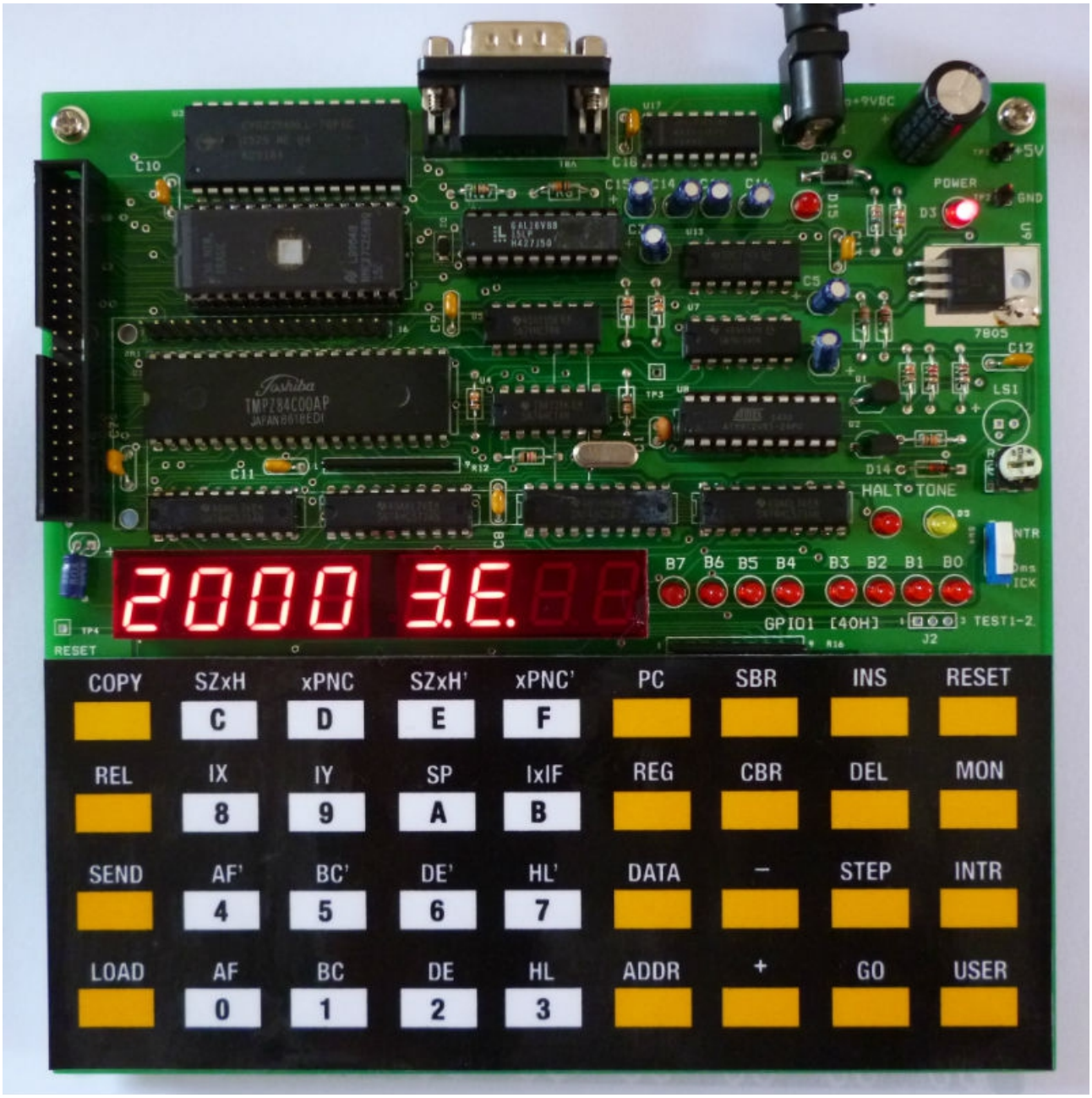
Z80 Microprocessor Kit
User's manual



Rev. 1.1, October 2015

Z80 MICROPROCESSOR KIT
USER'S MANUAL

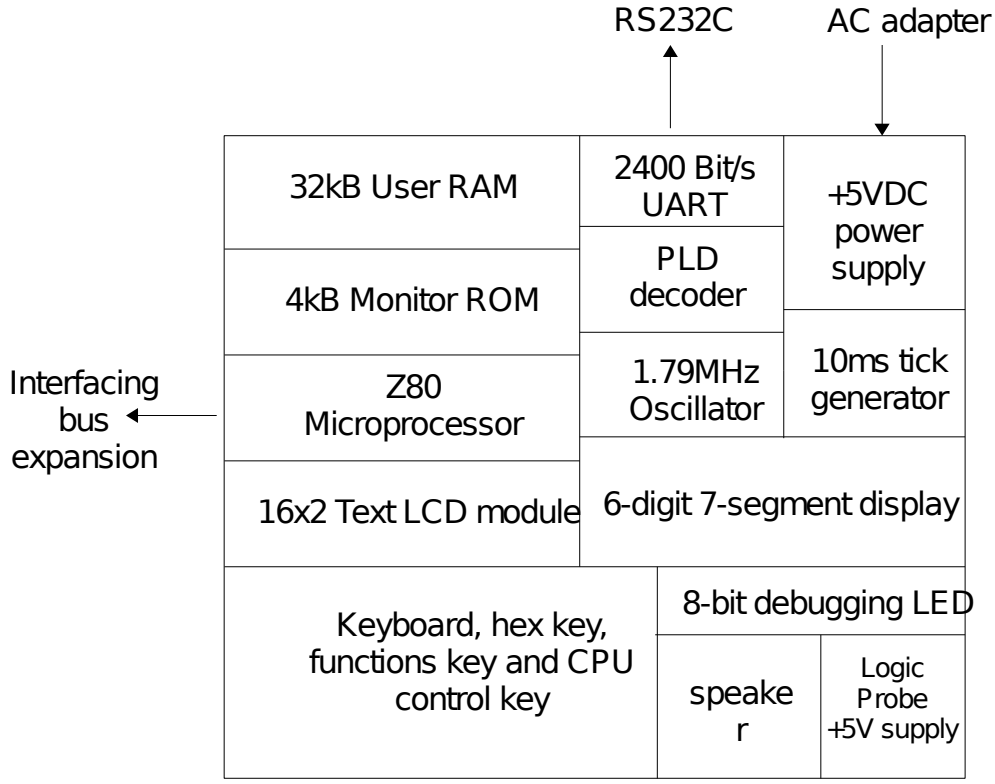
CONTENTS

OVERVIEW

The Z80 microprocessor kit is a single board microcomputer designed for self learning. Students will learn how to build the computer using the 1976, Z80 microprocessor with memory and simple I/O chips. The kit can be assembled at home without the need of special tools. The kit itself is the real microcomputer. Students will learn how to program the Z80 microprocessor in machine language with instruction hex code. The hex code can be entered to memory and tested it directly. Results of CPU operations can be checked in the memory or user registers display easily.

The kit manuals are 1) user's manual for hardware details, 2) programming lab book and 3) construction guide for kit assemble steps.

Z80 KIT FUNCTIONAL BLOCK DIAGRAM



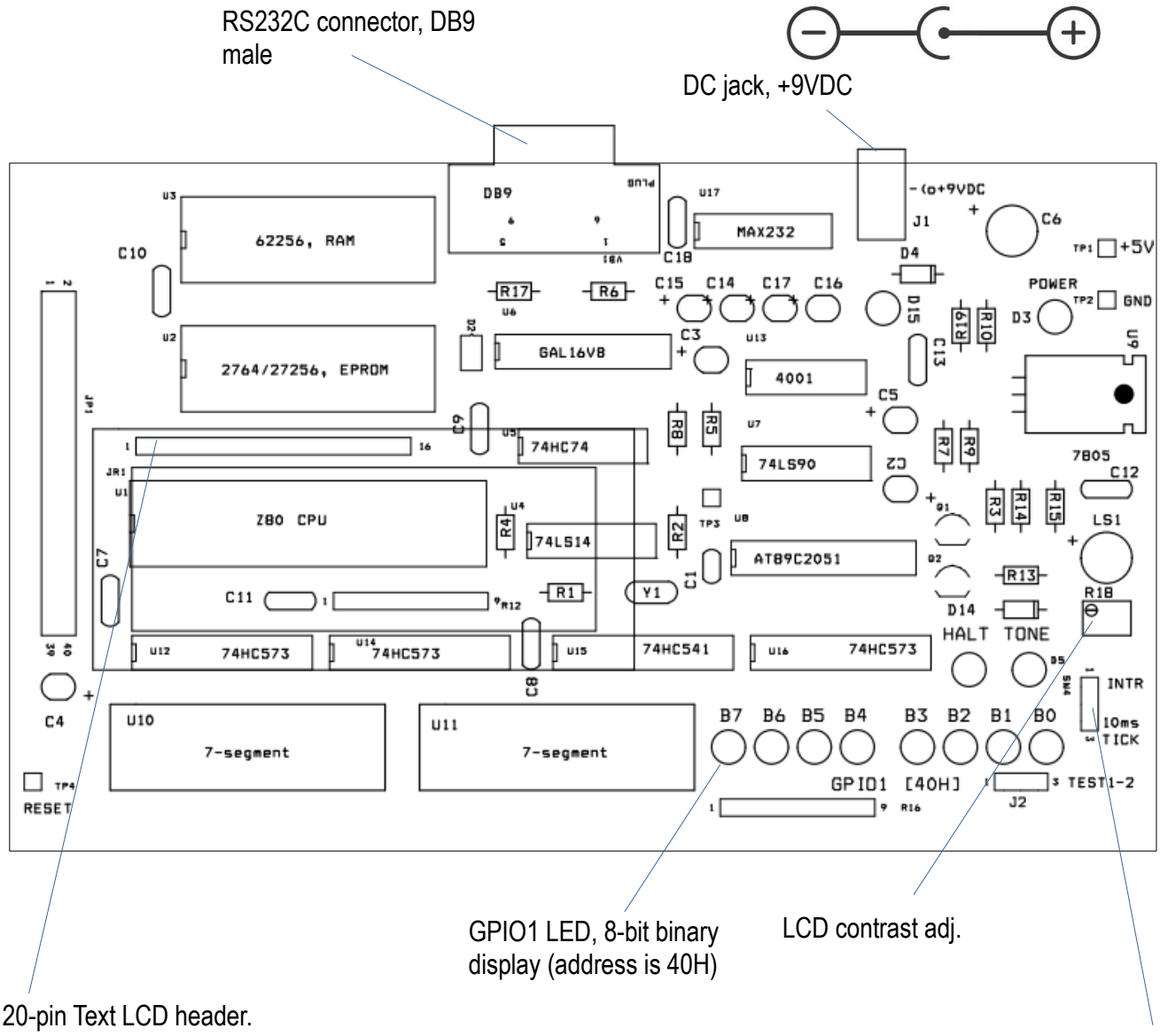
Notes

1. UART is software control for low speed asynchronous communication.

2. LCD module is optional, the kit provides a 16-pin header.

3. 10ms Tick generator is for a maskable interrupt source. NMI is used by single stepping circuit.

HARDWARE LAYOUT



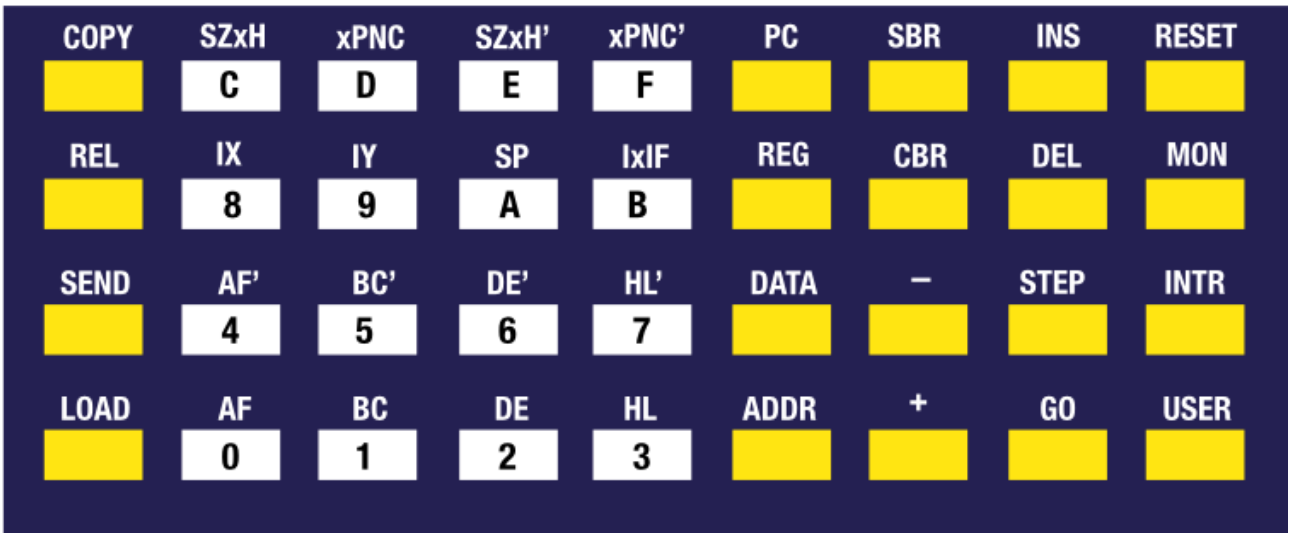
Important Notes

1. Plugging or removing the LCD module must be done when the kit is powered off!

2. AC adapter should provide approx. +9VDC, higher voltage will cause the voltage regulator chip, U9 becomes hot.

3. The kit has diode protection for wrong polarity of adapter jack. If the center pin is not the positive (+), the diode will reverse bias, preventing wrong polarity to feed to voltage regulator.

KEYBOARD LAYOUT



HEX keys Hexadecimal number 0 to F with associated user registers and flag bits when press REG

CPU control keys

RESET Reset the CPU, Z80 will begin fetch the code from location 0000
MON Force CPU to jump back to monitor program
INTR Make INT pin to logic low, used for experimenting with interrupt process
USER User key for lab test, active low

Monitor function keys

INS Insert byte to memory, current byte will be shifting down.

DEL Delete current byte, the next byte will be shifting up.

STEP Execute user code only single instruction and return to save CPU registers

GO Jump from monitor program to user code

SBR Set break address

CBR Clear break address

PC Set current display address with user Program Counter

REG Display user registers or flags with HEX key for a given register.

DATA Set entry mode of hex keys to Data field
ADDR Set entry mode of hex keys to Address field

COPY Copy block of memory, used with key + for Start, End, Destination and key GO

REL Compute relative byte, used with key + for Start, Destination and key GO

SEND Send Intel hex file at 2400 bit/s using serial port

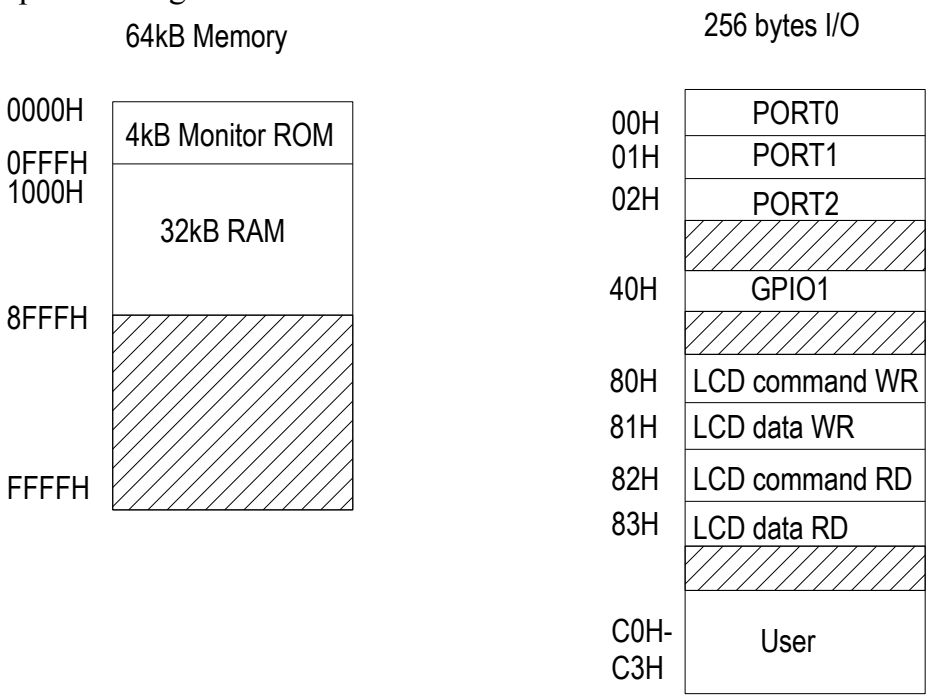
LOAD Load Intel hex file at 2400 bit/s using serial port

HARDWARE FEATURES

- **Microprocessor: Zilog Z80 @1.79MHz, 40-pin DIP package**
- **Memory: 4kB monitor ROM, 32KB user RAM**
- **Memory& I/O Decoder logic: GAL16V8**
- **System tick: 10ms produced by AT89C2051 microcontroller**
- **GPIO: 8-bit LED display**
- **DISPLAY: 6-digit super bright 7-segment display**
- **Keypad: 36-key**
- **Serial port: 2400 Bit/s RS232C using software controlled UART**
- **Expansion slot: 40-pin header**
- **LCD bus: 16x2 text display direct bus interface**

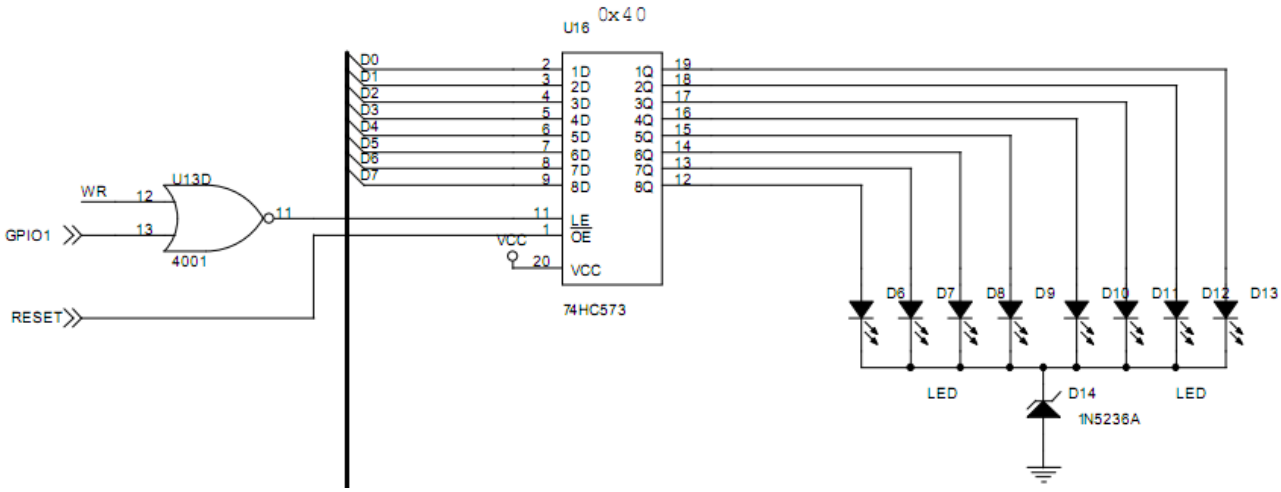
MEMORY AND I/O MAP

The Z80 kit has separated memory and I/O space. Memory space is selected with MREQ signal, I/O space with IORQ signal. The bank areas for both memory and I/O are available for experimenting.



GPIO1 LED

The Z80 kit provides a useful 8-bit binary display. It can be used to debug the program or code running demonstration. The I/O address is 40H. U16 is 8-bit data latch. Logic 1 at the output will make LED lit.



The GPIO1 LED can be used to display accumulator register easily. Let us take a look the sample code below.

Address	Hex code	Label	Instruction	comment
1800	3E 01	MAIN	LD A,1	Load register A with 1
1802	D3 40		OUT (40H),A	Write A to GPIO1@ 40H

The test code has only two instructions. Each instruction has machine code of two bytes. Enter the hex code to memory from 1800 to 1803. Then press PC, and execute the instruction with single step by pressing key STEP. The 2nd pressed STEP key that executes instruction out (40H),A will make the GPIO1 LED showing the content of register A.

Another sample is with JUMP instruction. The JUMP instruction will change the Program Counter to 1800, to repeat program running.

Address	Hex code	Label	Instruction	comment
1800	3C	MAIN	INC A	Increment register A
1801	D3 40		OUT (40H),A	Write A to GPIO1@ 40H
1803	C3 00 18		JP MAIN	Jump back to main

Again enter the hex code to memory and test it with single step. Every time when instruction out(40H),A was executed, did you see the LED display changed?

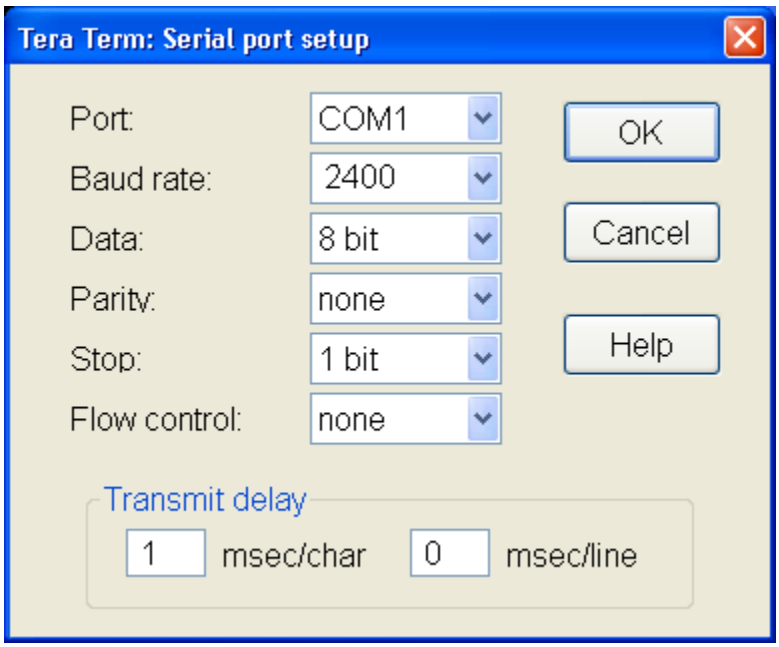
We will learn more the use of GPIO1 with Z80 Programming Lab Book.

CONNECTING Z80 KIT TO TERMINAL

All of monitor commands are compatible with MPF-1, except SEND and LOAD keys. For LOAD key, we can connect the Z80 kit to a terminal by RS232C cross cable. You may download free terminal program, teraterm from this URL,
http://ttssh2.sourceforge.jp/index.html.en

The example shows connecting laptop with COM1 port to the RS232C port of the Z80 kit.

To download Intel hex file that generated from the assembler or c compiler, **set serial port speed to 2400 bit/s, 8-data bit, no parity, no flow control, one stop bit.**

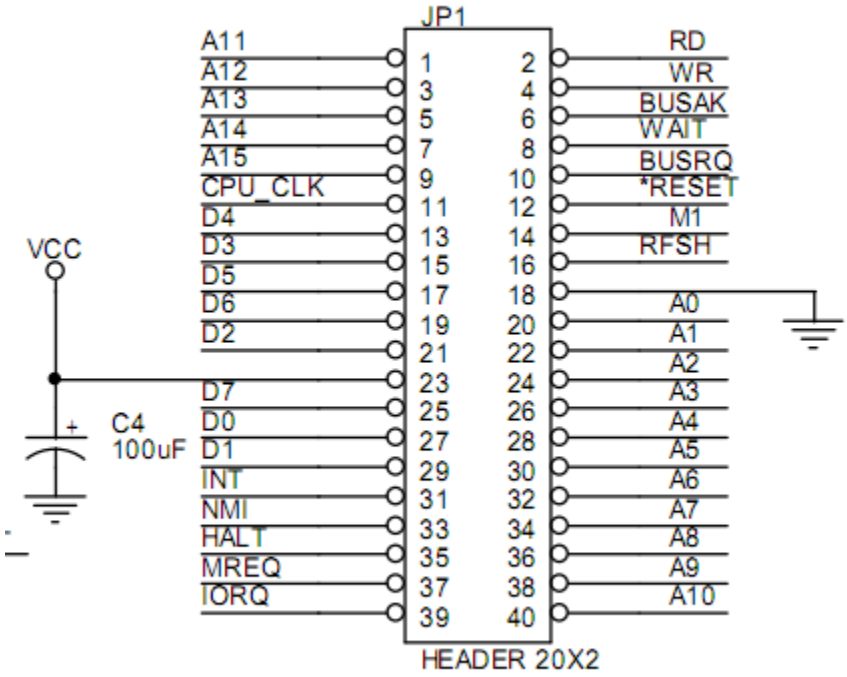


Press key LOAD, then key GO. The kit will wait for the data stream from terminal. On PC, Click file>Send File>LED.HEX. The kit will read the hex file, write to memory, when completed the start message will be displayed.

SEND key will need Start, End address then press GO, the kit will send Intel hex file to terminal.

EXPANSION BUS HEADER

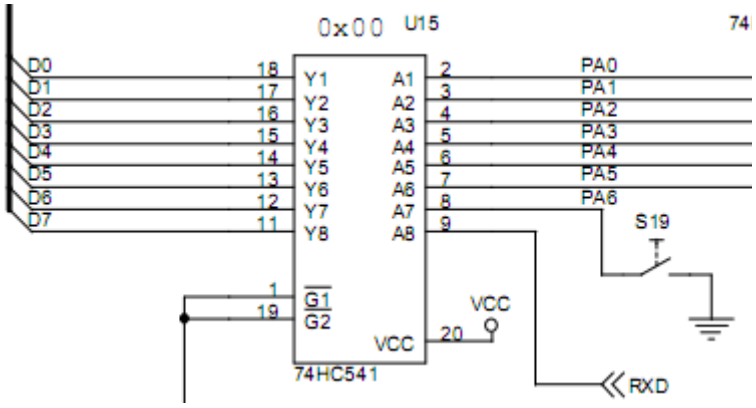
JP1, 40-pin header provides CPU bus signals for expansion or I/O interfacing. Students may learn how to make the simple I/O port, interfacing to Analog-to-Digital Converter, interfacing to stepper motor or AC power circuits.



Expansion Bus

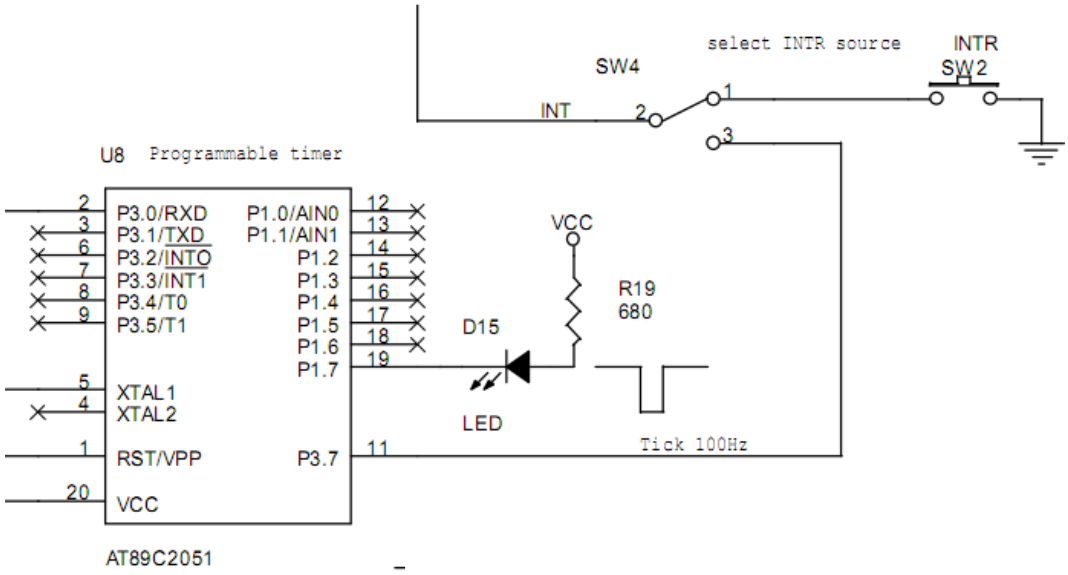
USER KEY

User key, S19 is one bit active low key switch connected to bit 6 of Port 0. To test the logic of S19, we can use IN A,(0) and check bit 6 of the accumulator with test bit instruction.



10ms TICK GENERATOR

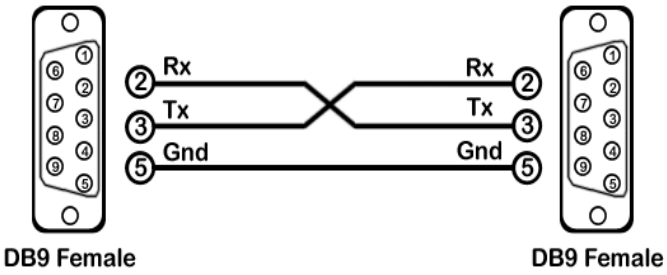
SW4 is a selector for interrupt source between key INTR or 10ms tick produced by 89C2051 microcontroller. Tick generator is software controlled using timer0 interrupt in the 89C2051 chip. The active low tick signal is sent to P3.7. For tick running indicator, P1.7 drives D15 LED.



Tick is a 10ms periodic signal for triggering the Z80 interrupt pin. When select SW4 to Tick, the Z80 CPU can be triggered by a maskable interrupt. The 100Hz tick or 10ms tick can be used to produce tasks that executed with multiple of tick. The Z80 kit lab look will show how to use 10ms tick to make a digital timer.

RS232C PORT

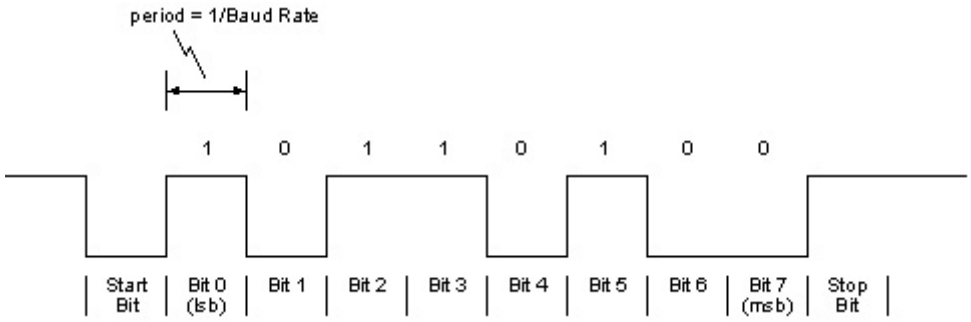
The RS232C port is for serial communication. We can use a cross cable or null MODEM cable to connect between the kit and terminal, or kit #1 to kit #2 for sending or receiving hex file. The connector for both sides are DB9 female. We may build it or buying from computer stores.



For new PC or laptop computer with USB port, we may have the RS232C port by using the USB to RS232 converter.

DATA FRAME for UART COMMUNICATION

Serial data that communicated between kit and terminal is asynchronous format. The Z80 kit has no UART chip, instead it uses software controlled to produce bit rate of 2400 bit/s. The data frame is composed of start bit, 8-data bit and stop bit. For our kit, period = 1/2400 = 417 microseconds.



Since bit period is provided by machine cycle delay. Thus to send/receive serial data correctly, all interrupts must be disabled.

CONNECTING LCD MODULE

JR1 is 20-pin header for connecting the LCD module. The example shows connecting the 16x2 lines text LCD module. R17 is a current limit resistor for back-light. R18 is trimmer POT for contrast adjustment. The LCD module is interfaced to the Z80 bus directly. The command and data registers are located in I/O space having address from 80H to 83H.

Be advised that plugging or removing the LCD module must be done when the kit is powered off.

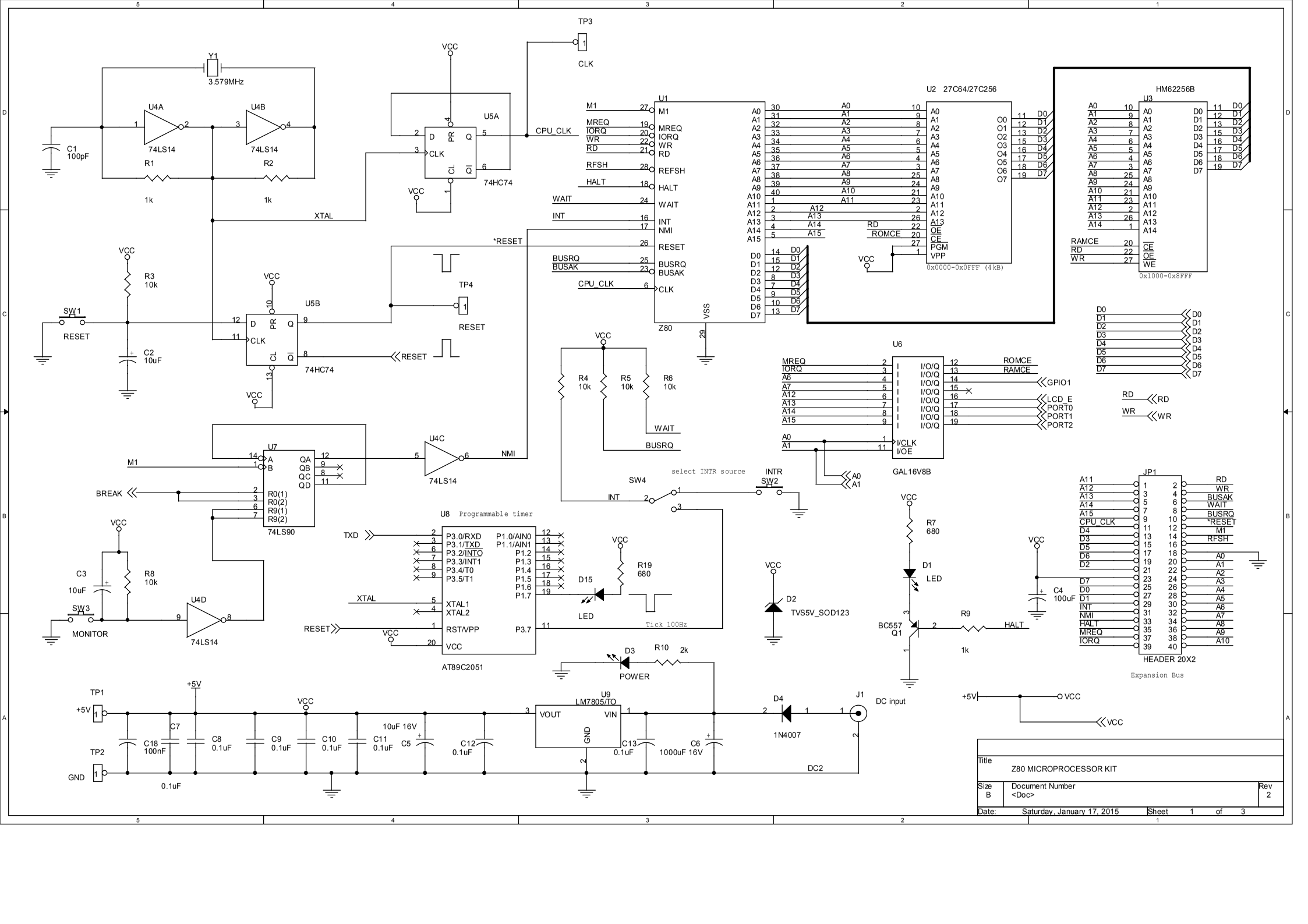
Text LCD module accepts ASCII codes for displaying the message on screen.

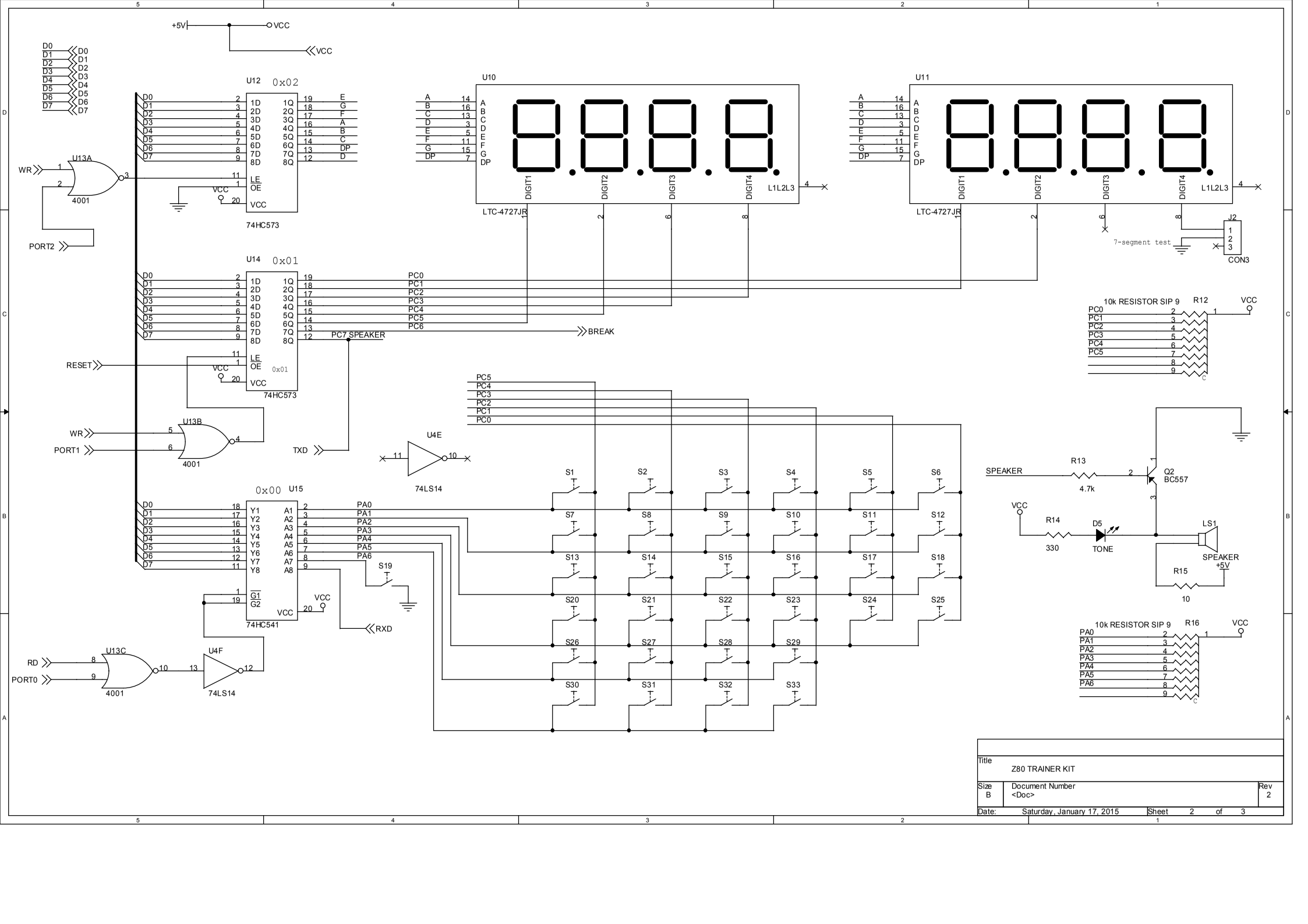
Without settings the LCD by software, no characters will be displayed. The first line will be black line by adjusting the R18 for contrast adjustment.

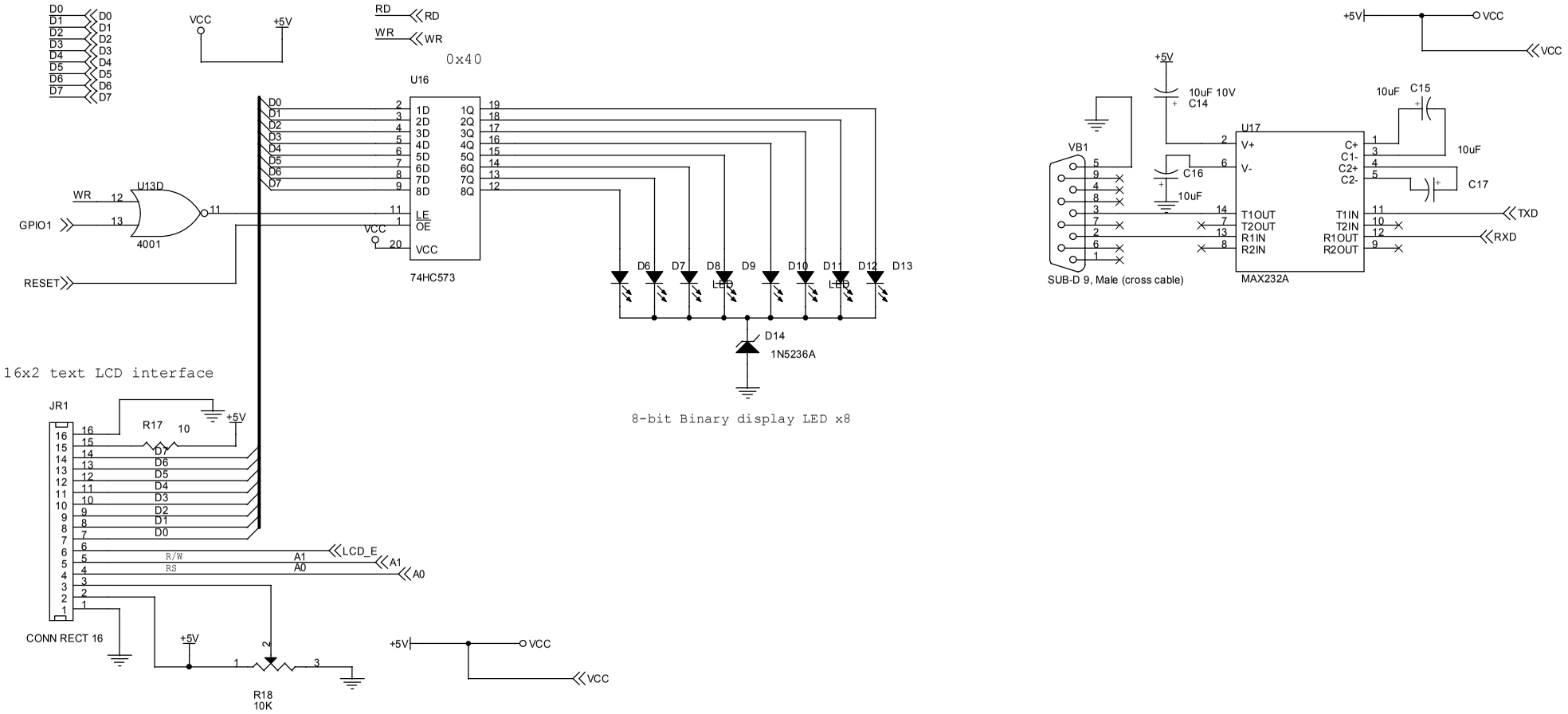
LOGIC PROBE POWER SUPPLY

The kit provides test points TP1(+5V) and TP2(GND) for using the logic probe. Students may learn digital logic signals with logic probe easily. The important signals are RESET (TP4) and CPU clock (TP3). Tick signal, however indicated by D15 LED blinking. Logic probe can test it at P3.7 of the 89C2051 microcontroller directly. Red clip is for +5V and Black clip for GND.

HARDWARE SCHEMATIC, PARTS LIST AND PCB LAYOUT


TP3
CLK
VCC
Y1
3.579MHz
U4A
U4B
74LS14
74LS14
C1
100pF
R1
R2
1k
1k
XTAL
U5A
74HC74
CPU_CLK
U1
Z80
M1
MREQ
IORQ
WR
RD
REFSH
HALT
WAIT
INT
NMI
RESET
BUSRQ
BUSAK
CPU_CLK
*RESET
U2 27C64/27C256
0x0000-0x0FFF (4kB)
HM62256B
U3
0x1000-0x8FFF
RAMCE
ROMCE
R3
10k
SW1
RESET
C2
10uF
U5B
74HC74
TP4
RESET
R4
10k
R5
10k
R6
10k
WAIT
BUSRQ
U6
GAL16V8B
ROMCE
RAMCE
GPIO1
LCD_E
PORT0
PORT1
PORT2
U7
74LS90
BREAK
U4C
74LS14
NMI
SW4
INT
select INTR source
INTR
SW2
U8 Programmable timer
AT89C2051
TXD
XTAL
RESET
R19
680
D15
LED
Tick 100Hz
C3
10uF
R8
10k
SW3
MONITOR
U4D
74LS14
D2
TVS5V_SOD123
R7
680
D1
LED
BC557
Q1
R9
1k
HALT
C4
100uF
JP1
HEADER 20X2
Expansion Bus
D3
POWER
R10
2k
TP1
TP2
+5V
GND
C18
100nF
C7
0.1uF
C8
0.1uF
C9
0.1uF
C10
0.1uF
C11
0.1uF
C5
10uF 16V
C12
0.1uF
U9
LM7805/TO
VOUT
VIN
GND
C13
0.1uF
C6
1000uF 16V
D4
1N4007
J1
DC input
DC2
Title
Z80 MICROPROCESSOR KIT
Size
B
Document Number
<Doc>
Rev
2
Date:
Saturday, January 17, 2015
Sheet
1
of
3


+5V
VCC
D0
D1
D2
D3
D4
D5
D6
D7
U12
0x02
74HC573
U10
U11
LTC-4727JR
DIGIT1
DIGIT2
DIGIT3
DIGIT4
L1L2L3
J2
7-segment test
CON3
U13A
4001
WR
PORT2
U14
0x01
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7 SPEAKER
BREAK
RESET
10k RESISTOR SIP 9
R12
U13B
PORT1
TXD
U4E
74LS14
U15
0x00
74HC541
PA0
PA1
PA2
PA3
PA4
PA5
PA6
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
S14
S15
S16
S17
S18
S19
S20
S21
S22
S23
S24
S25
S26
S27
S28
S29
S30
S31
S32
S33
RXD
U13C
U4F
RD
PORT0
SPEAKER
R13
4.7k
Q2
BC557
R14
330
D5
TONE
LS1
SPEAKER
+5V
R15
10
R16
Title
Z80 TRAINER KIT
Size
B
Document Number
<Doc>
Rev
2
Date:
Saturday, January 17, 2015
Sheet
2
of
3



Title	Z80 TRAINER KIT	
Size B	Document Number <Doc>	Rev 1
Date: Saturday, January 17, 2015	Sheet 3 of 3	

PARTS LIST

Semiconductors

U1 Z80 40-pin DIP microprocessor
U2 27C64 or 27C256 EPROM
U3 HM62256B 32kB SRAM
U4 74LS14 inverter
U5 74HC74 Dual D-type flip-flop
U6 GAL16V8B programmable logic device
U7 74LS90 decade counter
U8 AT89C2051 20-pin microcontroller
U9 LM7805/TO voltage regulator
U11,U10 LTC-4727JR 4-digit 7-segment LED
U12,U14,U16 74HC573 8-bit latch
U13 CD4001 quad NOR gate
U15 74HC541 tri-state buffer
U17 MAX232A rs232 converter
D1,D6,D7,D8,D9,D10,D11, 3mm LED
D12,D13,D15
D2 TVS5V_SOD123 transient voltage suppressor
D4 1N4007 rectifying diode
D14 1N5226A +3.3V zener 500mW
Q2,Q1 BC557 PNP transistor
D3 POWER 3mm LED
D5 TONE 3mm LED

Resistors (all resistors are 1/8W +/-5%)

R1,R2,R9 1k
R3,R4,R5,R6,R8,R18 10K
R7,R19 680 Ohms
R10 2k
R16,R12 10k RESISTOR SIP 9
R13 4.7k
R14 330 Ohms
R17,R15 10 Ohms

Capacitors

C1 100pF ceramic
C2,C3,C15,C16,C17 10uF electrolytic
C4 100uF electrolytic
C5 10uF 16V electrolytic
C6 1000uF 16V electrolytic
C7,C8,C9,C10,C11 0.1uF multi layer
C12,C13 0.1uF multilayer
C14 10uF 10V electrolytic
C18 0.1uF multilayer

Additional parts

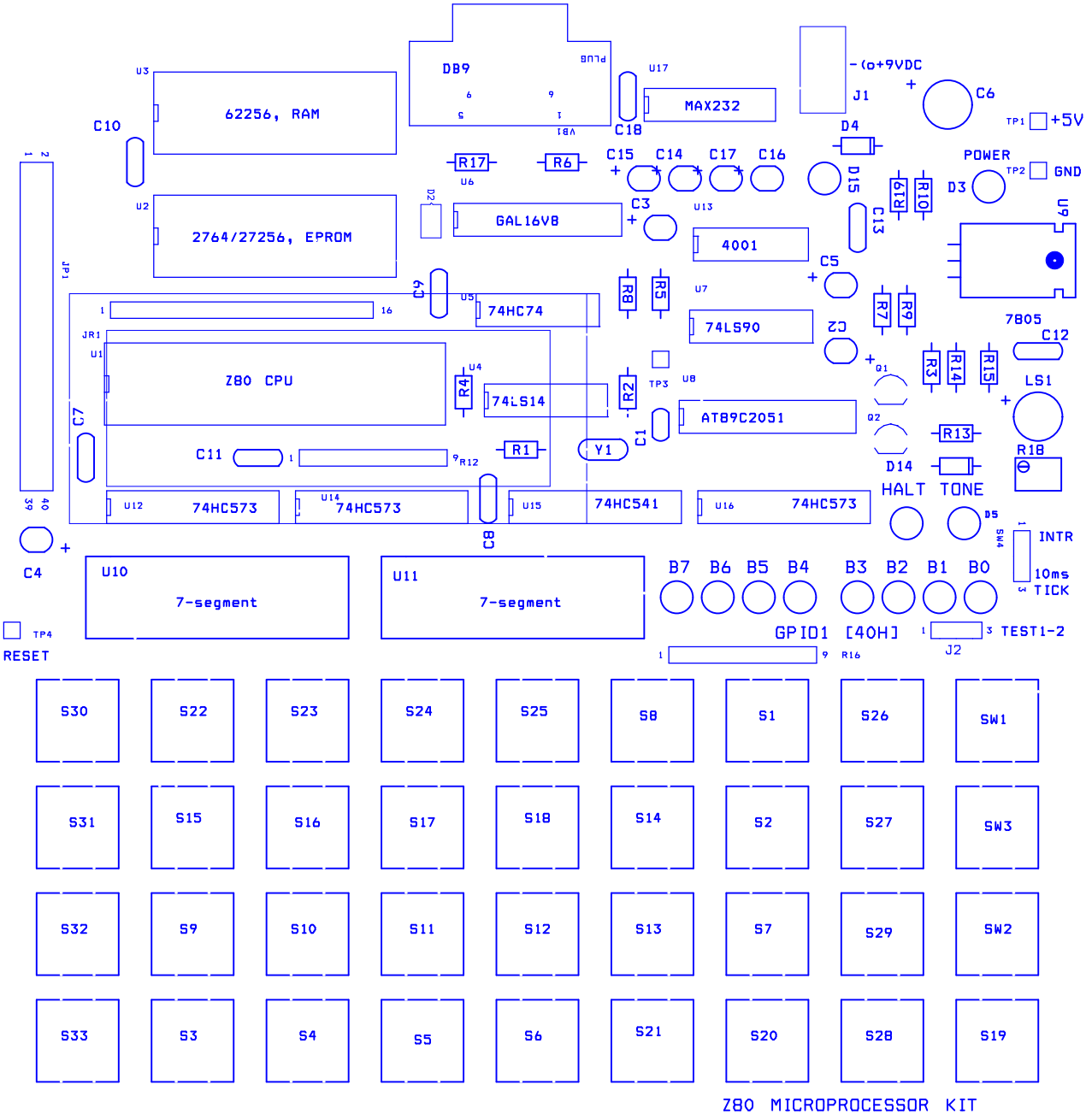
JP1 HEADER 20X2
JR1 CONN RECT 16 pins
J1 DC input JACK
J2 HEADER 3 pins
LS1 8 Ohms SPEAKER

SW1 RESET 12mm tact switch
SW2 INTR 12mm tact switch
SW3 MONITOR 12mm tact switch
SW4 slide switch-SPDT

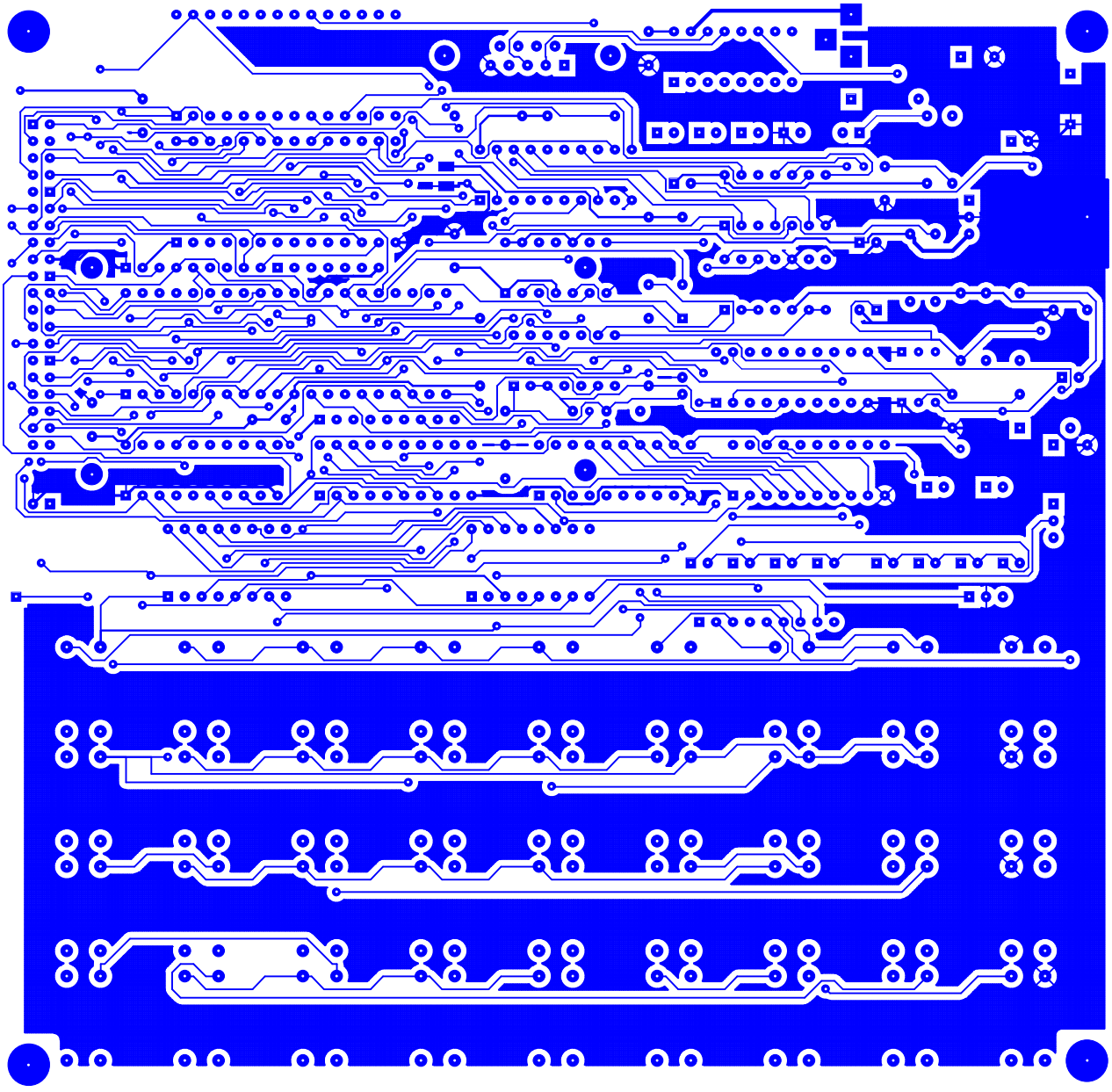
S1,S2,S3,S4,S5,S6,S7,S8, S9,S10,S11,S12,S13,S14, S15,S16,S17,S18,S19,S20, S21,S22,S23,S24,S25,S26, S27,S28,S29,S30,S31,S32, S33 12mm TACT switch

VB1 DB 9, Male connector
Y1 XTAL 3.579MHz
PCB double side plate through hole
LED cover Clear RED color acrylic plastic
Keyboard sticker printable SVG file

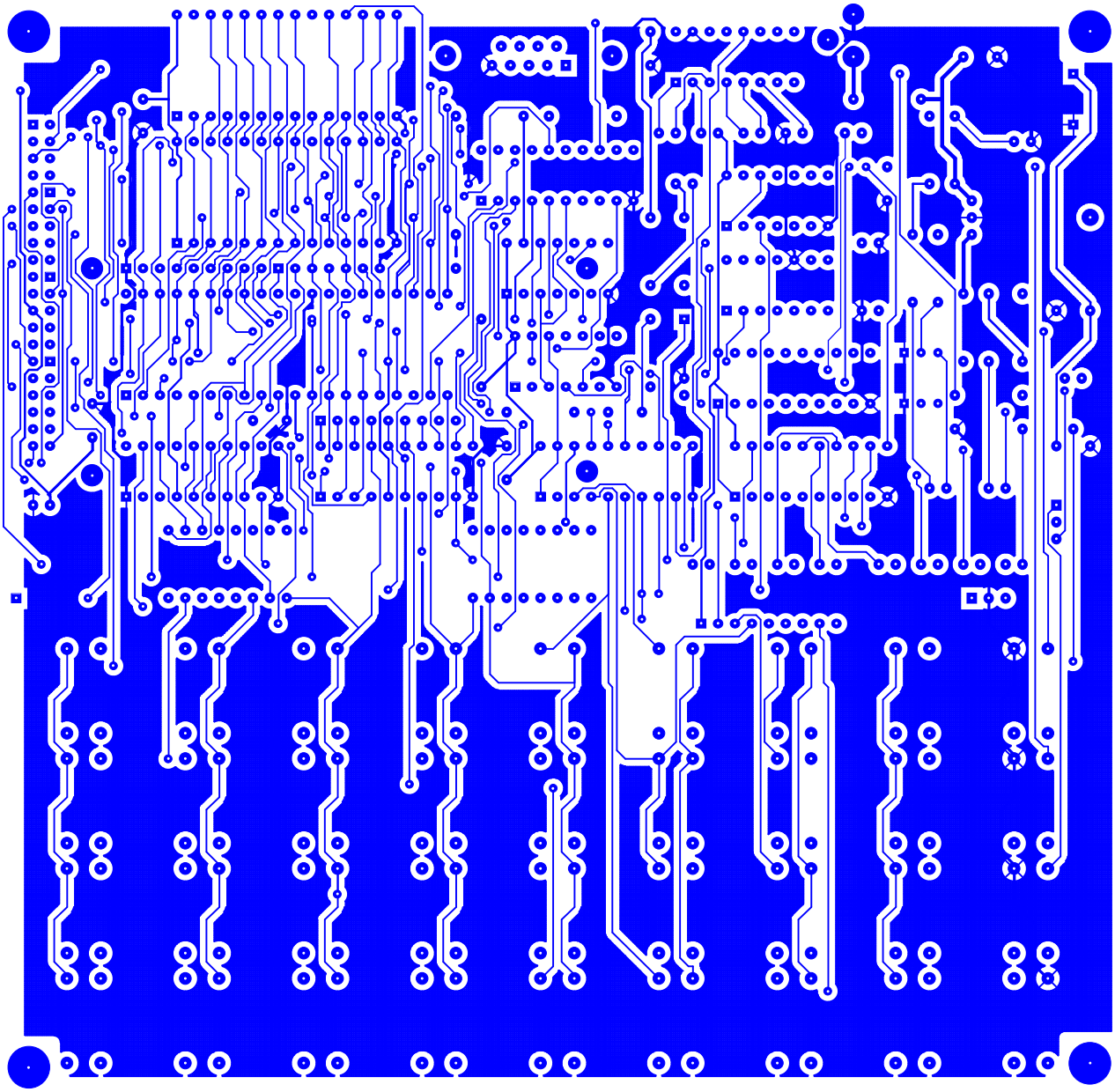
SILK SCREEN TOP LAYER



TOP LAYER



BOTTOM LAYER



NOTE