CS8964 EVB user guide

Rev.B

2018-12-20

DESCRIPTION

CS8964 is a general-purpose microcontroller with extensive peripherals suitable for a wide range of applications. The CPU utilizes an enhanced 1-cycle 8051 core equivalent to ten times the speed of a conventional 12-T 8051. The total on-chip memory includes a 2KB SRAM and 64KB embedded flash memory that can be used as program memory which may also serve as the flash memory. The 8051 core has built-in T0/T1/T2/T3/T4/T5 timers and a 30-bit watchdog timer. Embedded in the CPU core are also a full-duplex UART ports, one I2C master/slave and one I2C pure slave controllers, up to 28 GPIO pins with each GPIO pin configurable as external interrupt and wake up.

The flexibility in clock setting includes an on-chip precision oscillator with the accuracy deviation of +/-2%, or a slow power internal 100K Hz oscillator, and an external 4MHz to 24MHz crystal oscillator, or an ultra-low power precision real time clock (RTC). Unused clock sources can be disabled or used as GPIO pins for system optimization. The clock selections are combined with flexible power management schemes, including PMM, IDLE, and STOP, SLEEP modes to balance CPU speed and power consumption.

On-chip peripherals include one SPI control interface, one I2C master/slave and one I2C pure slave controllers. A Programmable Counter Array (PCA) with 6 channels of Capture/Compare/PWM modules can be used for varieties purposes controlling external devices. There is an additional 3 channel complementary 16-Bit center-aligned PWM for driving various kinds of motor.

Analog peripherals include a high performance 12-Bit Analog to Digital Converter (ADC) with 3.5us conversion time, 4 analog comparators with programmable threshold levels, and a 10-bit Voltage Output Digital to Analog Converter (VDAC).

CS8964 also provides a flexible means of flash programming that supports ISP and IAP. The protection of data loss is implemented in hardware. Access restriction on critical registers and low supply voltage detection allow reliable operations under harsh environments. The code security is reinforced with sophisticated writer commands and ISP commands. The on-chip break point processor also allows easy debugging which can be integrated with ISP.

Intended applications of CS8964 include battery operated systems, home appliances, industrial control, motor control and other embedded applications.

FEATURES

CPU and Memory

- 1-Cycle 8051 CPU core up to 24MHz (16MHz Zero Wait State)
- 16-bit Timers T0/T1/T2/T3/T4 and 24-bit T5
- Programmable 30-bit Watch Dog Timer
- Integrated break point controller and debug port through I2C slave
- One full-duplex UART0 port
- Up to 8 external interrupts shared with GPIO pins
- Power saving modes – PMM, IDLE, STOP, and SLEEP modes
- 256B Internal SRAM and 1792B XSRAM
- 64KB Flash Memory and 128B Information Block
  - Configured to be shared by ISP code, program code, and data flash
  - Code security and data loss protection
  - Endurance: 100K cycles
  - Retention: 10 years @85°C

Clock Sources

- Adjustable Internal oscillator from 8MHz to 16MHz
- Internal low power OSC of 100KHz
- Crystal oscillator 4MHz – 24MHz
- RTC - 32KHz of low power consumption

Digital Peripherals

- 16-bit PCA and 6 channel of CCP modules
  - Capture/Compare/Timer Mode
  - 8-Bit and 16-bit PWM Mode
  - 8-Bit Windowed PWM Mode
- 16-bit PWM Controller
  - 3 channels of center PWM with complementary outputs
  - 2 channels ADC and interrupt triggering
  - Dead time setting
  - Emergency control
- Two I2C Slave Controllers
- One Master/Slave SPI Controller
- One full-duplex LIN-capable EUART2

Analog Peripherals

- 12-Bit monotonic SAR ADC
  - 1 channels with a built-in 7X PGA
  - 4us conversion time
  - 2 S/H channels triggered by PWM16 or software
  - 15 inputs multiplexed with GPIO
  - On-chip temperature sensor
- 4 analog comparators
  - Two 8-bit programmable threshold or external threshold

  - Linked to PWM16 module for emergency
- 10-bit Voltage Output DAC
  - Source resistance < 1KOhm
  - 0 – VDD output range
- Power on reset
- Low voltage detection on supply voltage

Miscellaneous

- Up to 28 GPIO pins
- 2.5V to 5.5V single supply with on-chip regulator
- Low power standby (< 10uA) in SLEEP mode
- Operating temperature -40°C to 85°C
- LQFP-32, QFN-32, and TSSOP-24 package and RoHS compliant

QUICK START

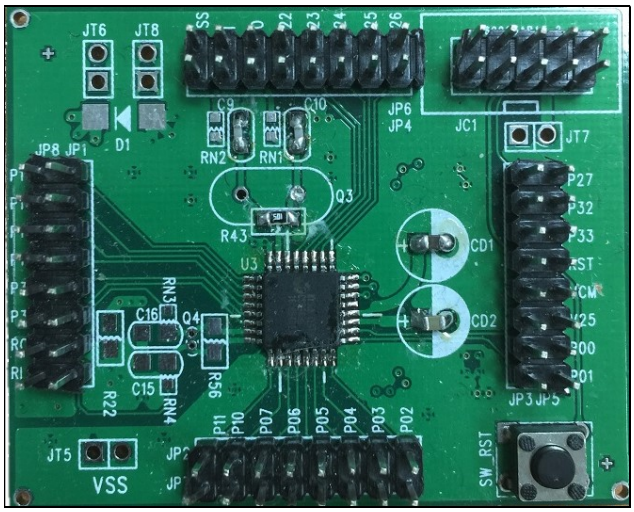


Figure 1: Photo of CS8964 Evaluation Board

RECOMMENDED EQUIPMENT

- 5V power supply

ABSOLUTE MAXIMUM RATINGS

- ≤ 5.5V power supply

Caution: Do not exceed the conditions listed above, otherwise the board will be damaged.

PROCEDURE

The CS8964 evaluation board is fully assembled and tested. Follow the steps listed below to verify board operation.

Caution: Do not turn on the power supply until all connections are completed.

1) Connect the power supply. Pay attention to the supply current.

eZISP BURNING BOARD OPERATION

CS8964 provides a flexible means of flash programming, it can be programmed through Write Mode or Fast Writer Mode of eZISP Burning Board. Use Write Mode and Fast Writer Mode of eZISP Burning Board requires connecting hardware 7 pins (RST, CK, CS, MO, MI, GND, and VDD etc.).



Figure 2: Photo of eZISP Burning Board

ORDERING INFORMATION

Part No.	Temperature Range	Package
CS8964G	-40°C ~ +85°C (Industrial)	LQFP-32, Lead-free
CS8964W		QFN-32, Lead-free
CS8964A		TSSOP-24, Lead-free

Table 1: Ordering Information

For pricing, delivery, and ordering information, please contacts ISSI's analog marketing team at analog@issi.com or (408) 969-6600.

SOFTWARE CONTROL

Needs to install the USB driver and related files (ex: Microsoft Framework and C++ Library) on your PC before using eZISP Burning Board. eZISP software support XP/Win7/Win8/Win10 operation system.

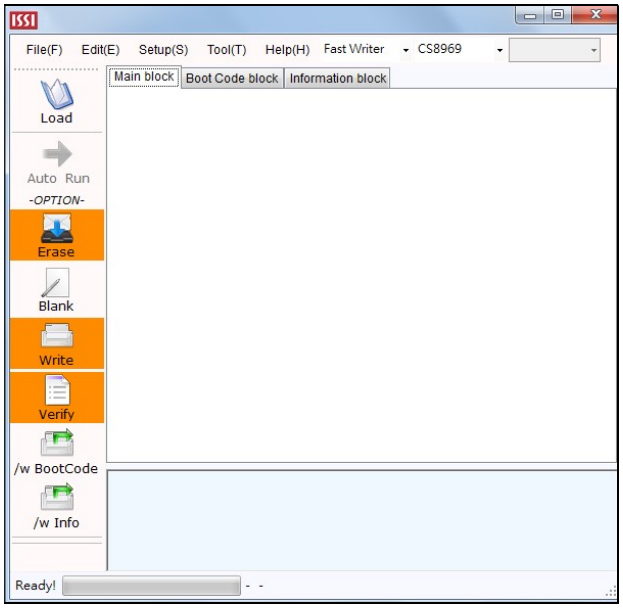


Figure 3: Photo of eZISP software operation interface

eZISP programming operation process is as follows:

1) Connect USB cable between write connector of the eZISP Burning Board and USB port of your PC.
2) Using IDC cable of 10-pins 2x5 Socket-Socket 2.54mm to connect from the writer connector on eZISP Burning Board to the writer connector on the CS8964 Evaluation Board.
3) Execute eZISP software (file name: eZISP-Plus V3.X.X.exe).
4) Choose MCU chip type and writer mode (ex: Writer Mode or Fast Writer Mode).
5) Single click 'Load' button, choose programming code (*.hex) to load.
6) Single click 'Auto Run' button, eZISP software will execute 'Erase', 'Write' and 'Verify' at once, indication message is shown below the windows. The indication message includes programming results and run time.

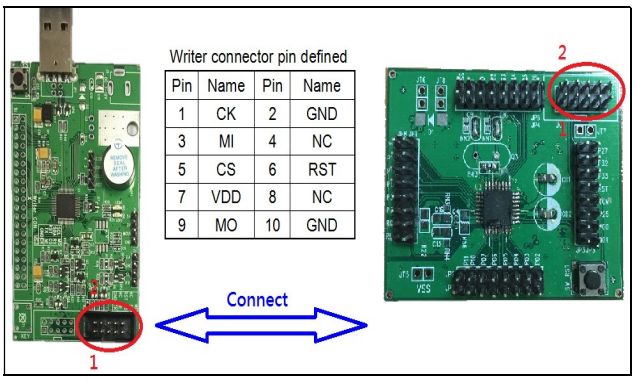


Figure 4: Photo of eZISP Burning Board and CS8964 Evaluation Board connection

The CS8964 Evaluation Board requires a 5V supply voltage.

The steps listed below are an example using the CS8964 for GPIO control.

1) Use the test code in appendix I and compiling test code in Keil C51 development environment (IDE, Keil µVision).
2) Create Hex file of test code in Keil C51 and load hex file of test code in eZISP software for firmware update into CS8964 flash.
3) After firmware update finish, The CS8964 chip will auto reset and executing the program.
4) In this example, you can measure that P02 pin has been toggled on the CS8964 Evaluation Board.

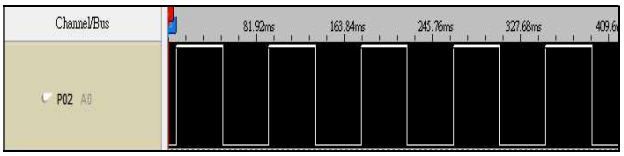


Figure 5: Photo of P02 pin toggle on CS8964 Evaluation Board

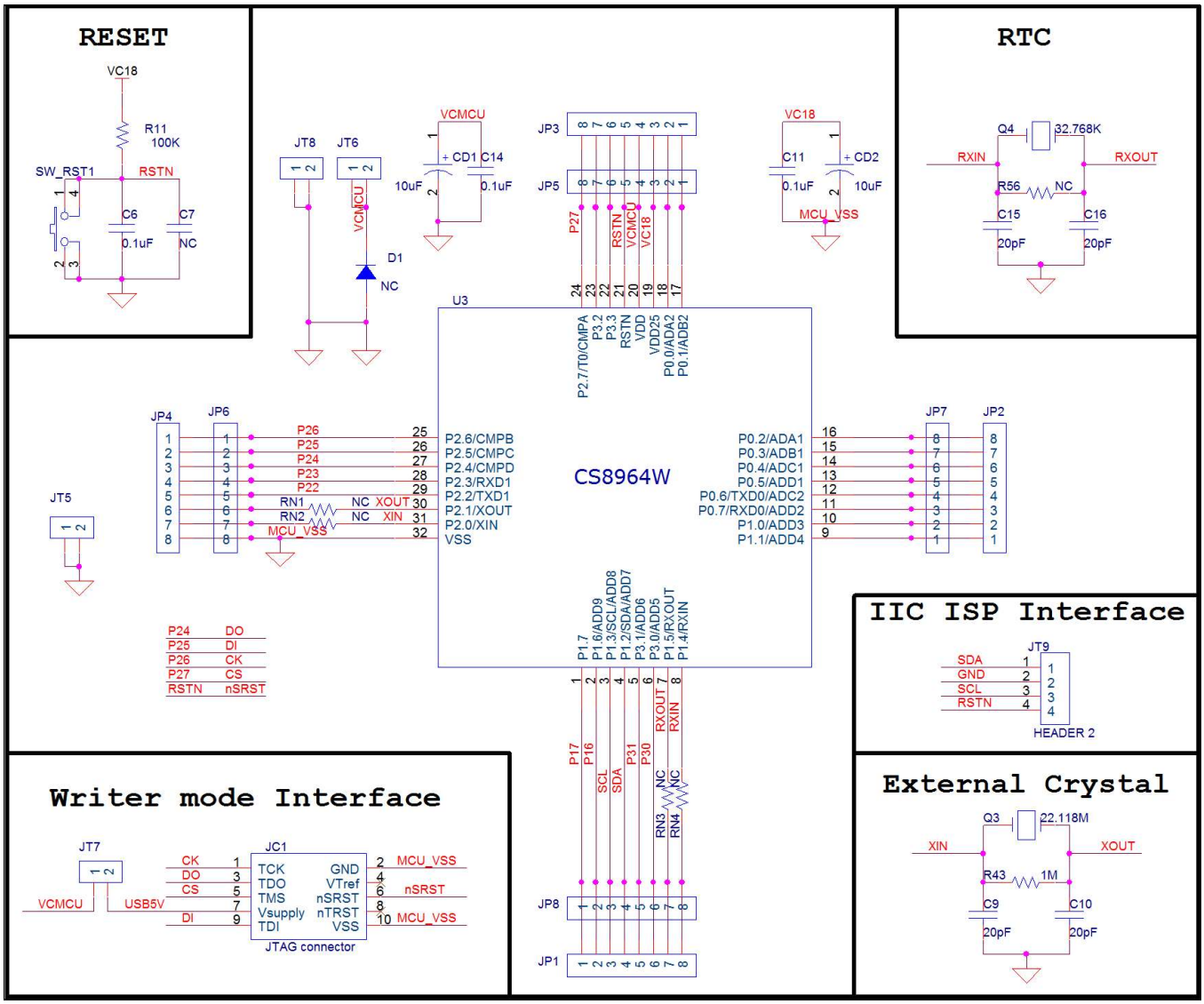


Figure 6: CS8964 Evaluation Board Application Schematic

BILL OF MATERIALS

Name	Symbol	Description	Qty	Supplier	Part No.
MCU	U3	Microcontroller	1	ISSI	CS8964W QFN-32
Button	SW_RST1	Push Button	1		
Crystal	Q3	22.118MHz Crystal	1		
Crystal	Q4	32.768KHz Crystal	1		
Capacitor	CD1,CD2	CAP,10uF,16V,±20%,SMD	2		
Capacitor	C9,C10,C15, C16	CAP,22pF,16V,±20%,SMD	4		
Capacitor	C6,C11,C14	CAP,0.1μF,16V,±20%,SMD	3		
Resistor	R11	RES,100K,1/10W,±5%,SMD	1		
Resistor	R43	RES,1M,1/10W,±5%,SMD	1		

Bill of Materials, refer to Figure 6 above.

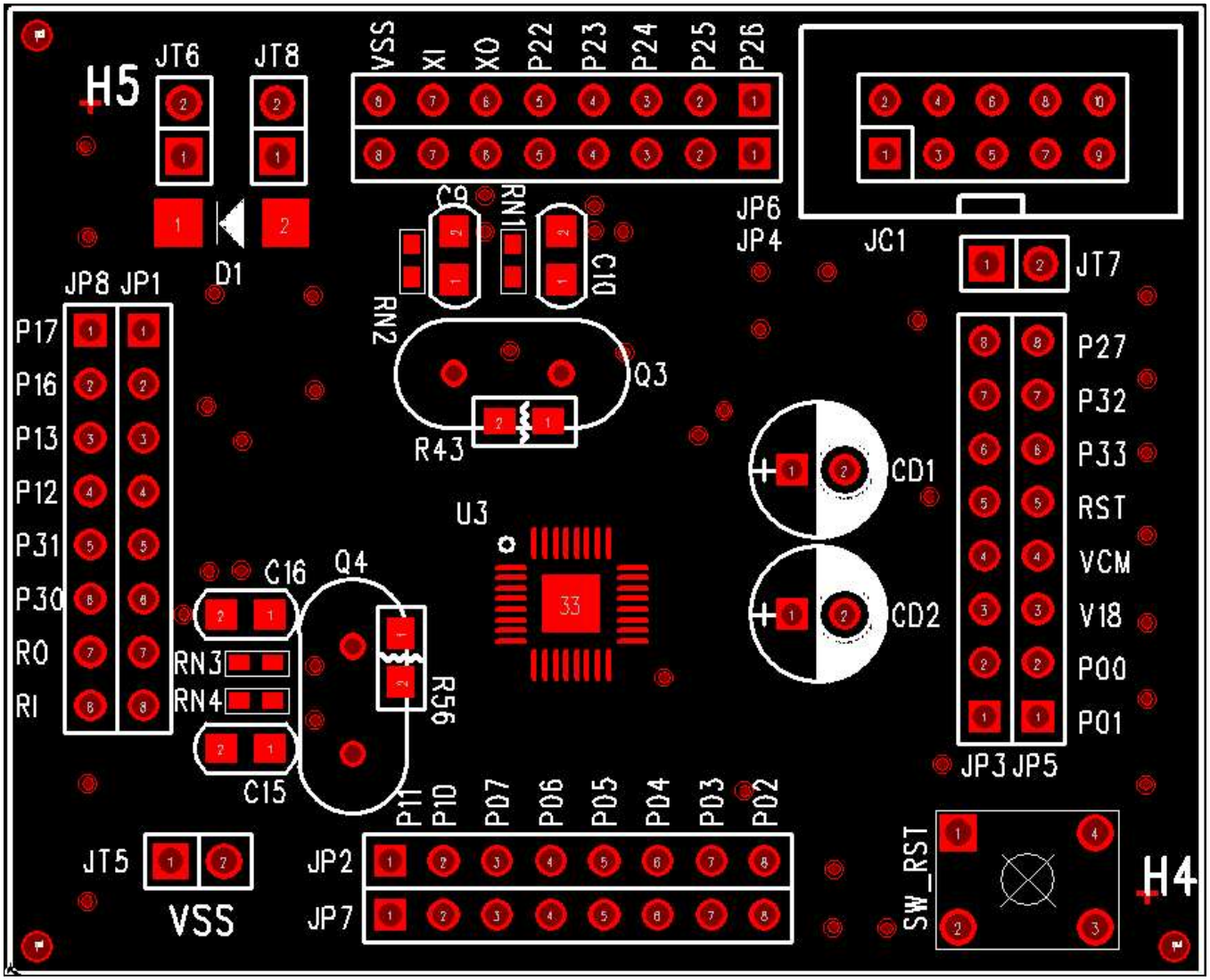


Figure 7: CS8964 Evaluation Board Component Placement Guide - Top Layer

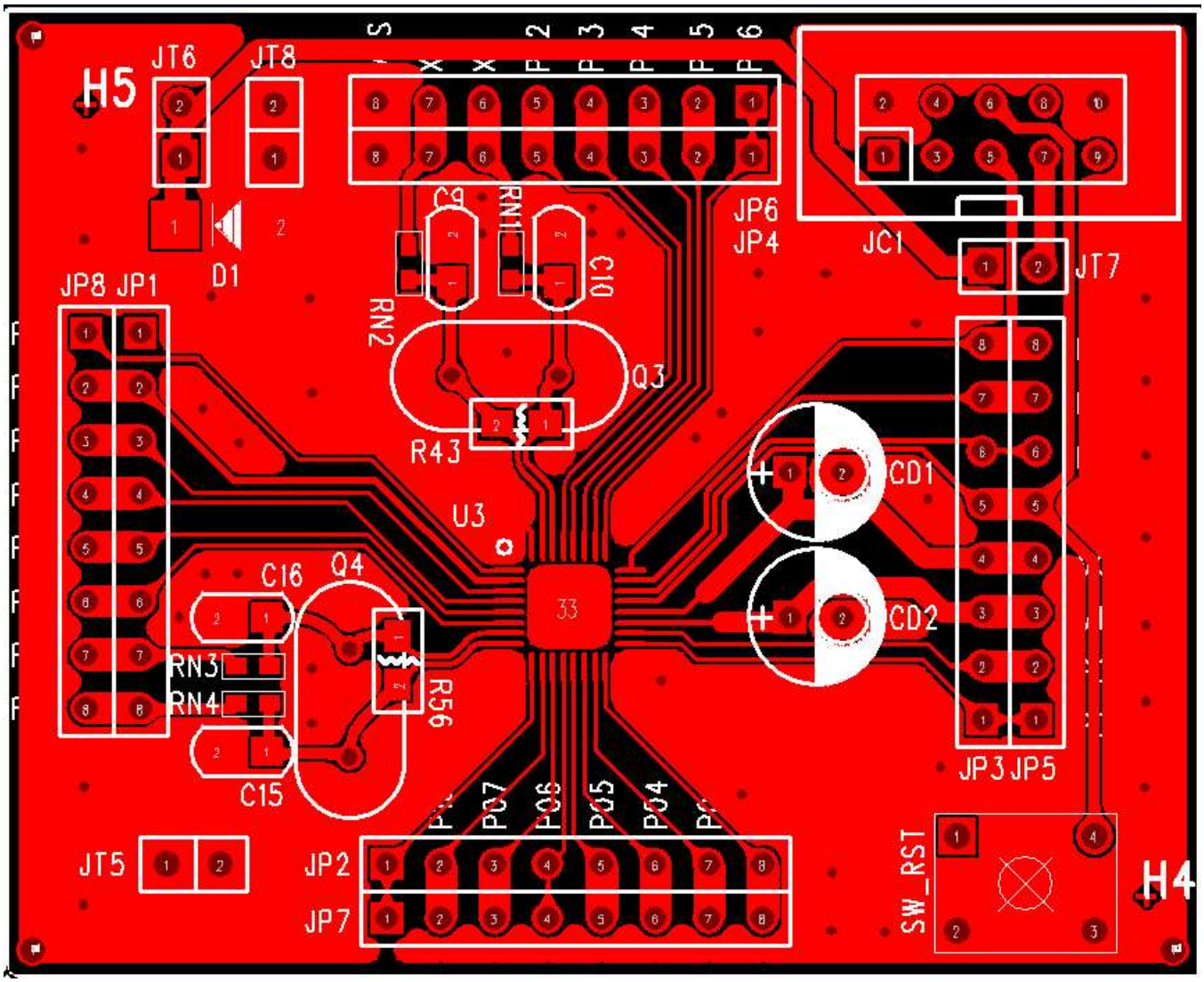


Figure 8: CS8964 Evaluation Board PCB Layout - Top Layer

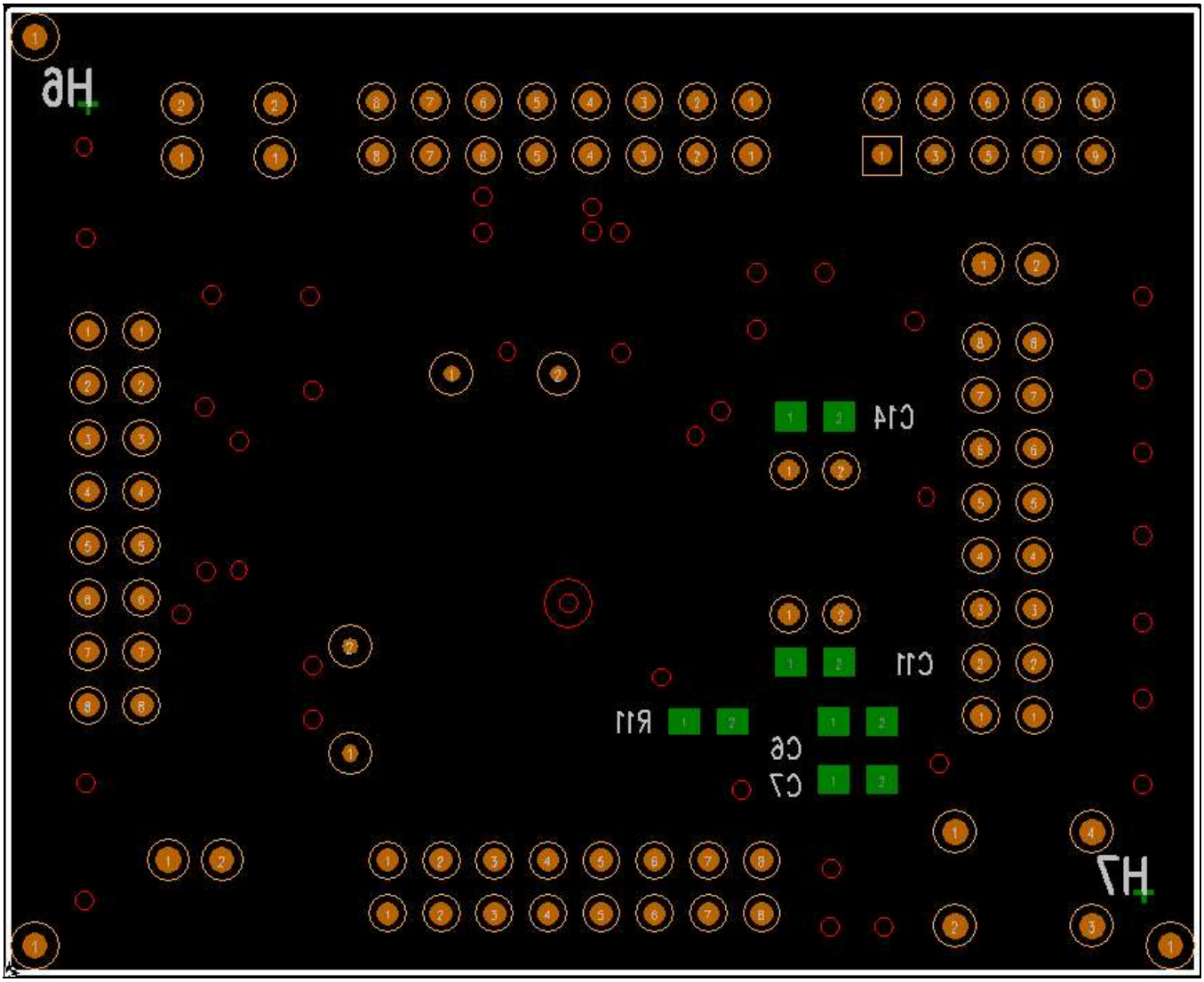


Figure 9: CS8964 Evaluation Board Component Placement Guide - Bottom Layer

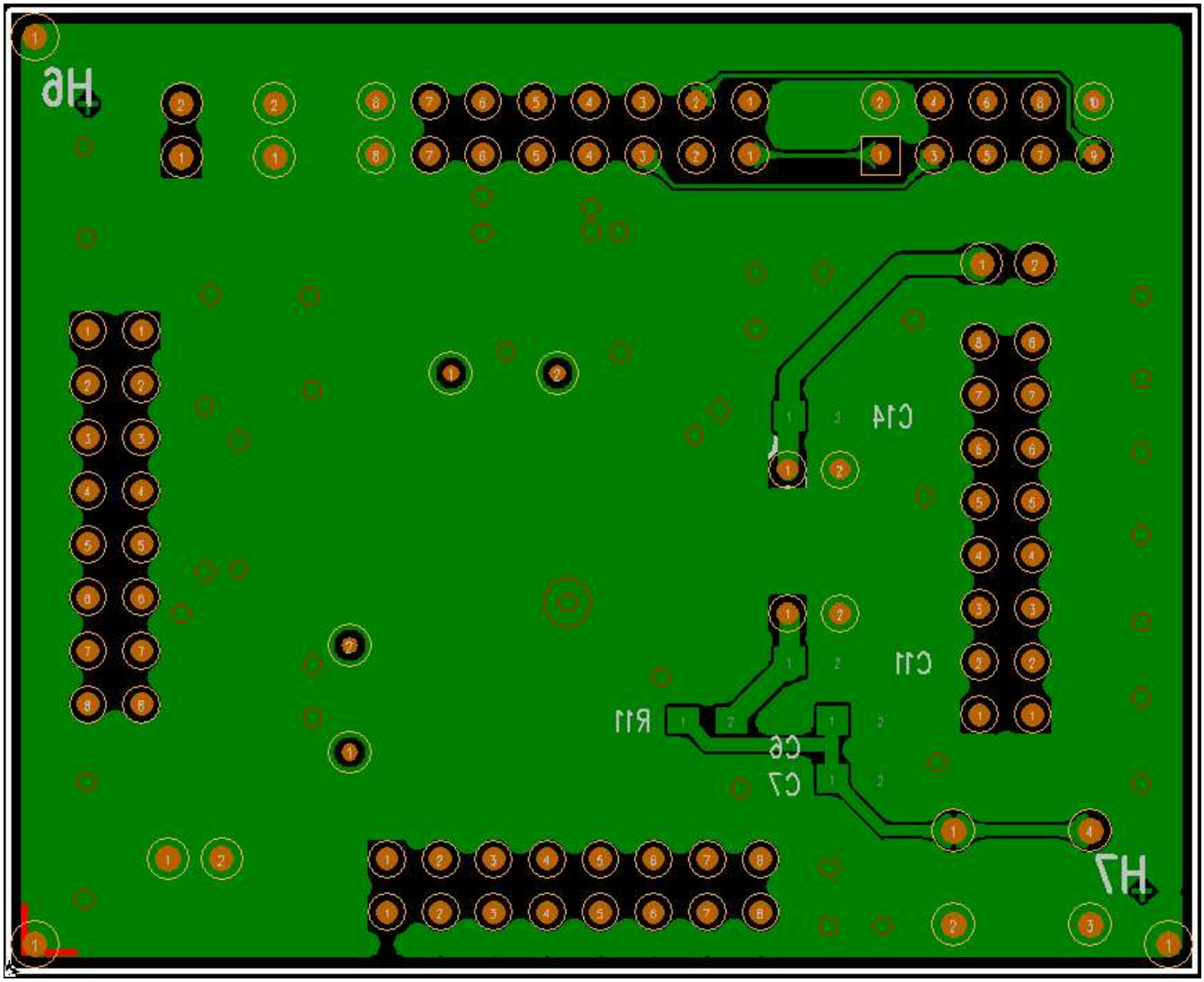


Figure 10: CS8964 Evaluation Board PCB Layout - Bottom Layer

REVISION HISTORY

Revision	Detail Information	Data
A	Initial release	2018.08.27
B	Add a home page.	2018.12.20

Copyright © 2018 Integrated Silicon Solution, Inc. All rights reserved. ISSI reserves the right to make changes to this specification and its products at any time without notice. ISSI assumes no liability arising out of the application or use of any information, products or services described herein. Customers are advised to obtain the latest version of this device specification before relying on any published information and before placing orders for products. Integrated Silicon Solution, Inc. does not recommend the use of any of its products in life support applications where the failure or malfunction of the product can reasonably be expected to cause failure of the life support system or to significantly affect its safety or effectiveness. Products are not authorized for use in such applications unless Integrated Silicon Solution, Inc. receives written assurance to its satisfaction, that:
a.) the risk of injury or damage has been minimized;
b.) the user assume all such risks; and
c.) potential liability of Integrated Silicon Solution, Inc is adequately protected under the circumstances

APPENDIX Ⅰ: CS8964 Test Code - IOSC test

```
/*
Main_IOSC.c

IOSC function for cs8964
Version 1.0
Copyright(c) Integrated Silicon Solution Inc.
All right reserved
*/


#include "CS8964SFR.h"
#include "CS8964xram.h"
#include "HexToBin.h"
#include "Global.h"



void Delay10ms(unsigned char delay)
{
   unsigned char i, j, k;

        for(i=0; i<delay; i++)
                for(j=0; j<100; j++)
                        for(k=0; k<120; k++);
}

void printS(unsigned char p)
{
        ES0 = 0;
        SBUF0 = p;
        while (!TI0);
        TI0 = 0;
        ES0 = 1;
}

void Initial_REGTRM(unsigned char regtrm)                       //Initial REGTRM
{
        TB = 0xAA;
        TB = 0x55;
        REGTRM = regtrm;
        TB = 0x00;
}

void Initial_IOSC(unsigned char ITRM, unsigned char VTRM)       //Initial IOSC
{
        TB = 0xAA;
        TB = 0x55;
        IOSCITRM = ITRM;
        TB = 0x00;

        Delay10ms(1);

        TB = 0xAA;
        TB = 0x55;
        IOSCVTRM = VTRM;
        TB = 0x00;
}

unsigned char IFB_Read_1Byte(unsigned char ADD)                 //IFB Read byte
{
    unsigned char IFB_DAT;

    TB = 0xAA;
    TB = 0x55;
    FLSHADH = 0x00;
    FLSHADL = ADD;
    FLSHCMD = IFB_ByteRead;                                     //IFB read enable
    TB = 0x00;

    TB = 0xAA;
    TB = 0x55;
    IFB_DAT = FLSHDAT;
```

```
    TB = 0x00;

    return IFB_DAT;
}

void Initial_UART0(unsigned long BR, unsigned long XTAL)
{
        IOCFGP0_6 = b00000110;                          //CMOS output(TXD0)
        IOCFGP0_7 = b10100000;                          //input only(RXD0)
        MFCFGP0_6 = b00010000;                          //UART0 TXD0    ENABLE
        MFCFGP0_7 = b00010000;                          //UART0 RXD0    ENABLE

        CKCON |= 0x10;                                  //timer1 divid by 4
        SCON0 = 0x50;                                   //uart0 mode1
        TMOD = 0x20;                                    //time1 mode2(8bit auto reload mode)
        PCON = PCON|0x80;                               //smod0 = 1
        TL1 = TH1 = 256-(XTAL/64/BR);                   // CKCON |= 0x10
        TR1 = 1;
}

void test_io_initial()
{
        IOCFGP0_2 = AllSeting;
        MFCFGP0_2 = _GPIOEN_;

        P0_2 = 0x00;
}

void main (void)
{
        Initial_REGTRM(IFB_Read_1Byte(0x20));                           //Initial IOSC:
        Initial_IOSC(IFB_Read_1Byte(0x21), IFB_Read_1Byte(0x22));       //Default IOSC: 16MHz
        Initial_UART0(19200,16000000);                                  //initial UART0
        test_io_initial();                                              //initial test IO

        while(1)
        {
                printS(0x56);
                P0_2 = ~P0_2;                                           //P02 toggle
                Delay10ms(100);
        }
}
```