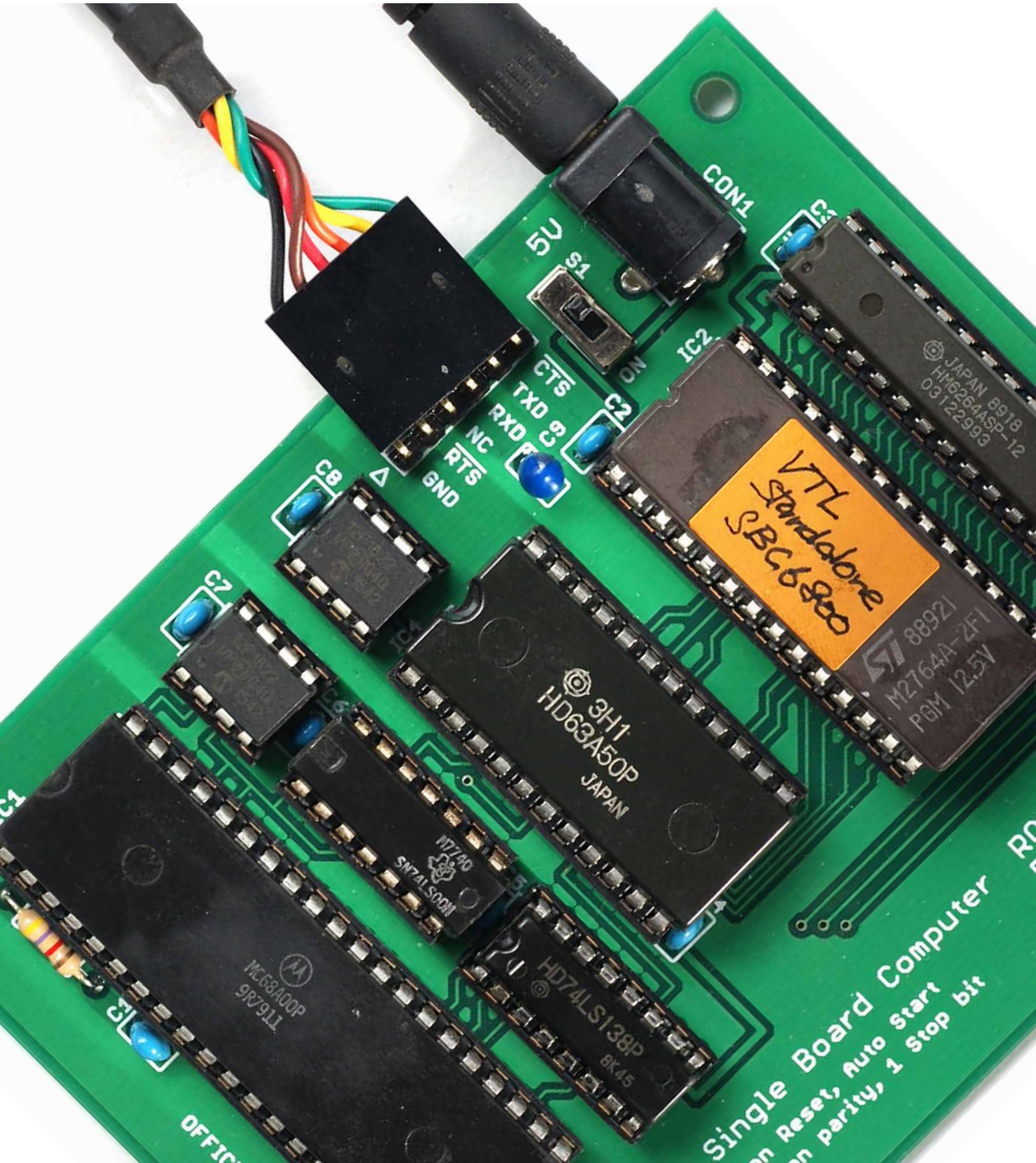


# SBC6800 Technical Data

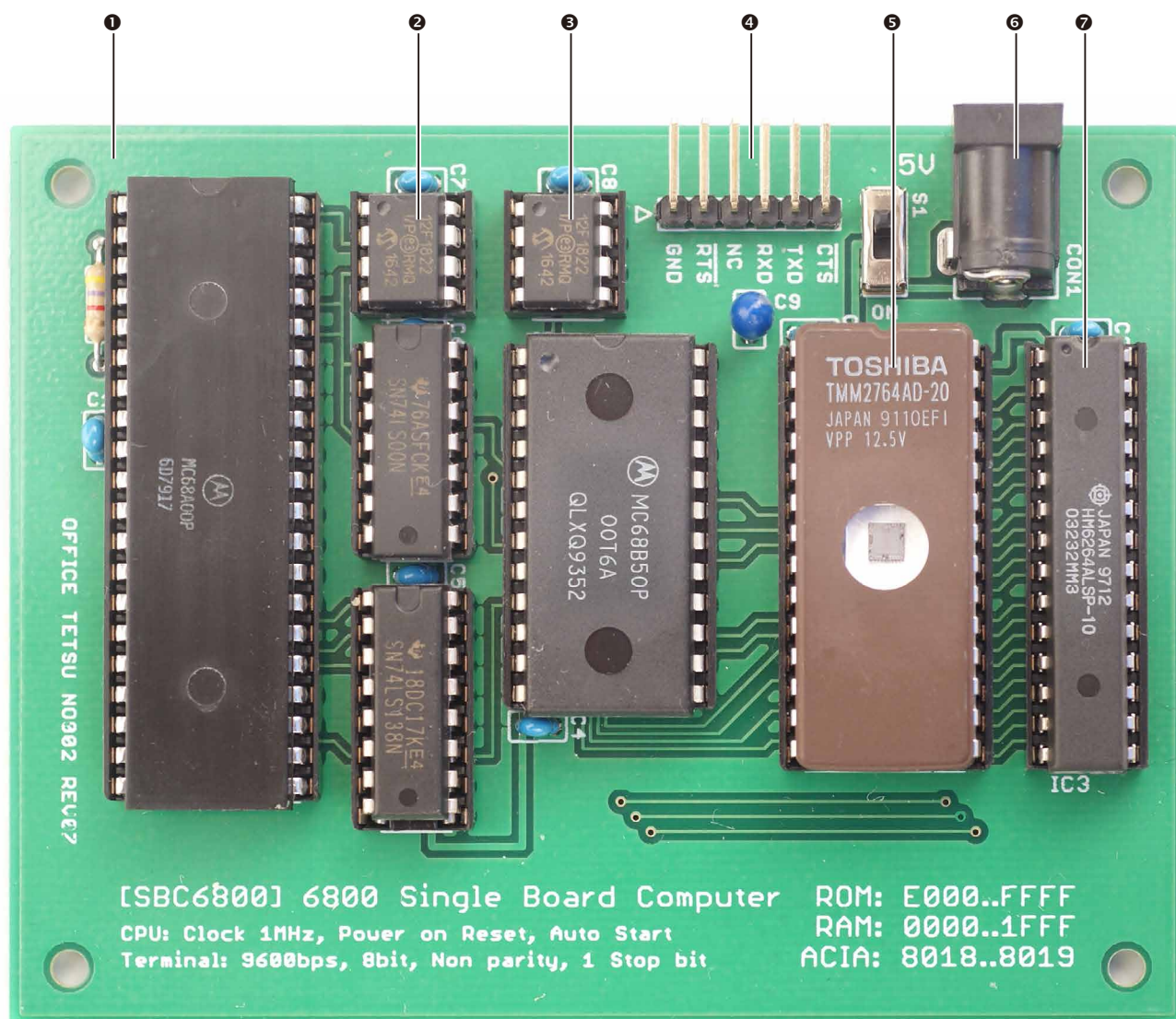
The SBC 6800 is a single board computer operating with the Motorola MC6800. You can run and fun the many popular software developed in the 1970s, such as Mikbug, VTL, MicroBASIC etc.



# Overview

SBC 6800 is an open source project. Technical data are described in this document. Gerber data is exist in Fusion Gallery SBC6800 project. Required firmware and Old software, such as Mikbug, VTL, MicroBASIC, are included in the separately distributed sbc6800\_datapack.zip. When assembling the SBC 6800, it will be as shown below.

- ❶ PCB-Order by 'Order boards' button in Fusion Gallery SBC6800 project.
- ❷ Clock generator-Write mc6800crngen.hex included in SBC6800\_datapack.zip to PIC12F1822, and installing it.
- ❸ Baud rate generator-Write osc1536.hex included in SBC6800\_datapack.zip to PIC12F1822, and installing it.
- ❹ Serial I/O-Connect FTDI TTL-232R-5V or compatible cable.
- ❺ ROM-Intel 2732, 2764, 27128 or 27256 compatible EPROM.
- ❻ DC Jack-Connect 5V/2A, 2.1mm, Center Plus AC adapter.
- ❼ RAM-HM6264ASP/HM6264ALSP. When solder pad is processed, it corresponds to HM6116ASP/HM6116ALSP.



# Parts List

The parts list is shown below. Collate the parts number of the parts list with the part number of the PCB and attach it to the same number position. TTL (IC5 and IC6) must be standard type or LS type (HC type is not acceptable).

Parts List

Parts number	Device name	Qty	Description
IC1	MC68A00P* <sup>1</sup>	1	Micro processor
IC2	2732, 2764, 27128 or 27256	1	EPROM
IC3	HM6264ASP/HM6264ALSP or HM6116ASP/HM6116ALSP* <sup>2</sup>	1	SRAM
IC4	MC68A50P	1	ACIA
IC5	SN74LS138N	1	TTL
IC6	SN74LS00N	1	TTL
IC7-IC8	PIC12F1822-I/P* <sup>3</sup>	2	Micro controller
R1	4.7kohm, 1/4W, carbon resister	1	resister
C1-C8	0.1uF, 50V, laminated ceramic capacitor	8	capacitor
C9	10-100uF, 16V, Tantalum/Electrolytic capacitor	1	capacitor
CON1	DC Jack, 2.1mm	1	DC Jack
S1	slide switch. ex.SS-12D00-G5	1	slide switch
-	IC socket, 40pin, 600mil. ex.2227-40-06	1	IC socket
-	IC socket, 28pin, 600mil. ex.2227-28-06	1	IC socket
-	IC socket, 28pin, 300mil. ex.2227-28-03	1	IC socket
-	IC socket, 24pin, 600mil. ex.2227-24-06	1	IC socket
-	IC socket, 16pin, 300mil. ex.2227-16-03	1	IC socket
-	IC socket, 14pin, 300mil. ex.2227-14-03	1	IC socket
-	IC socket, 8pin, 300mil. ex.2227-08-03	2	IC socket
-	L type 1line 6pin header. ex.2545-1X40* <sup>4</sup>	1	pin header

\*1.Probably, standerd MC6800 will work, but we recommend MC68A00 or MC68B00.

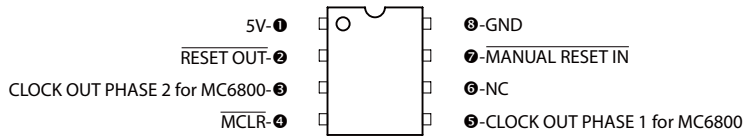
\*2.When using HM6116ASP/HM6116ALSP, processing of solder pad is necessary.

\*3.Write mc6800crngen.hex to IC7 and osc1536.hex to IC8.

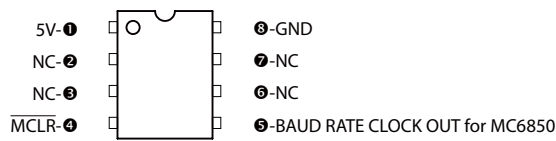
\*4.Cut the 6 pins out of 40 pins and use it.

# Writing Firmware

The clock and the reset signal for the MC6800 are generated by the PIC12F1822 (IC7). So write mc6800crgen.hex to PIC12F1822 before installing. When writing mc6800crgen.hex, PIC12F1822 performs the function shown below.



The baud rate clock for the MC6850 is generated by the PIC12F1822 (IC8). So write osc1536.hex to PIC12F1822 before installing. When writing osc1536.hex, PIC12F1822 performs the function shown below.



# Writing Program

The application program of SBC6800 writes to EPROM (IC2). There are some machine language files such as Mikbug (Mikbug.HEX) or VTL (VTL.SA.HEX) in the sbc6800\_datapack.zip. Depending on the EPROM used, how to write differs as follows.

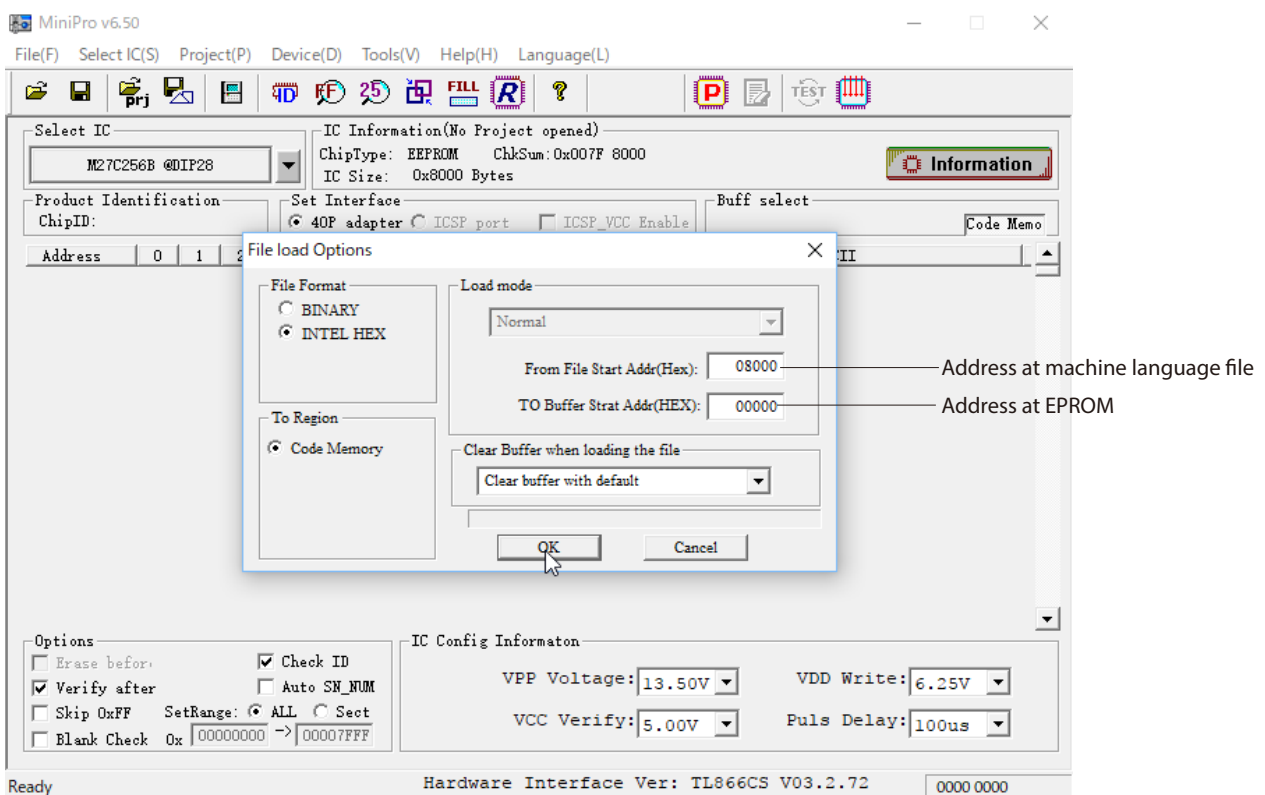
2732 type-This is explained in the next page "Using Alternate Memory".

2764 type-Write From \$E000 at machine language file to \$0000 at EPROM. This is standard. Others are alternative.

27128 type-Write From \$C000 at machine language file to \$0000 at EPROM. Only valid after \$ E000.

27256 type-Write From \$8000 at machine language file to \$0000 at EPROM. Only valid after \$ E000.

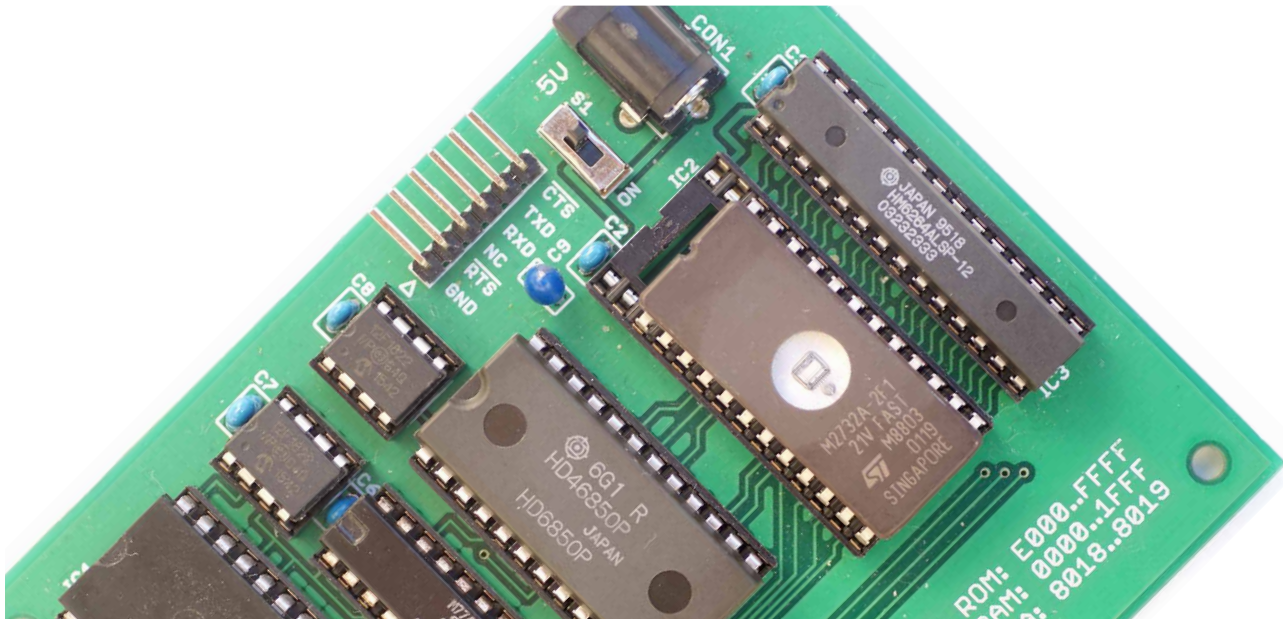
Specification example when writing to 27256 type with writing device TL866CS, write software MiniPro.



# Using Alternate Memory

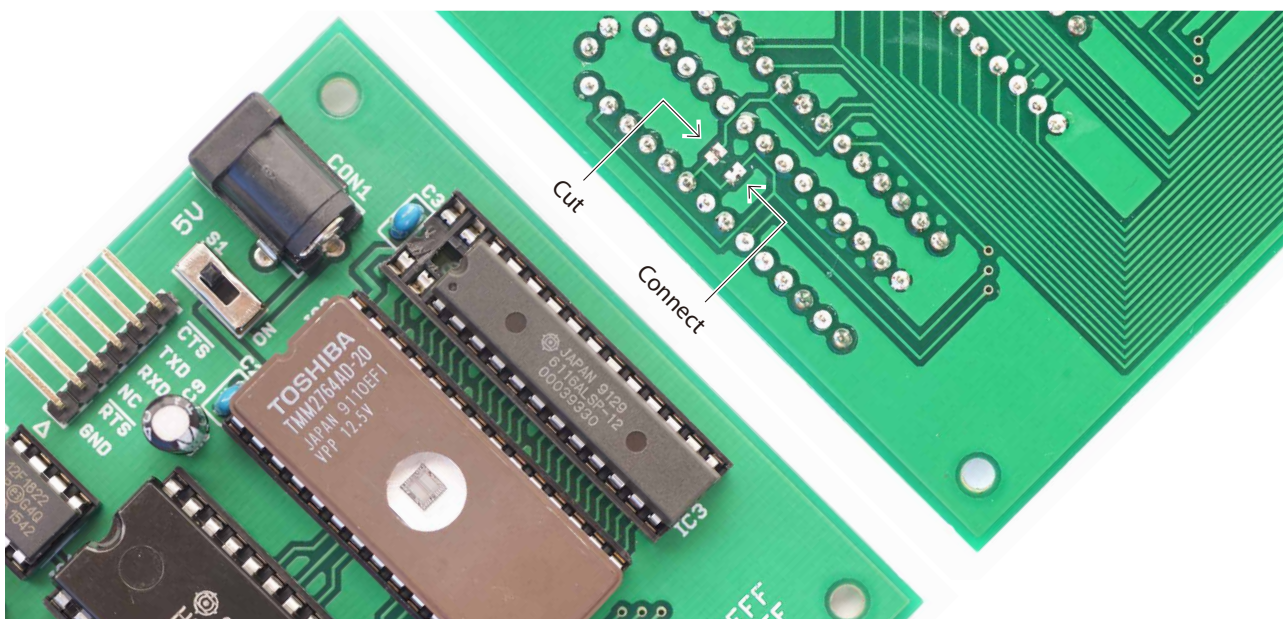
When using 2732 type for EPROM (IC2), write and install it as follows.

- ❶ Write From \$E000 at machine language file to \$0000 at EPROM.
- ❷ Disable erase check and verify of writing software.
- ❸ Overwrite From \$F000 at machine language file to \$0000 at EPROM.
- ❹ Attach leave the index side of the IC socket open.



When using HM6116ASP/ALSP for RAM (IC3), process the PCB and attach as follows.

- ❶ Cut the connected solder pad and connect(solder bridge) the oder solder pad.
- ❷ Attach leave the index side of the IC socket open.



## Items for operation

To operate the SBC6800, you need an AC adapter and USB to serial conversion cable.

AC adapter-5V, 2A or more, 2.1 mm, center plus(Ex. Go Forward Enterprise corp. GF12-US0520).

USB to serial conversion cable-FTDI TTL-232R-5V or compatible cable.

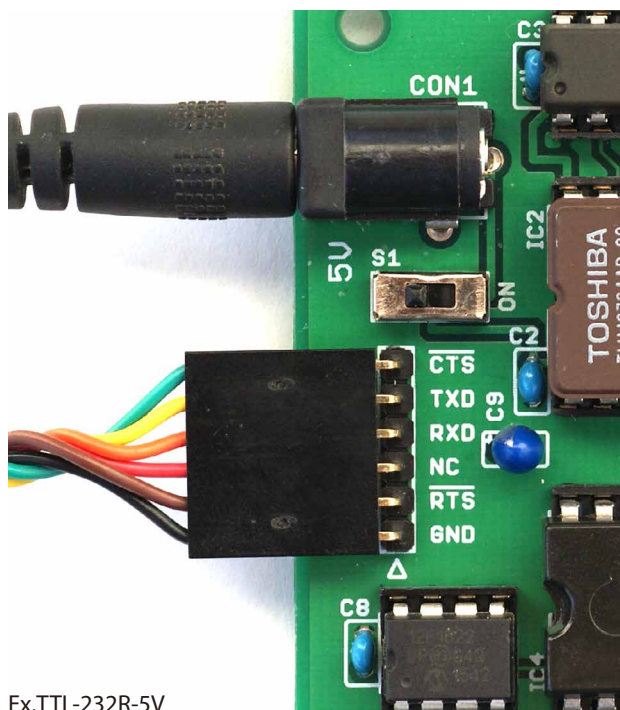


GF12-US0520

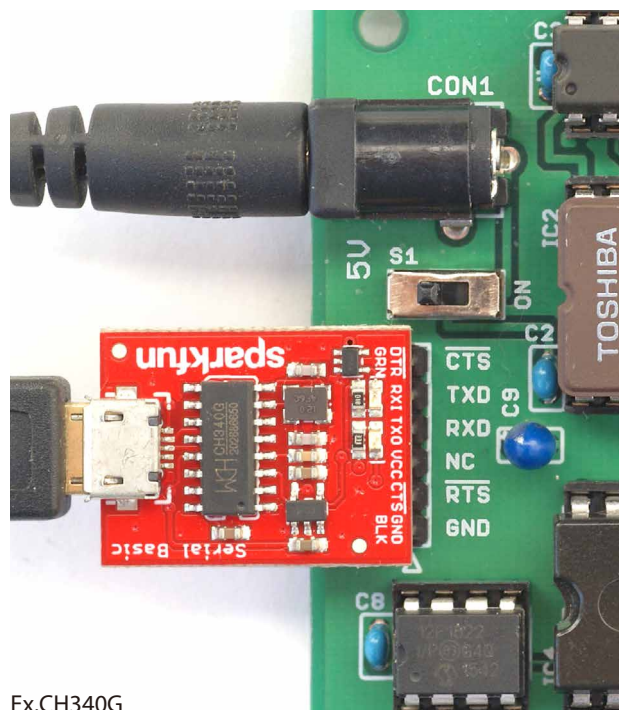


TTL-232R-5V

The signal name printed on the PCB serial terminal is the output of the MC6850. Connect this so that the signal of the USB - serial conversion cable is cross - referenced. It is normal to be TXD-RXD, CTS-RTS(or DTR), GND- GND. If using a USB-serial conversion cable supporting 3.3V / 5V signal voltage, please set it to 5V.

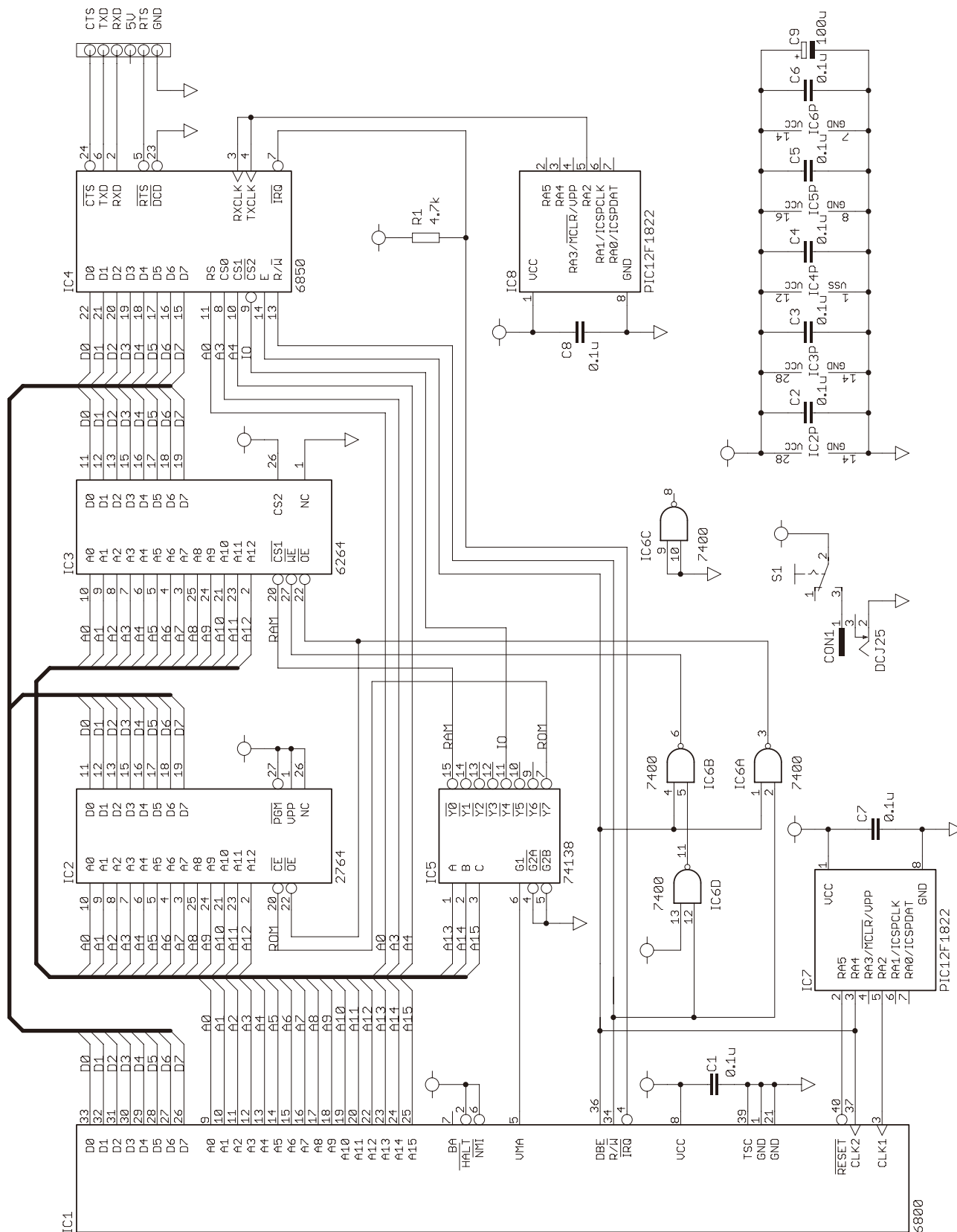


Ex.TTL-232R-5V



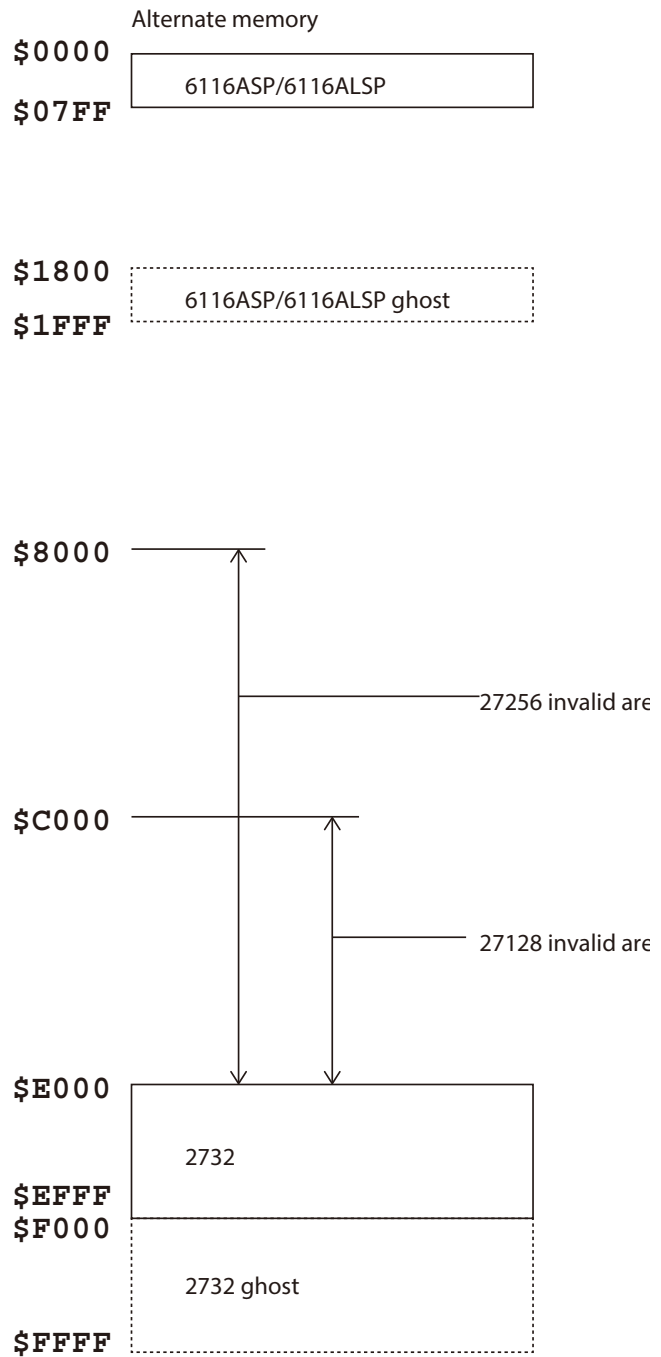
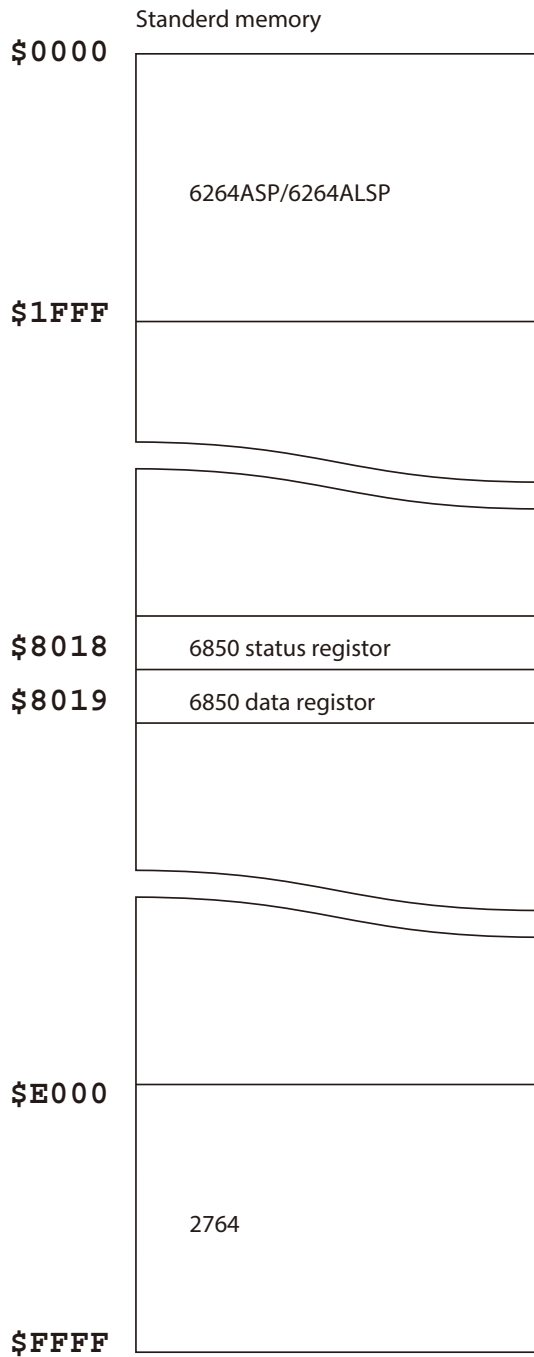
Ex.CH340G

The schematic of the SBC6800 is shown below. The part number is consistent with the PCB and parts list.



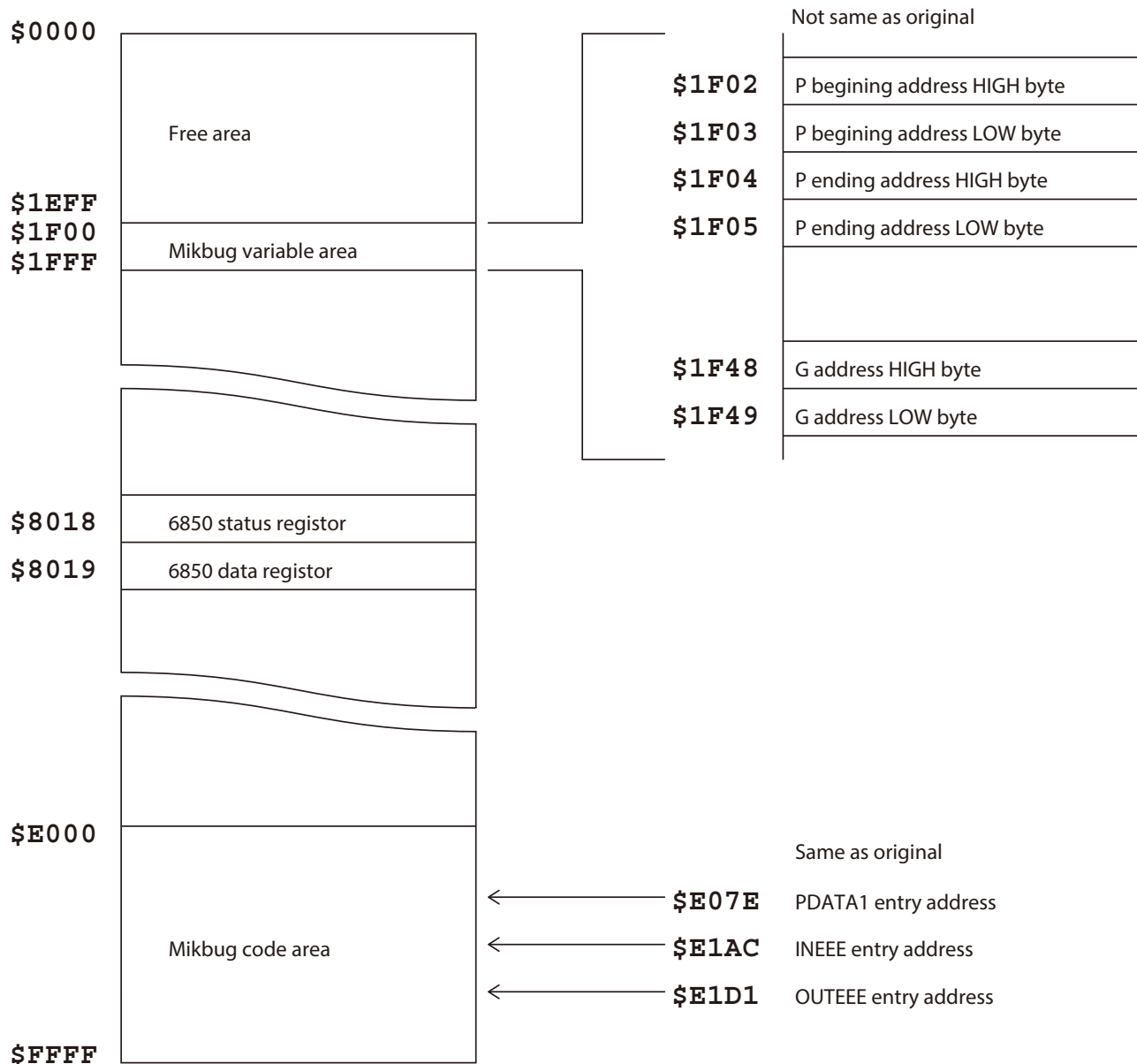
# Address map

The address map of the SBC6800 is shown below.



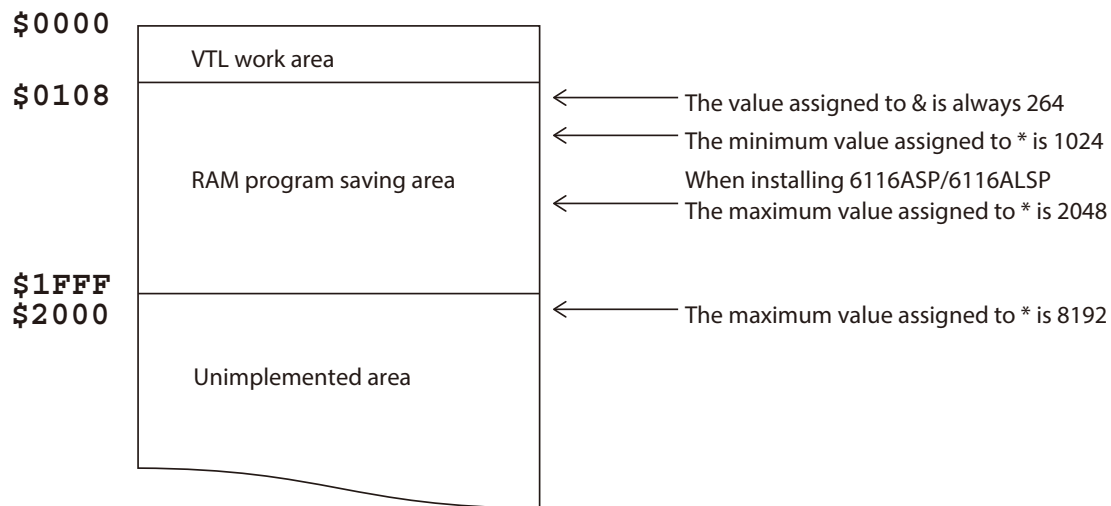
# Mikbug SBC6800 Edition

MIKBUG.\* included in the sbc6800\_datapack.zip is Mikbug SBC6800 edition. For details on how to use Mikubug, please refer to the manual distributed on the MIKBUG Operating System ([http://www.swtpc.com/mholley/MP\\_A/MIKBUG\\_Index.htm](http://www.swtpc.com/mholley/MP_A/MIKBUG_Index.htm)). However, the original variable area is start from \$A000, but the SBC6800 edition is start from \$1F00. Therefore, please change read the address of the variable area related to the P command and G command from \$A000- in that manual to \$1F00-.

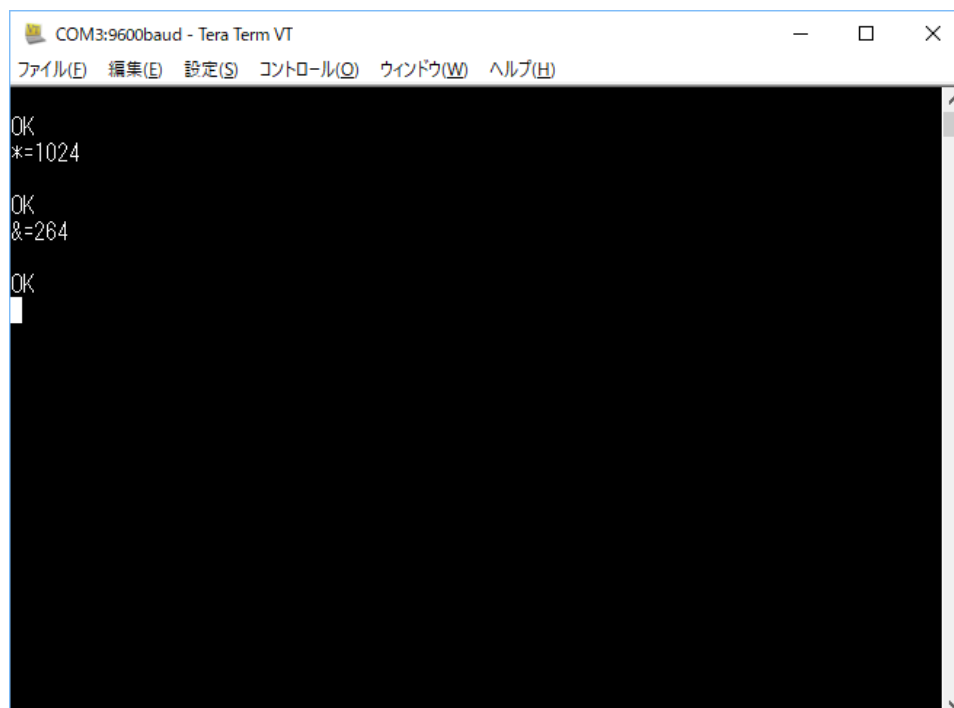


# VTL SBC6800 Stand Alone Edition

VTLSA.\* included in the sbc6800\_datapack.zip is VTL (officially named VTL-2) SBC6800 stand alone edition. The original VTL starts with the monitor of MITS Altair680, but VTLSA.\* has been modified to start directly from EPROM. The usage of VTLSA.\* is the same as the original VTL. As for VTL, there are a lot of information on the net, please search.



After VTL start and before using, specify the range of the program saving area of the RAM. First, assign the end address to the system variable \*. This is at least 1024, up to 8192 (2048 if RAM is HM6116ASP/HM6116ALSP). Next, assign the begin address to the system variable &. This is always 264. If you use it without specifying it will runaway.



# Separately Distribution List

sbc6800\_datapack.zip contains the files shown below.

filelist.txt - File list. It is the same content as this page.

SBC6800.zip - SBC6800 garber data.

SBC6800eagle - SBC6800 EAGLE data.

mc6800crgen.hex - PIC12F1822 clock and reset firmware.

mc6800crgen.X - mc6800crgen.hex MPLAB XIDE project.

osc1536.hex - PIC12F1822 baud rate generator firmware.

osc1536.X - osc1536.hex MPLAB XIDE project.

TESTPOL.\* - A test program for keyin and echo back with terminal.

TESTINT.\* - A test program for keyin and echo back with terminal by recieve interrupt.

MIKBUG.\* - Mikbug SBC6800 edition.

MICBAS13.\* - MicroBASIC Ver.1.3 which operates under Mikbug SBC6800 edition.

RNDSORT.BAS - Sample program of MicroBASIC Ver.1.3.

VTLSA.\* - VTL SBC6800 stand alone edition.

VTLALPHA.VTL - Sample program of VTL SBC6800 stand alone edition.

PROM680.\* - 680 monitor SBC 6800 edition(MITS Altair 680 monitor ported version).

VTL.\* - Original VTL(Started with J FC00 by 680monitor SBC6800 edition).

SBC6800.zip and SBC6800eagle are CC BY-SA 3.0(Copyright © 2017 Tetsuya Suzuki).

mc6800crgen.\*, osc1536.\*, TESTPOL.\*, TESTINT.\*, RNDSORT.BAS are public domain.

Other files license are declared by the original author.

SBC6800 Technical Data

Issued on December 30, 2017

Author-Tetsuya Suzuki

Copyright © 2017 Tetsuya Suzuki

CC BY-SA 3.0