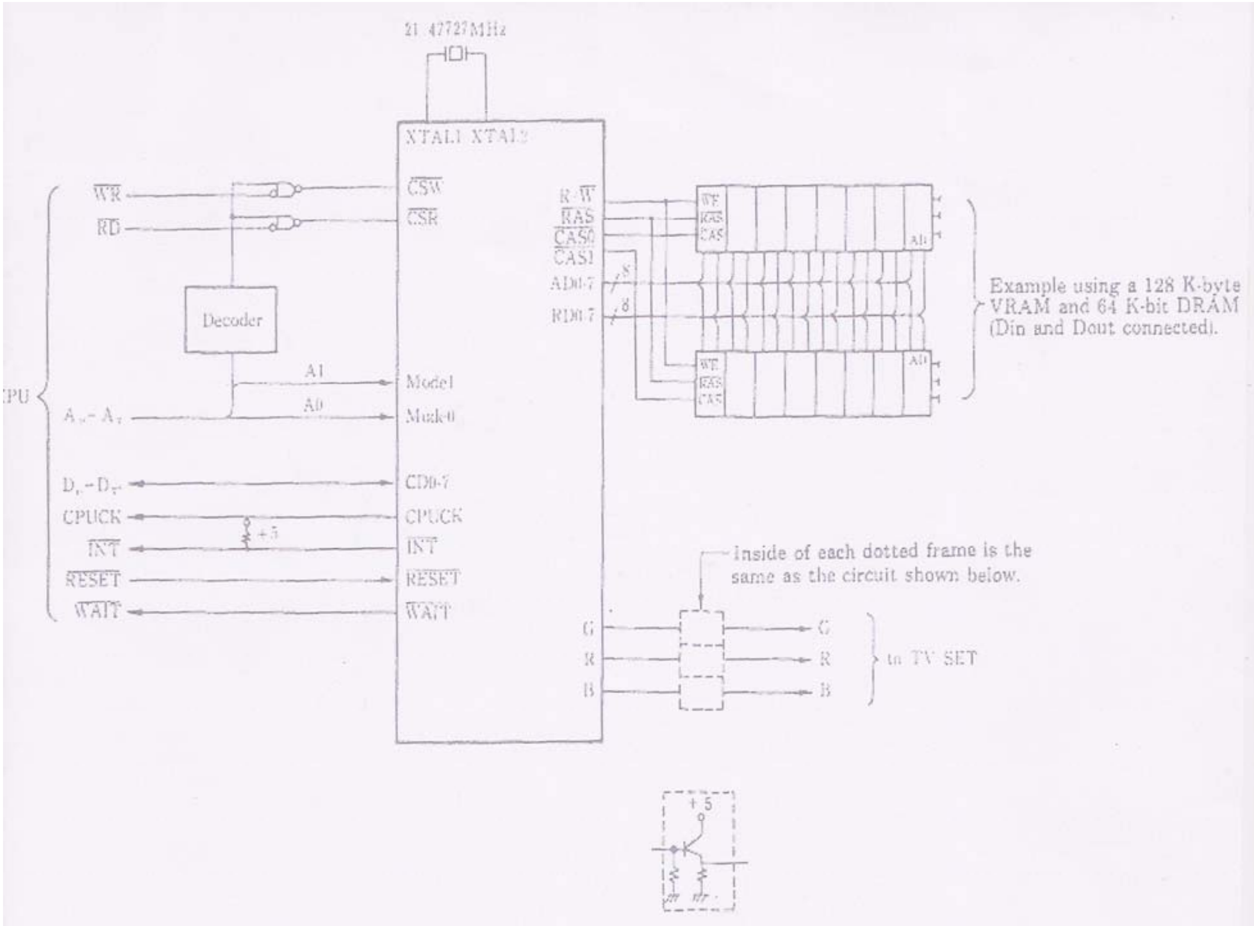



## Block diagram of Video Circuitry

This is probably the most complicated and time consuming part of the circuit! It's best to use an existing circuit from the Einstein256 schematics or an MSX Turbo R schematic. You can use 128K (4 x 41464 DRAMs) or 192K (6 x 41464 DRAMS) obviously using six is more complicated.

Use your datasheets and service manuals to help. The MSX turbo R manuals are online, the text is in Japanese, but this does not affect the schematics.






# <u>Part6</u>
# Sound Chip

## General Instruments - AY-3-8910, AY-3-8912, AY-3-8913
## Programmable Sound Generator

**FEATURES**

- Full Software Control of Sound Generation
- Interface to Most 8-bit and 16-bit Microprocessors
- Three independently Programmed Analogue Outputs
- Two 8-bit General Purpose I/O ports (AY-3-8910)
- One 8-bit General Purpose I/O port (AY-3-8912)
- Single +5 Volt Supply

**DESCRIPTION**

The AY-3-8910/8912/8913 Programmable Sound Generator (PSG) is a LSI Circuit which can produce a wide variety of complex sounds under software control. The AY-3-8910/8912/8913 is manufactured in the General Instrument N-Channel Ion Implant Process. Operation requires a single +5V power supply, a TTL compatible clock, and a microprocessor controller such as the General Instrument 16-bit CP1610 or one of the PIC1650 series of 8-bit microcomputers.

The PSG is easily interfaced to any bus orientation system, its flexibility makes it useful in applications such as music synthesis, sound effects generation, audible alarms, tone signalling and FSK modems. The analogue sound outputs can each provide 4bits of logarithmic digital to analogue conversion greatly enhancing the dynamic range of the sounds produced.

In order to perform sound effects while allowing the processor to continue its other tasks, the PSG can continue to produce sound after the initial commands have been given by the control processor. The fact that realistic sound production often involves more than one effect is satisfied by the three independently controllable channels available in the PSG .

All of the circuit control signals are digital in nature and intended to be provided directly by a microprocessor/microcomputer. This means that one PSG can produce the full range of required sounds with no change in external circuitry. Since the frequency response of the PSG ranges from sub-audible at its lowest frequency to post-audible at its highest frequency, there are few sounds which are beyond reproduction ith only the simplest electrical connections.

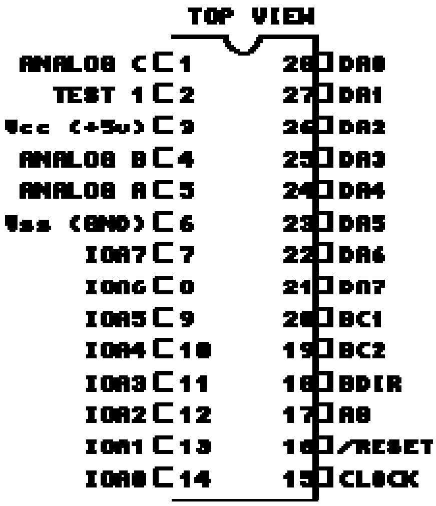
Since most applications of a microprocessor/PSG system would also require interfacing between the outside world and the microprocessor, this facility has been designed into the PSG. The AY-3-8910 has two general purpose 8-bit I/O ports and is supplied in a 40 lead package; the AY-3-8912 has one port and 28 leads; the AY-3-8913 has no ports and 24-leads.

# See AY-3-8912 Datasheet for wiring details.

### PIN CONFIGURATIONS

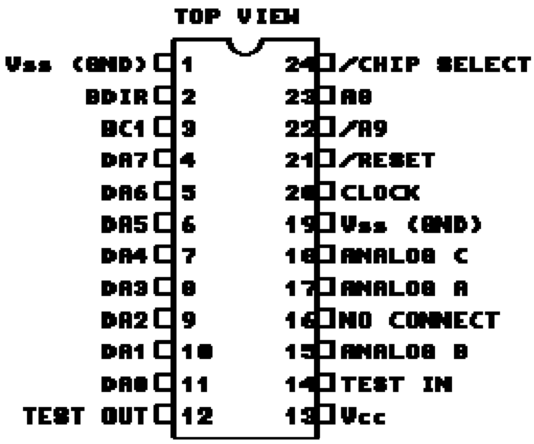
## 28 LEAD DUAL IN LINE

### AY-3-8912




## 24 LEAD DUAL IN LINE

### AY-3-8913




**PIN FUNCTIONS**

**DA7--DA0 (input/output/high impedance) pins 30--37 (AY-3-8910) pins 21--28 (AY-3-8912) pins 4--11 (AY-3-8913)**

**Data/Address 7--0:**

These 8 lines comprise the 8-bit bidirectional bus used by the microprocessor to send both data and addresses to the PSG and to receive data from the PSG. In the data mode, DA7--DA0 correspond to Register Array bits B7--B0. In the address mode, DA3--DA0 select the register number ($0\text{--}17_8$) and a DA7-DA4 in conjunction with address inputs /A9 and A8 for the high order address (chip select).

**A8 (input): pin 25 (AY-3-8910) pin 17 (AY-3-8912) pin 23 (AY-3-8913)**

**/A9 (input): pin 24 (AY-3-8910) pin 28 (AY-3-8912) (not provided on AY-3-8913)**

**/Address 9,Address 8**

These "extra" address bits are made available to enable the positioning of the PSG (assigning a 16 word memory space) in a total 1,024 word memory area rather than in a 256 word memory area as defined by address bits DA7--DA0 alone. If the memory size does not require the use of these extra address lines they may be left unconnected as each is provided with either an on-chip pull down (/A9) or pull-up (A8) resistor. In "noisy" environments, however, it is recommended that /A9 and A8 be tied to an external ground and +5V, respectively, if they are not to be used.

**/RESET (input): pin 23 (AY-3-8910) pin 21 (AY-3-8913) pin 16 (AY-3-8912)**

For initialization/power on purposes, applying a logic "0" (ground) to the /Reset pin will reset all registers to "0". The /Reset pin is provided with an on-chip pull-up resistor.

**CLOCK (signal): pin 22 (AY-3-8910) pin 20 (AY-3-8913) pin 15 (AY-3-8912)**

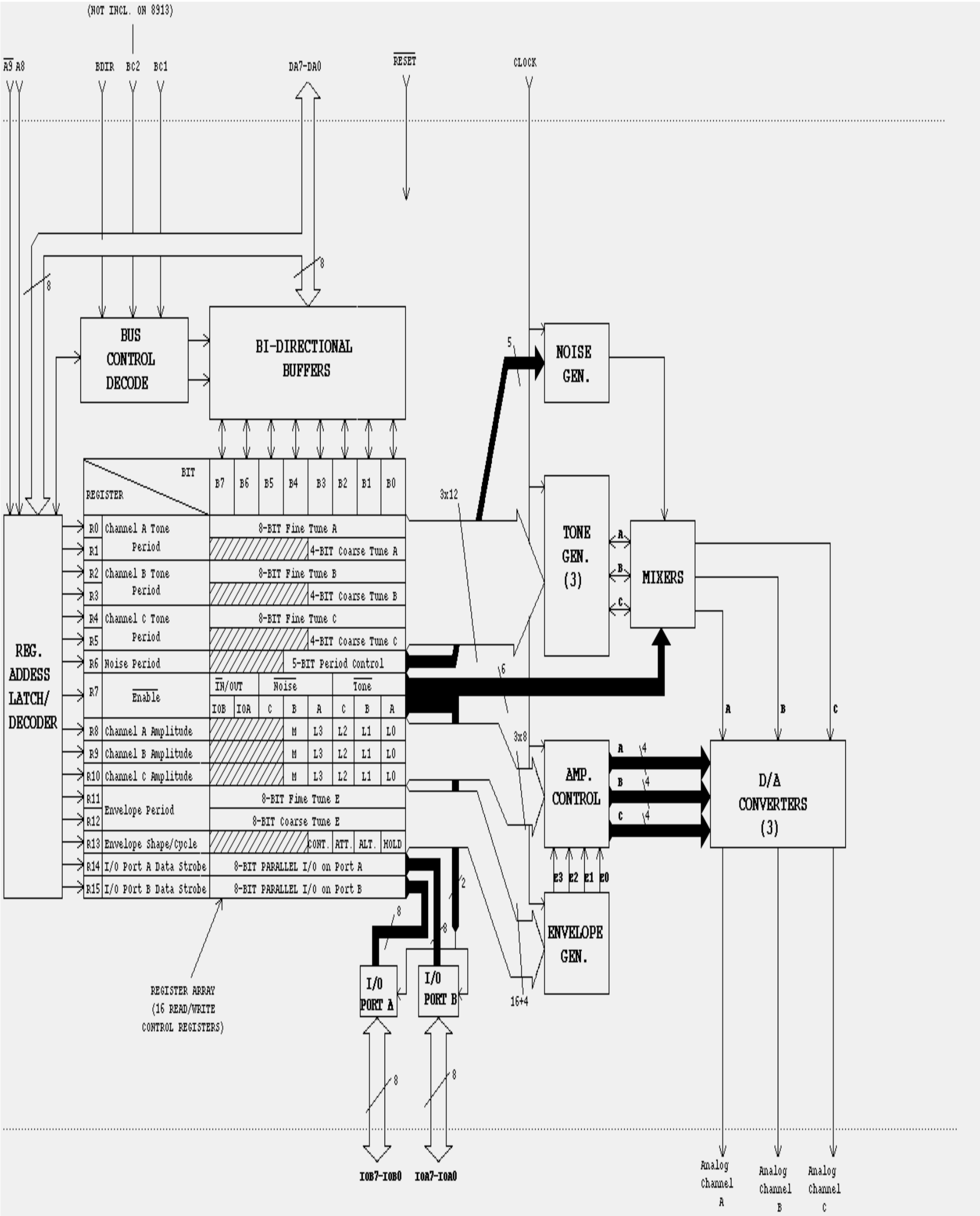
This TTL-compatible input supplies the timing reference for the Tone, Noise and Envelope Generators.

**BDIR,BC2,BC1 (inputs): pins 27,28,29 (AY-3-8910) pins 18,19,20 (AY-3-8912) pins 2,3 (No BC2 on AY-3-8913 see below)**

**I/O PORTS**

Two additional blocks are shown in the PSG Block Diagram which have nothing directly to do with the production of sound -- these are the two I/O ports (A and B). Since virtually all uses of microprocessor-based sound would require interfacing between the outside world and the processor, this facility has been included in the PSG. Data to/from the CPU bus may be read/written to either of two 8-bit I/O Ports without affecting any other function of the PSG. The I/O Ports are TTL-compatible and are provided with internal pull-ups on each pin. Both Ports are available on the AY-3-8910; only I/O Port A is available on the AY-3-8912; no ports are available on the AY-3-8913.

## PSG BLOCK DIAGRAM




PSG BLOCK DIAGRAM

# How to make your own PCBs

Printed Circuit Boards -- how to make nice looking ones from your own designs

There are a handful of ways available to the hobbyist to turn your own designs into PCBs. They yield results of different qualities. Homemade board tend to be of a much lower quality than professionally manufactured ones. Any process that involves making your own board will have a number of steps in common. At a high level, here's what you're doing:

1. Procure a bare copper coated board. These can be bare copper or photosensitive types.

2. There are loads of excellent videos on YouTube showing you how to design a 'homebrew' PCB. I recommend you spent a couple of hours researching this online. I prefer the laser jet method and a hot iron.  Where the design is printed off using a laser jet printer and then the ink from the paper is 'Ironed' onto the copper board.. Again there's You tube videos showing this very well !

3. There are various methods for using an expensive UV box to a simple water resistant maker pen and even transfers. If you have a plain board,

4. All homemade boards are prone to going wrong, lining up both sides on a double sided board can also pose problems and in general I'd only recommend doing it yourself if your experience and confident to do it or the board design is very simple!

5. Cost: the cost of buying all the materials, Ferric Chloride, gloves, goggles,  copper board etc can be quite expensive and sometimes it works out cheaper just to send your design off and get one made from a budget no frills PCB manufacturer.

# Spend a couple of hours on You Tube !!

Souring a PCB manufacturer to produce your board.:

UK : there are literally dozens of small companies in the UK that will make you a PCB, Just go online and see ! There are some simple rules to follow.

1. Send your design to at least three companies for a quotation. They will tell you whether they can do it and how much it will cost. I've seen quite a few offer no frills boards for £15-£20 ! Once you've source a company just send of the money and they'll send you your board in about a week. Then you can solder your components onto it !

2. China : Again there are lots og companies in China that will make a PCB, some will even design the board from a generalized specification, for a price! However it takes at least three weeks fro stuff to come from China so you've got to think of the time delay !

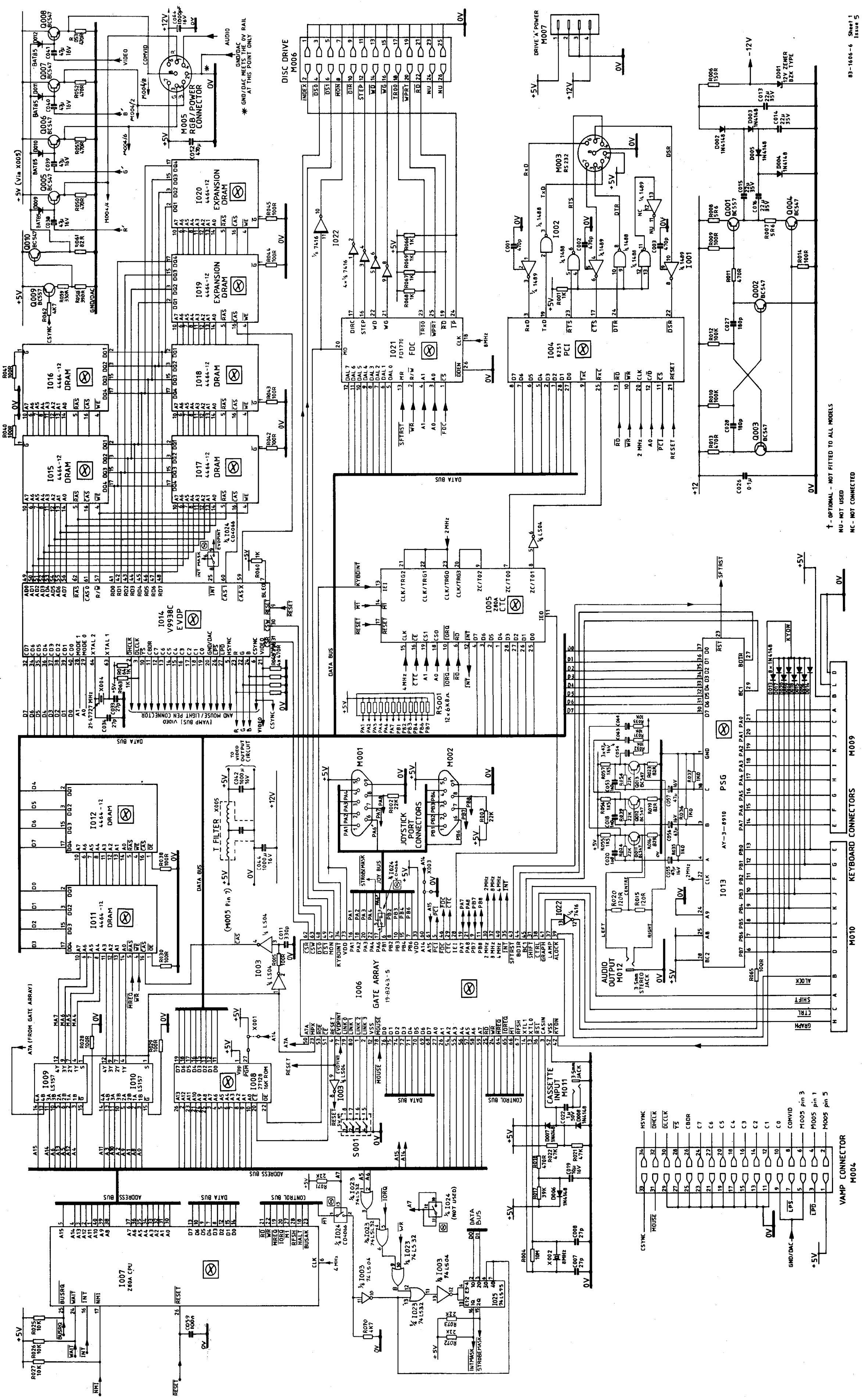
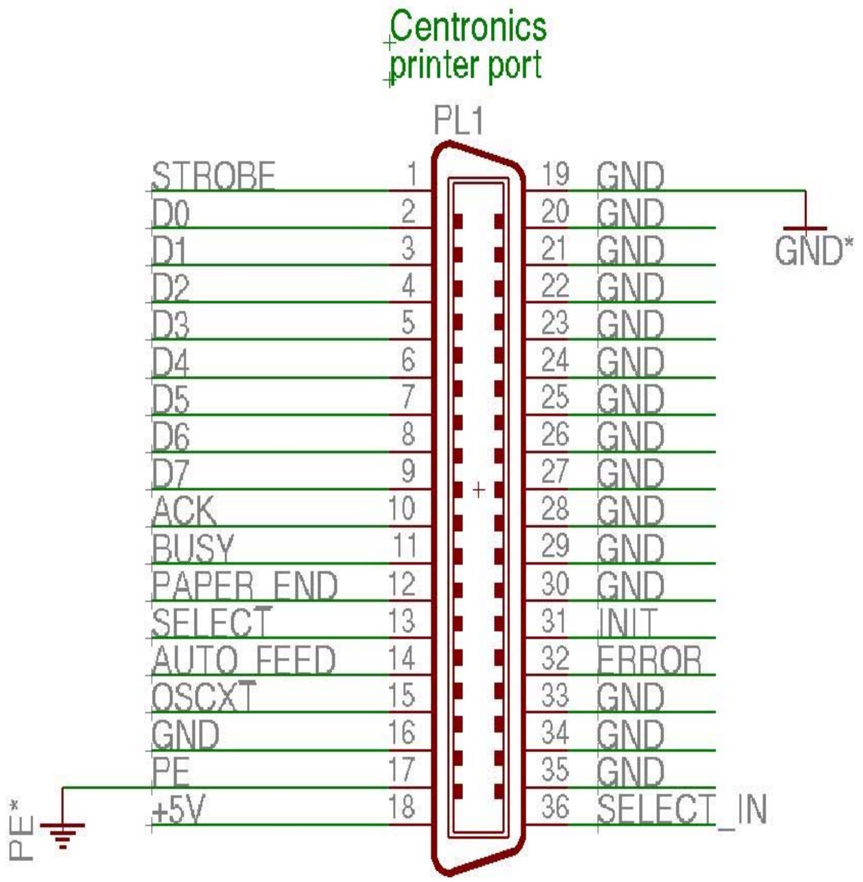


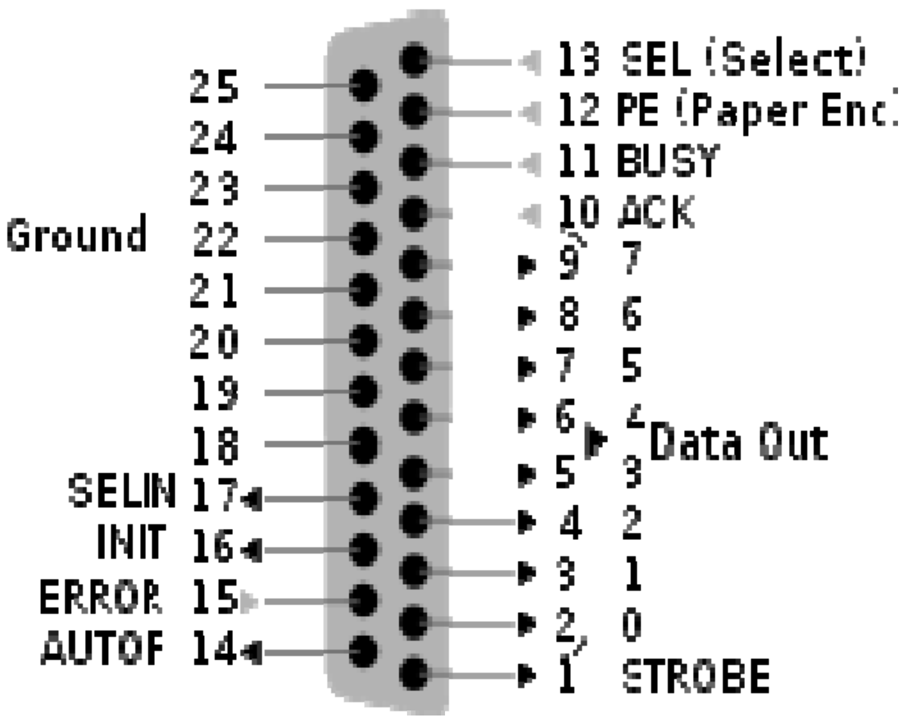

Fig. P.7. Schematic Diagram TCS256 Home Computer



***Centronics Printer Connector***




LPT1 DB25 Printer pinouts

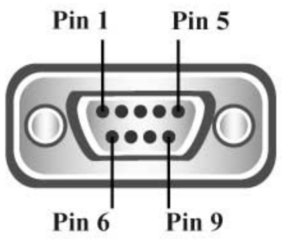


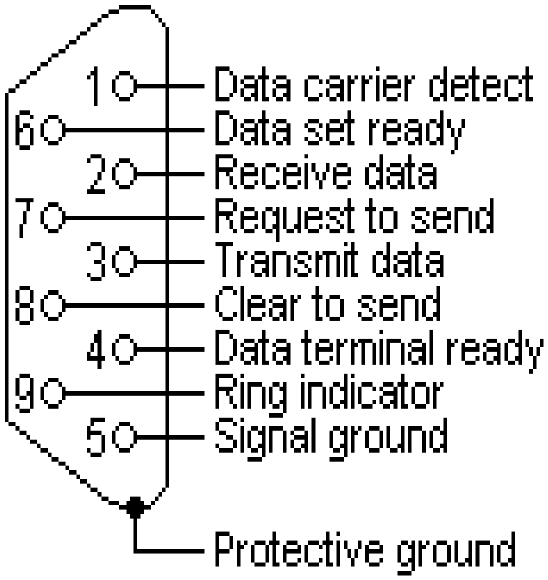
# Serial Port   COM1 / COM2      DB9

## RS232

| | |
|---|---|
| Pin 1 | DCD |
| Pin 2 | RXD |
| Pin 3 | TXD |
| Pin 4 | DTR |
| Pin 5 | GND |
| Pin 6 | DSR |
| Pin 7 | RTS |
| Pin 8 | CTS |
| Pin 9 | RI |

**RS232 Pinout (9 Pin Male)**



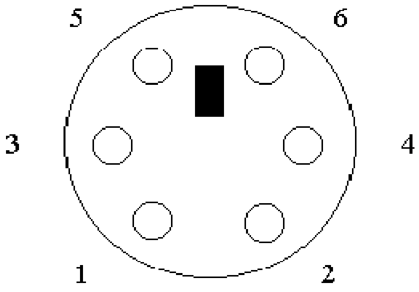




## Pinouts

The following table lists commonly-used RS-232 signals and pin assignments.

| Signal | | Origin | | | | | 8P8C ("RJ45") | | | 10P10C ("RJ50") | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Abbreviation | DTE | DCE | DB-25 | DE-9 (TIA-574) | TIA-561 | Yost | MMJ | Cyclades[4] | National Instruments[5] | Cyclades[4] | Digi[6] |
| Transmitted Data | TxD | • | | 2 | 3 | 6 | 3 | 2 | 3 | 8 | 4 | 5 |
| Received Data | RxD | | • | 3 | 2 | 5 | 6 | 5 | 6 | 9 | 7 | 6 |
| Data Terminal Ready | DTR | • | | 20 | 4 | 3 | 2 | 1 | 2 | 7 | 3 | 9 |
| Carrier Detect | DCD | | • | 8 | 1 | 2 | 7 | — | 7 | 10 | 8 | 10 (alt 2) |
| Data Set Ready | DSR | | • | 6 | 6 | 1 | | 6 | 8 | 5 | 9 | 2 (alt 10) |
| Ring Indicator | RI | | • | 22 | 9 | | — | — | — | 2 | 10 | 1 |
| Request To Send | RTS | • | | 4 | 7 | 8 | 1 | — | 1 | 4 | 2 | 3 |
| Clear To Send | CTS | | • | 5 | 8 | 7 | 8 | — | 5 | 3 | 6 | 8 |
| Common Ground | G | common | | 7 | 5 | 4 | 4,5 | 3,4 | 4 | 6 | 5 | 7 |
| Protective Ground | PG | common | | 1 | — | — | — | — | — | — | 1 | 4 |

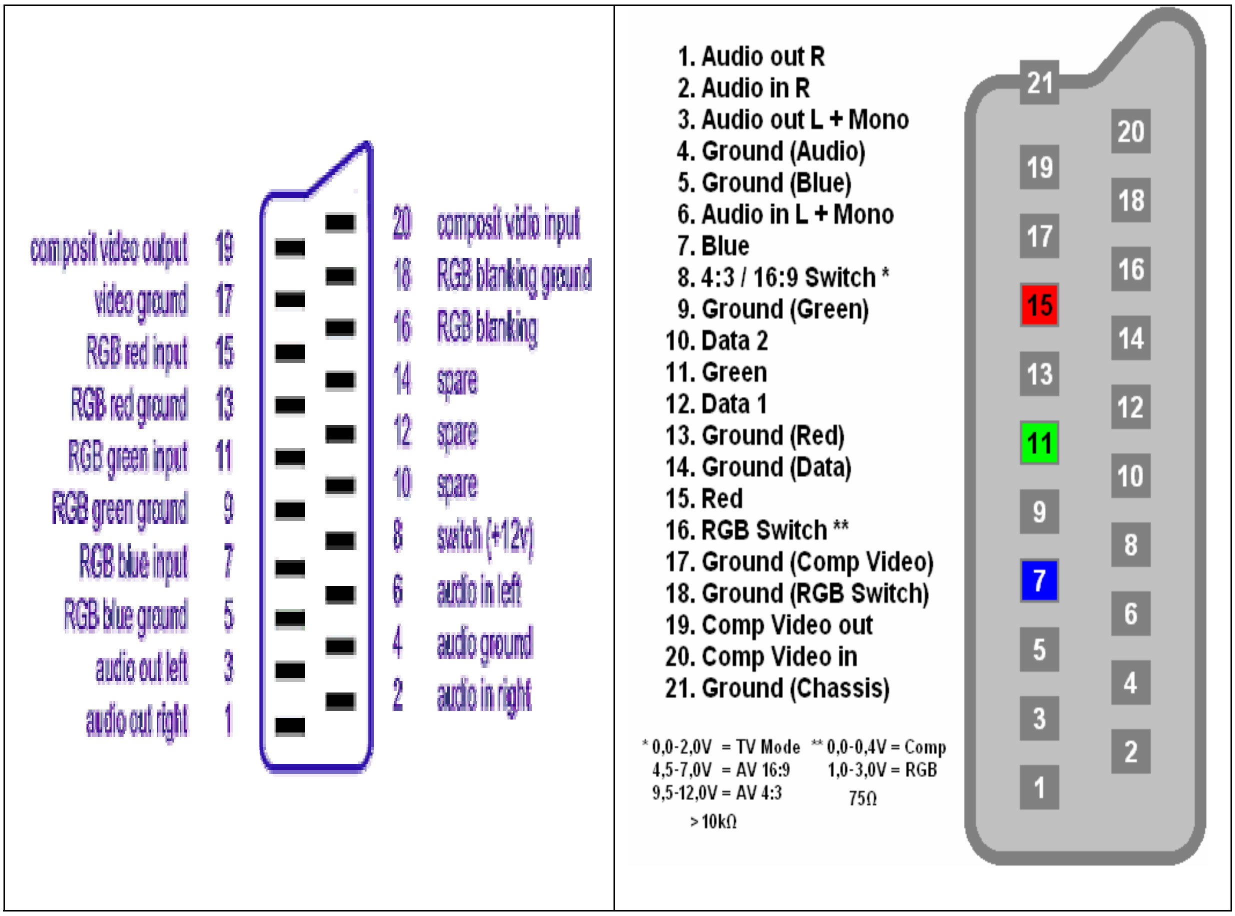
PS/2 Keyboard and Mouse Cable



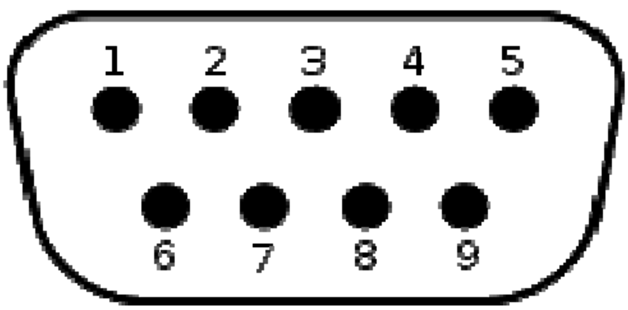

| Pin | Name |
|---|---|
| 1 | +Keyboard Data |
| 2 | Unused |
| 3 | Ground |
| 4 | +5 Volts |
| 5 | Clock |
| 6 | Unused |

Cable (male) pinout

# SCART TV Connector




## Atari joystick



Computer port view of the Atari standard connector: **1.** up, **2.** down, **3.** left, **4.** right, **5.** (pot y), **6.** fire button, **7.** +5 V DC, **8.** ground, **9.** (pot x).

# Bill of Materials(BOM) list

## Integrated Circuits

| *Item* | *Code* | *Qty* | *Price each* | *Total* |
|---|---|---|---|---|
| Z80 CPU | Z84C0020PEC | 1 | £1.75 | £1.75 |
| SRAM | HM628128BLP-7 | 1 | £1.00 | £1.00 |
| EEPROM | W27C512-45Z | 1 | £0.90 | £0.90 |
| RS232 driver | MAX232 IC | 1 | £0.50 | £0.50 |
| Sound Generator | AY-3-8912A | 1 | £2.50 | £2.50 |
| CPLD | EPM7128-15 | 2 | £3.20 | £6.40 |
| V9958 VDP | V9958 | 1 | £5.75 | £5.75 |
| DRAM 64k | D41464 | Max 6 | £2.00 | £12.00 (Max) |
| Flash 512k | AM29F040 | 1 | £1.00 | £1.00 |
| 5V, 1.5A regulator | 7805 | 1 | £0.15 | £0.15 |

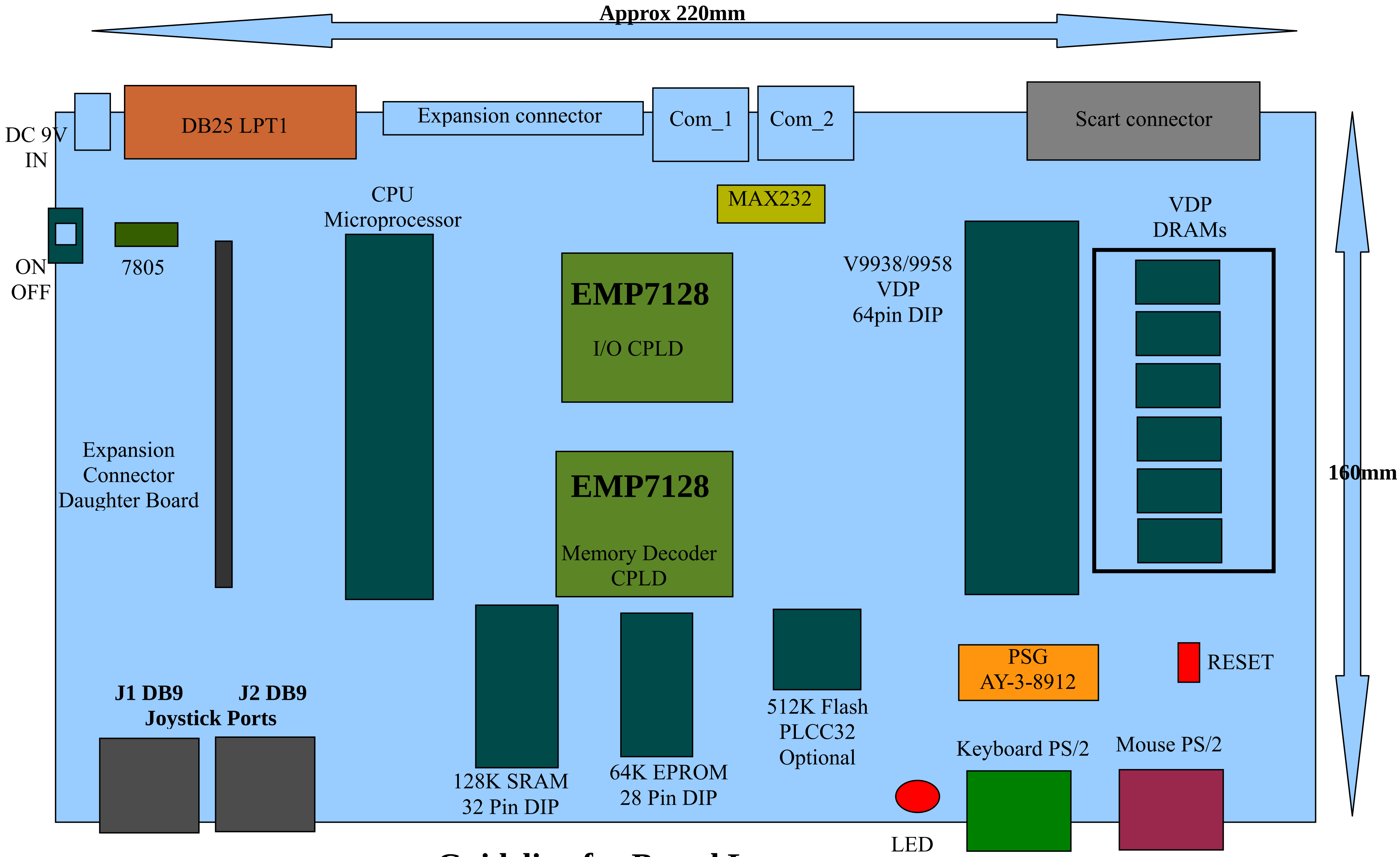
## Connectors

| Item | Code | Qty | Price each | Total |
|---|---|---|---|---|
| LPT1 DB25 | DB25 female | 1 | £0.45 | £0.45 |
| PS2 DIN 6 pin | DIN 6 pin female | 2 | £0.25 | £0.50 |
| Joystick | DB9 male | 2 | £0.25 | £0.50 |
| Scart TV | SCART | 1 | £0.90 | £0.90 |
| DC power | DC Jack socket | 1 | £0.10 | £0.10 |
| Switch On/Off | SWITCH | 1 | £0.05 | £0.05 |
| Expansion connector | Double SIL | 1 | £0.10 | £0.10 |
| Auxiliary Connector | DB15 ? | Optional | N/A | N/A |
| Reset Button | Single Momentary | 1 | £0.05 | £0.05 |

## Miscellaneous

| Item | Code | Qty | Price each | Total |
|---|---|---|---|---|
| Crystal 20MHz | XTAL 20 | 1 | £0.30 | £0.30 |
| Crystal | XTAL 21.4772 | 1 | £0.35 | £0.35 |
| Socket 84 LCC | PLCC84 | 2 | £0.70 | £1.40 |
| Socket 32 LCC | PLCC32 | 1 | £0.25 | £0.25 |
| Socket 40 | DIP 40 | 1 | £0.25 | £0.25 |

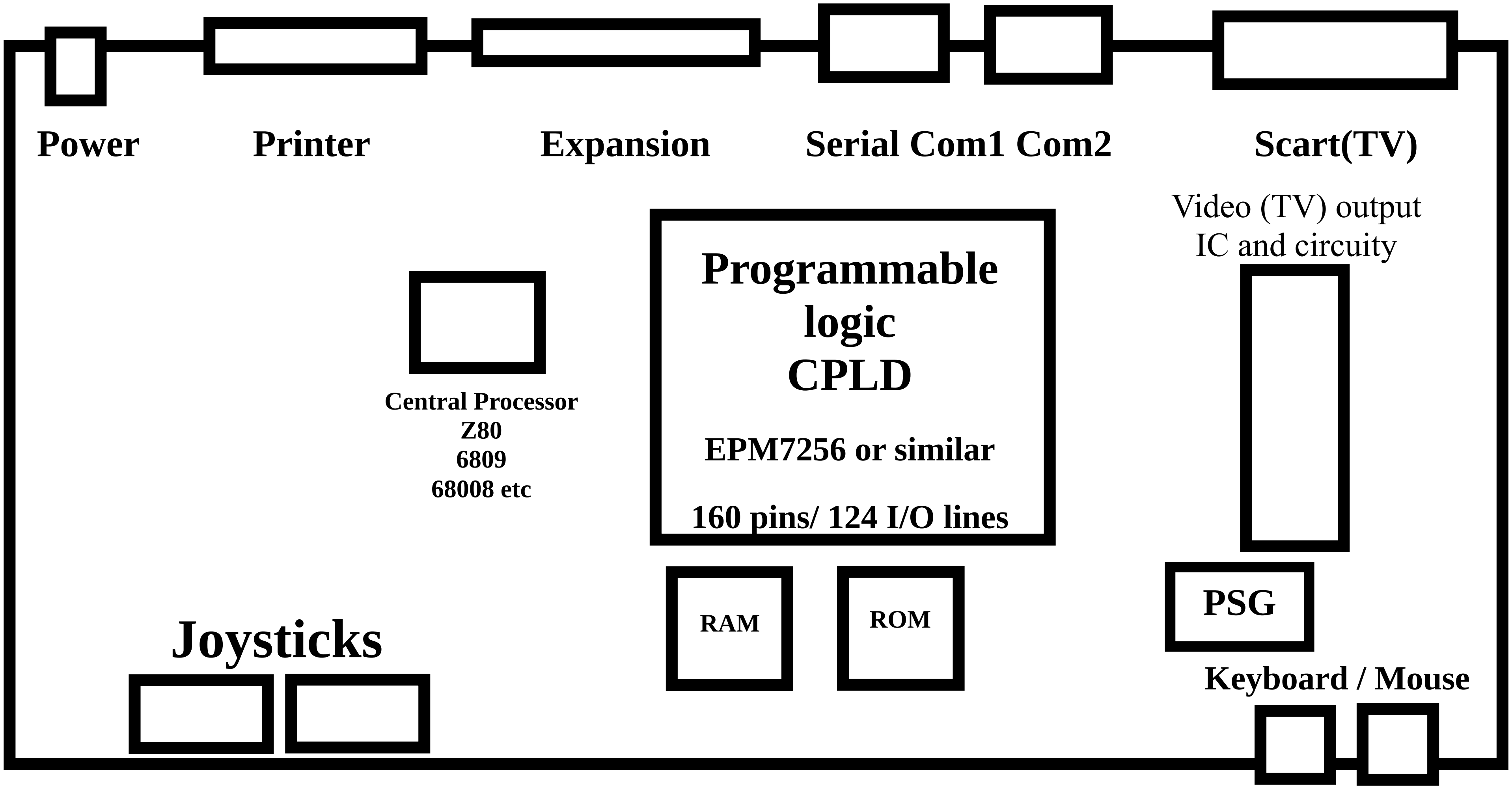
| | | | | |
|---|---|---|---|---|
| Socket 32 | DIP 32 | 1 | £0.15 | £0.15 |
| Socket 28 | DIP 28 | 1 | £0.15 | £0.15 |
| Socket 64 | DIP 64 | 1 | £0.15 | £0.15 |
| Socket 16 | DIP 16 | 7 (Max) | £0.10 | £0.70 |
| | | | | |

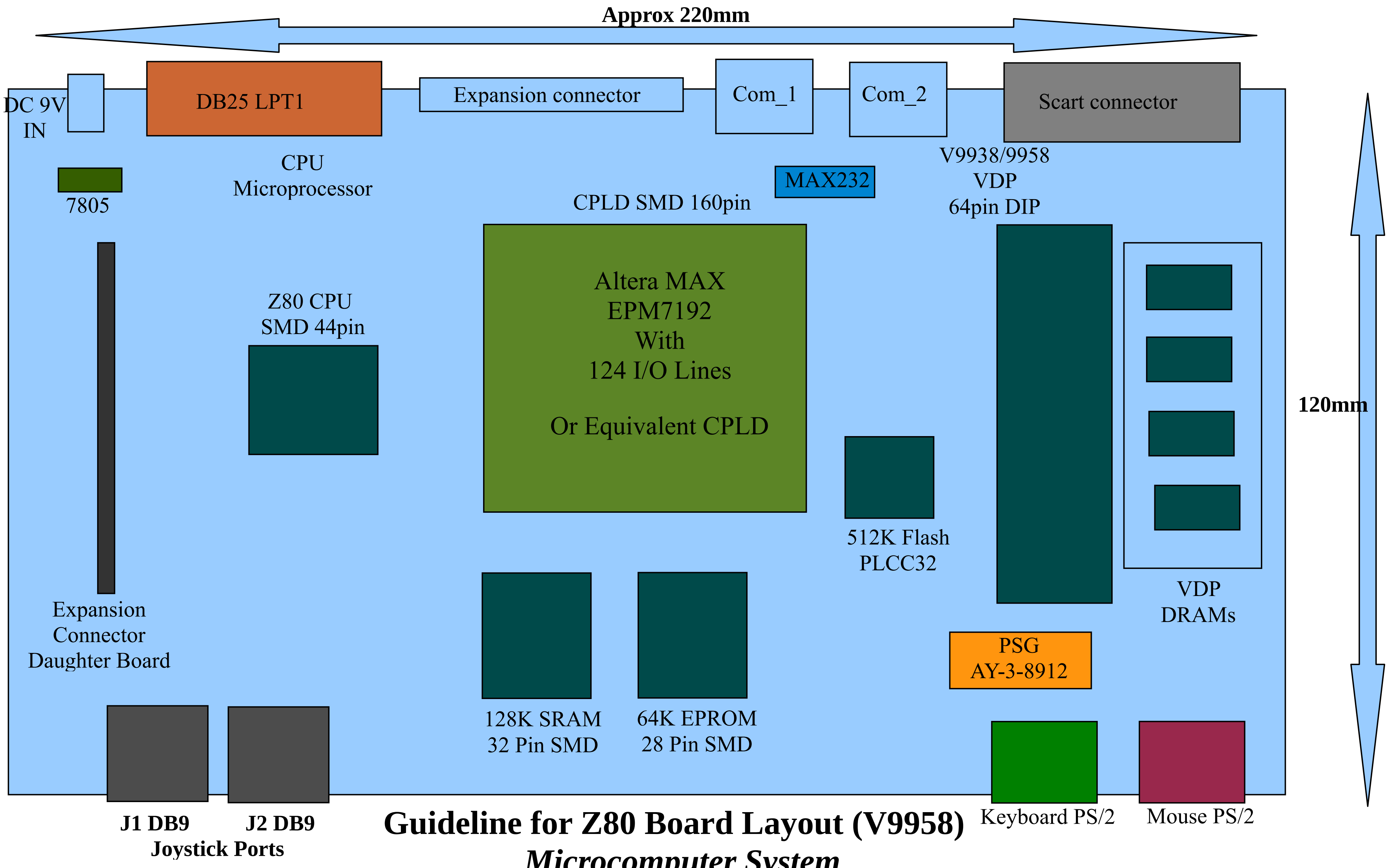



**Guideline for Board Layout**

***Microcomputer System***

# Z80 CPLD Board






**Guideline for Z80 Board Layout (V9958)**
***Microcomputer System***