IS31CS8968A EVB user guide

Rev.C

2019-07-09

DESCRIPTION

IS31CS8968A is a general-purpose microcontroller with extensive peripherals suitable for a wide range of applications. The CPU is based on an enhanced 1-cycle 8051 core equivalent to ten times the speed of a conventional 12-T 8051. The total on-chip memory include 6KB SRAM and a total of 128KB embedded flash memory that can be used as program memory and portion of this can be used as data flash. The 8051 core has built-in T0/T1/T2 timers, 24-bit T3 timer, 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 42 GPIO pins.

The flexibility in clock setting includes an on-chip precision oscillator with the accuracy deviation of +/-2%, or a slow power internal 100KHz 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.

Other on-chip peripherals include one SPI control interface, one I2C master/slave and one I2C pure slave controllers, as well as a Programmable Counter Array (PCA) with 6 channels of Capture/Compare/PWM modules.

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

A built-in programmable CEC (Consumer Electronics Control) Controller allows users to control all of the various audiovisual products in a user's environment easily.

IS31CS8968A also includes a CAN bus controller compliant to CAN2.0A/B and CAN OPEN specifications. It supports standard 11-bit and 29-bit identification modes. CAN receive path contains 4 acceptance filter for ID filtering and a parameterized FIFO (shared with the CPU SRAM storage) for received message buffering. The transmit path includes a dedicated transmit message buffer and also three dispatch message buffers. The dispatch messages can be used for periodic messaging without CPU intervention with programmable durations and priorities. The CAN controller supports normal operation mode with auto-recovery that compliant to CAN bus error handling standard. It also supports listen-only mode, and test mode which allows external loop-back test of the CAN functions.

IS31CS8968A also provides a flexible means of flash programming that supports ISP and IAP. The protections of loss of Flash contents are implemented in hardware. There is also access restriction on critical registers and low supply voltage detection that allows IS31CS8968A reliable operations under harsh environment. The code security is extremely secure based on sophisticated writer commands and ISP commands. The on-chip break point processor also allows easy debug environments that can be integrated with ISP.

Intended application fields of IS31CS8968A include LCD/PDP TV, LCD Monitor, automotive AV system, home appliance, and other embedded applications.

FEATURES

CPU and Memory

- 1-Cycle 8051 CPU core up to 24MHz operation frequency (One Wait State); 16MHz (Zero Wait State)
- 16-bit Timers T0/T1/T2 and 24-bit Timer T3
- Programmable 30-bit Watch Dog Timer
- Integrated break point controller for software debugger
- Software debugging port through I2C slave
- One full-duplex UART0 port
- Up to 10 external interrupts shared with GPIO pins
- Power saving mode – PMM, IDLE, STOP, and SLEEP modes
- 256B Internal SRAM and 5888B XSRAM
- 128KB Flash Memory and 256B Information Block
  - Configured to be shared by ISP code, program code, and data flash
  - Code security and content loss protection
  - Endurance: 100K cycles

Clock Sources

- Internal oscillator at 16MHz of +/- 2% accuracy
- Internal low power OSC of 100KHz
- Crystal oscillator 4MHz – 24MHz
- RTC - 32KHz of low power consumption

Digital Peripherals

- 16-bit PCA and 6 channel CCP modules
  - Capture/Compare/Timer Mode
  - 8-Bit and 16-bit PWM Mode
  - 8-Bit Windowed PWM Mode

- One 16-bit PWM output with programmable base frequency and duty cycles
- Two I2C Slave Controllers
- One Master/Slave SPI Controller
- One full-duplex LIN-capable EUART
- CEC Controller
- CAN Controller

Analog Peripherals

- 12-bit monotonic SAR ADC
  - 4us conversion time
  - 4 intrinsic time-multiplexed channels and dedicated result registers. 2 of which have sample and hold.
  - 10 inputs multiplexed with GPIO
  - On-chip temperature sensor
- 4 analog comparators
  - Two 8-bit programmable threshold or external threshold
- 10-bit Voltage Output DAC
  - 2mA full scale
  - Configurable Sink/Source
- Power on reset
- Low voltage detection on supply voltage

Miscellaneous

- Up to 42 GPIO pins
- 2.5V to 5.5V single supply with on-chip regulator or 1.8V direct single supply
- Low power standby (< 20uA) in SLEEP mode
- Operating temperature -40°C – 85°C
- LQFP-48 package and RoHS compliant

QUICK START

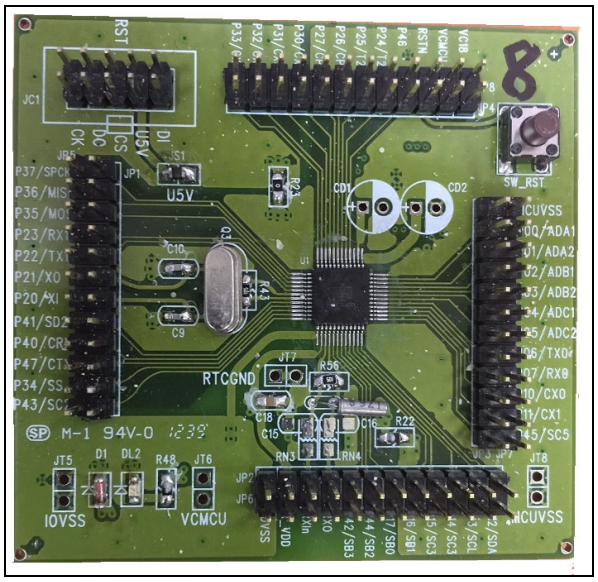

Figure 1: Photo of IS31CS8968A Evaluation Board

RECOMMENDED EQUIPMENT

- 5V power supply

ABSOLUTE MAXIMUM RATINGS

- ≤ 5.5V power supply

PROCEDURE

The IS31CS8968A 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

IS31CS8968A 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
IS31CS8968AG-LQLS2-EB	-40°C ~ +85°C (Industrial)	LQFP-48, 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.

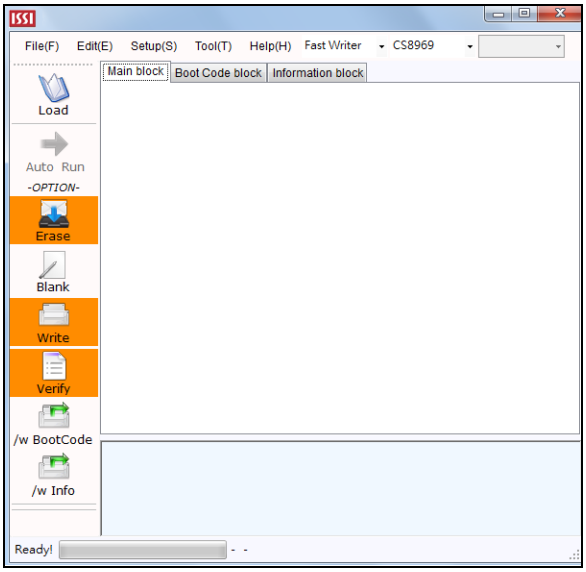

Figure 3: Photo of eZISP software operation interface

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 1.27mm to connect from the writer connector on eZISP Burning Board to the writer connector on the IS31CS8968A 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.

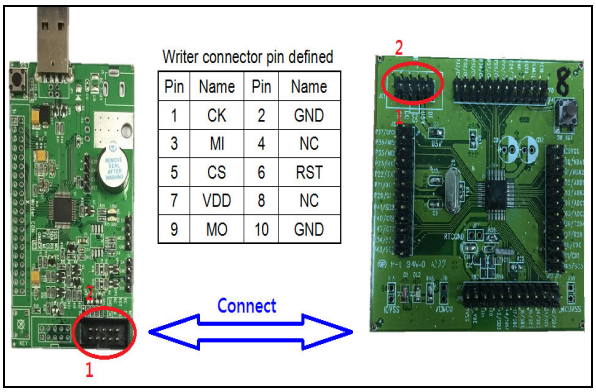

Figure 4: Photo of eZISP Burning Board and IS31CS8968A Evaluation Board connection

The IS31CS8968A Evaluation Board requires a 5V supply voltage.

The steps listed below are an example using the IS31CS8968A 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 IS31CS8968A flash.
3) After firmware update finish, The IS31CS8968A chip will auto reset and executing the program.
4) In this example, you can measure that P02 pin has been toggled on the IS31CS8968A Evaluation Board.

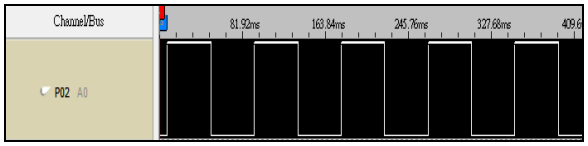

Figure 5: Photo of P02 pin toggle on IS31CS8968A Evaluation Board

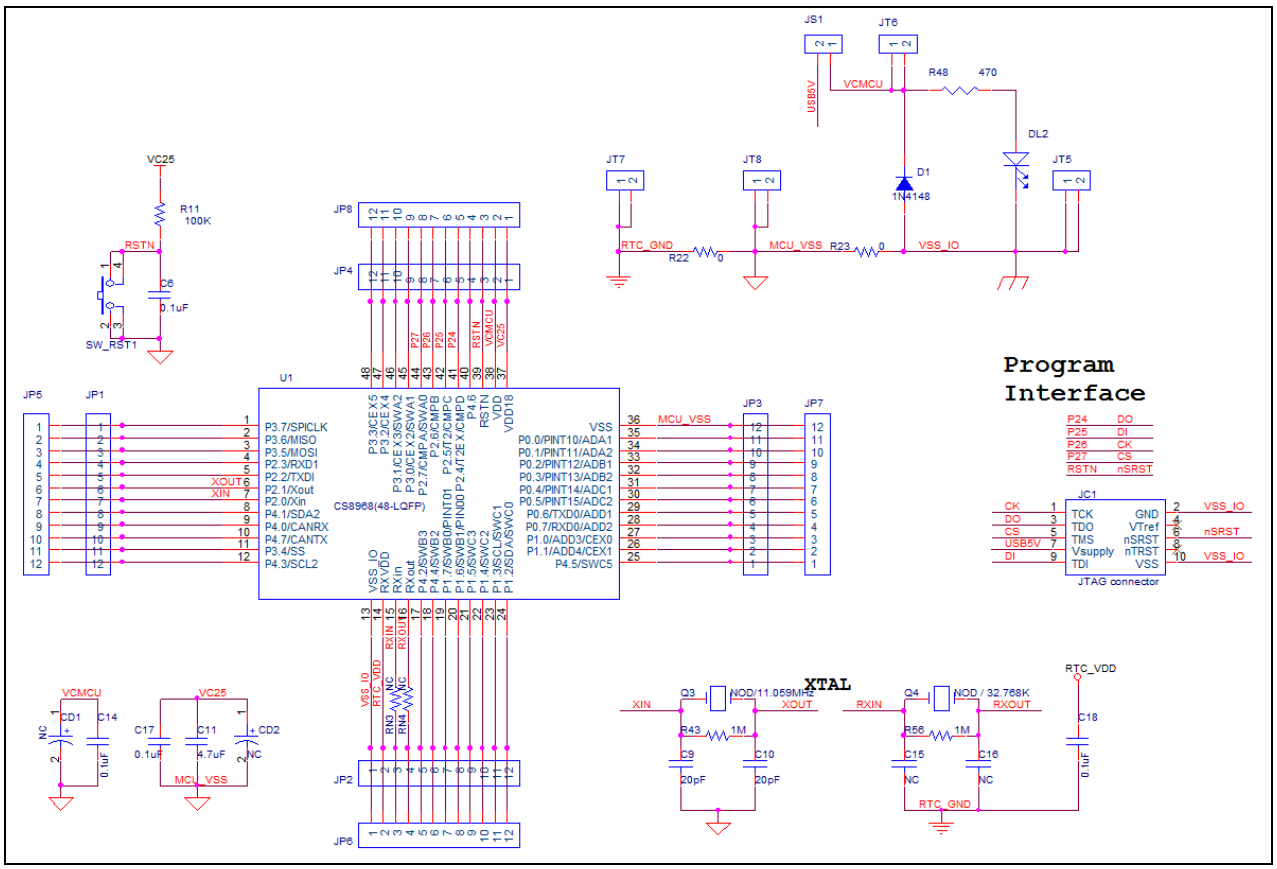

Figure 6: IS31CS8968A Evaluation Board Application Schematic

BILL OF MATERIALS

Name	Symbol	Description	Qty	Supplier	Part No.
MCU	U1	Microcontroller	1	ISSI	IS31CS8968AG-LQLS2
Button	SW_RST1	Push Button	1		
Crystal	Q3	11.059MHz Crystal	1		
Crystal	Q4	32.768KHz Crystal	1		
Diode	D1	1N4148,SMD	1		
LED	DL2	LED, SMD	1		
Capacitor	C11	CAP,4.7uF,16V,±20%,SMD	1		
Capacitor	C9,C10	CAP,20pF,16V,±20%,SMD	2		
Capacitor	C6,C14,C17, C18	CAP,0.1μF,16V,±20%,SMD	4		
Resistor	R22,R23	RES,0R,1/10W,±5%,SMD	2		
Resistor	R48	RES,470R,1/10W,±5%,SMD	1		
Resistor	R11	RES,100K,1/10W,±5%,SMD	1		
Resistor	R43,R56	RES,1M,1/10W,±5%,SMD	2		

Bill of Materials, refer to Figure 6 above.

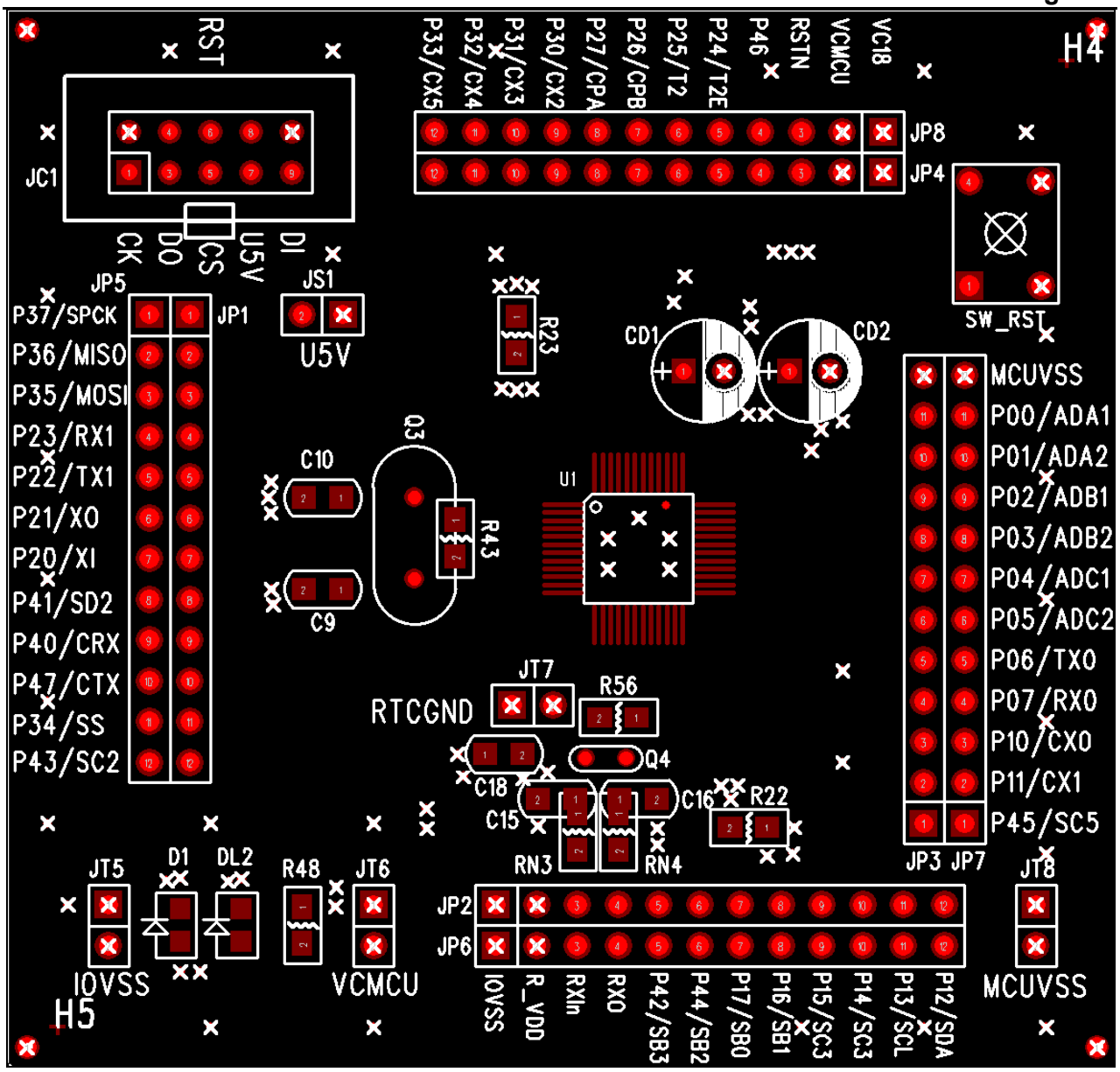

Figure 7: IS31CS8968A Evaluation Board Component Placement Guide - Top Layer

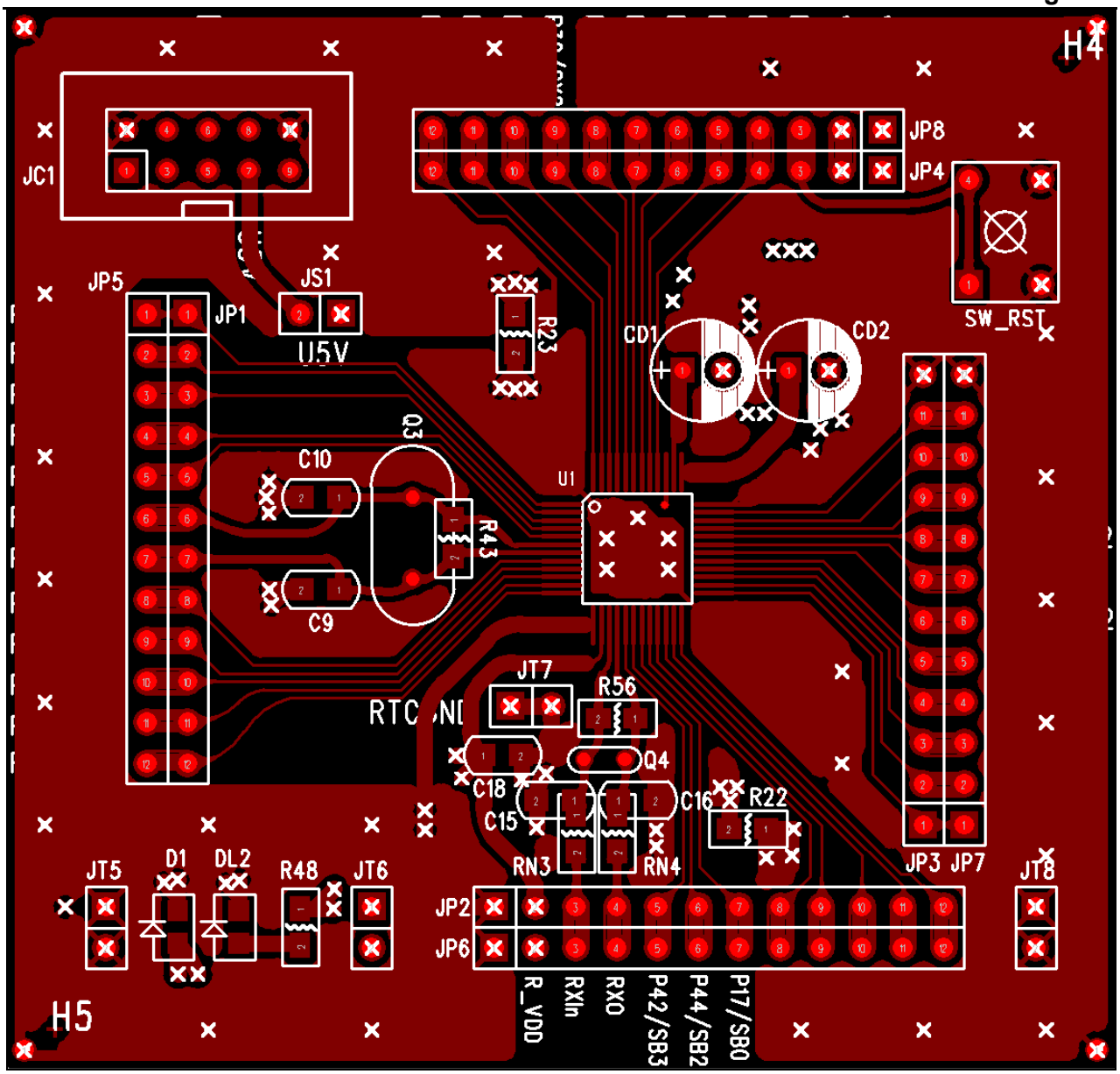

Figure 8: IS31CS8968A Evaluation Board PCB Layout - Top Layer

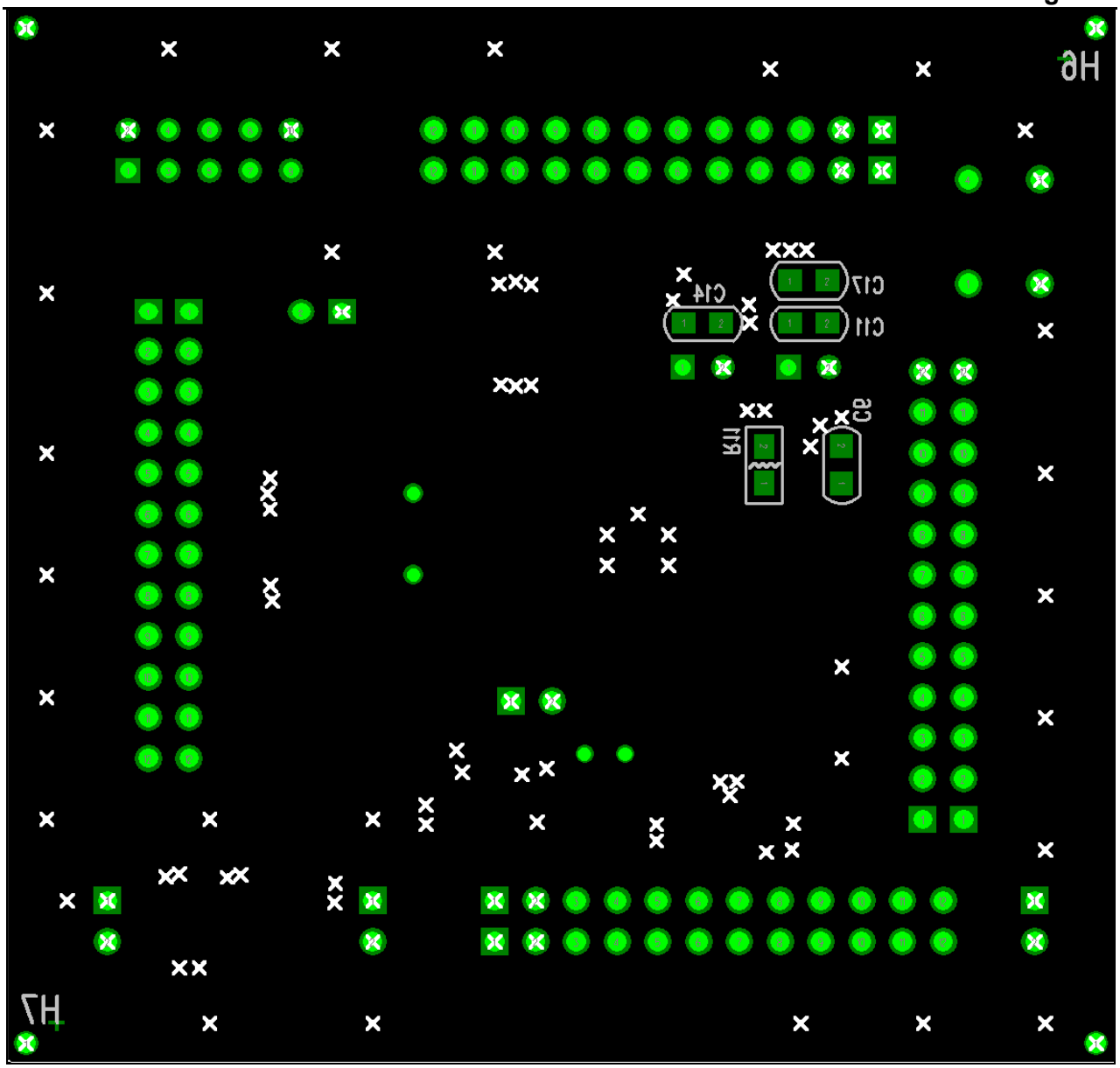

Figure 9: IS31CS8968A Evaluation Board Component Placement Guide - Bottom Layer

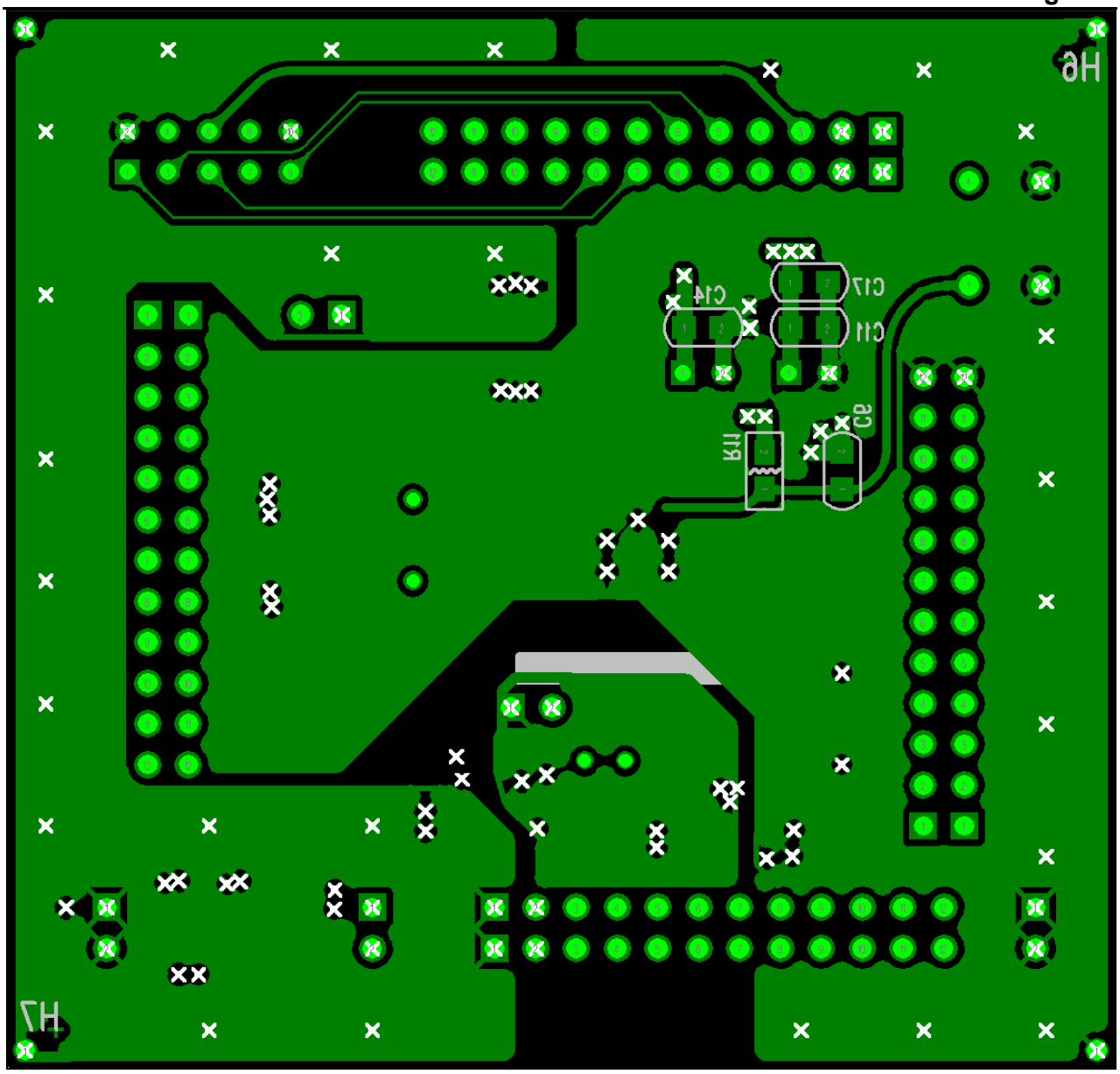

Figure 10: IS31CS8968A Evaluation Board PCB Layout - Bottom Layer

REVISION HISTORY

Revision	Detail Information	Data
A	Initial release	2018.08.28
B	Add a home page.	2018.12.20
C	Modify the ordering information.	2019.07.09

APPENDIX Ⅰ: IS31CS8968A Test Code - IOSC test

```
/*
IOSC function for IS31CS8968a
Version 1.0
Copyright(c) Integrated Silicon Solution Inc.
All right reserved
*/

#include "CS8968Asfr.h"
#include "CS8968Axram.h"
#include "HexToBIn.h"
#include "Global.h"

void main (void)

{
        Initial_REGTRM(IFB_Read_1Byte(0x20));                           //Initial IOSC:
        Initial_IOSC(IFB_Read_1Byte(0x21), IFB_Read_1Byte(0x22));       //Default IOSC: 16MHz
        TA=0xAA;
        TA=0x55;
        WDCON = 0x01;                                                   // reset watchdog timer
        TA=0x00;
//      Initial_UART0(19200,iSYSCLK);                                   //Initial UART0
        Test_IO_initial();                                              //Initial test IO

        EA = 1;

        while(1)
        {
//              printS(0x56);
                P0_2 = ~P0_2;
                Delay10ms(1);
        }
}

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 = IOSCITRM;          //ITRM;
    TB = 0x00;

    Delay10ms(1);

    TB = 0xAA;
    TB = 0x55;
```

```
    IOSCVTRM = 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)           //Initial UART0 : XOSC,19200BAUD
{
        IOCFGP0_6 = b00000110;                                     //CMOS output(TXD0)
        IOCFGP0_7 = b10100000;                                     //input with pull-up(RXD0)
        MFCFGP0_6 = b00000010;                                     //UART0 TXD0    ENABLE
        MFCFGP0_7 = b00000010;                                     //UART0 RXD0    ENABLE

        CKCON |= 0x10;                                             //timer1 divided 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;
}
```