STC32 USER MANUAL

(Part of manual only, full manual on request)

1、 STC32 DSP Board Specification

1.1 STC32 DSP board specification

(1) T.I. TMS320C32PCM50、50MHz 32-bit floating-point CPU
(2) 32K words SRAM (zero wait states)
(3) 32K or 64K bytes ROM (2 wait states)
(4) 2 channels ADC, 12-bit resolution, 100KHz sampling rate, ±10 V input
(5) 2 channels DAC, 12-bit resolution, 100KHz setting rate, ±10 V output
(6) 2 channels encoder input (A/A-, B/B-, Z/Z-), 24-bit counter each
(7) 1 channel RS-232C UART serial port
(8) 24 bits digital input/output, 16×2 LCD display, 4×4 keyboard
(9) 2 channels PWM output, 16-bit resolution each
(10) one DSPLINK3 expansion slot
(11) system function extendable with additional daughter board

1.2 STC32 system block diagram

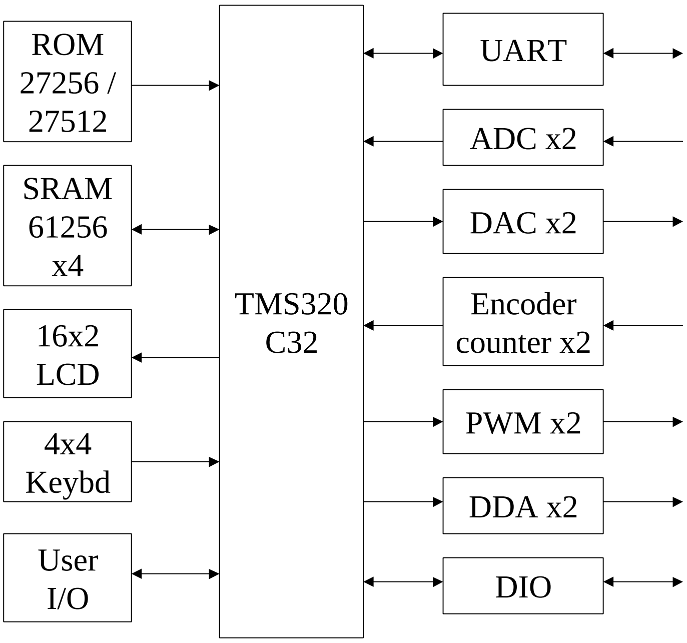


Figure 1：STC32 system block diagram.

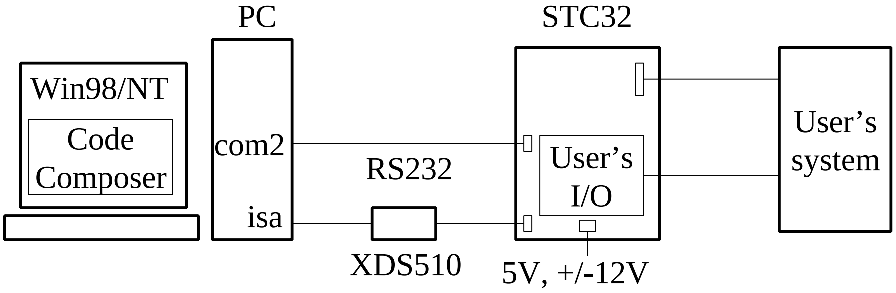
2、 System Requirement

2.1 System requirement

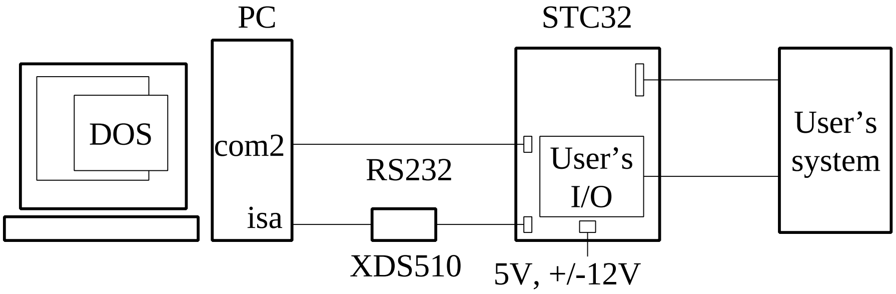
(1) STC32 DSP Board
(2) ROM simulator (for 27256, 27512) or TI XDS510 Emulator
(3) Power supply 5V (1A), +12V (0.1A) and -12V (0.1A)
(4) PC 486 and above with TI TMS320C3x/C4x Optimizing C Compiler for DOS or C3x Code Composer for Windows
(5) RS232 cable (3 wires)

2.2 Environment to develop STC32 Software

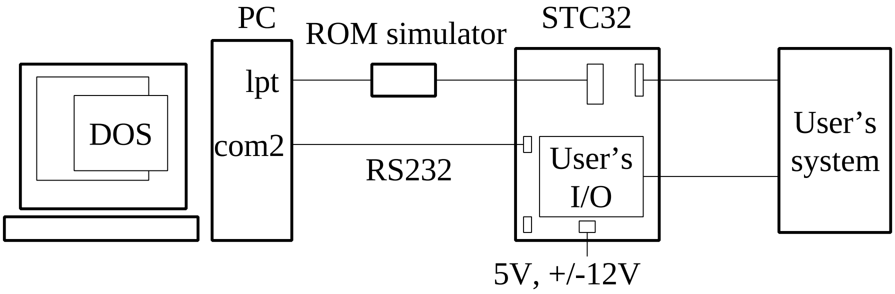
There are four different configurations to develop STC32 DSP board software (Figure 2):
(1) use C3x Code Composer and T.I. XDS510 emulator,
(2) use T.I. XDS510 emulator,
(3) use ROM simulator, and
(4) use EPROM writer.



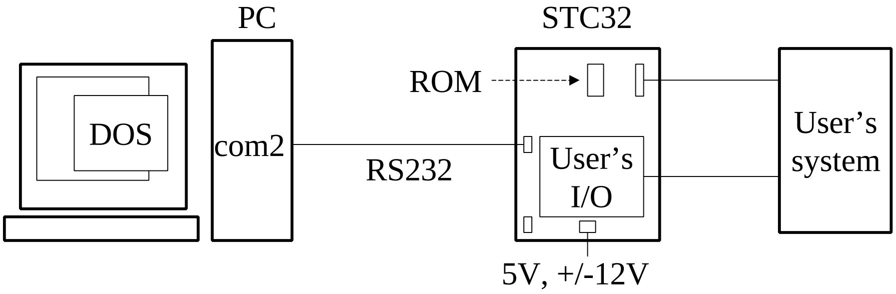
(1) Use C3x Code Composer and T.I. XDS510 Emulator



(2) Use T.I. XDS510 emulator



(3) Use ROM simulator



(4) Use EPROM writer

Figure 2：Configurations for developing STC32 software: (1) use C3x Code Compose and T.I. XDS510 emulator, (2) use T.I. XDS510 emulator, (3) use ROM simulator, and (4) use EPROM writer.

3、 Hardware Description

3.1 TMS320C32 50MHz DSP

STC32 board's CPU is a 32-bit floating-point digital signal processor TMS320C32PCM 50MHz (or C32 in short). The system has 32K×32 bits Static RAM 61256-15 (zero wait state) and 32K× 8 bits EPROM 27256 (two wait states). With C32's boot loader the program stored in the EPROM will be loaded into the system RAM immediate after the system starts. The TMS320C3X(30, 31, 32, and 33) are 32-bit digital signal processors, capable of performing float-point, integer and logical operations. Its architecture allows four levels of pipelining. While an instruction is being executed, the next three instructions are being consequently fetched, decoded, and read. Many instructions can be executed in parallel, such as load with store, multiply with add, and so

on.　The TMS320C32 is a different version of the TMS320C30 processor, with the same execution speed, but with only one primary bus and two serial ports.　The C32 boot loader can load and execute programs received from EPROM, and support 8-bit, 16-bit, and 32-bit data type sizes.

4、　STC32 External Connectors

4.1　STC32's external connectors

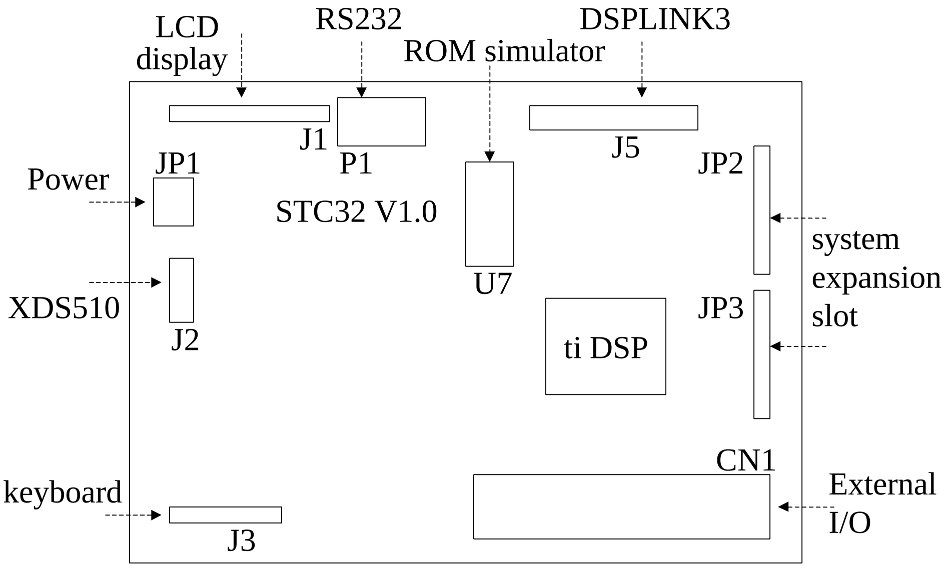


Figure 3：STC3's external connectors.

6、System Basic I/O Functions

The STC32system basic I/O functions for C-Language is saved in the fils STC32IO.H, STC32IO.C and STC32IO.OBJ.

6.1　System initialization and interrupt routines

(1)　void　stc32_init(void) ;

Function：STC32 system initialization routine, must call at beginning.

Input：none

Output：none

(2) void c_int01(void) ;

Function：External interrupt INT 0 service routine, system reserved and not open for the user.

Input：none

Output：none

(3) void c_int07(void) ;

Function：internal timer 0 interrupt service routine, system reserved and not open for the user.

Input：none

Output：none

(4) void c_int08(void) ;

Function：internal timer 1 interrupt service routine, open for the user.

Input：none

Output：none

(5) void settimer1clk(int freq) ;

Function：setting the internal timer 1 interrupt frequency.

Input：interrupt frequency value.

Output：none

(6) void inittime1isr(void) ;

Function：start execution timer 1 ISR c_int08() routine.

Input：none

Output：none

(7) void disabletimer1isr(void) ;

Function：stop execution timer 1 ISR c_int08() routine.

Input：none

Output：none

(8) void enabletimer1isr(void) ;

Function：restart execution timer 1 ISR c_int08() routine.

Input：none

Output：none

6.2 RS232 serial communication

The default serial communication format is as follows：9600 baud rate, 8 bits, 1 start bit, 1 stop bit, and no parity.

(1) void rs232_flush(void) ;
Function：reset RS232 serial communication and clear receiver/transmitter buffers.
Input：none
Output：none

(2) char rs232_in(void) ;
Function：get one character from RS232 receiver buffer.
Input：none
Output：character received, or 0 if no character received.

(3) int rs232_out (char chr) ;
Function：send one character to the transmitter buffer, and then send out by ISR c_int07().
Input：character to be send.
Output：1 if success (i.e. the buffer is not full), and 0 if fail.

(4) int string2PC (char *str) ;
Function：send character string to the transmitter buffer.
Input：character string to be sent.
Output：1 if success, and 0 if fail.

6.4 DAC

(1) int set_dac_cmd(int n, real x) ;
Function：output an analog voltage to the assigned channel.
Input：channel n = 0 or 1; output voltage x, $-10 \le x \le 10$.
Output：none

6.5 Encoder

(1) int get_encoder(int n) ;
Function：read the position count of the assigned motor channel.

Input：channel n = 0 or 1.

Output：position count.

(2) void set_encoder(int n, int offset) ;

Function：set the initial position count of the assigned motor channel.

Input：channel n = 0 or 1.

Output：none

6.6 ADC

(1) void get_adc(int ain[]) ;

Function：read the input voltage of channels 0 and 1.

Input：none

Output：ain[0] and ain[1] are the converted ADC value, respectively.

6.7 PWM（8254 Timer）

(1) void set_pwm(int n, int duty) ;

Function：send PWM output

Input： channel n = 1 or 2；0 < duty < 1000 represents the PWM's duty cycle and 1000 represents 100%.

Output： PWM output (with frequency 8.33 Khz).

6.8 DI/O（8255 Port A bit7--bit4 / Port C bit 3--bit 0）

(1) void set_servo_on(int n) ;

Function：output motor driver enable signal.

Input：channel n = 0 or 1.

Output：none

(2) void set_servo_off(int n);

Function：output motor driver disable signal.

Input：channel n = 0 or 1.

Output：none

(3) int get_positive_limit (int n) ;

Function：sense the status of the assigned positive limit switch.

Input：channel n = 0 or 1.

Output：1 if senses the assigned limit switch, and 0 if not.

(4) int get_negative_limit (int n) ;

Function：sense the status of the assigned negative limit switch.

Input：channel n = 0 or 1.

Output：1 if senses the assigned limit switch, and 0 if not.

6.9 LCD（8255 Port B）

(1) void LCD_flush(void) ;

Function：clear and reset the LCD display.

Input：none

Output：none

(2) int LCD_out(char chr) ;

Function：display one character in the LCD display at the current cursor position.

Input：character to be display.

Output：1if success, and 0 if fail.

(3) int string2LCD(char chr[]) ;

Function：display a character string to the LCD display.

Input：string to be display.

Output：1if success, and 0 if fail.

(4) void LCD_HOME0(void), LCD_HOME1(void);

Function：set the LCD display cursor in the top-left home position.

Input：none

Output：none

6.10 4×4 keyboard（8255 Port A bit3--bit0 / Port C bit7--bit4）

(1) void keybd_flush(void) ;

Function：clear and reset keyboard.

Input：none

Output：none

(2) int kbhit (void) ;

Function：check is any key pressed ?
Input：none
Output：1 if a key has been pressed, and 0 if no key pressed.

(3) int getkbt (void) ;
Function：read the key code of the key pressed.
Input：none
Output：the key code, and 0 if no key pressed.

8、 Applications

8.1 Single-axis table

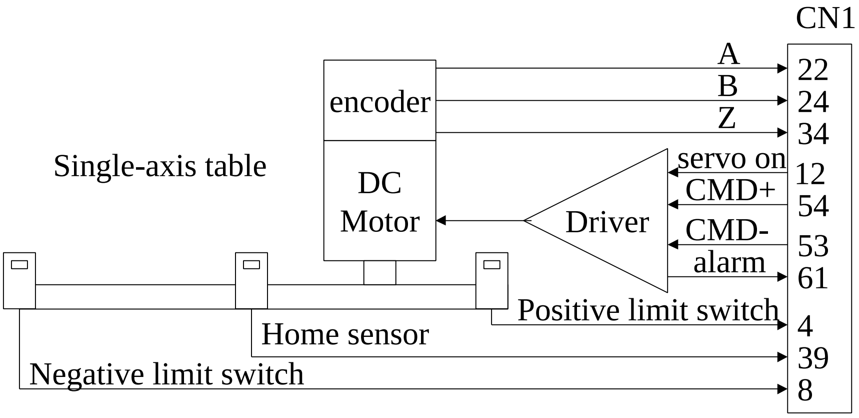


Figure 4：Single-axis motion control example.

8.2 XY table

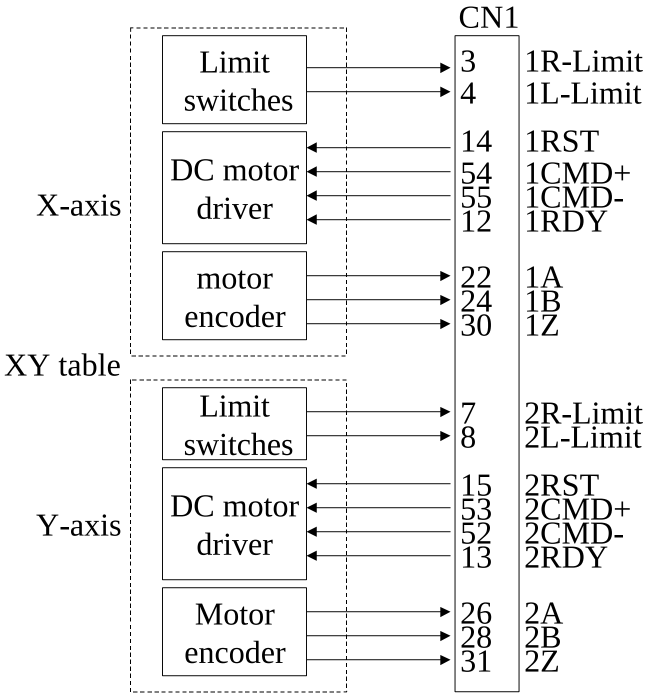


Figure 5：XY table motion control example.

8.3 DC motor

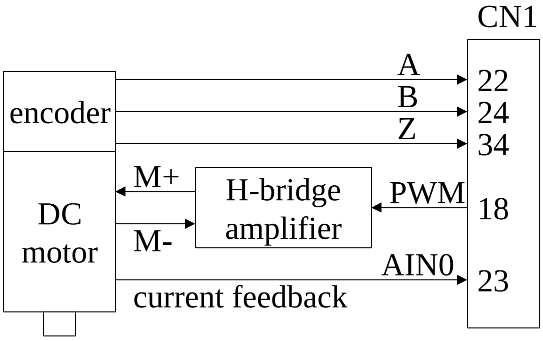


Figure 6：DC motor control example.

8.4 Blushless DC Motor

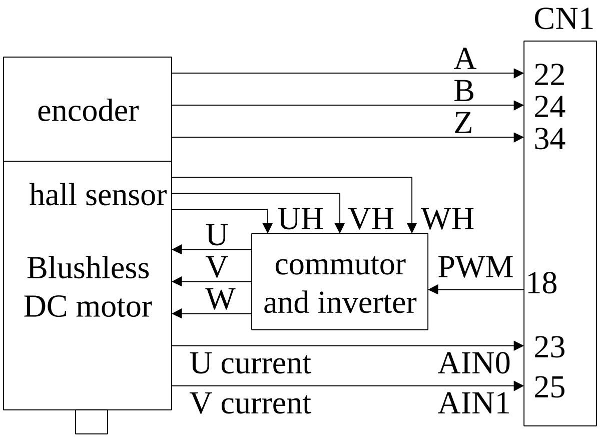


Figure 7：Blushless DC motor control example.

8.5 MR sensor (or sin/cos encoder) position decoder

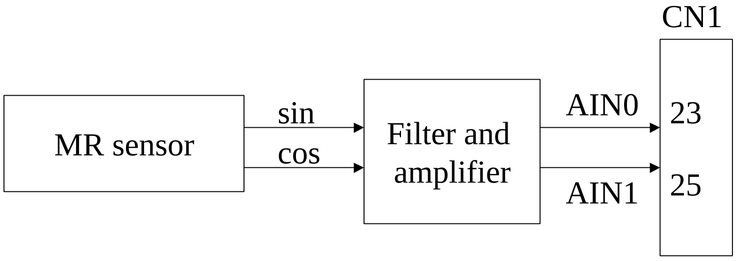
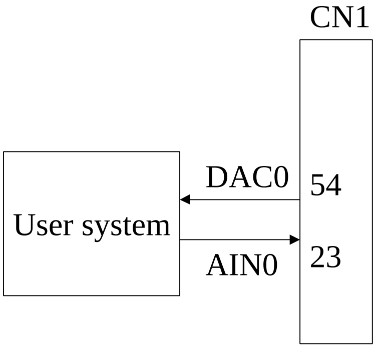


Figure 8：MR sensor position decoder example.

8.6 System frequency response identification



Figuire 9：System frequency response identification example.

(Please inquiry for full user's manual !!)