

文档编号: AN_049

上海东软载波微电子有限公司

用户手册

HRSDK 通用 MCU 开发套件

修订历史

| 版本 | 修订日期 | 修改概要 |
|------|------------|---|
| V1.0 | 2014-10-31 | 初版 |
| V1.1 | 2014-12-25 | 增加 HRSDK-GDB-7P201 子板说明 |
| V1.2 | 2015-07-28 | 增加 HRSDK-GDB-7P275 子板说明 修改 4.17 TK（触摸按键）软件设计 统一修改公司名称、logo 及网址等 |
| V1.3 | 2016-02-25 | 增加 HRSDK-GDB-8P287/HRSDK-GDB-7P169B 子板说明 |
| V1.4 | 2017-03-07 | 增加 HRSDK-GDB-8P506 子板说明 |
| V1.5 | 2017-06-27 | 增加 HRSDK-GDB-7P2953 子板说明 |
| V1.6 | 2018-01-22 | 1.增加 HRSDK-GDB-7P179/8P286/HW2170/8P296R/7P169BR 子板说明 2.增加 HRSDK-GDB-ES8P508 子板说明 |
| V1.7 | 2018-09-05 | 1.增加 ESSDK-GDB-ES7P003 子板说明 |
| V1.8 | 2020-04-09 | 1.增加 HRSDK-GDB-SSC1645 子板说明 |
| V1.9 | 2021-10-18 | 1.增加 ES-GDB-ES7P169CFGTF1 子板说明 2.增加 ES-GDB-ES8P5066 子板说明 |
| | | |

地 址：中国上海市徐汇区古美路 1515 号凤凰园 12 号楼 3 楼

邮 编：200235

E-mail: support@essemi.com

电 话：+86-21-60910333

传 真：+86-21-60914991

网 址：<http://www.essemi.com/>

版权所有©

上海东软载波微电子有限公司

本资料内容为上海东软载波微电子有限公司在现有数据资料基础上慎重且力求准确无误编制而成，本资料中所记载的实例以正确的使用方法和标准操作为前提，使用方在应用该等实例时请充分考虑外部诸条件，上海东软载波微电子有限公司不承担或确认该等实例在使用方的适用性、适当性或完整性，上海东软载波微电子有限公司亦不对使用方因使用本资料所有内容而可能或已经带来的风险或后果承担任何法律责任。基于使本资料的内容更加完善等原因，上海东软载波微电子有限公司保留未经预告的修改权。使用方如需获得最新的产品信息，请随时用上述联系方式与上海东软载波微电子有限公司联系。

目 录

内容目录

| | |
|--------------------------------|-----------|
| HRSDK 通用 MCU 开发套件 | 1 |
| 第 1 章 HRSDK 系统 | 7 |
| 1.1 概述..... | 7 |
| 1.2 系统组成..... | 7 |
| 1.3 系统框图..... | 8 |
| 1.4 开发套件..... | 9 |
| 1.5 子板选型..... | 10 |
| 1.6 开发环境..... | 12 |
| 第 2 章 通用母板 | 13 |
| 2.1 硬件资源..... | 13 |
| 2.2 使用说明..... | 14 |
| 2.3 原理图..... | 15 |
| 第 3 章 核心 MCU 子板 | 16 |
| 3.1 硬件资源..... | 16 |
| 3.2 HRSDK-GDB-7P194..... | 17 |
| 3.2.1 使用说明..... | 17 |
| 3.2.2 原理图..... | 17 |
| 3.3 HRSDK-GDB-7P195..... | 18 |
| 3.3.1 使用说明..... | 18 |
| 3.3.2 原理图..... | 18 |
| 3.4 HRSDK-GDB-7P92..... | 19 |
| 3.4.1 使用说明..... | 19 |
| 3.4.2 原理图..... | 19 |
| 3.5 HRSDK-GDB-7P275..... | 20 |
| 3.5.1 使用说明..... | 20 |
| 3.5.2 原理图..... | 20 |
| 3.6 HRSDK-GDB-7P170..... | 21 |
| 3.6.1 使用说明..... | 21 |
| 3.6.2 原理图..... | 21 |
| 3.7 HRSDK-GDB-7P167..... | 22 |
| 3.7.1 使用说明..... | 22 |
| 3.7.2 原理图..... | 22 |
| 3.8 HRSDK-GDB-7P160..... | 23 |
| 3.8.1 使用说明..... | 23 |
| 3.8.2 原理图..... | 23 |
| 3.9 HRSDK-GDB-7P156..... | 24 |
| 3.9.1 使用说明..... | 24 |
| 3.9.2 原理图..... | 24 |
| 3.10 HRSDK-GDB-7P201..... | 25 |
| 3.10.1 使用说明..... | 25 |
| 3.10.2 原理图..... | 25 |

| | | |
|--------|--------------------------|----|
| 3.11 | HRSDK-GDB-7P169B..... | 26 |
| 3.11.1 | 使用说明..... | 26 |
| 3.11.2 | 原理图 | 26 |
| 3.12 | HRSDK-GDB-8P287 | 27 |
| 3.12.1 | 使用说明..... | 27 |
| 3.12.2 | 原理图 | 27 |
| 3.13 | HRSDK-GDB-8P506 | 28 |
| 3.13.1 | 使用说明..... | 28 |
| 3.13.2 | 原理图 | 28 |
| 3.14 | HRSDK-GDB-7P295x | 29 |
| 3.14.1 | 使用说明..... | 29 |
| 3.14.2 | 原理图 | 29 |
| 3.15 | HRSDK-GDB-7P179 | 30 |
| 3.15.1 | 使用说明..... | 30 |
| 3.15.2 | 原理图 | 30 |
| 3.16 | HRSDK-GDB-8P286 | 31 |
| 3.16.1 | 使用说明..... | 31 |
| 3.16.2 | 原理图 | 31 |
| 3.17 | HRSDK-GDB- HW2170 | 32 |
| 3.17.1 | 使用说明..... | 32 |
| 3.17.2 | 原理图 | 32 |
| 3.18 | HRSDK-GDB-8P296R | 33 |
| 3.18.1 | 使用说明..... | 33 |
| 3.18.2 | 原理图 | 33 |
| 3.19 | HRSDK-GDB-HR7P169BR..... | 34 |
| 3.19.1 | 使用说明..... | 34 |
| 3.19.2 | 原理图 | 34 |
| 3.20 | HRSDK-GDB-ES8P508x | 35 |
| 3.20.1 | 使用说明..... | 35 |
| 3.20.2 | 原理图 | 35 |
| 3.21 | ESSDK-GDB-ES7P003..... | 36 |
| 3.21.1 | 使用说明..... | 36 |
| 3.21.2 | 原理图 | 36 |
| 3.22 | HRSDK-GDB-SSC1645..... | 37 |
| 3.22.1 | 使用说明..... | 37 |
| 3.22.2 | 原理图 | 37 |
| 3.23 | ES-GDB-ES7P169C | 38 |
| 3.23.1 | 使用说明..... | 38 |
| 3.23.2 | 原理图 | 38 |
| 3.24 | ES-GDB-ES8P5066 | 39 |
| 3.24.1 | 使用说明..... | 39 |
| 3.24.2 | 原理图 | 39 |
| 3.25 | LIGHT（指示灯） | 40 |
| 3.25.1 | 模块说明..... | 40 |

| | | |
|--------|------------------------------|----|
| 3.25.2 | 硬件设计..... | 40 |
| 3.25.3 | 软件设计 (light_demo) | 40 |
| 3.26 | MOTOR (步进电机) | 41 |
| 3.26.1 | 模块说明..... | 41 |
| 3.26.2 | 硬件设计..... | 41 |
| 3.26.3 | 软件设计 (motor_demo) | 41 |
| 3.27 | ZERO (过零检测) | 43 |
| 3.27.1 | 模块说明..... | 43 |
| 3.27.2 | 硬件设计..... | 43 |
| 3.27.3 | 软件设计 (zero_demo) | 43 |
| 3.28 | ADC (模数转换) | 44 |
| 3.28.1 | 模块说明..... | 44 |
| 3.28.2 | 硬件设计..... | 44 |
| 3.28.3 | 软件设计 (adc_demo) | 44 |
| 3.29 | BUZZER (蜂鸣器) | 46 |
| 3.29.1 | 模块说明..... | 46 |
| 3.29.2 | 硬件设计..... | 46 |
| 3.29.3 | 软件设计 (buzzer_demo) | 47 |
| 3.30 | LED (数码管) | 49 |
| 3.30.1 | 模块说明..... | 49 |
| 3.30.2 | 硬件设计..... | 49 |
| 3.30.3 | 软件设计..... | 49 |
| 3.31 | LCD (液晶) | 52 |
| 3.31.1 | 模块说明..... | 52 |
| 3.31.2 | 硬件设计..... | 52 |
| 3.31.3 | 软件设计 (lcd_demo) | 53 |
| 3.32 | KINT (独立按键) | 55 |
| 3.32.1 | 模块功能..... | 55 |
| 3.32.2 | 硬件设计..... | 55 |
| 3.32.3 | 软件设计 (key_demo) | 55 |
| 3.33 | KEY (矩阵按键) | 57 |
| 3.33.1 | 模块功能..... | 57 |
| 3.33.2 | 硬件设计..... | 57 |
| 3.33.3 | 软件设计 (key_demo) | 57 |
| 3.34 | SPI (主控) | 60 |
| 3.34.1 | 模块说明..... | 60 |
| 3.34.2 | 硬件设计..... | 60 |
| 3.34.3 | 软件设计 (spi_demo) | 60 |
| 3.35 | I2CM (主控) | 64 |
| 3.35.1 | 模块说明..... | 64 |
| 3.35.2 | 硬件设计..... | 64 |
| 3.35.3 | 软件设计—模拟方式 (i2cm_demo1) | 64 |
| 3.35.4 | 软件设计—硬件方式 (i2cm_demo2) | 68 |
| 3.36 | FLASH (内部存储器) | 69 |

| | | |
|--------------|------------------------------|-----------|
| 3.36.1 | 模块说明..... | 69 |
| 3.36.2 | 软件设计 (flash_demo) | 69 |
| 3.37 | I2CS (从动) | 71 |
| 3.37.1 | 模块说明..... | 71 |
| 3.37.2 | 硬件设计..... | 71 |
| 3.37.3 | 软件设计—模拟方式 (i2cs_demo1) | 71 |
| 3.37.4 | 软件设计—硬件方式 (i2cs_demo2) | 74 |
| 3.38 | UART (串口) | 75 |
| 3.38.1 | 模块说明..... | 75 |
| 3.38.2 | 硬件设计..... | 76 |
| 3.38.3 | 软件设计 (uart_demo) | 76 |
| 3.39 | CARD (IC 卡) | 79 |
| 3.39.1 | 模块说明..... | 79 |
| 3.39.2 | 硬件设计..... | 79 |
| 3.40 | RM (红外收发) | 80 |
| 3.40.1 | 模块说明..... | 80 |
| 3.40.2 | 硬件设计..... | 80 |
| 3.40.3 | 软件设计 (rm_demo) | 81 |
| 3.41 | AMCU (辅助 MCU) | 85 |
| 3.41.1 | 模块功能..... | 85 |
| 3.41.2 | 硬件设计..... | 85 |
| 3.41.3 | 软件设计 (amcu_demo) | 86 |
| 3.42 | TK (触摸按键) | 87 |
| 3.42.1 | 模块说明..... | 87 |
| 3.42.2 | 硬件设计..... | 87 |
| 3.42.3 | 软件设计 (tk_demo) | 87 |
| 第 4 章 | 其他硬件..... | 90 |
| 4.1 | 电源设计 | 90 |
| 4.1.1 | 模块说明..... | 90 |
| 4.1.2 | 硬件设计..... | 90 |
| 4.2 | ICD (调试端口) | 91 |
| 4.2.1 | 模块说明..... | 91 |
| 4.2.2 | 硬件设计..... | 91 |
| 4.3 | 外部复位 | 92 |
| 4.3.1 | 模块说明..... | 92 |
| 4.3.2 | 硬件设计..... | 92 |
| 4.4 | 外部时钟 | 93 |
| 4.4.1 | 模块说明..... | 93 |
| 4.4.2 | 硬件设计..... | 93 |
| 4.5 | 连接插座 | 94 |
| 4.5.1 | 模块说明..... | 94 |
| 4.5.2 | 硬件设计..... | 94 |

第1章 HRSDK系统

1.1 概述

HRSDK 是入门开发套件，它主要用于指导工程师快速学习 MCU 的各种外设功能。通过该开发套件的学习，再结合各系列芯片的数据手册、应用笔记和例程，可以在短时间内构建起应用系统所需的软硬件开发平台。

该开发套件由通用母板、核心 MCU 子板以及各种功能软件例程包组成。通用母板包含了按键、蜂鸣、LED、LCD、I2C、SPI、UART、IrDA 等常见应用电路模块，核心 MCU 子板采用了各系列 MCU 中的典型芯片进行设计，在这两个硬件平台下，通过功能软件例程包来实现 MCU 外设和应用电路模块的功能演示。

1.2 系统组成

HRSDK 硬件由通用母板（GMB）和各种核心 MCU 子板（GDB）组成。

通用母板集电源输入、按键输入、显示输出、串行通讯、扩展存储器、红外通讯、模数转换、音乐蜂鸣、步进电机和辅助 MCU 等丰富的资源于一体，各功能模块清晰明了，便于用户分别开发调试。

核心 MCU 子板通过 2 排 DIN32 插座与通用母板连接，组合成 HRSDK 完整的硬件系统，另外核心 MCU 子板亦可作为该系列 MCU 的最小系统单独进行开发调试。

HRSDK 软件提供了支持通用母板各功能模块的应用例程包，根据不同的核心 MCU 子板选择相应的例程包。

例程包的编程风格力求结构化，模块化和规范化，主要采用了主程序和模块程序分离的形式，定义了一系列标准接口的模块函数，每个模块函数相对独立，可移植性好，方便用户在构建应用系统中根据需要直接调用，也可为用户设计自己的程序提供借鉴的思路。

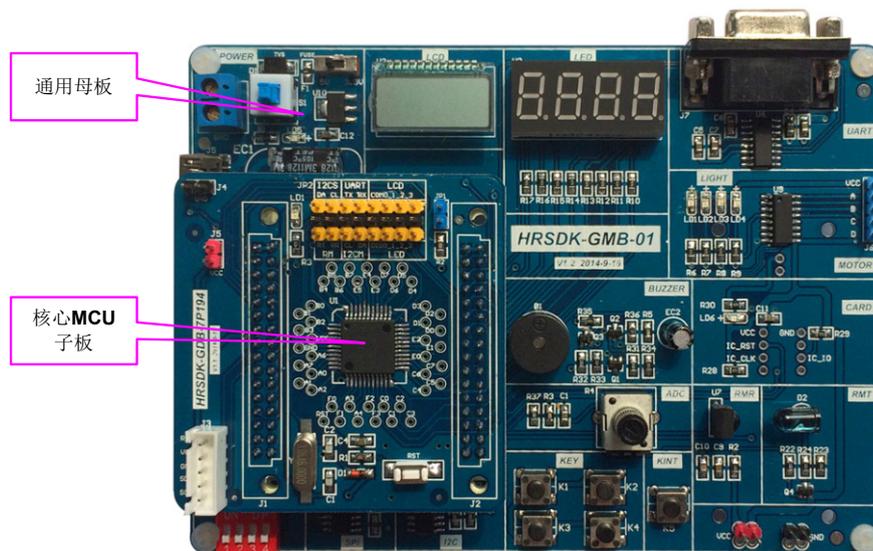


图 1-1 通用母板搭载子板图

1.3 系统框图

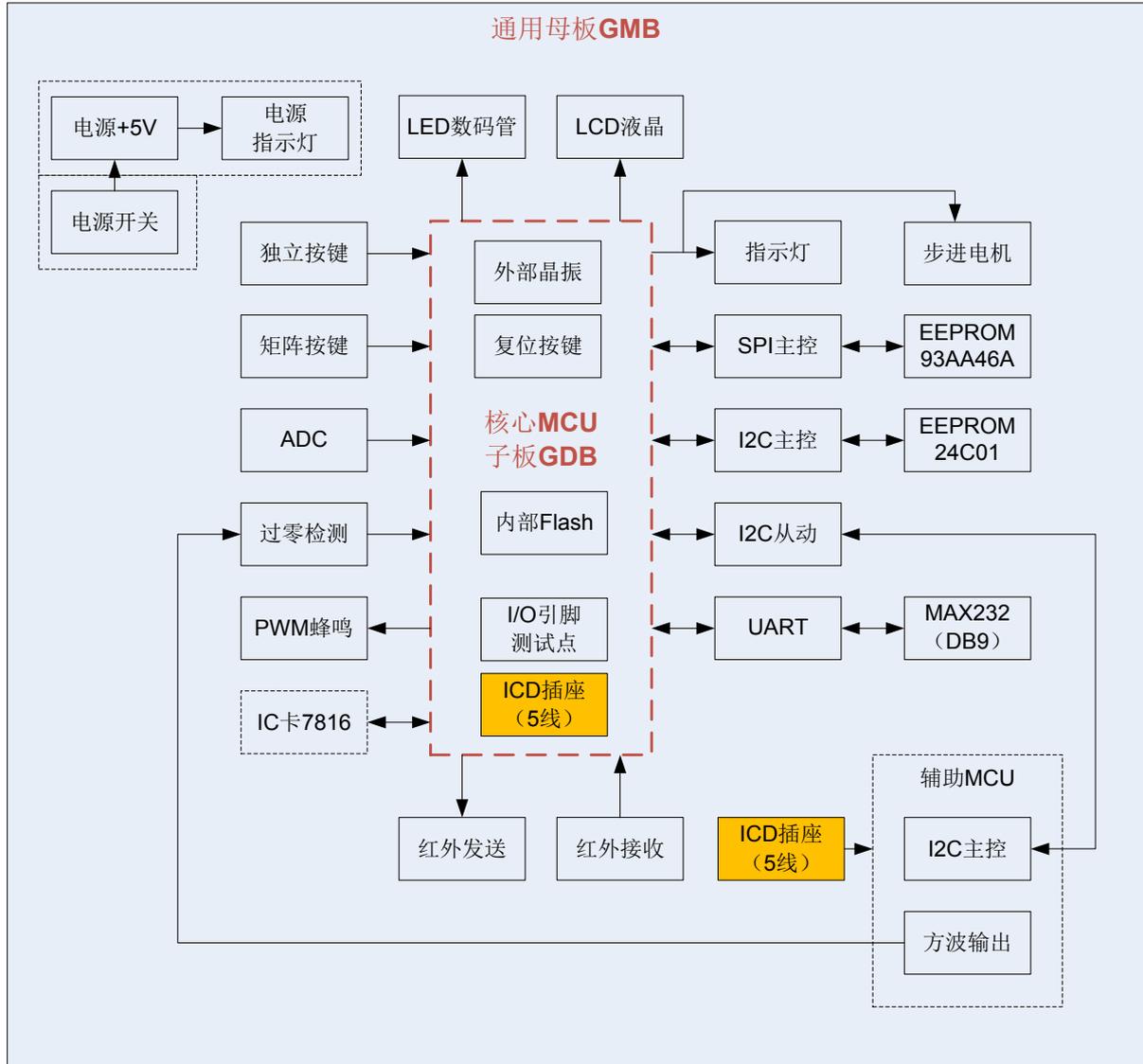
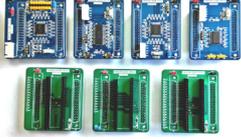


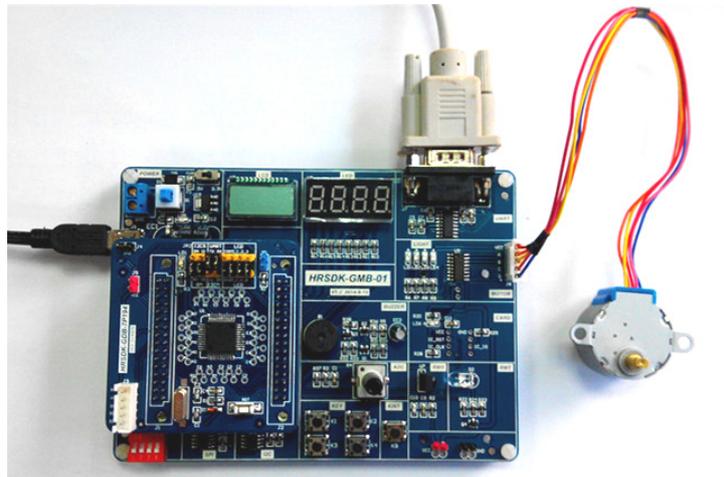
图 1-2 HRSDK 系统框图

1.4 开发套件

HRSDK 开发套件中包含了以下基本硬件配置：

| 序号 | 名称 | 实物 | 数量 | 备注 |
|----|-----------------------|---|----|----------------------|
| 1 | 母板 HRSDK-GMB-01 |  | 1 | |
| 2 | 子板 HRSDK-GDB-XXXXX |  | 若干 | 根据需要选配型号， 详见子板选型表 |
| 3 | Mini USB 线缆 |  | 1 | 供电电源，若接仿真器供电，可不用连接 |
| 4 | 5V 步进电机 |  | 1 | 因功耗较大，建议调试其它模块时，不要安装 |
| 5 | 16MHz 晶振 |  | 1 | 若使用片内晶振，可不用安装 |
| 6 | 短接帽 |  | 若干 | 用于板上跳线开关和复用选择 |

连接示意图



1.5 子板选型

| 芯片型号 | 类型 | MCU 子板 | LIGHT | MOTOR | ZERO | ADC | BUZZER | LED | LCD | KINT | KEY | SPI | I2CM 模拟 | I2CM 硬件 | I2CS 模拟 | I2CS 硬件 | UART | RM | Flash | 备注 |
|----------|-------|------------|-------|-------|------|-----|--------|-----|-----|------|-----|-----|------------|------------|------------|------------|------|----|-------|------------|
| HR7P193 | 8bit | GDB-7P194 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | ● | ● | ● | |
| HR7P194 | 8bit | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | ● | ● | ● | |
| HR7P195 | 8bit | GDB-7P195 | ● | ● | ● | ● | ● | ● | | ● | ● | | | | | | ● | | ● | |
| HR7P90 | 8bit | | | | | | | | | | | | | | | | | | | |
| HR7P91 | 8bit | GDB-7P92 | ● | ● | ● | ● | ● | ● | | ● | ● | | ● | | | | ● | | ● | |
| HR7P92 | 8bit | | ● | ● | ● | ● | ● | ● | ● | | ● | ● | | ● | | | | ● | | ● |
| HR7P275 | 8bit | GDB-7P275 | ● | ● | ● | ● | ● | ● | | ● | ● | | ● | | | | ● | | ● | EERPOM |
| HR7P169 | 8bit | GDB-7P170 | ● | ● | | ● | ● | | | ● | ● | | | | | ● | ● | | ● | |
| HR7P170 | 8bit | | ● | ● | | ● | ● | | | | ● | ● | | | | | | | | ● |
| HR7P166 | 8bit | GDB-7P167 | ● | ● | ● | ● | ● | | | ● | ● | | | | | | | | | ME310 仿真 |
| HR7P167 | 8bit | | ● | ● | ● | ● | ● | ● | | | ● | ● | | | | | | | | |
| HR7P159 | 8bit | GDB-7P160 | ● | ● | ● | | ● | | | ● | | | | | | | | | | ME303 仿真 |
| HR7P160 | 8bit | | ● | ● | ● | | ● | | | | ● | | | | | | | | | |
| HR7P155 | 8bit | GDB-7P156 | ● | | ● | ● | ● | | | ● | | | | | | | | | | ME303 仿真 |
| HR7P156 | 8bit | | ● | | ● | ● | ● | | | | ● | | | | | | | | | |
| HR7P201 | 8bit | GDB-7P201 | | | | | ● | ● | | | | | | | | | | | | TK |
| HR7P169B | 8bit | GDB-7P169B | ● | ● | | ● | ● | | | | ● | | | | | ● | ● | | ● | |
| HR8P287 | 32bit | GDB-8P287 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | ● | | ● | ● | | ● | CARD(7816) |
| HR8P506 | 32bit | GDB-8P506 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | ● | | ● | ● | | ● | |
| ES7P2953 | 8bit | GDB-7P2953 | ● | | | ● | ● | ● | | | | | | | | ● | ● | | ● | |
| HR7P179 | 8bit | GDB-7P179 | ● | ● | | ● | ● | | | | ● | | | | | ● | ● | | | |
| HR8P286 | 32bit | GDB-8P286 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | ● | | ● | ● | ● | ● | |

| 芯片型号 | 类型 | MCU 子板 | LIGHT | MOTOR | ZERO | ADC | BUZZER | LED | LCD | KINT | KEY | SPI | I2CM 模拟 | I2CM 硬件 | I2CS 模拟 | I2CS 硬件 | UART | RM | Flash | 备注 |
|-----------|-------|---------------|-------|-------|------|-----|--------|-----|-----|------|-----|-----|------------|------------|------------|------------|------|----|-------|----|
| HW2170 | 8bit | GDB-HW2170 | | | | ● | ● | | | | ● | | | | | | ● | ● | ● | |
| HR8P296R | 32bit | GDB-8P296R | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | ● | | ● | ● | ● | ● | |
| HR7P169BR | 8bit | GDB-HR7P169BR | | | | | | | | | ● | | | | | | ● | ● | ● | |
| ES8P508x | 32bit | GDB-ES8P508x | ● | | | ● | ● | ● | | ● | ● | ● | | ● | | ● | ● | | ● | |
| ES7P003 | 8bit | GDB-ES7P003 | ● | | | ● | | ● | | ● | ● | ● | | | | ● | ● | | ● | |
| SSC1645 | 32bit | GDB-SSC1645 | ● | | | ● | ● | | | ● | ● | ● | | | | | ● | | ● | |
| ES8P5066 | 32bit | GDB-ES8P5066 | ● | | | ● | ● | | | | ● | ● | | ● | | | ● | | | |

表 1-1 核心 MCU 子板系列功能表

1.6 开发环境

开发调试 HRSDK 时，8 位芯片需使用开发工具 HR10M，32 位芯片需使用 HR-Link 或 Jlink，如下图所示：

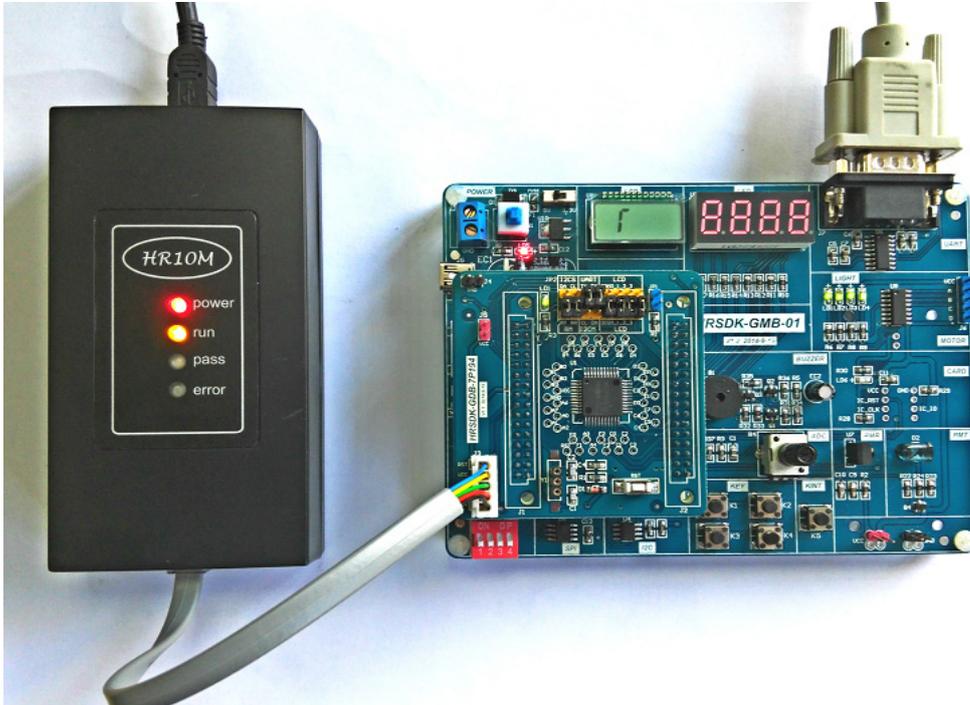


图 1-3 HRSDK 开发调试连接图

集成开发环境（IDE）采用 iDesigner，如下图所示：

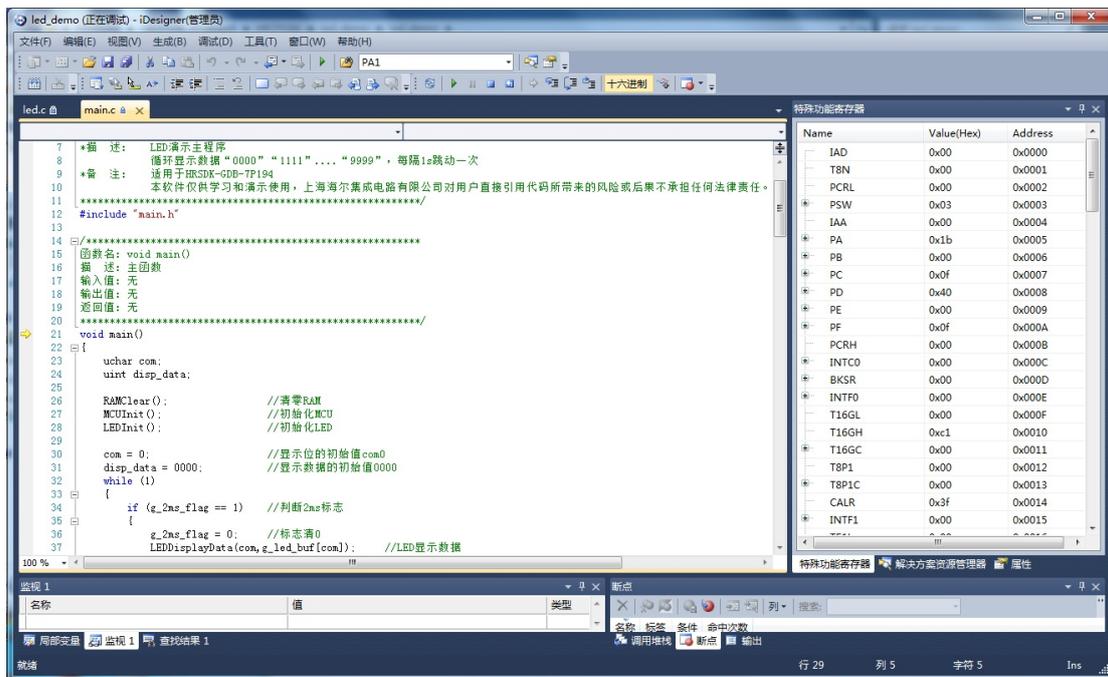


图 1-4 iDesigner 软件界面

第2章 通用母板

2.1 硬件资源

通用母板 HRSDK-GMB-01 包含以下硬件模块：

- ◆ 电源输入：Mini USB 或 2 芯端子电源输入
- ◆ 电源输出：5V 或 3.3V 电压选择输出（连接仿真器时无法选择），带电源指示灯。
- ◆ I/O：输入输出端口，2 排 DIN32 插座。
- ◆ KEY：矩阵按键，4 个（2 行 2 列）。
- ◆ KINT：独立按键，1 个，兼用于按键中断或唤醒。
- ◆ ADC：模数转换，1 路，带可调电位器。
- ◆ ZERO：过零检测，1 路。
- ◆ BUZZER：交流蜂鸣器，1 路，PWM 控制。
- ◆ MOTOR：步进电机驱动，28BYJ-48（4 相）。
- ◆ LIGHT：指示灯，4 个，发光二极管。
- ◆ LED：数码管，4 位 8 段（4com*8seg）。
- ◆ LCD：液晶显示，4 位 8 段（4com*8seg）。
- ◆ SPI：SPI 通讯，读写 EEPROM，93C46（128 字节）。
- ◆ I2CM：I2C 主机，读写 EEPROM，24C01（128 字节）。
- ◆ I2CS：I2C 从机，通讯主机为辅助 MCU。
- ◆ UART：串行通讯，标准 RS232 电平接口。
- ◆ RM：IrDA 红外收发，38KHz 载波协议。
- ◆ AMCU：辅助 MCU，模拟过零方波输出，模拟 I2C 主机，带工作指示灯和 ICD 调试端口。
- ◆ CARD（暂不提供）：IC 卡接口，ISO7816 协议，带卡在位指示灯。

2.2 使用说明

通用母板包括常用模块和外围电路，可支持多款 MCU 系列子板，结构图如下：

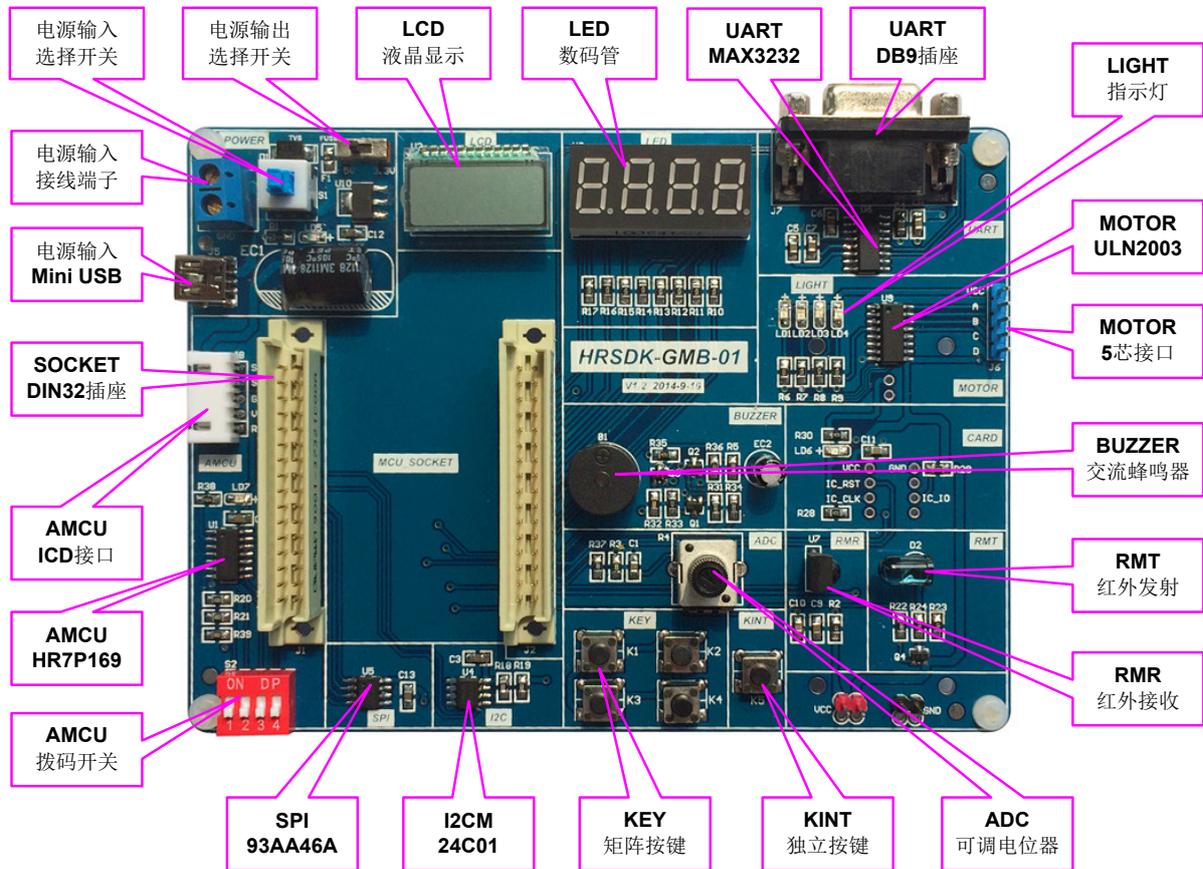


图 2-1 通用母板实物图

其中，2 排 DIN32 插座用于搭载核心 MCU 子板。

当采用外部电源时，可选择 Mini USB 或 2 芯端子电源输入，以及选择 5V 或 3.3V 电压输出。

当连接仿真器调试时，无需连接外部电源输入，也无需选择输出电压。

2.3 原理图

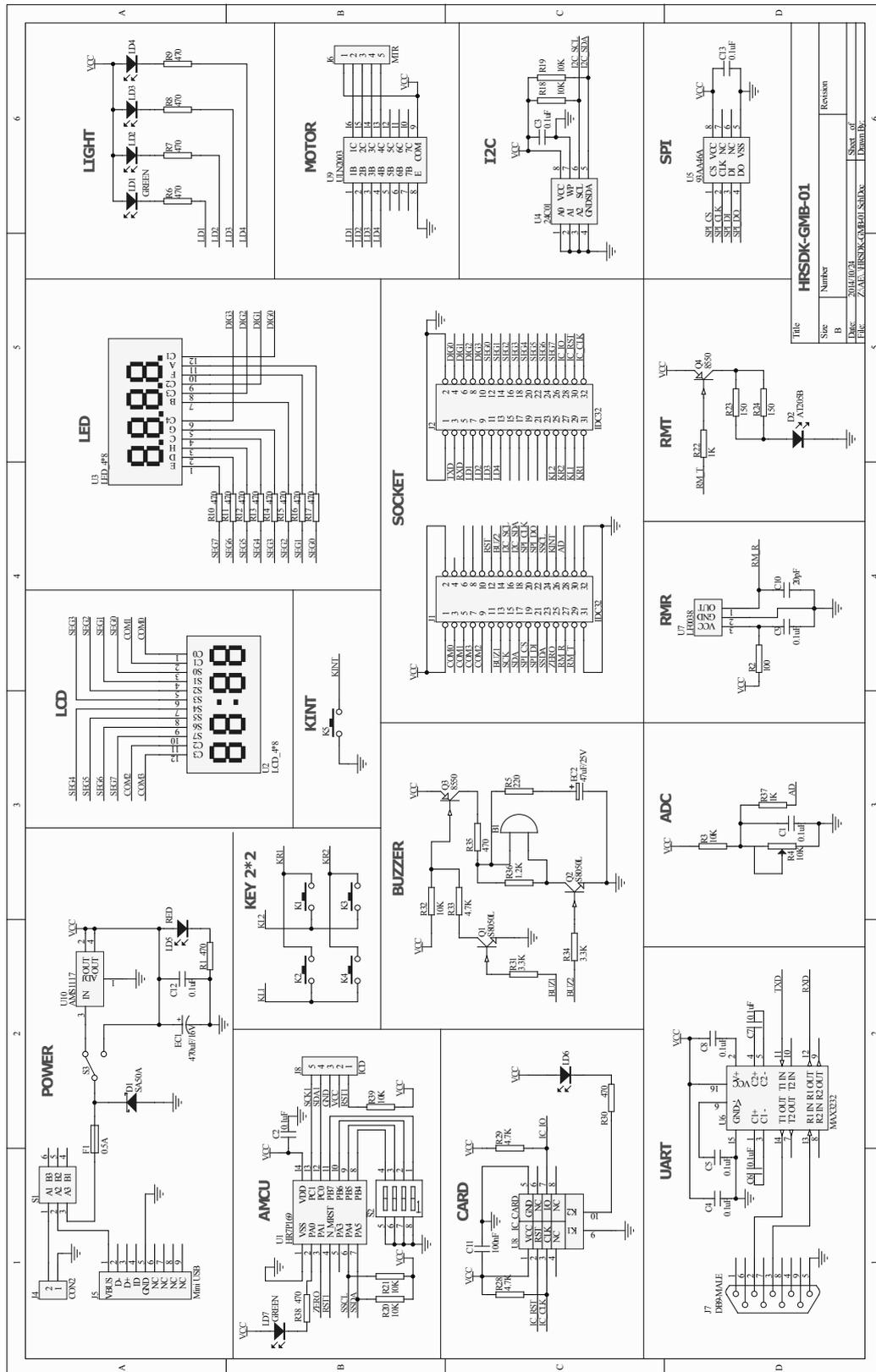


图 2-2 通用母板原理图

第3章 核心MCU子板

3.1 硬件资源

HRSDK 针对不同的 MCU，设计了各种核心 MCU 子板，每款子板包含 MCU 芯片（或插座）和部分外围电路：

- ◆ MCU 芯片：8-bit 或 32-bit MCU
- ◆ 外部晶振：当系统配置为外部晶振时使用
- ◆ 外部复位：阻容上电复位和按键复位
- ◆ ICD 调试端口：MRST、VCC、GND、SDA、SCK
- ◆ 引脚测试点：用于测量信号或外接电路
- ◆ 选择跳线（短接帽）：设置复用功能或通断信号

例如，HRSDK-GDB-7P194 的结构图如下：

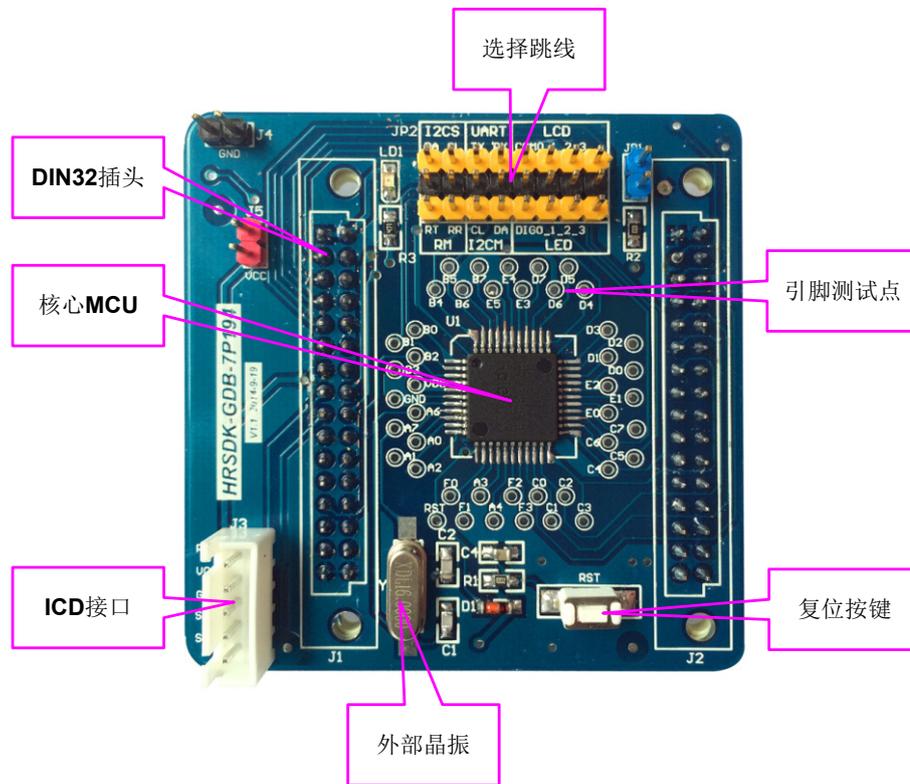


图 3-1 核心 MCU 子板结构图

3.2 HRSDK-GDB-7P194

3.2.1 使用说明

- ◆ JP2 复用: LED / LCD, UART / I2CM, RM / I2CS (短接帽选择)。
- ◆ JP1 短接: PD5 引脚下拉至 GND, 自定义输入功能。

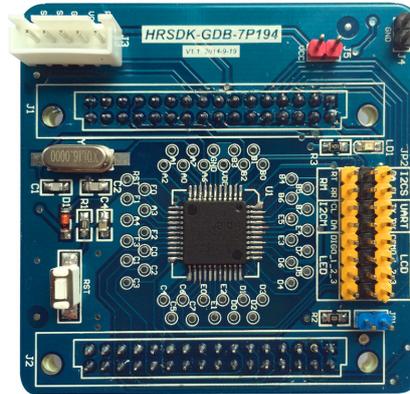


图 3-2 HRSDK-GDB-7P194 实物图

3.2.2 原理图

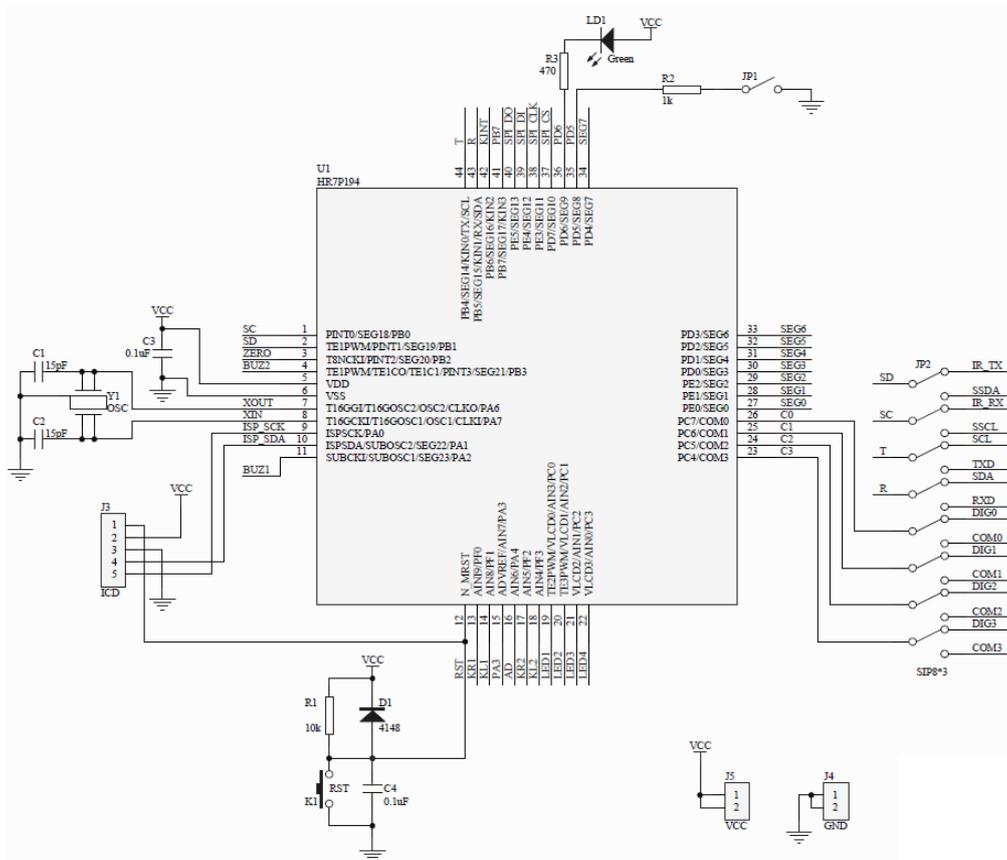


图 3-3 HRSDK-GDB-7P194 原理图

3.3 HRSDK-GDB-7P195

3.3.1 使用说明

- ◆ JP1 和 JP2 复用：LD3 / SEG6，LD4 / SEG7（短接帽选择）。

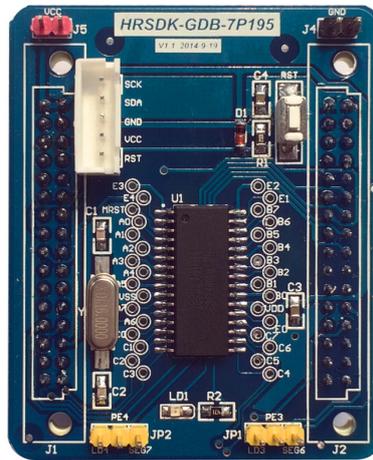


图 3-4 HRSDK-GDB-7P195 实物图

3.3.2 原理图

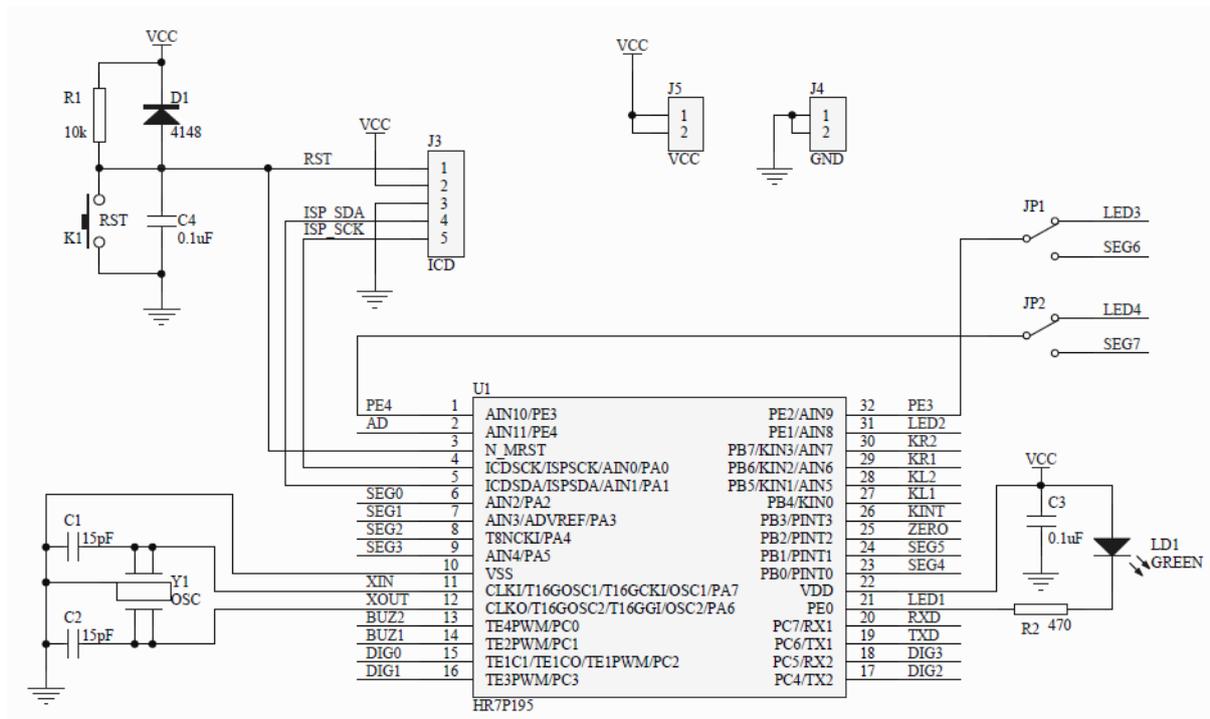


图 3-5 HRSDK-GDB-7P195 原理图

3.4 HRSDK-GDB-7P92

3.4.1 使用说明

- ◆ JP1 短接：PA3 引脚下拉至 GND，自定义输入功能。

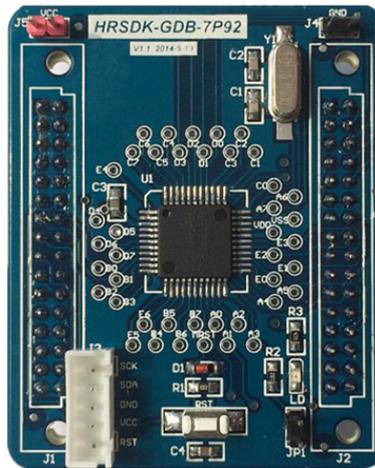


图 3-6 HRSDK-GDB-7P92 实物图

3.4.2 原理图

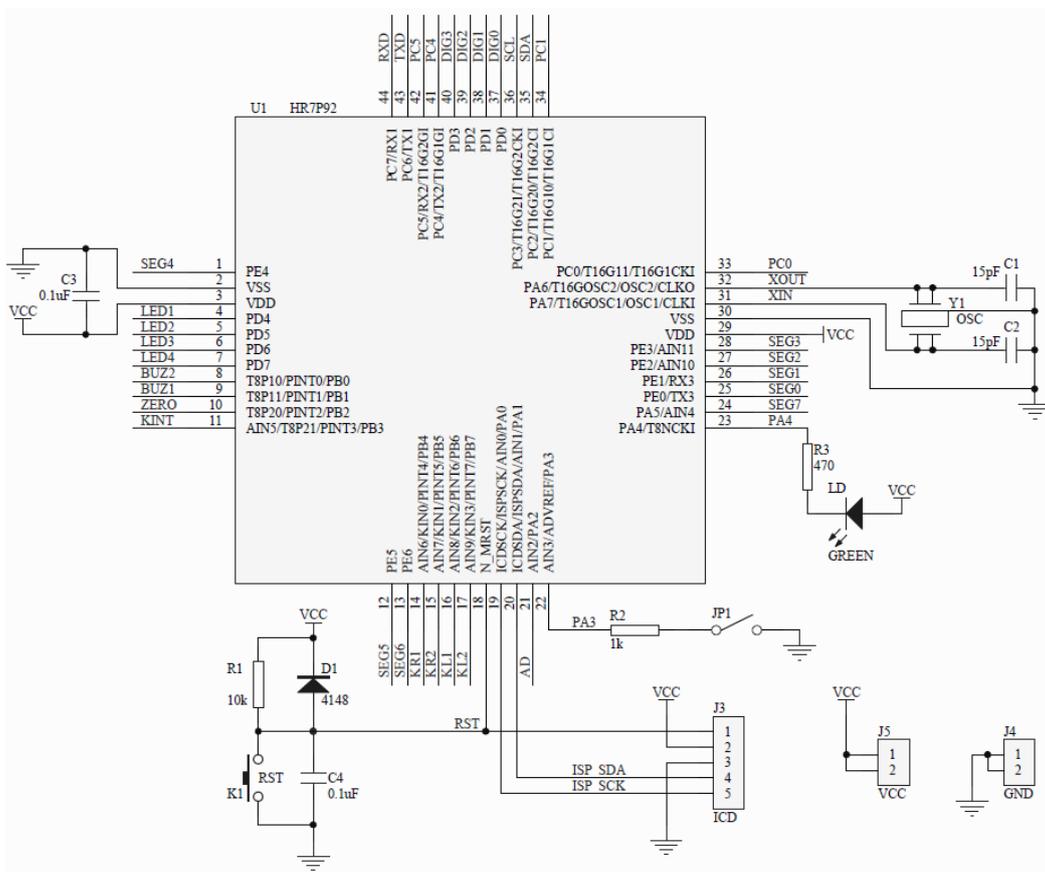


图 3-7 HRSDK-GDB-7P92 原理图

3.5 HRSDK-GDB-7P275

3.5.1 使用说明

- ◆ JP1 短接：PD5 引脚下拉至 GND，自定义输入功能。



图 3-8 HRSDK-GDB-7P275 实物图

3.5.2 原理图

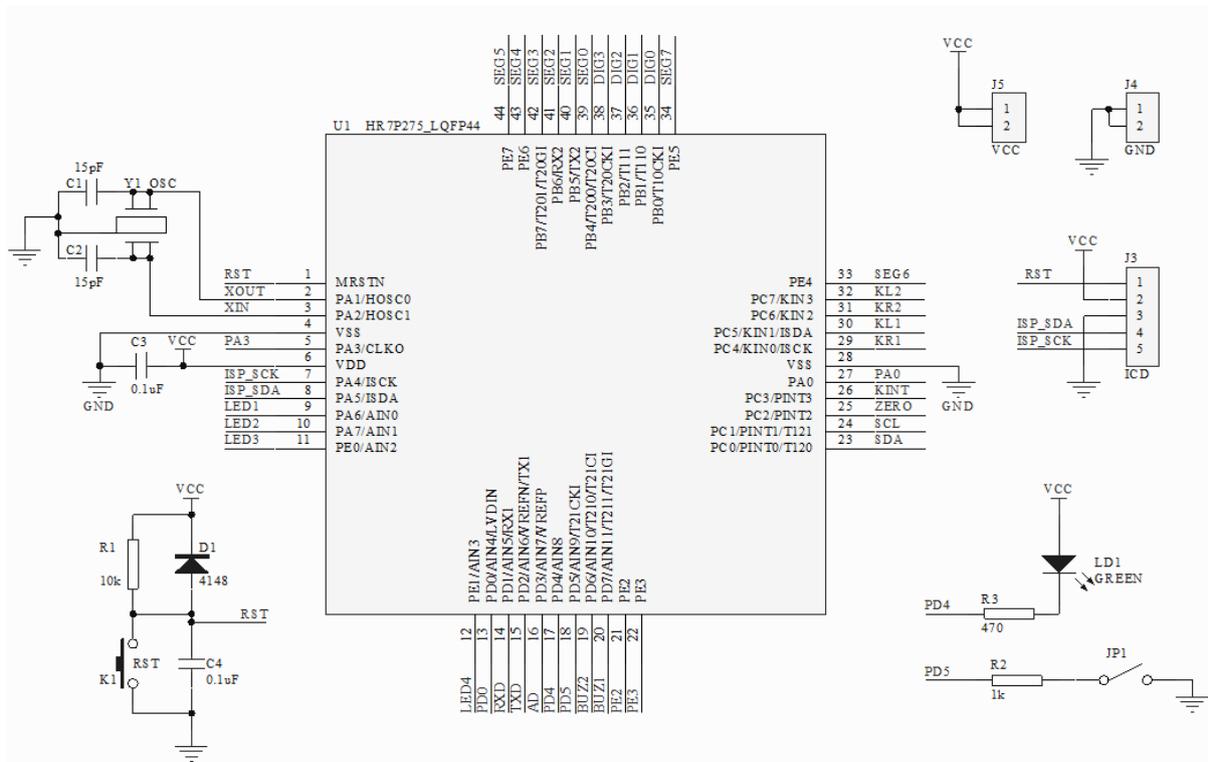


图 3-9 HRSDK-GDB-7P275 原理图

3.6 HRSDK-GDB-7P170

3.6.1 使用说明

- ◆ JP1 和 JP2 复用: SSCL / TXD, SSSDA / RXD (短接帽选择), 因为这两根口线还占用了 ICD 仿真调试通讯口线, 所以在下载与仿真时, 需注意断开短接帽, 而且由于口线冲突, 无法仿真调试 I2CS 和 UART 功能, 必须禁止 IC DEN, 全程运行。

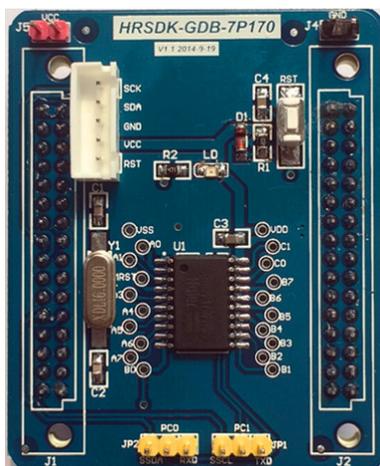


图 3-10 HRSDK-GDB-7P170 实物图

3.6.2 原理图

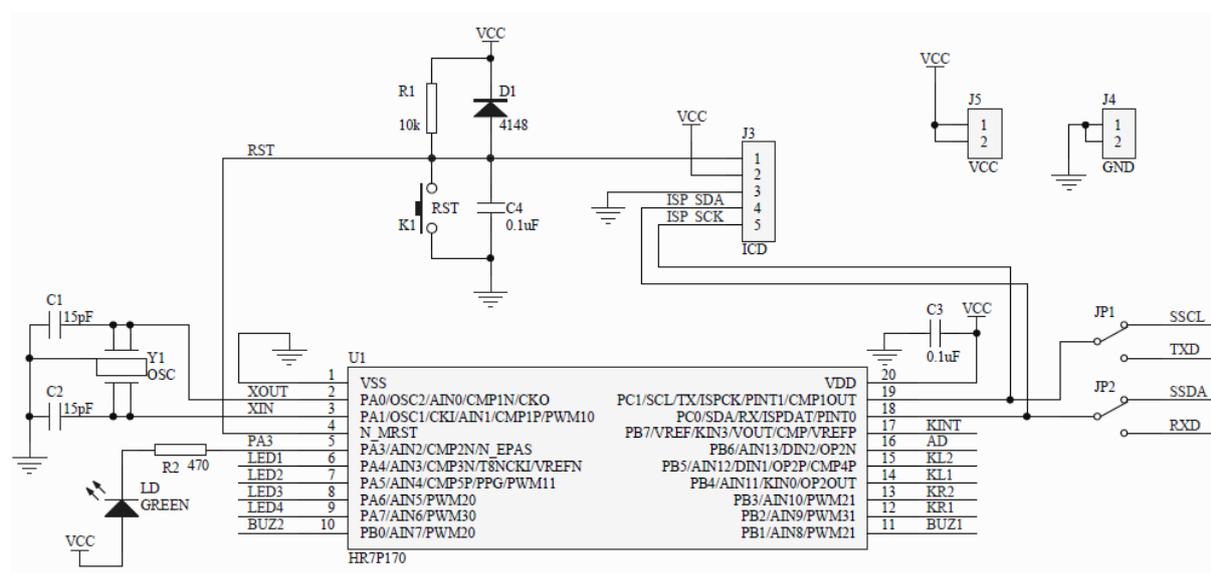


图 3-11 HRSDK-GDB-7P170 原理图

3.7 HRSDK-GDB-7P167

3.7.1 使用说明

- ◆ 在使用时，需搭配 ME310-EMU 仿真头。

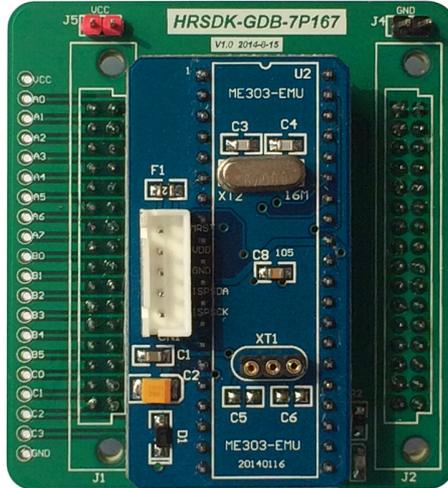


图 3-12 HRSDK-GDB-7P167 实物图

3.7.2 原理图

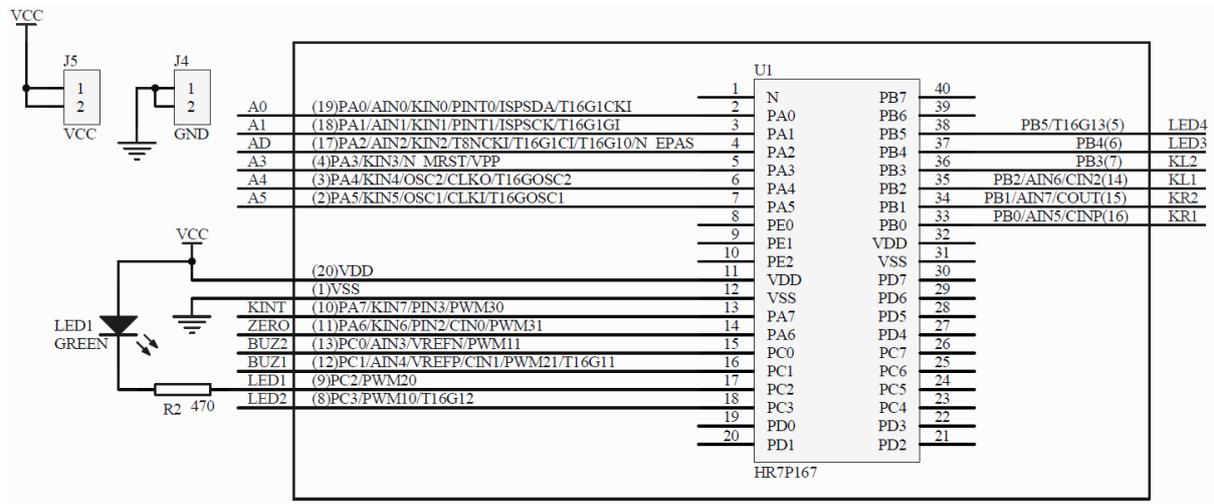


图 3-13 HRSDK-GDB-7P167 原理图

3.8 HRSDK-GDB-7P160

3.8.1 使用说明

- ◆ 在使用时，需搭配 ME303-EMU 仿真头。
- ◆ JP1 短接：用于过零检测输入开关。

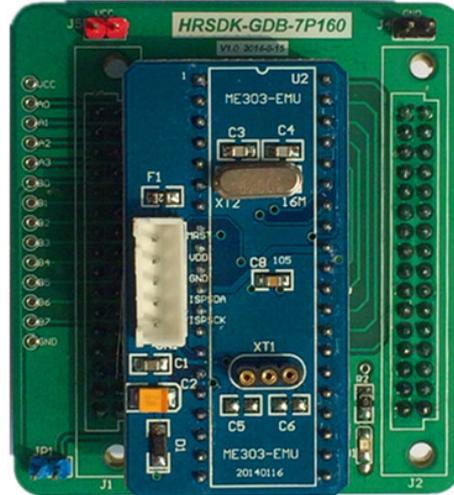


图 3-14 HRSDK-GDB-7P160 实物图

3.8.2 原理图

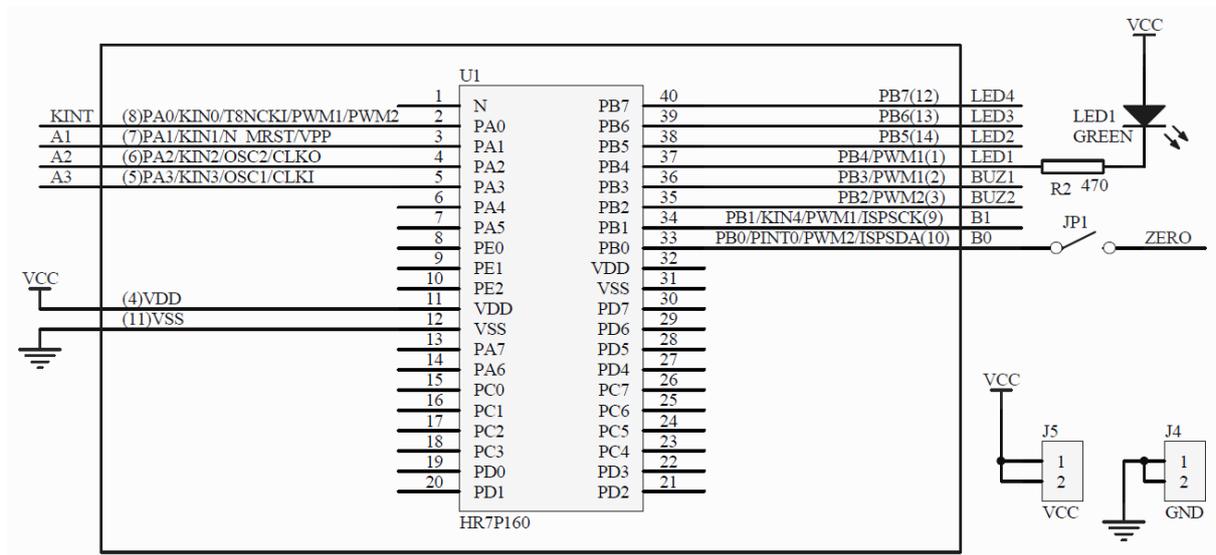


图 3-15 HRSDK-GDB-7P160 原理图

3.9 HRSDK-GDB-7P156

3.9.1 使用说明

- ◆ 在使用时，需搭配 ME303-EMU 仿真头。

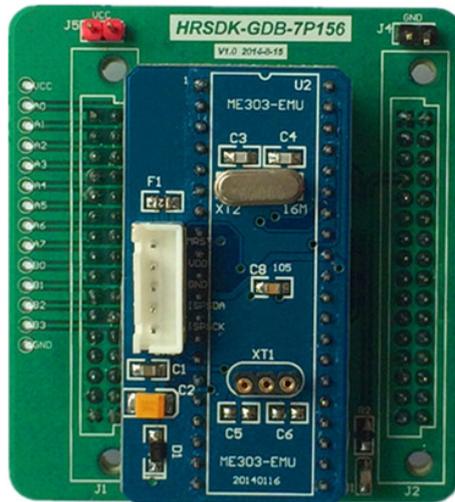


图 3-16 HRSDK-GDB-7P156 实物图

3.9.2 原理图

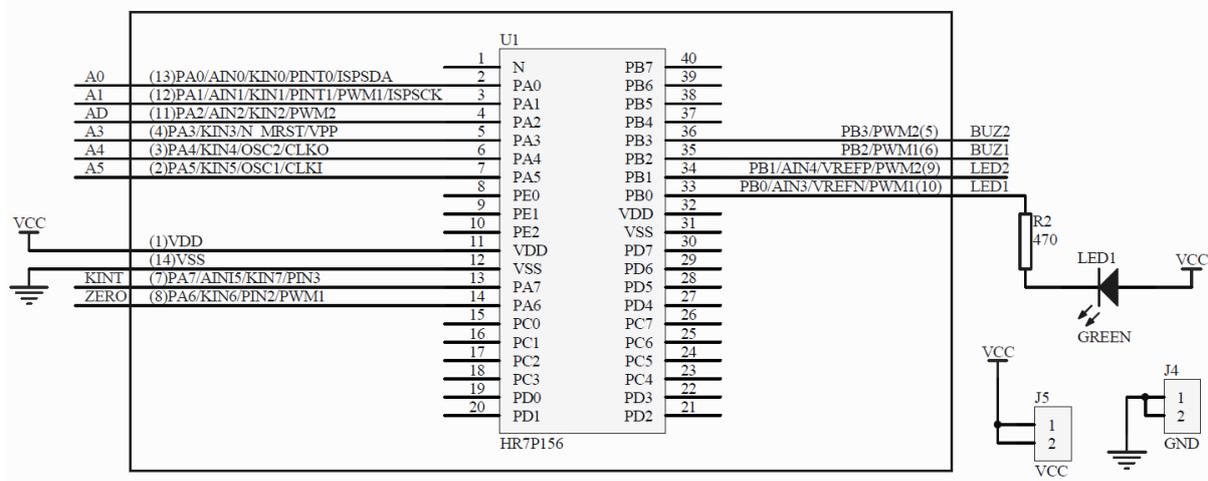


图 3-17 HRSDK-GDB-7P156 原理图

3. 10 HRSDK-GDB-7P201

3. 10. 1 使用说明

- ◆ HRSDK-GDB-7P201 子板实现触摸按键功能

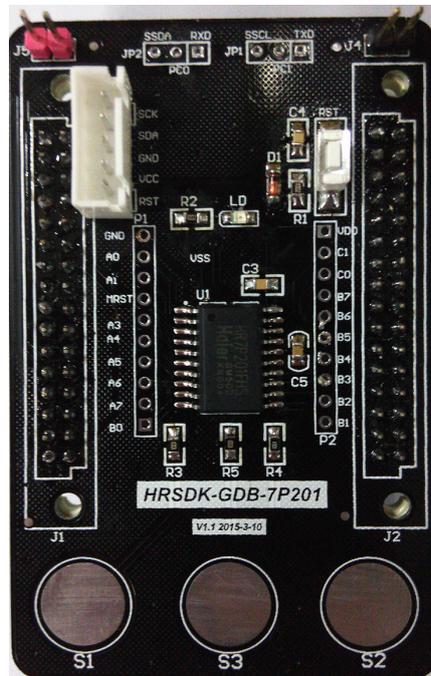


图 3-18 HRSDK-GDB-7P201 实物图

3. 10. 2 原理图

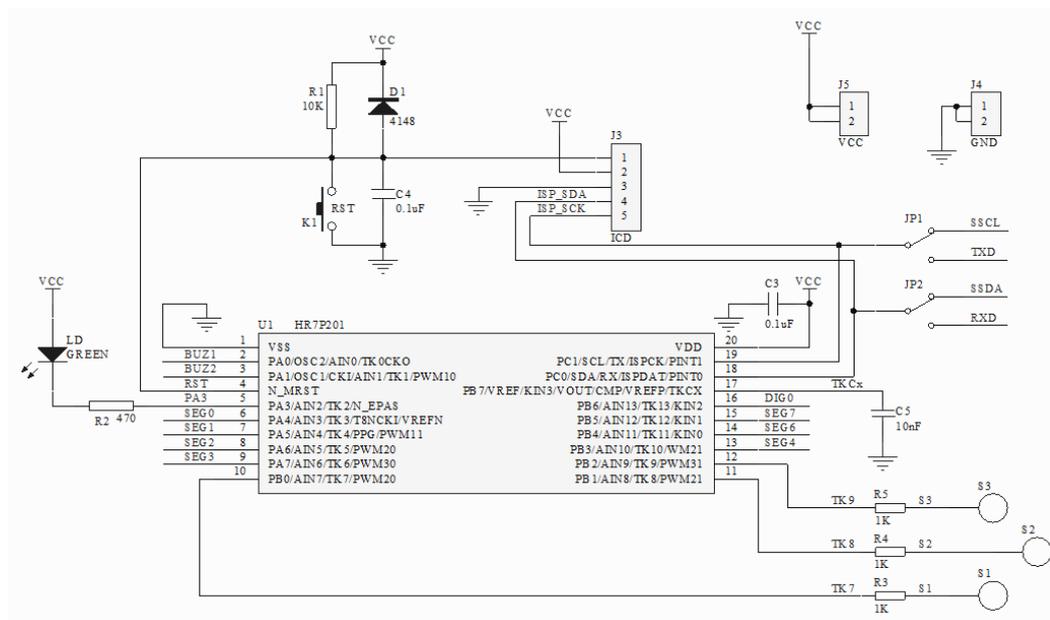


图 3-19 HRSDK-GDB-7P201 原理图

3. 11 HRSDK-GDB-7P169B

3. 11. 1 使用说明

- ◆ 仿真口采用 PB6 (ISPDAT) 和 PB7 (ISPCK), 不影响仿真调试 UART 和 I2C 通讯接口 PC0 和 PC1。

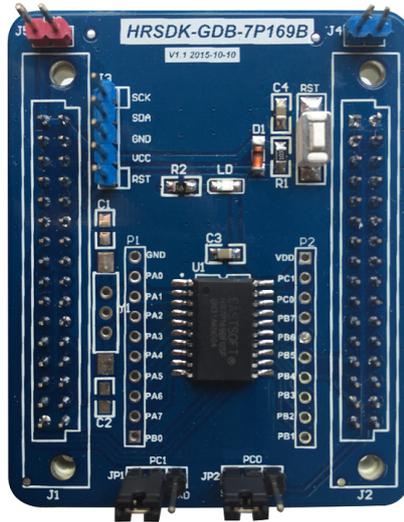


图 3-20 HRSDK-GDB-7P169B 实物图

3. 11. 2 原理图

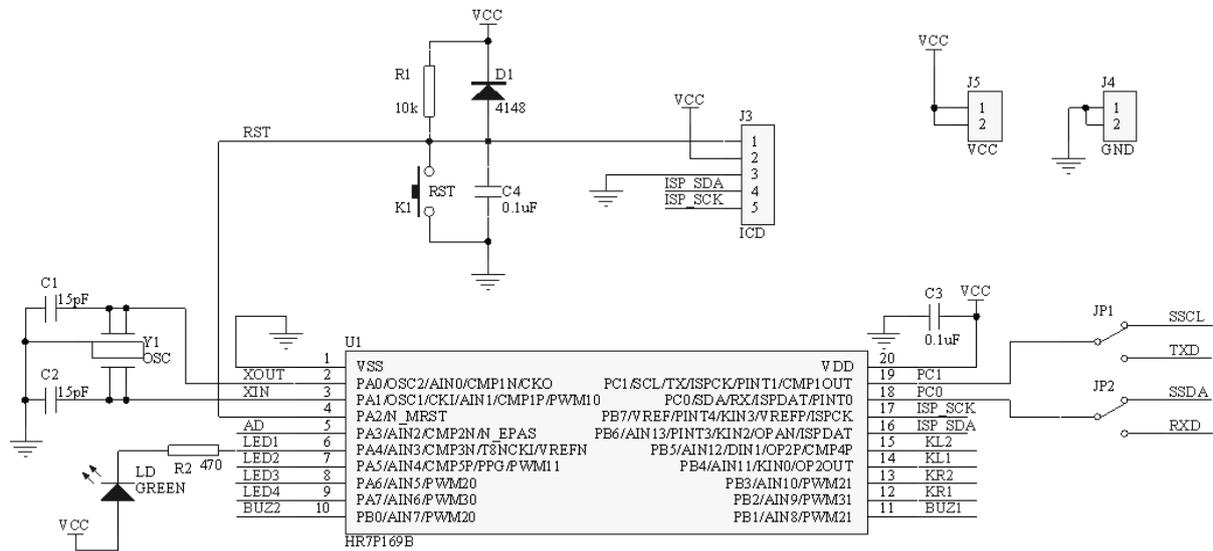


图 3-21 HRSDK-GDB-7P169B 原理图

3. 12 HRSDK-GDB-8P287

3. 12. 1 使用说明

◆ HRSDK-GDB-8P287 子板

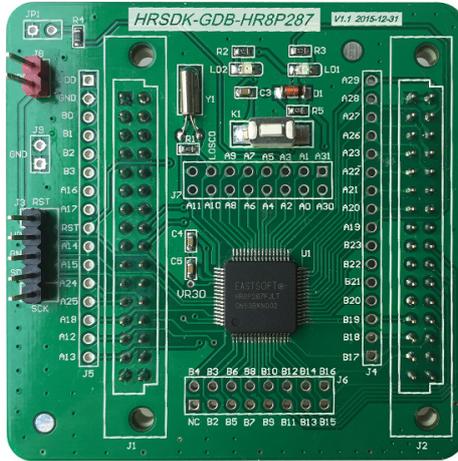


图 3-22 HRSDK-GDB-8P287 实物图

3. 12. 2 原理图

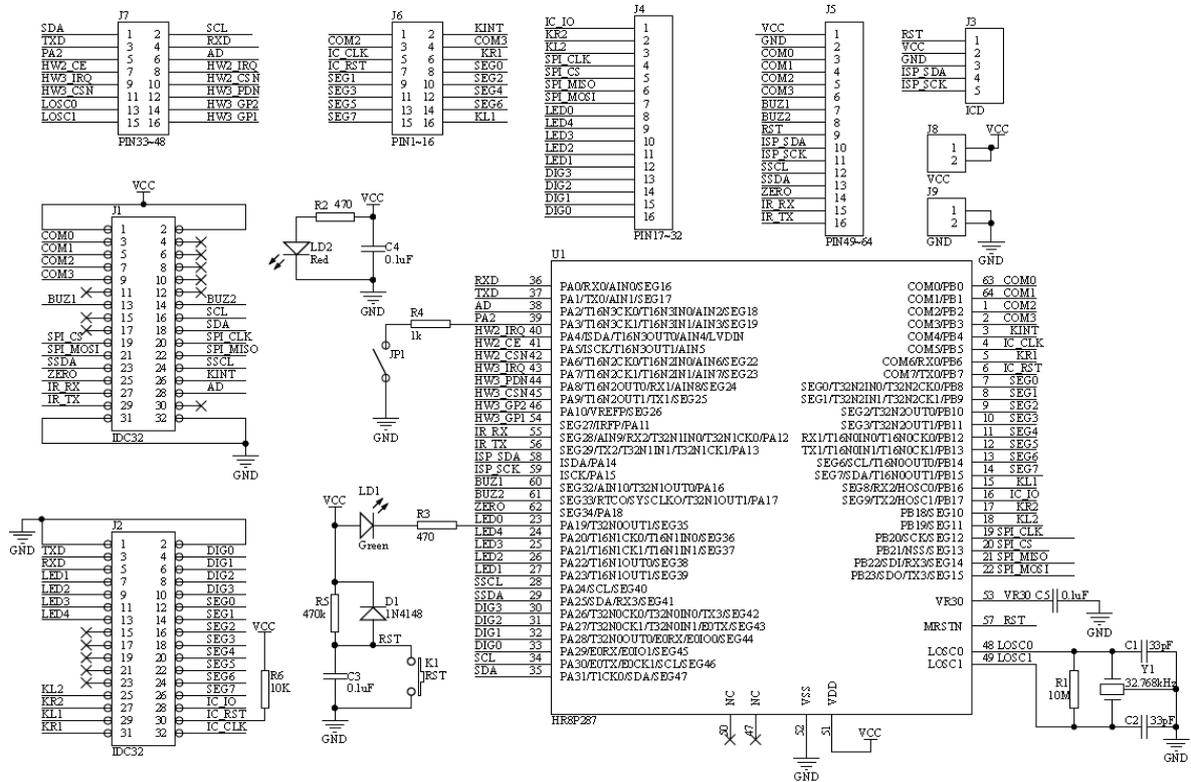


图 3-23 HRSDK-GDB-8P287 原理图

3. 13 HRSDK-GDB-8P506

3. 13. 1 使用说明

◆ HRSDK-GDB-8P506 子板

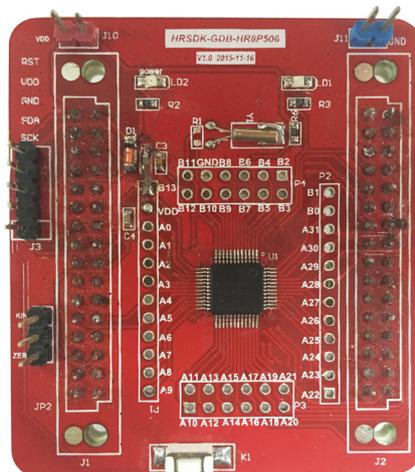


图 3-24 HRSDK-GDB-8P506 实物图

3. 13. 2 原理图

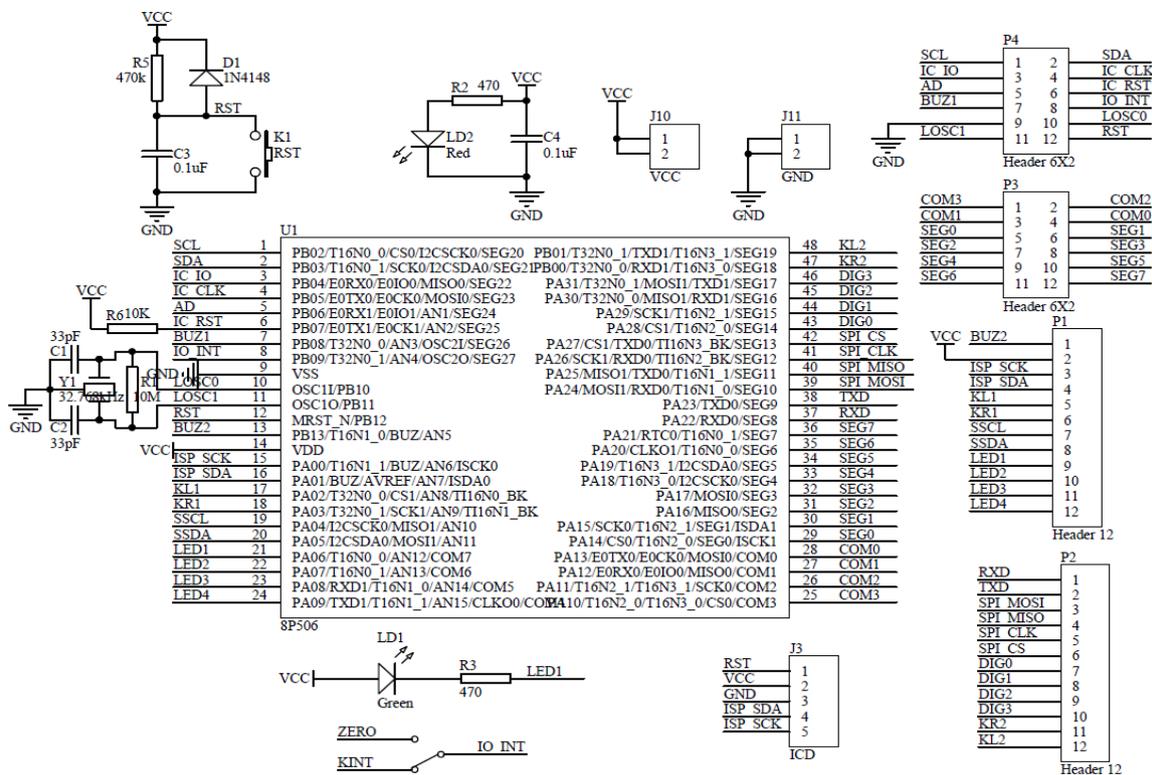


图 3-25 HRSDK-GDB-8P506 原理图

3. 14 HRSDK-GDB-7P295x

3. 14. 1 使用说明

◆ HRSDK-GDB-7P295x 子板

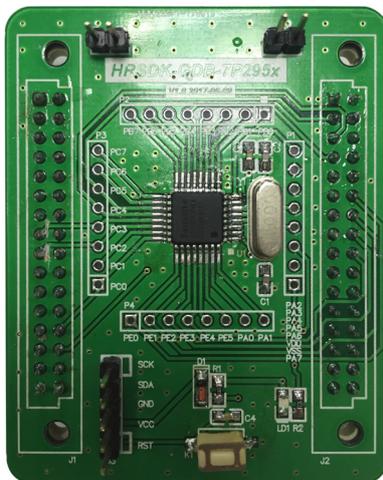
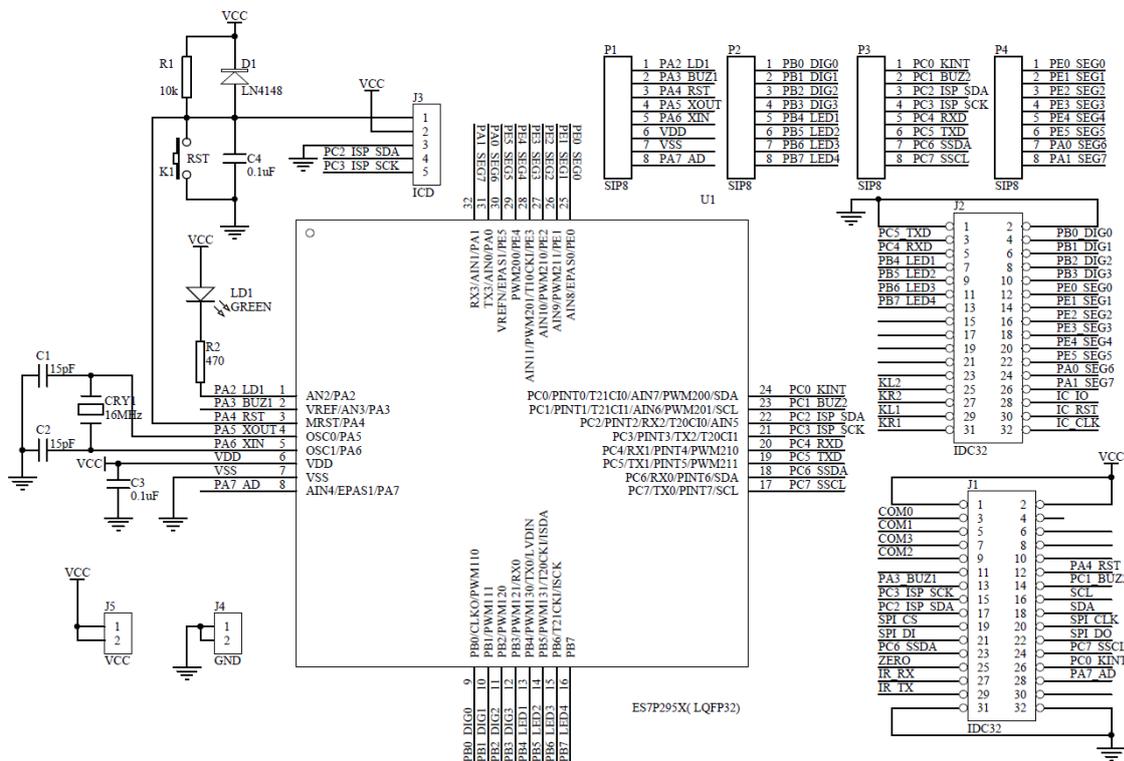


图 3-26 HRSDK-GDB-7P295x 实物图

3. 14. 2 原理图



3. 15 HRSDK-GDB-7P179

3. 15. 1 使用说明

◆ HRSDK-GDB-7P179 子板

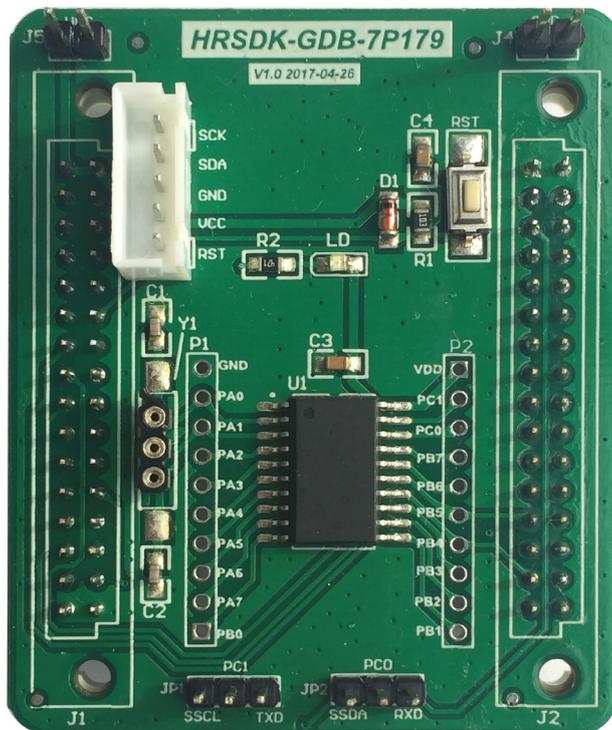


图 3-28 HRSDK-GDB-7P179 实物图

3. 15. 2 原理图

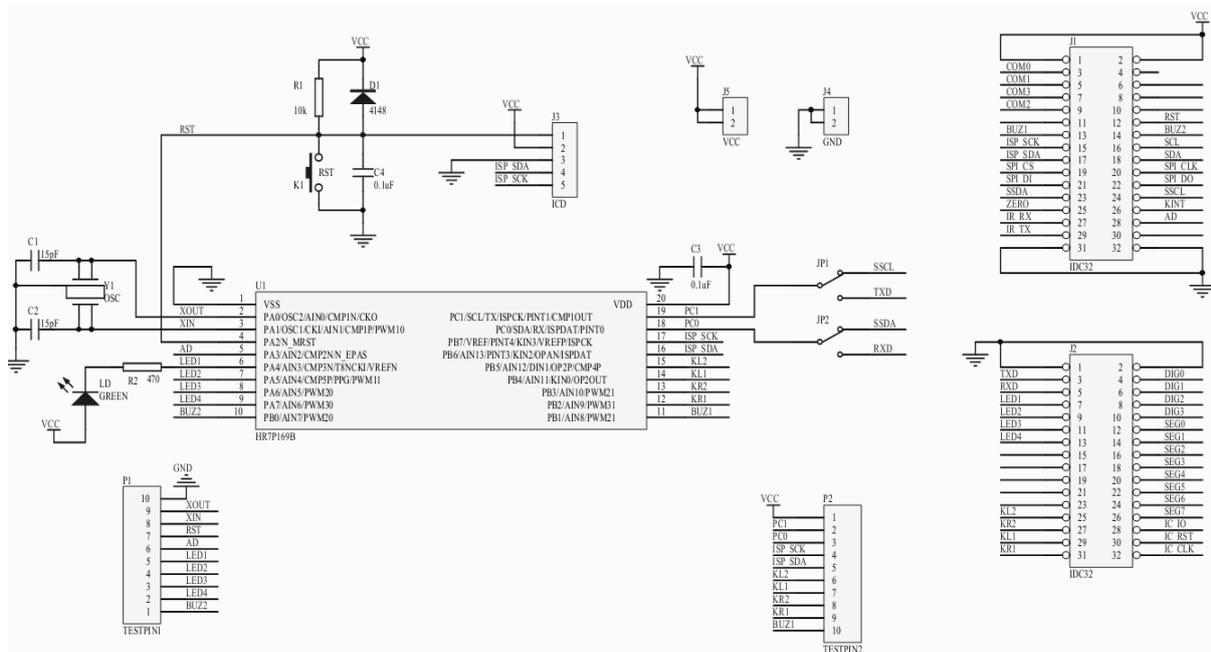


图 3-29 HRSDK-GDB-7P179 原理图模块功能

3. 16 HRSDK-GDB-8P286

3. 16. 1 使用说明

◆ HRSDK-GDB-8P286 子板

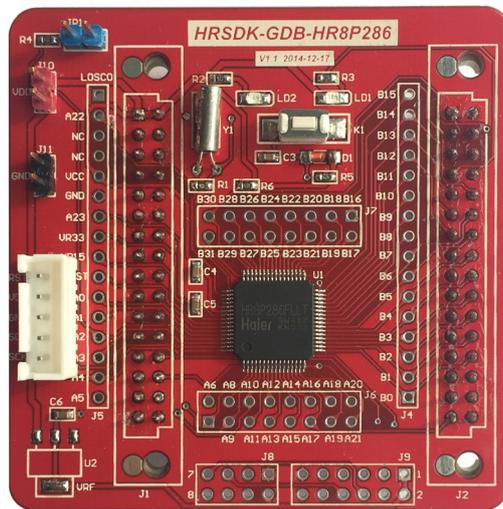


图 3-30 HRSDK-GDB-8P286 实物图

3. 16. 2 原理图

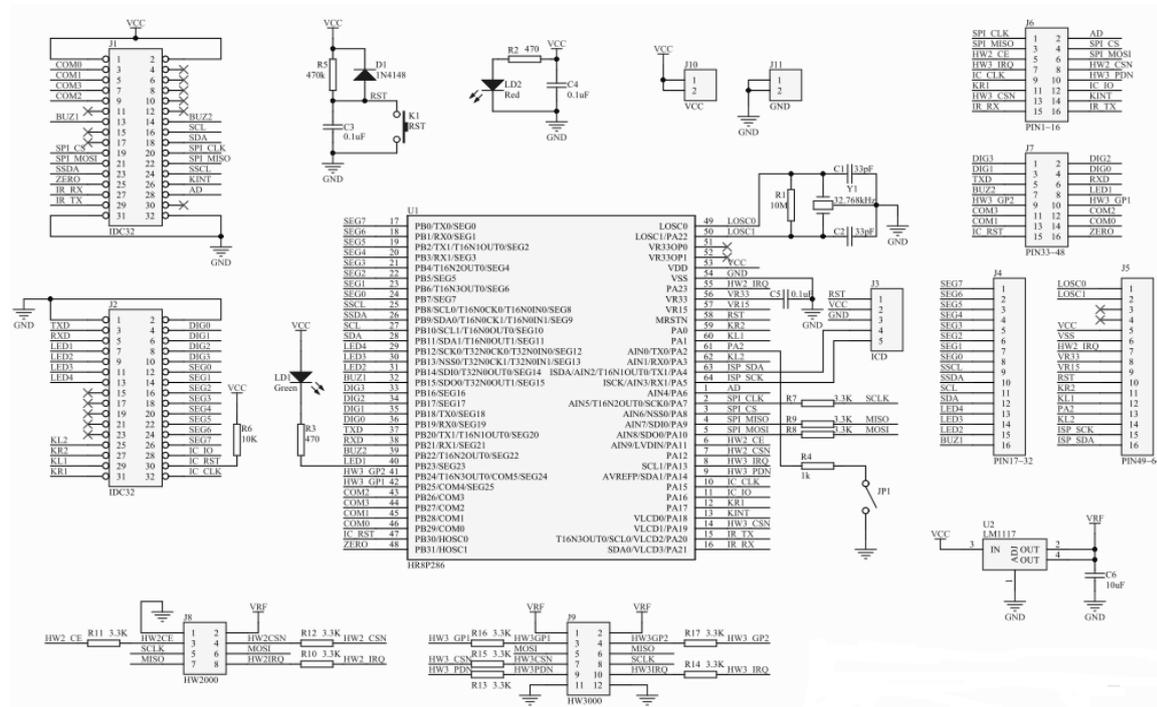


图 3-31 HRSDK-GDB-8P286 原理图模块功能

3. 17 HRSDK-GDB- HW2170

3. 17. 1 使用说明

◆ HW2170 子板

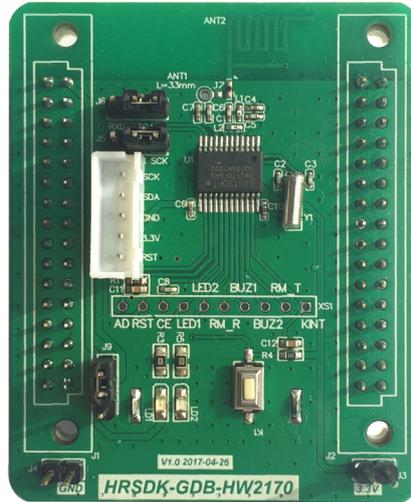


图 3-32 HW2170 实物图

3. 17. 2 原理图

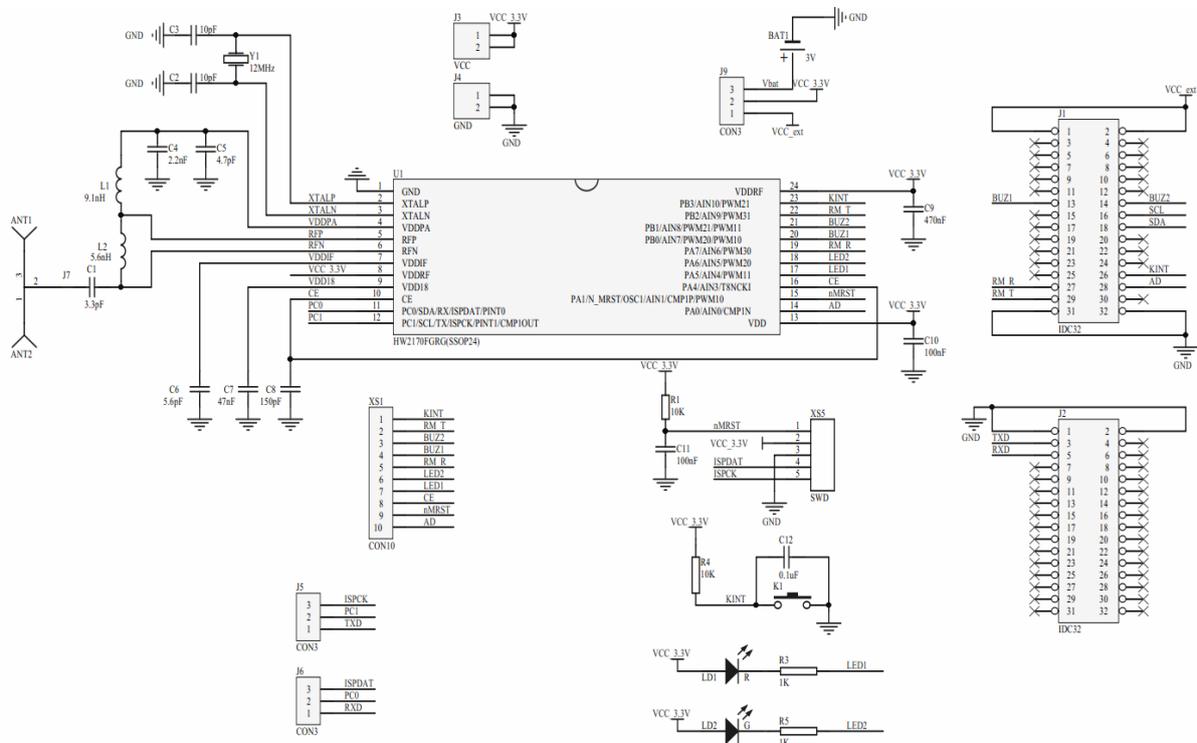


图 3-33 HW2170 原理图模块功能

3. 18 HRSDK-GDB-8P296R

3. 18. 1 使用说明

◆ HRSDK- GDB-8P296R 子板



图 3-34 HRSDK- GDB-8P296R 实物图

3. 18. 2 原理图

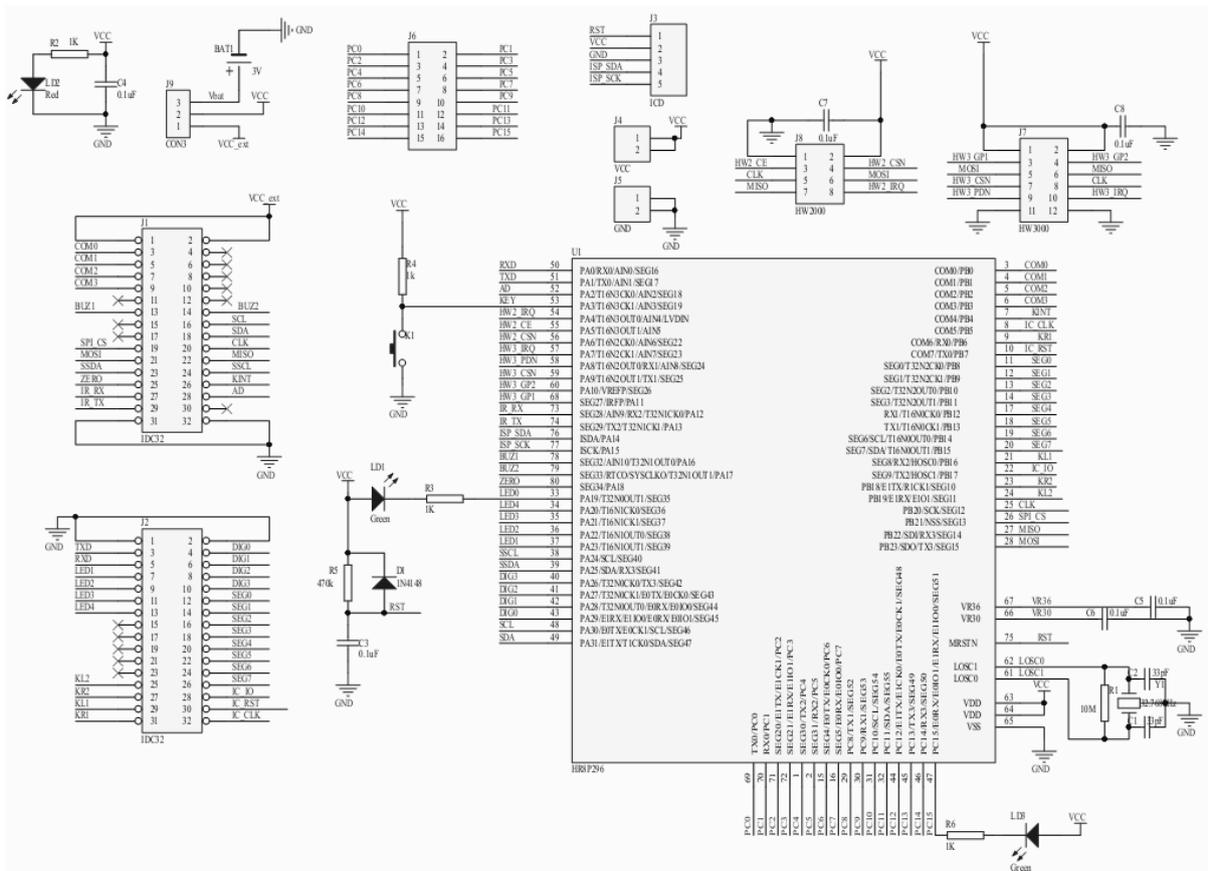


图 3-35 HRSDK- GDB-8P296R 原理图模块功能

3. 19 HRSDK-GDB-HR7P169BR

3. 19. 1 使用说明

◆ HRSDK- GDB-HR7P169BR 子板

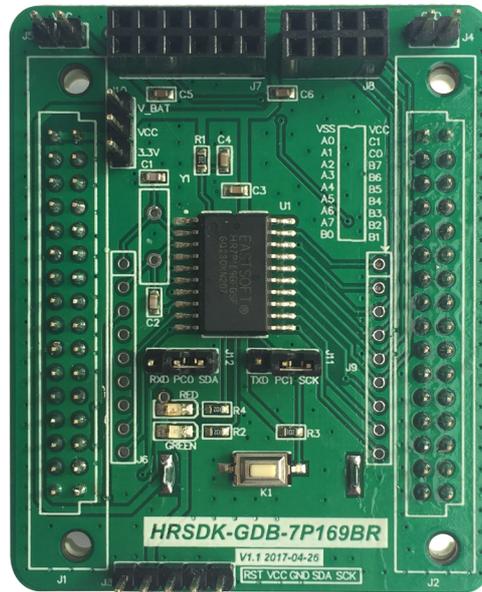


图 3-36 HRSDK- GDB-HR7P169BR 实物图

3. 19. 2 原理图

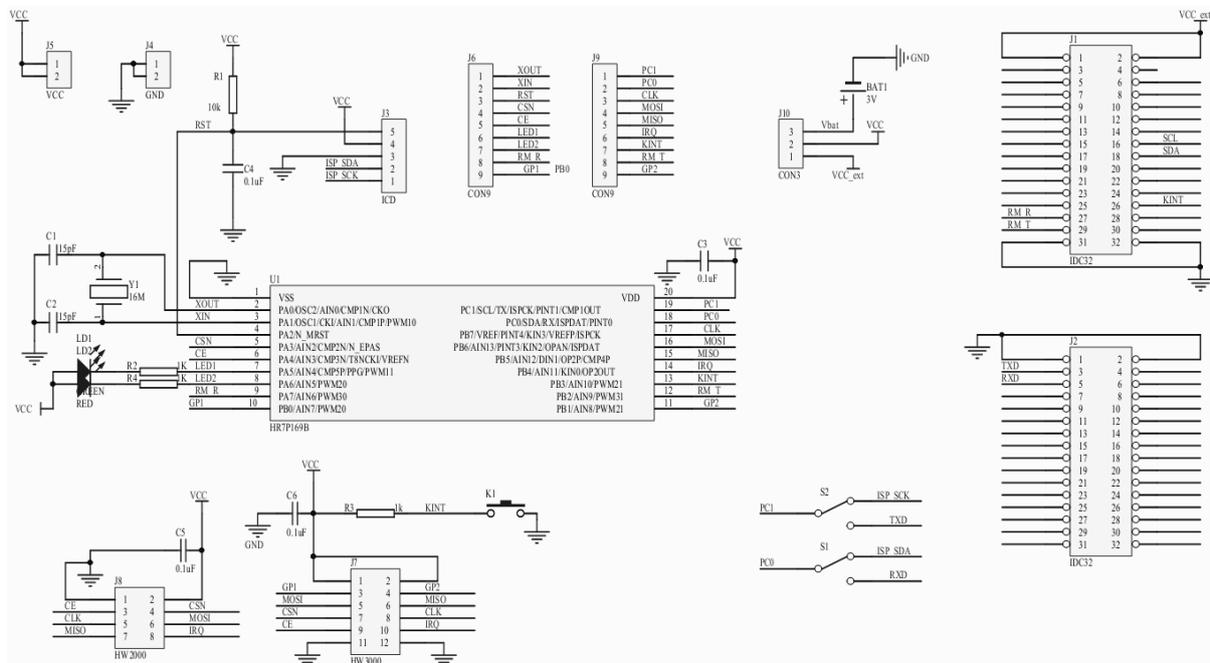


图 3-37 HRSDK- GDB-HR7P169BR 原理图模块功能

3. 20 HRSDK-GDB-ES8P508x

3. 20. 1 使用说明

◆ HRSDK- GDB-ES8P508x 子板

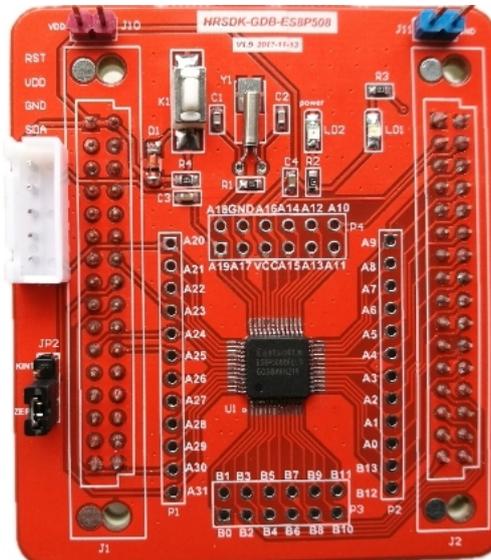


图 3-38 HRSDK- GDB-ES8P508x 实物图

3. 20. 2 原理图

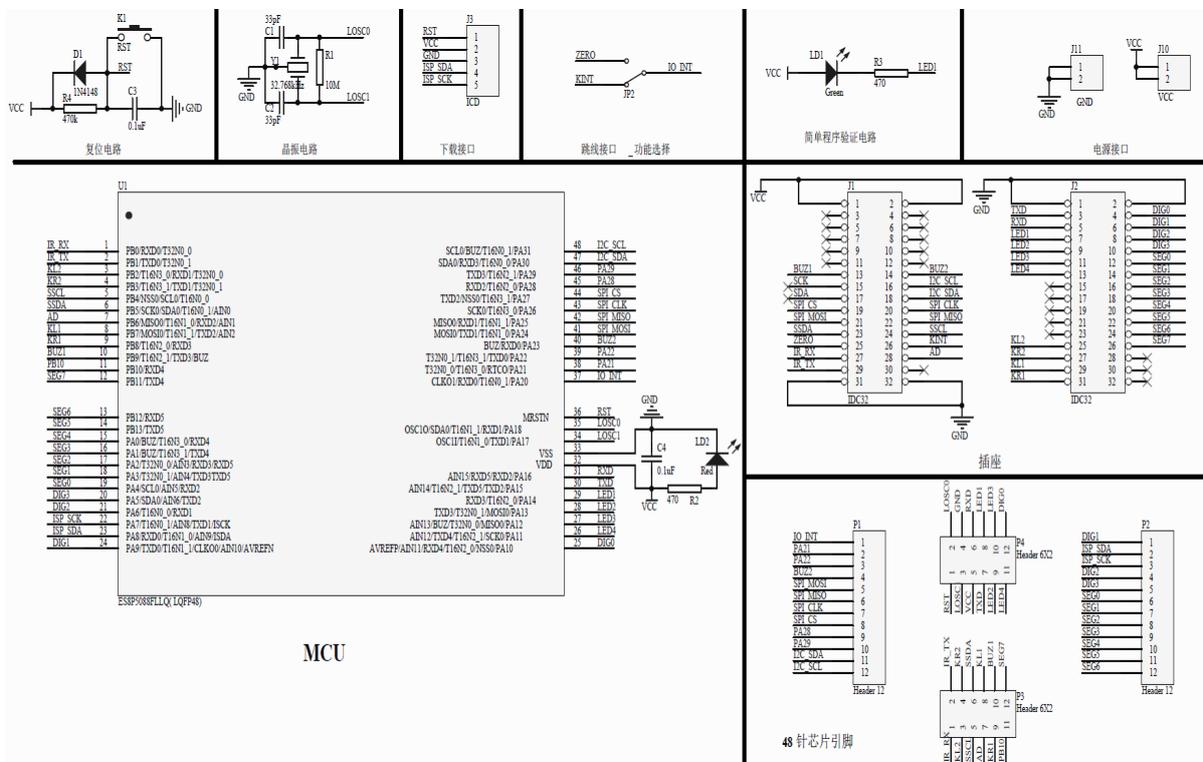


图 3-39 HRSDK- GDB-ES8P508x 原理图模块功能

3. 21 ESSDK-GDB-ES7P003

3. 21. 1 使用说明

◆ ESSDK- GDB-ES7P003 子板



图 3-40 ESSDK-GDB-ES7P003 实物图

3. 21. 2 原理图

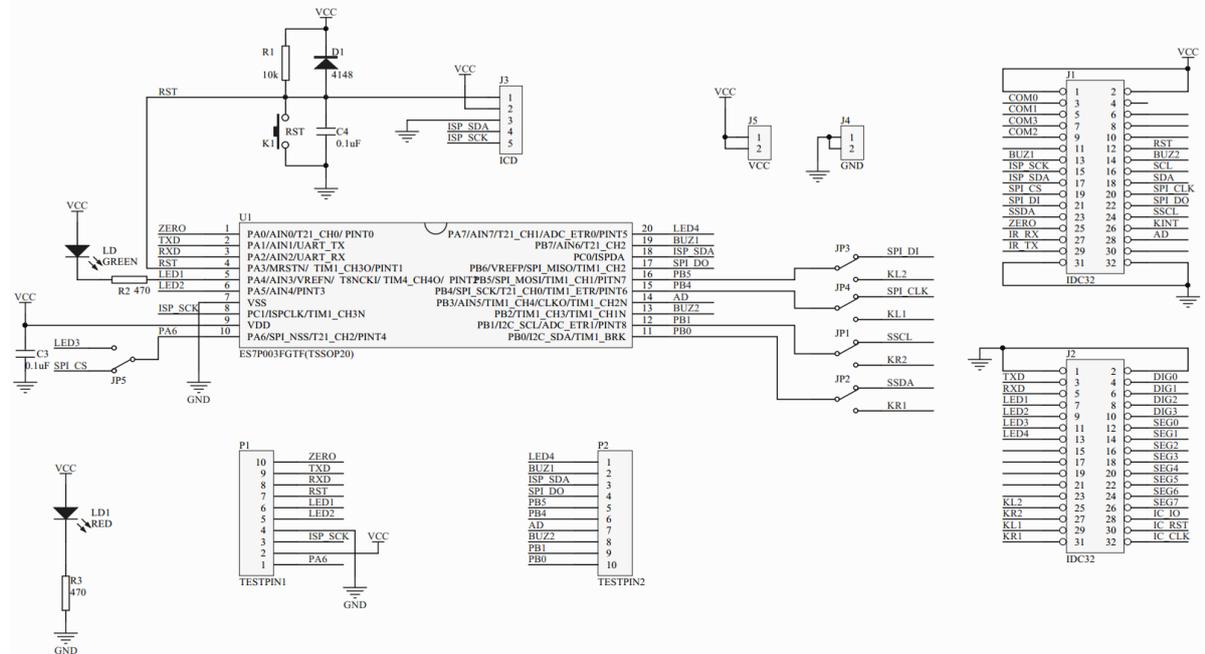


图 3-41 ESSDK-GDB-ES7P003 原理图模块功能

3. 22 HRSDK-GDB-SSC1645

3. 22. 1 使用说明

◆ HRSDK-GDB-SSC1645 子板

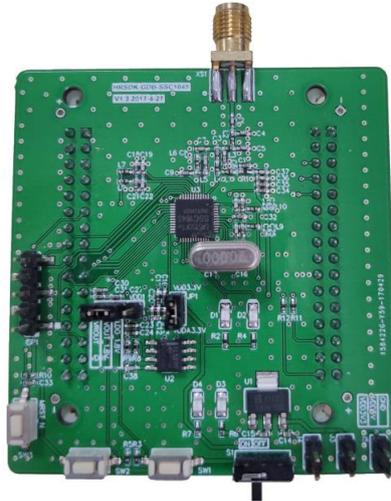


图 3-42 HRSDK-GDB-SSC1645 实物图

3. 22. 2 原理图

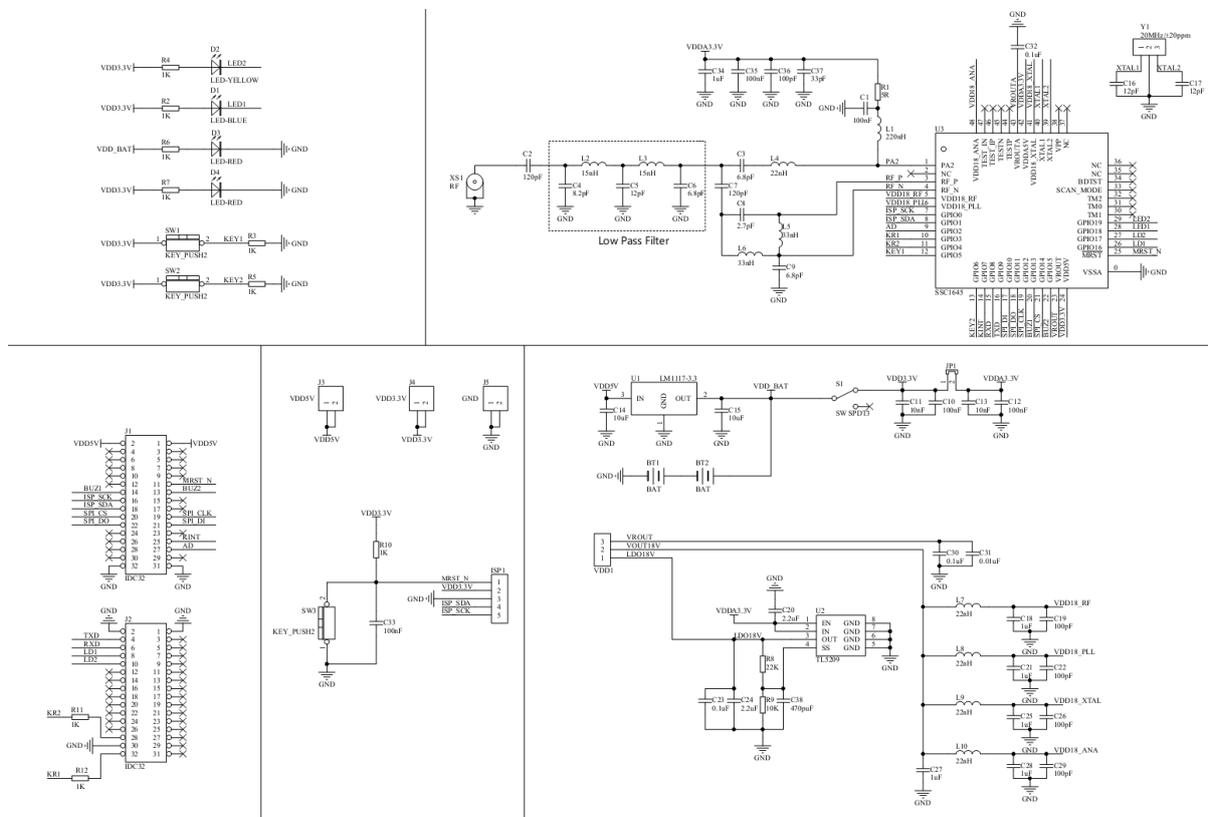


图 3-43 HRSDK-GDB-SSC1645 原理图模块功能

3.23 ES-GDB-ES7P169C

3.23.1 使用说明

◆ ES-GDB-ES7P169C 子板

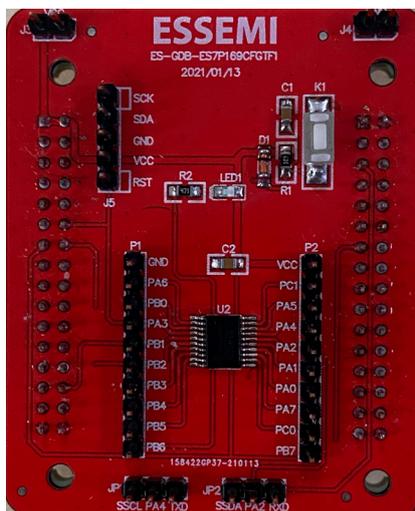


图 3-44 ES-GDB-ES7P169C 实物图

3.23.2 原理图

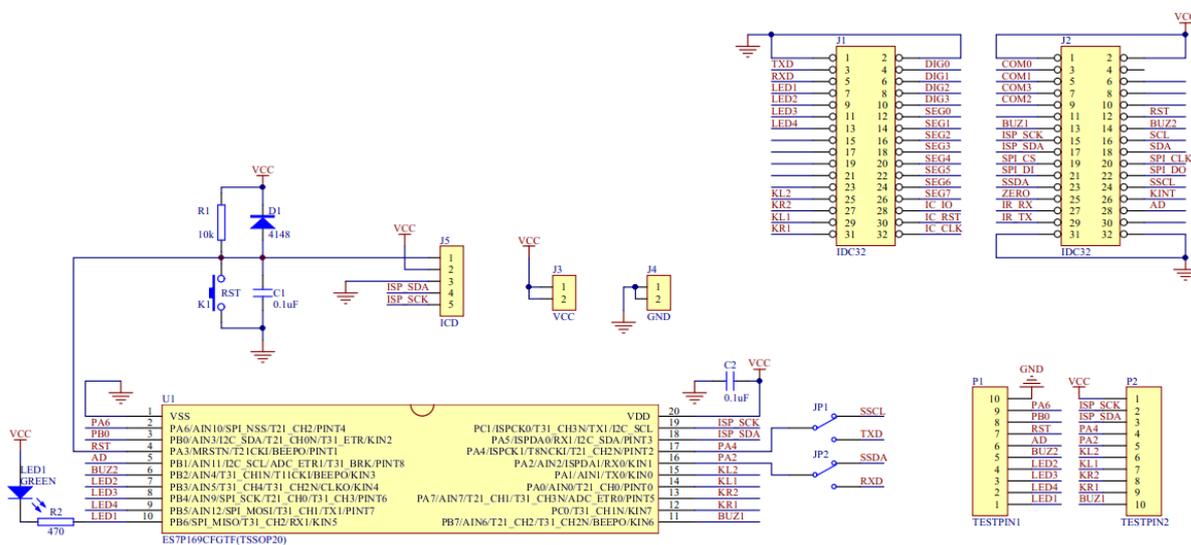


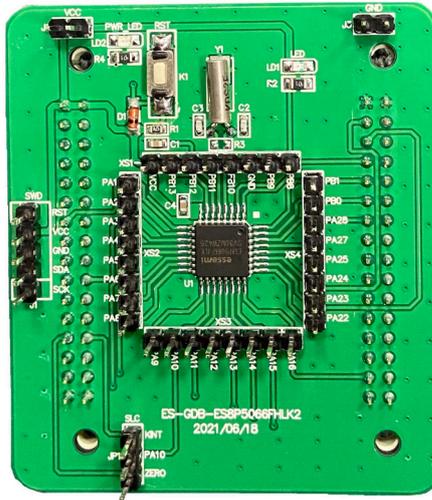
图 3-45 ES-GDB-ES7P169C 原理图模块功能

注：PA2 作 UART RXD 时，对应的 TXD 为 PA1，所以需要串口通信时，用短接帽将 JP2 的 PA2 接 RXD，JP1 的 TXD 用杜邦线接 P2 的 PA1。SSCL 和 SSDA 不受影响。

3. 24 ES-GDB-ES8P5066

3. 24. 1 使用说明

◆ ES-GDB-ES8P5066 子板



◆ 图 3-46 ES-GDB-ES8P5066 实物图

3. 24. 2 原理图

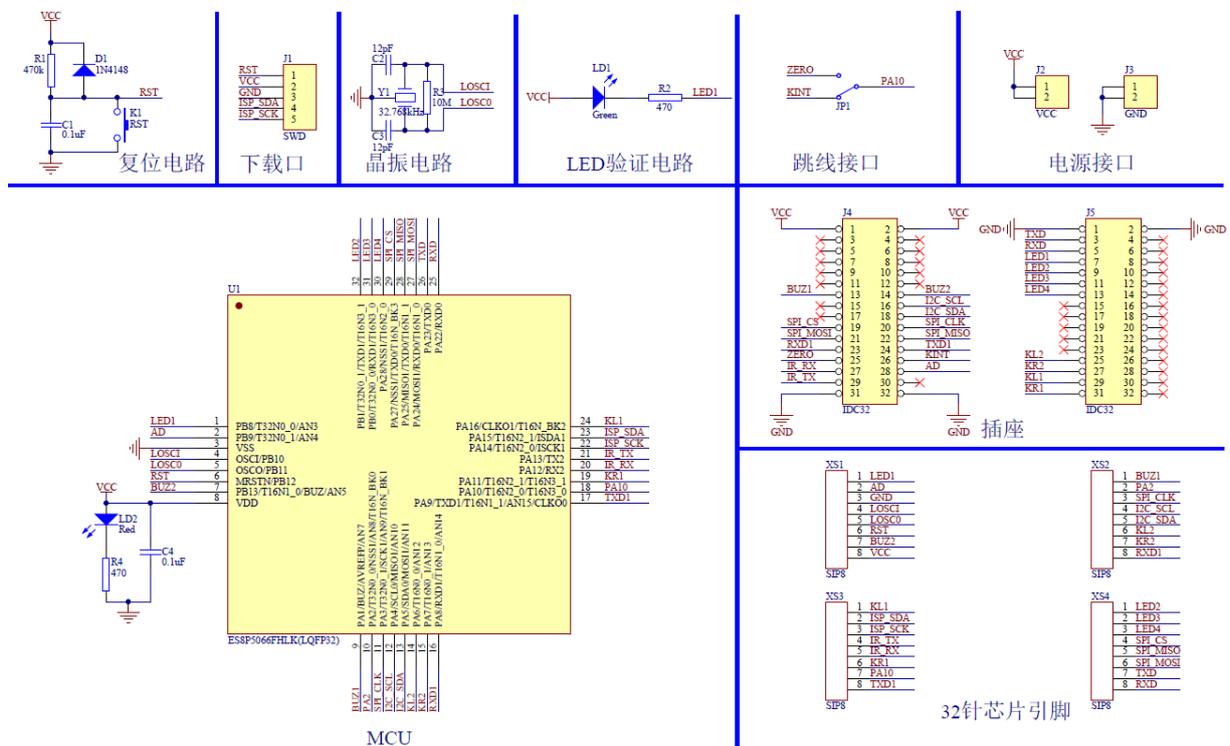


图 3-47 ES-GDB-ES8P5066 原理图模块功能

3. 25 LIGHT（指示灯）

3. 25. 1 模块说明

该模块主要演示 I/O 控制 LED 指示灯状态变化。

3. 25. 2 硬件设计

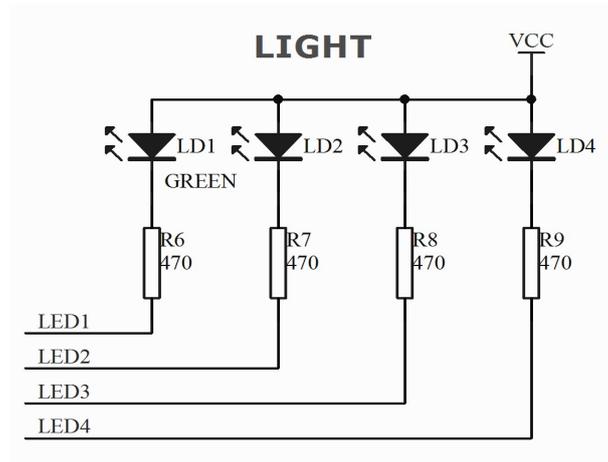


图 3-42 LIGHT 原理图

3. 25. 3 软件设计（light_demo）

3. 25. 3. 1 模块函数

无模块函数，直接在主程序中控制。

3. 25. 3. 2 流程图

无流程图。

3. 26 MOTOR（步进电机）

3. 26. 1 模块说明

该模块主要演示步进电机的 4 相 8 拍驱动。

4 个 I/O 分别对应步进电机的 4 线，按照 8 拍相序编码循环输出电平，控制电机旋转。

按键 K5 切换电机运行状态，正转-停止-反转-停止。

3. 26. 2 硬件设计

步进电机型号：28BYJ48。

MCU 的 IO 口通过 ULN2003 驱动，再连接到相应的电机接口。

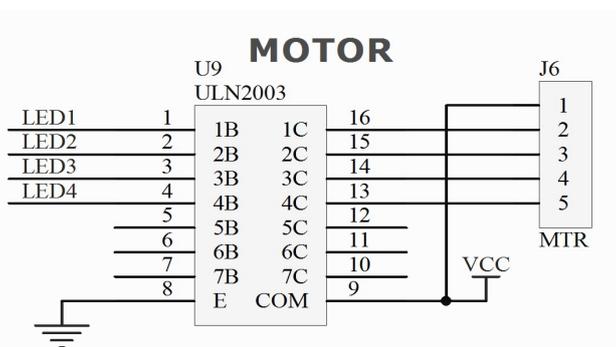


图 3-43 MOTOR 原理图

3. 26. 3 软件设计（motor_demo）

当对步进电机施加一系列连续不断的控制脉冲时，它可以连续不断地转动。每一个脉冲信号对应步进电机的某一相或两相绕组的通电状态改变一次，也就对应转子转过一定的角度（一个步距角）。当通电状态的改变完成一个循环时，转子转过一个齿距。4 相步进电机可以在不同的通电方式下运行，常见的通电方式有单 4 拍（单相绕组通电：A-B-C-D-A...），双 4 拍（双相绕组通电：AB-BC-CD-DA-AB...），8 拍（A-AB-B-BC-C-CD-D-DA-A...）。

电机顺时针旋转（正转）的 8 拍序列，端口控制电平如下：

| 步数 | A | B | C | D | 十六进制（正转） |
|----|---|---|---|---|----------|
| 1 | 1 | 0 | 0 | 0 | 0x08 |
| 2 | 1 | 1 | 0 | 0 | 0x0C |
| 3 | 0 | 1 | 0 | 0 | 0x04 |
| 4 | 0 | 1 | 1 | 0 | 0x06 |
| 5 | 0 | 0 | 1 | 0 | 0x02 |
| 6 | 0 | 0 | 1 | 1 | 0x03 |
| 7 | 0 | 0 | 0 | 1 | 0x01 |
| 8 | 1 | 0 | 0 | 1 | 0x09 |

表 3-1 电机正转 8 拍序列码表

逆时针旋转（反转）和正转相序相反，定义编码数组如下：

```
const uchar MotorRW[8] = {0x08, 0x0C, 0x04, 0x06, 0x02, 0x03, 0x01, 0x09}; //正转编码表
```

```
const uchar MotorFW[8] = {0x09, 0x01, 0x03, 0x02, 0x06, 0x04, 0x0C, 0x08}; //反转编码表
```

3. 26. 3. 1 模块函数

无模块函数，直接在主程序中控制。

3. 26. 3. 2 流程图

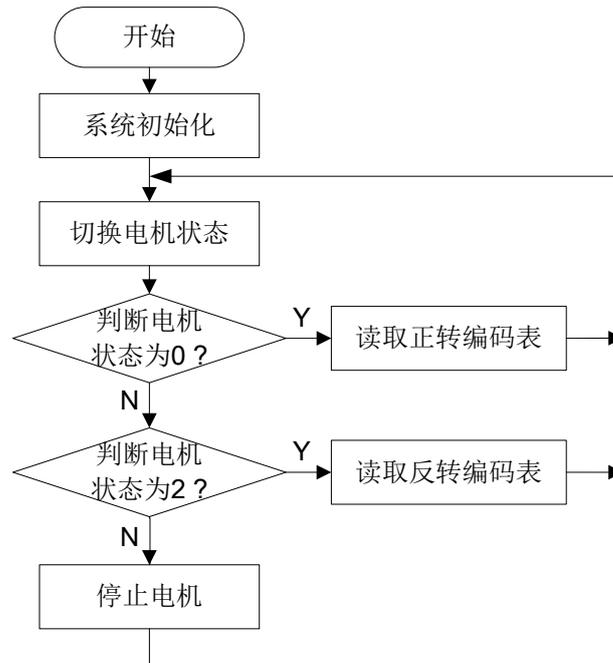


图 3-44 MOTOR 主程序流程图

3.27 ZERO（过零检测）

3.27.1 模块说明

该模块主要演示过零信号检测及处理功能。

过零信号由 AMCU 提供，过零信号触发 MCU 的外部沿中断 PINT，然后 MCU 处理过零事件，通过指示灯演示效果。

- (1) LED1 同步输出过零方波。
- (2) LED2 显示闪烁周期为过零周期的 50 倍。

3.27.2 硬件设计

由 AMCU 产生的过零信号，直连至 MCU 的 PINT 引脚。

3.27.3 软件设计（zero_demo）

3.27.3.1 模块函数

无模块函数。

对于过零事件，如果实时性要求较高，应在中断里进行相应的处理。

3.27.3.2 流程图

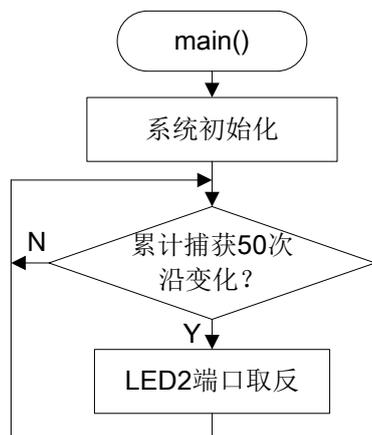


图 3-45 过零检测主程序流程图

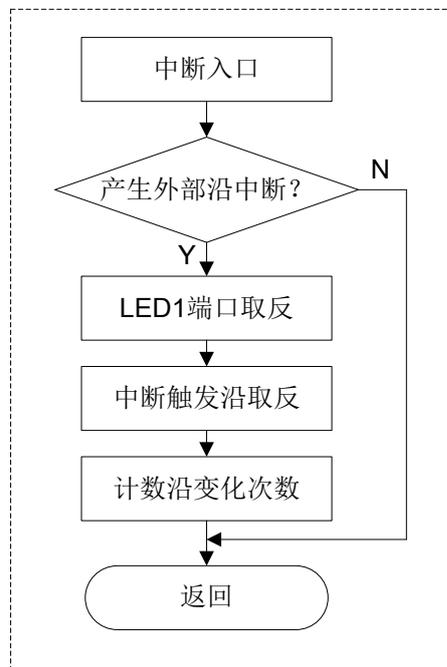


图 3-46 外部中断程序流程图

3. 28 ADC（模数转换）

3. 28. 1 模块说明

该模块主要演示 AD 转换功能和数字滤波算法。

采用 10KΩ 可调电位器，通过 4 位数码管显示 ADC 结果（10bits，右对齐格式）。

按键 K5 切换显示模式，分别显示 ADC 数值和模拟电压值。

注：由于 HR7P170 不支持 LED 数码管显示，因此通过示波器获取 PB0 端口(PWM20)输出的 PWM 波形的周期值 T(单位为 us)，每 10ms 刷新一次，然后通过公式 $V_{adc} = (T/8 - 1) * (VDD/256)$ 将其转换为模拟电压值 V_{adc} (单位为 V)。

3. 28. 2 硬件设计

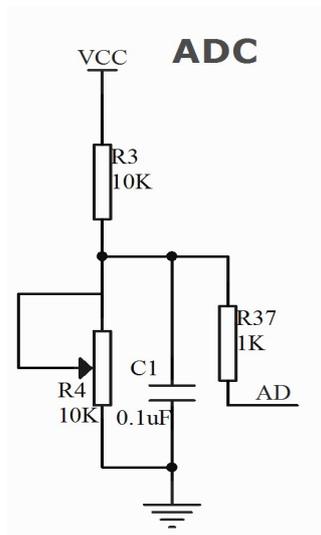


图 3-47 ADC 原理图

3. 28. 3 软件设计（adc_demo）

3. 28. 3. 1 模块函数

(1) ADC 初始化

/******

函数名: void ADCInit(void)

描述: ADC 初始化子程序

输入值: 无

输出值: 无

返回值: 无

*****/

(2) 读取 ADC 数据

/******

函数名: void ADCRead(uchar channel, uint *dat)

描述: ADC 转换程序, 转换 18 次, 去掉最小值和最大值, 求 16 次平均值

输入值: channel—模拟输入通道

输出值: dat—ADC 结果

返回值: 无

*****/

3.28.3.2 流程图

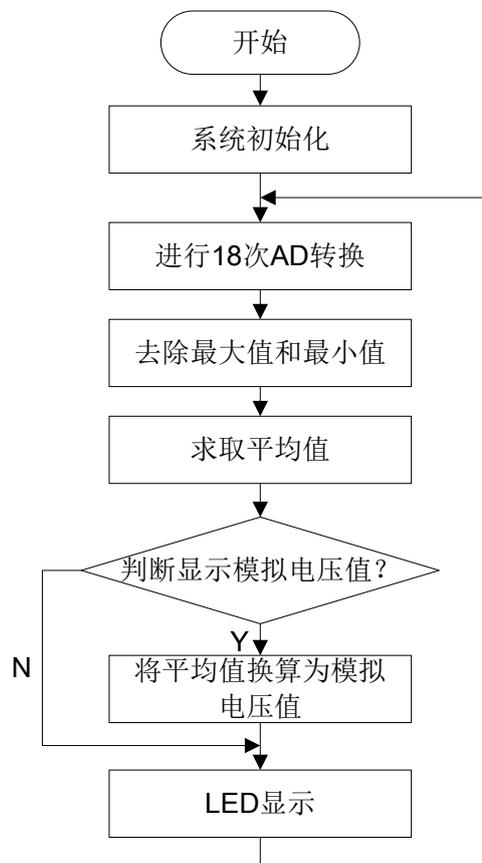


图 3-48 ADC 程序流程图

3. 29 BUZZER（蜂鸣器）

3. 29. 1 模块说明

该模块主要演示 PWM 输出的频率变化功能。

蜂鸣器根据不同的频率发出不同的音调，实现乐曲演奏。

- (1) 单击 K4: 演奏《两只老虎》
- (2) 单击 K3: 演奏《新年好》
- (3) 单击 K2: 演奏《粉刷匠》
- (4) 单击 K1: 演奏《拔萝卜》
- (5) 单击 K5: 演奏音符音阶

注：由于 HR7P156 和 HR7P160 无矩阵按键功能，因此通过 K5 按键代替。每单击 K5 一次，将循环演奏以上乐曲。

3. 29. 2 硬件设计

蜂鸣器选择无源电磁式交流蜂鸣器。

BUZ1 控制三极管 Q1 使能（高电平有效），BUZ2 输出 PWM，通过三极管 Q2 驱动蜂鸣器。

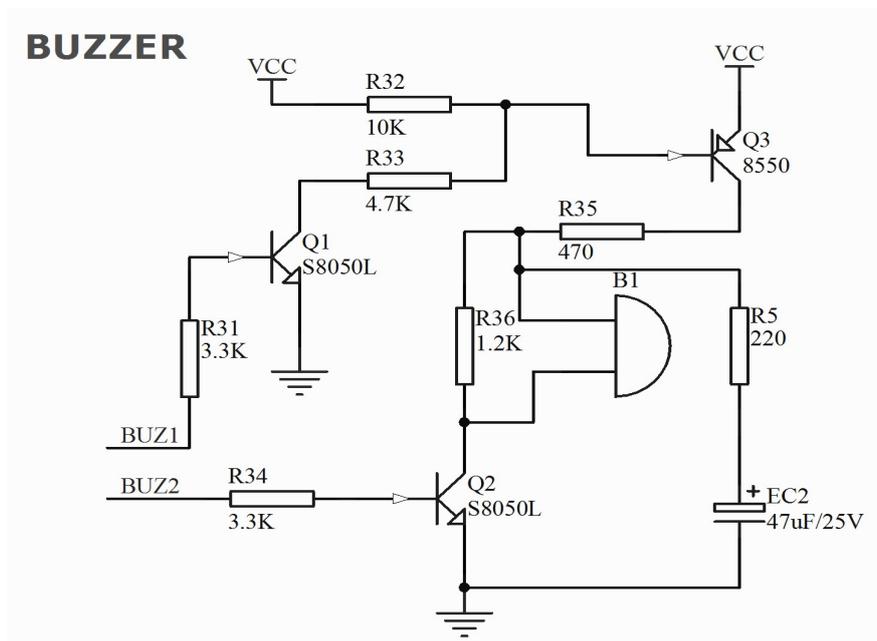


图 3-49 BUZZER 原理图

3. 29. 3 软件设计 (buzzer_demo)

定时器 T8Px 设为 PWM 模式，输出 PWM 波形，计算公式如下：

$$\text{PWM 周期} = (\text{T8PxP} + 1) \times \text{指令周期分频值} \times \text{Tosc} \times \text{T8PxP 预分频值}$$

$$\text{PWM 脉宽} = (\text{TEXL:TEXC} < 5:4 >) \times \text{Tosc} \times \text{T8PxP 预分频值}$$

3. 29. 3. 1 模块函数

(1) PWM 初始化

/******

函数名: void PWMInit(void)

描 述: 显示初始化子程序

输入值: 无

输出值: 无

返回值: 无

*****/

(2) PWM 输出

/******

函数名: void PWMOutput(uint period, uchar duty)

描 述: PWM 输出子程序

输入值: period—周期，单位 us，范围 4us~1024us (精度 4us)

duty—占空比，百分数

输出值: 无

返回值: 无

*****/

3.29.3.2 流程图

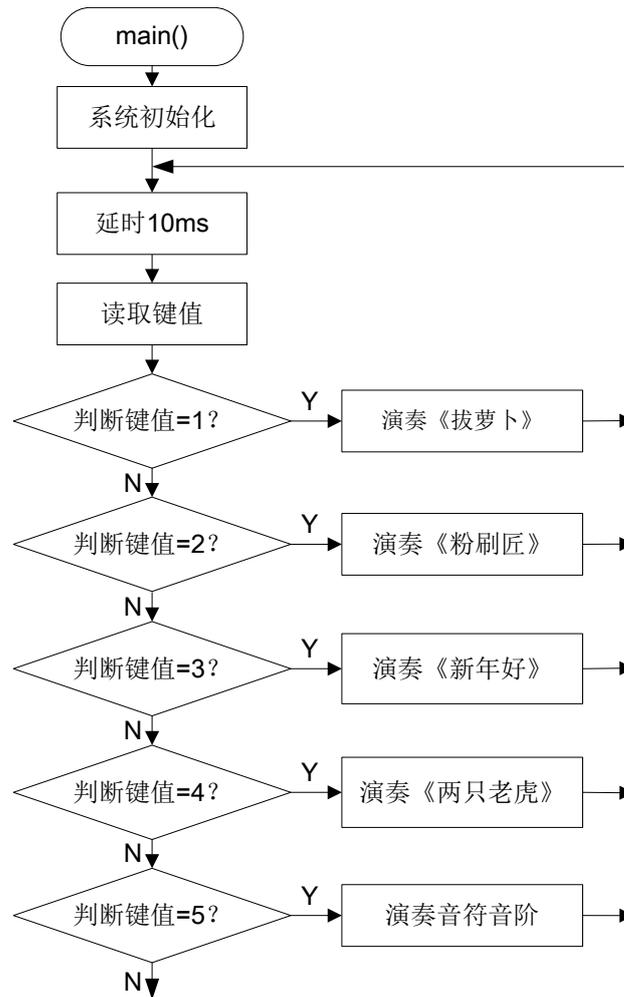


图 3-50 BUZZER 主程序流程图

3. 30 LED（数码管）

3. 30. 1 模块说明

该模块主要演示大电流口线的驱动功能。

循环显示数字“0000”“1111”....“9999”，每隔 1s 跳动一次。

3. 30. 2 硬件设计

4 位共阴 LED 数码管，0.36 英寸。

型号：LG3641AH。

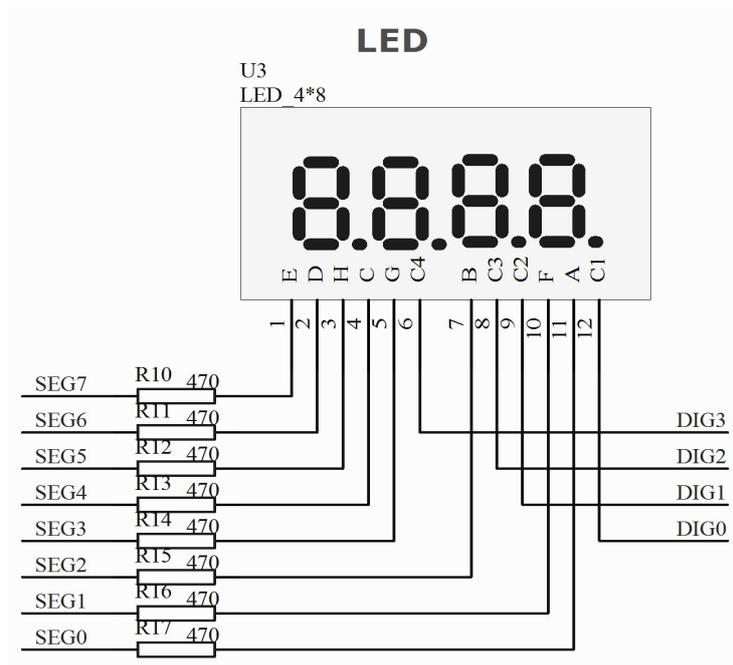


图 3-51 LED 原理图

3. 30. 3 软件设计

3. 30. 3. 1 模块函数

(1) LED 初始化

```
/******
```

函数名: void LEDInit(void)

描述: LED 显示初始化子程序

输入值: 无

输出值: 无

返回值: 无

```
*****/
```

(2) LED 显示

/******

函数名: void LEDDisplayData(uchar com, uchar dat)

描 述: LED 显示驱动子程序

输入值: com—显示位码

 dat—显示字符

输出值: 无

返回值: 无

*****/

3.30.3.2 流程图 (led_demo)

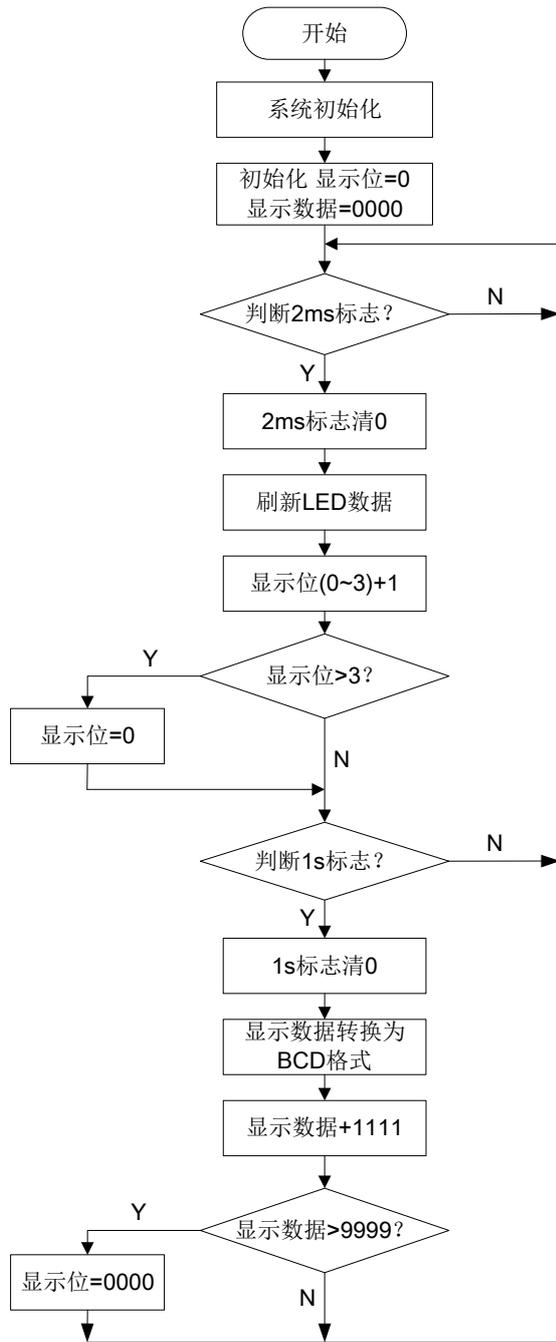


图 3-52 LED 主程序流程图

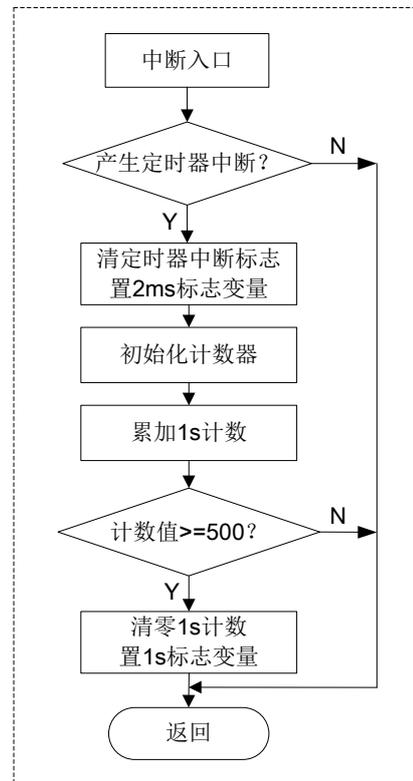


图 3-53 T8N 中断服务程序流程图

3.31 LCD（液晶）

3.31.1 模块说明

该模块主要演示 LCD 控制驱动功能。

上电显示“12:00”，然后显示实时时钟

- (1) 单击 K3：调节小时值
- (2) 单击 K4：调节分钟值
- (3) 单击 K1：切换显示模式，“小时:分钟” / “分钟:秒”

3.31.2 硬件设计

采用 4 位段码式 LCD，4COM * 8SEG。

型号：JRT-20491-RP-1。

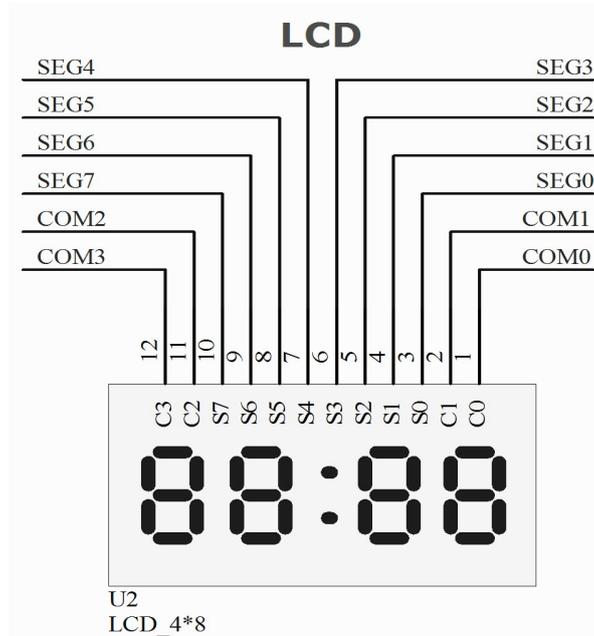


图 3-54 LCD 原理图

3.31.3 软件设计 (lcd_demo)

3.31.3.1 模块函数

(1) LCD 初始化

/******

函数名: void LCDInit(void)

描 述: LCD 显示初始化子程序

输入值: 无

输出值: 无

返回值: 无

*****/

(2) LCD 显示

/******

函数名: void LCDDisplayData(uchar high, uchar low, uchar colon)

描 述: LCD 显示驱动子程序

输入值: high_byte—显示高字节

low_byte—显示低字节

colon—显示冒号

输出值: 无

返回值: 无

*****/

3.31.3.2 流程图

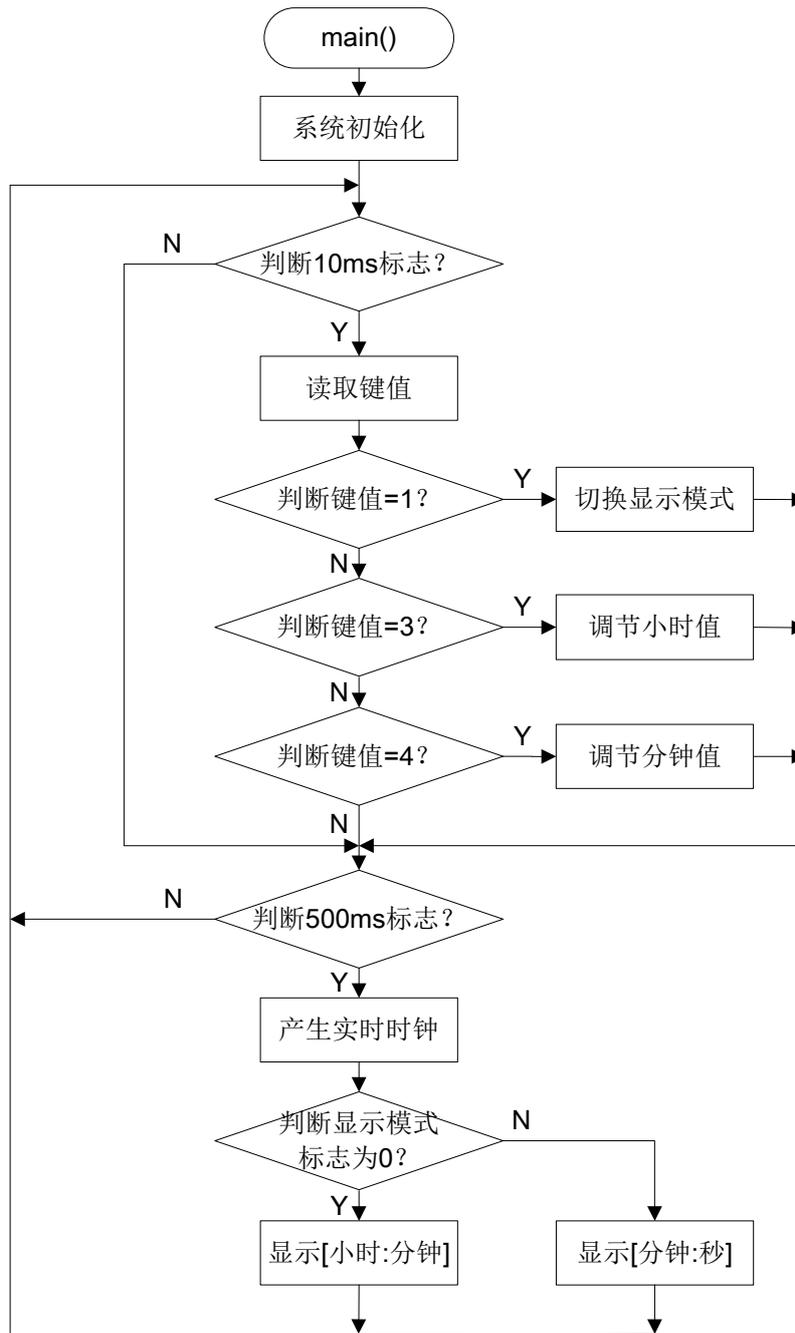


图 3-55 LCD 主程序流程图

3. 32 KINT（独立按键）

3. 32. 1 模块功能

主要实现外部按键中断的低功耗唤醒功能。

芯片上电后 LD1 指示灯首先闪烁 5 次，然后进入低功耗模式。

K5 按键提供低功耗唤醒功能。当按下 K5 后 LD1 闪烁 5 次，而后再进入低功耗模式。

3. 32. 2 硬件设计

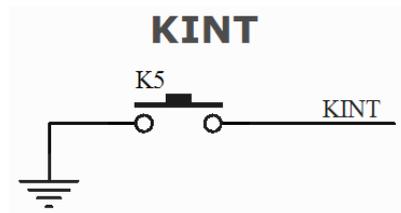


图 3-56 KINT 原理图

3. 32. 3 软件设计（key_demo）

3. 32. 3. 1 模块函数

（1）外部按键中断初始化

```
/*
*****
*/
```

函数名: void KintInit(void)

描 述: 外部按键中断初始化子程序

输入值: 无

输出值: 无

返回值: 无

```
*****
*/
```

（2）进入待机模式

```
/*
*****
*/
```

函数名: void Sleep(uchar mode)

描 述: 进入待机模式子程序

输入值: mode - IDLE 模式

输出值: 无

返回值: 无

```
*****
*/
```

3.32.3.2 流程图

无流程图。

3.33 KEY（矩阵按键）

3.33.1 模块功能

该模块主要实现行列式矩阵按键扫描的输入判别功能。

- (1) 单击 K1（按下有效），LED1 点亮
- (2) 单击 K2（释放有效），LED2 翻转
- (3) 长按 K3（按下有效），LED3 闪烁
- (4) 复按 K4 和 K2，LED4 翻转

3.33.2 硬件设计

4 个按键为：2 行输出 KR1~KR2，2 列输入 KL1~KL2。

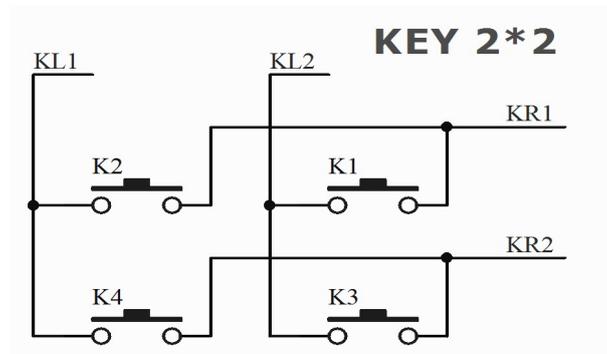


图 3-57 矩阵按键原理图

3.33.3 软件设计（key_demo）

3.33.3.1 模块函数

- (1) 按键初始化

/******

函数名: void KeyInit(void)

描述: 按键初始化子程序

输入值: 无

输出值: 无

返回值: 无

*****/

(2) 读取键值

函数名: uchar KeyGetValue(void)

描述: 按键驱动子程序, 判断按键状态, 读取键值

输入值: 无

输出值: 无

返回值: 1 - K1 (按下有效)

2 - K2 (释放有效)

3 - K3 (长按有效)

4 - K4 (复合 K2 有效)

*****/

3.33.3.2 流程图

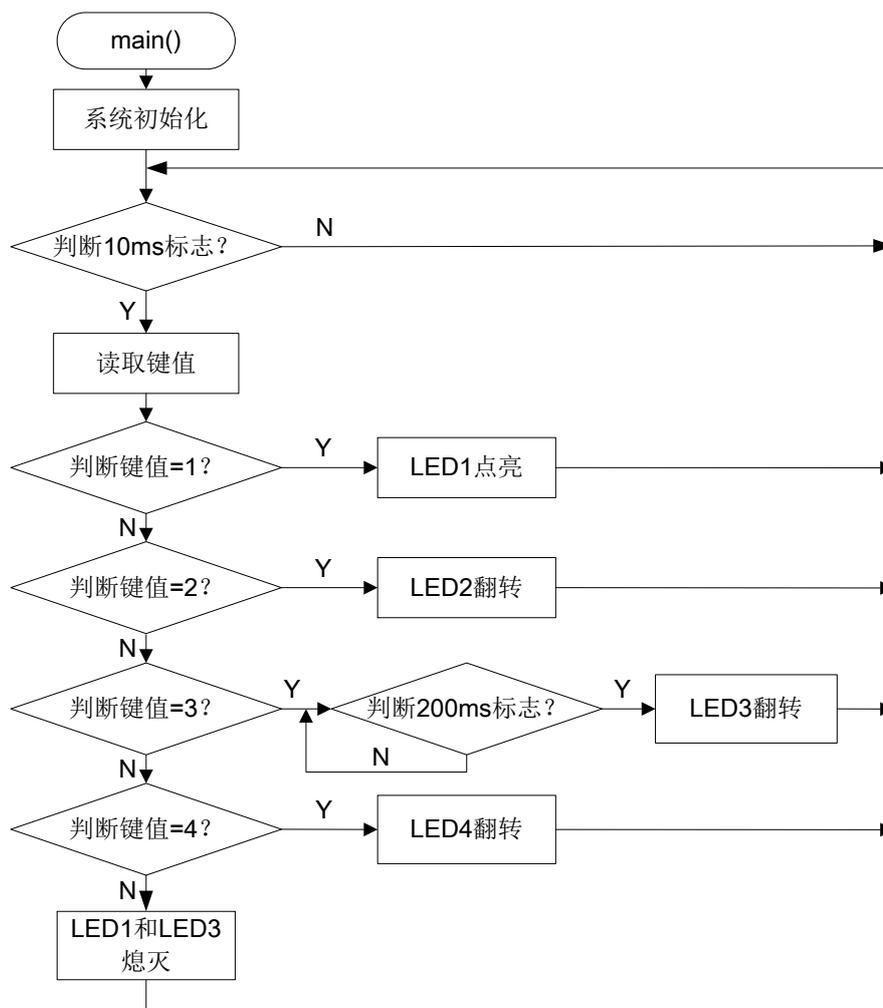


图 3-58 KEY 主程序流程图

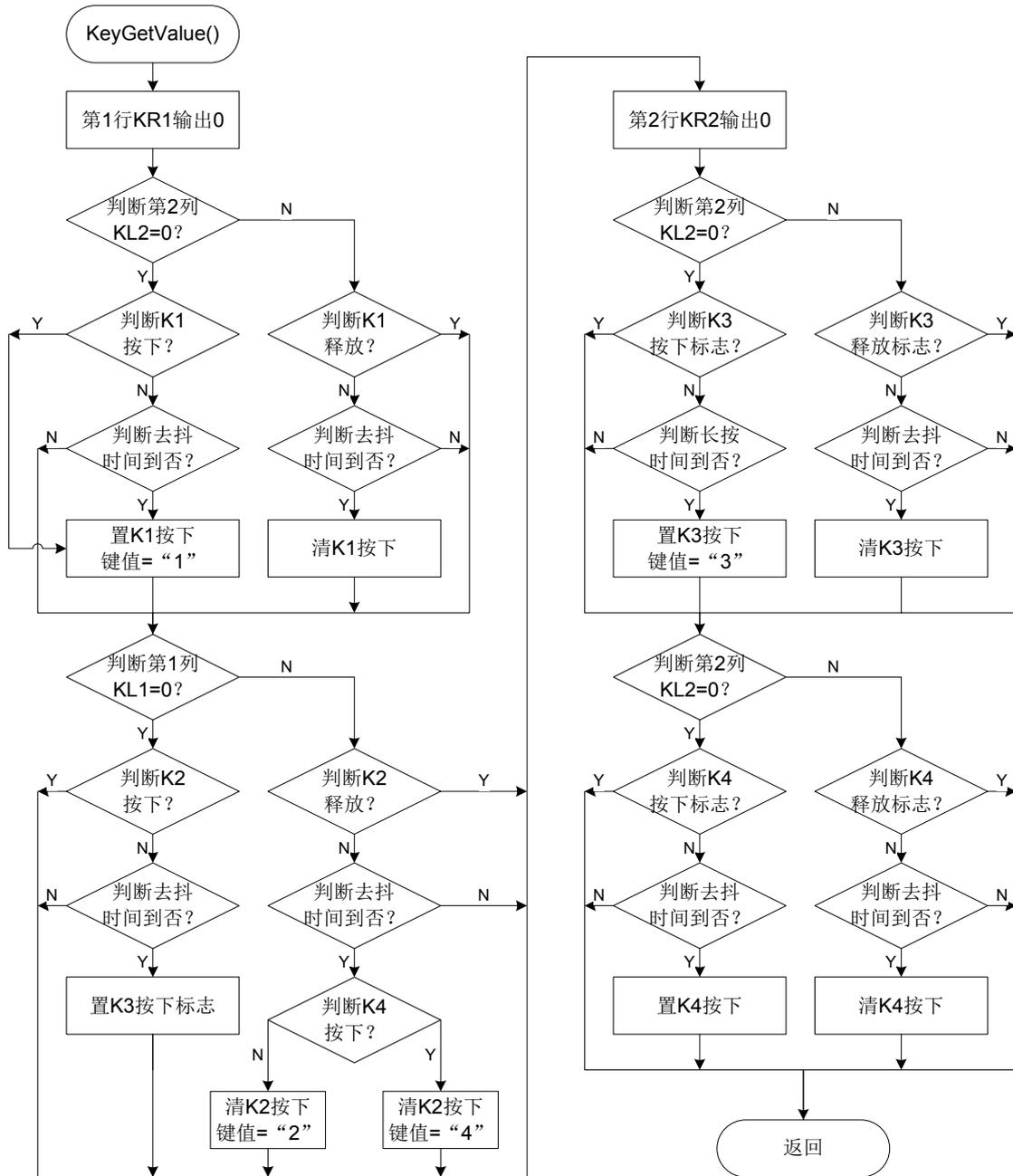


图 3-59 按键驱动子程序流程图

3.34 SPI (主控)

3.34.1 模块说明

该模块主要演示 SPI 接口访问外部 EEPROM 的功能。

EEPROM 型号: 93AA46A, 访问地址范围 0x00~0x7F。

按键选择地址或读写操作, LED 显示结果 (1 字节地址+1 字节数据)。

- (1) 单击 K3: 地址高半字节+1, 同时读出当前数据;
- (2) 单击 K4: 地址低半字节+1, 同时读出当前数据;
- (3) 单击 K2: 写入数据=当前地址;
- (4) 单击 K1: 写入数据=0x00;
- (5) 长按 K5: 全片擦除=0xFF。

3.34.2 硬件设计

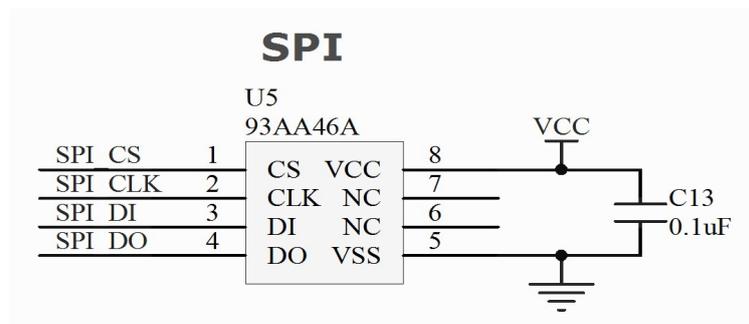


图 3-60 SPI 原理图

3.34.3 软件设计 (spi_demo)

3.34.3.1 模块函数

(1) SPI 初始化

```
/******
```

函数名: void SPInit(void)

描述: SPI 主机初始化子程序

输入值: 无

输出值: 无

返回值: 无

```
*****/
```

(2) SPI 写数据

函数名: uchar SPIWriteData(uchar memory_addr, uchar *buf, uchar n)

描 述: SPI 写数据, 逐个字节编程

输入值: addr—地址

buf—发送数据缓存

n—发送数据个数

输出值: 无

返回值: 0—TRUE

1—FALSE

*****/

(3) SPI 读数据

函数名: void SPIReadData(uchar addr, uchar *buf, uchar n)

描 述: SPI 读数据, 连续字节读出

输入值: addr—地址

n—接收数据个数

输出值: buf—接收数据缓存

返回值: 无

*****/

(4) SPI 擦除

函数名: uchar SPIEraseAll(void)

描 述: SPI 擦除全片 (填充 0xFF)

输入值: 无

输出值: 无

返回值: 0—TRUE

1—FALSE

*****/

3.34.3.2 流程图

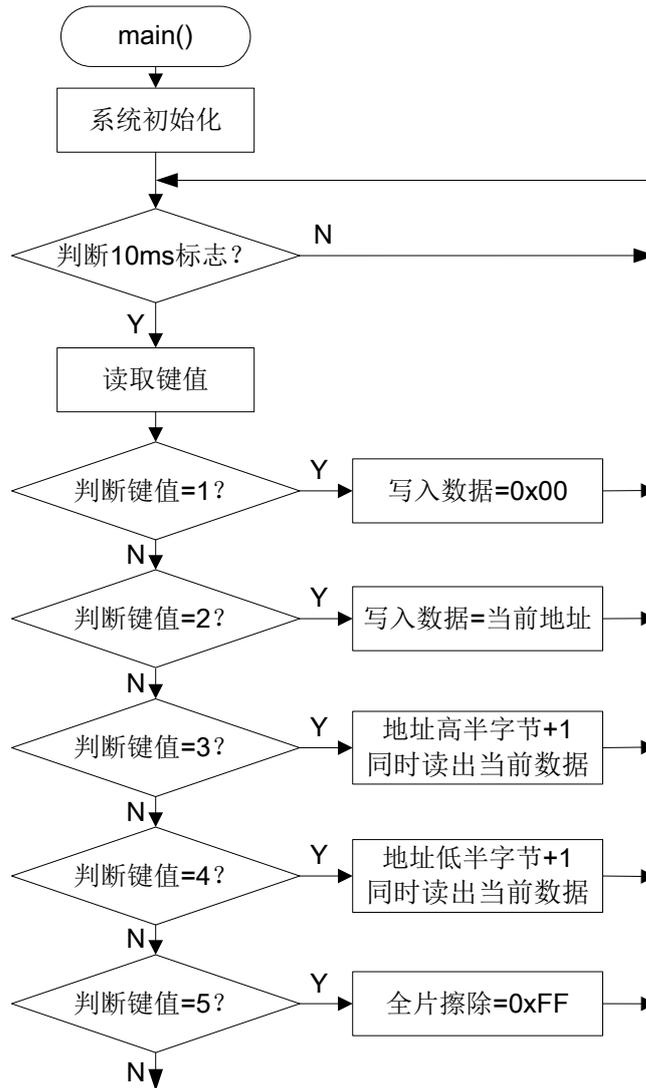


图 3-61 SPI 主程序流程图

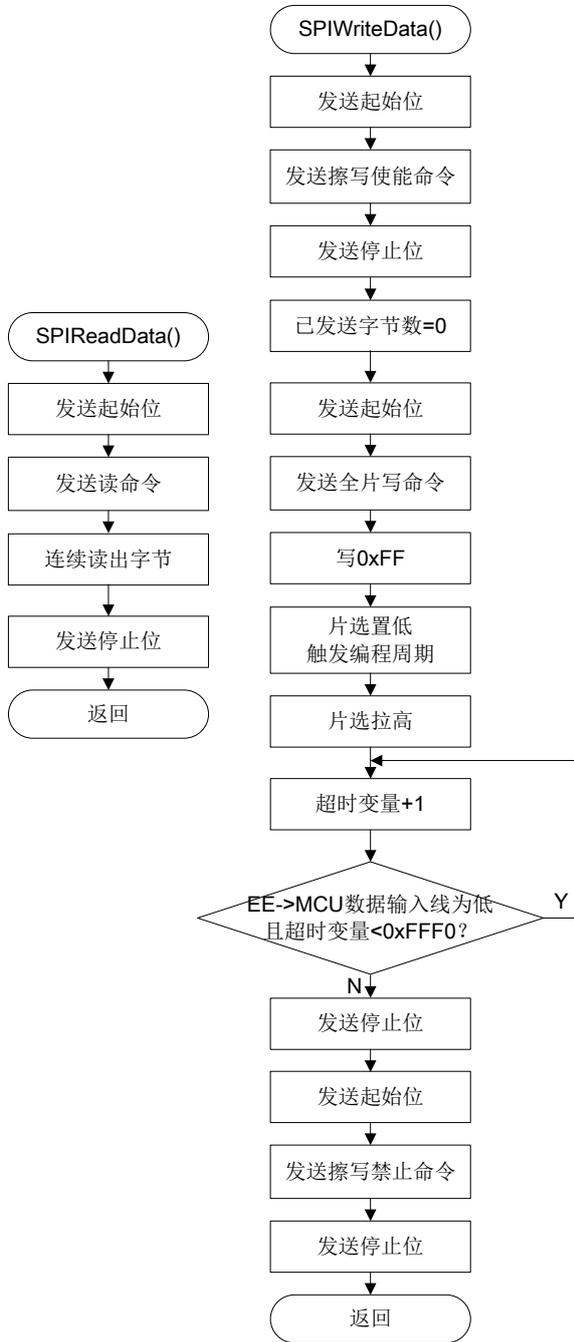


图 3-62 SPI 读/写数据流程图

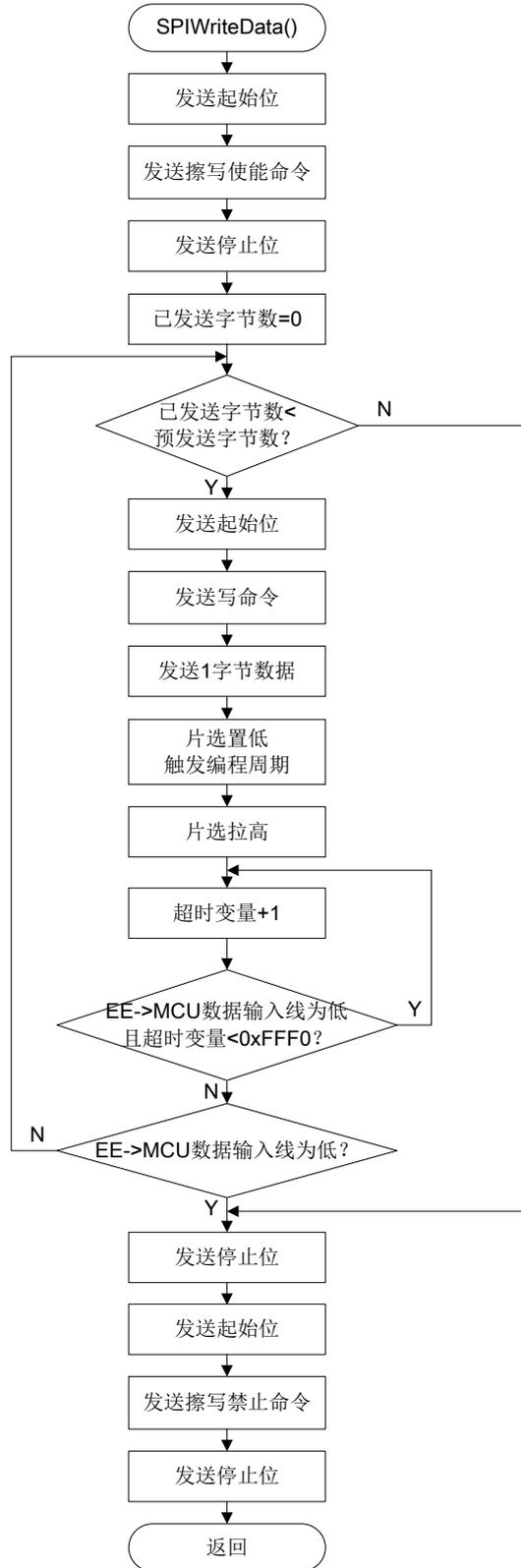


图 3-63 SPI 全片擦除流程图

3. 35 I2CM（主控）

3. 35. 1 模块说明

该模块主要演示 I2C 串行通讯访问外部 EEPROM 的功能。

EEPROM 型号：24C01，访问地址范围 0x00~0x7F。

按键选择地址或读写操作，LED 显示结果（1 字节地址+1 字节数据）。

- (1) 单击 K3：地址高半字节+1，同时读出当前数据；
- (2) 单击 K4：地址低半字节+1，同时读出当前数据；
- (3) 单击 K2：写入数据=当前地址；
- (4) 单击 K1：写入数据=0x00；
- (5) 长按 K5：全片擦除=0xFF。

3. 35. 2 硬件设计

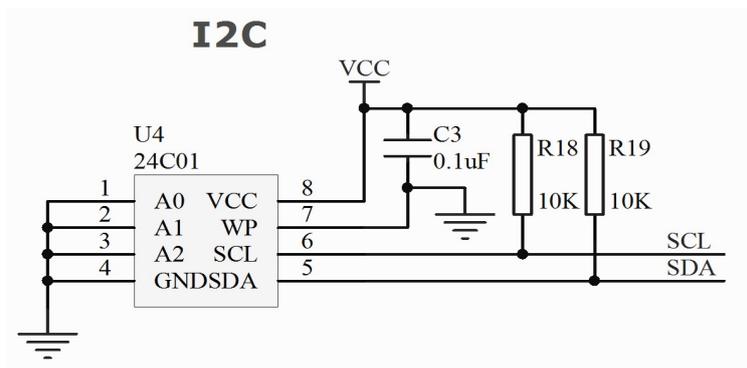


图 3-64 I2CM 原理图

3. 35. 3 软件设计—模拟方式（i2cm_demo1）

软件模拟 I2C 主机，SCL 为时钟线输出，SDA 为数据输入/输出，在写数据和发送应答时为输出，而在读数据或接收应答时作为输入。

数据写入 EEPROM 以后，需等待读取查询写入周期完成后，再继续操作 EEPROM。

3. 35. 3. 1 模块函数

(1) I2C 初始化

/******

函数名： void I2CMasterInit(void)

描述： I2CM 主机初始化子程序

输入值： 无

输出值： 无

返回值: 无

*****/

(2) I2C 写数据

/*****

函数名: uchar I2CMasterWriteData(uchar slave_addr, uchar memory_addr, uchar *buf, uchar n)

描 述: I2CM 写数据

输入值: slave_addr—从机地址

memory_addr—片内地址

buf—发送数据缓存

n—发送数据个数

输出值: 无

返回值: 0—TRUE

1—FALSE

*****/

(3) I2C 读数据

/*****

函数名: uchar I2CMasterReadData(uchar slave_addr, uchar memory_addr, uchar *buf, uchar n)

描 述: I2CM 读数据

输入值: slave_addr—从机地址

memory_addr—片内地址

n—接收数据个数

输出值: buf—接收数据缓存

返回值: 0—TRUE

1—FALSE

*****/

(4) I2C 擦除

/*****

函数名: uchar I2CMasterErasePage(uchar slave_addr, uchar memory_addr)

描 述: I2CM 擦除页面 (填充 0xFF), 8 字节/页

输入值: slave_addr—从机地址

memory_addr—片内地址

输出值: 无

返回值: 0—TRUE

1—FALSE

*****/

3.35.3.2 流程图

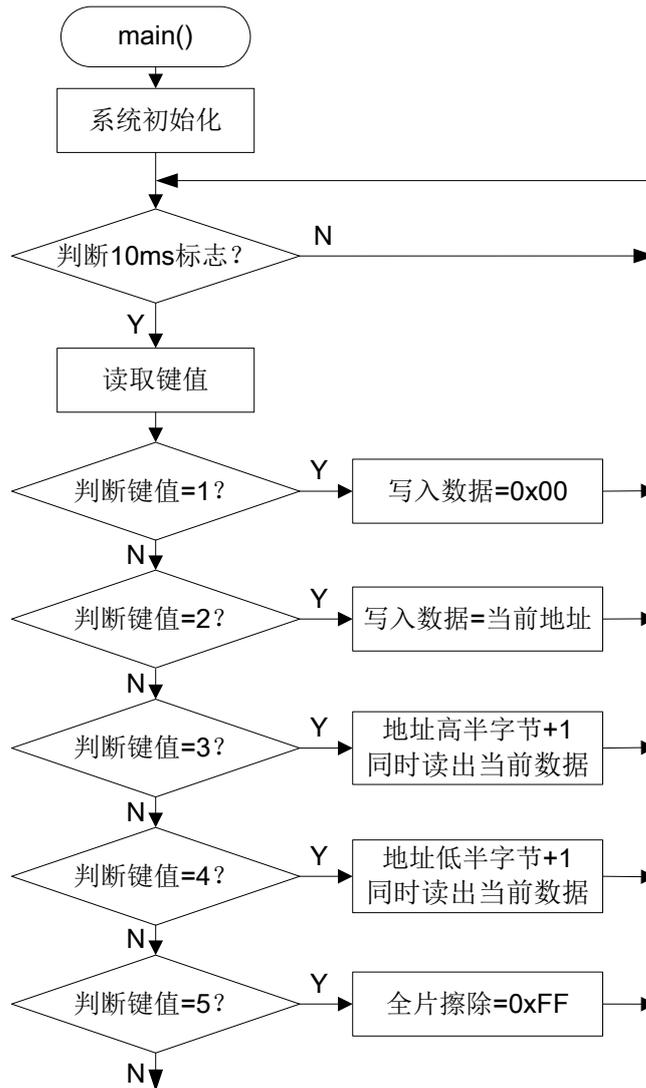


图 3-65 I2CM 主程序流程图

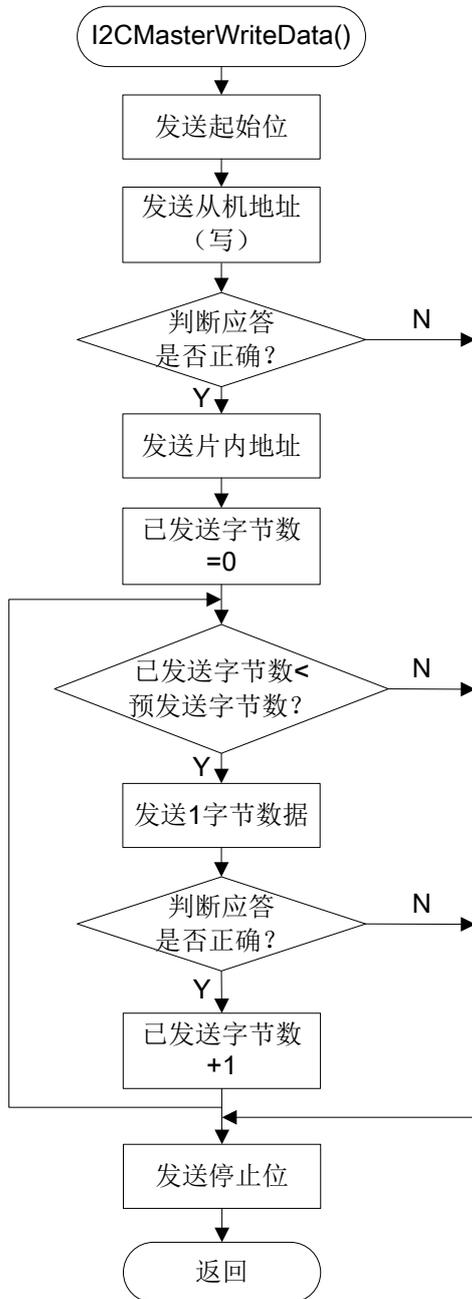


图 3-66 I2CM 写数据流程图

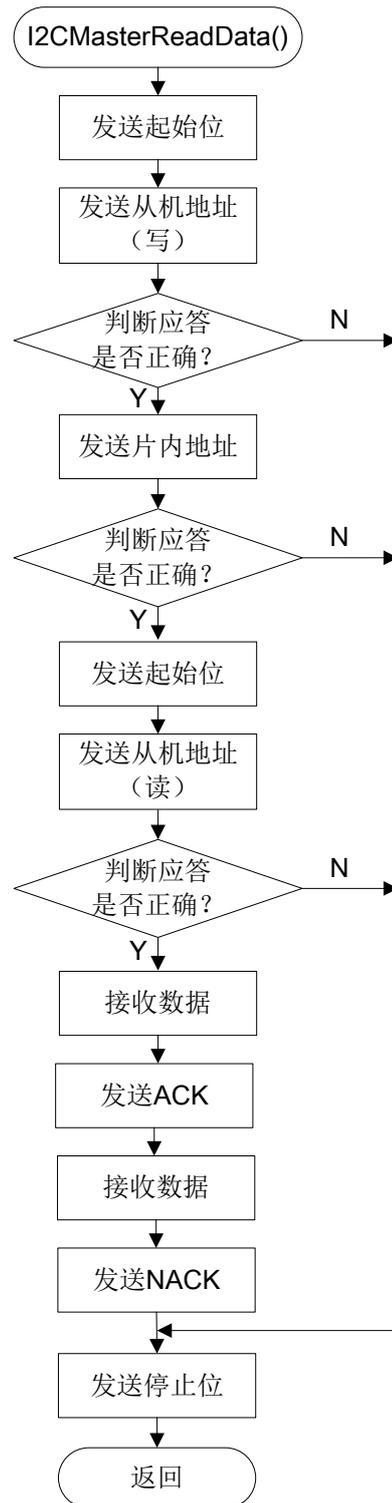


图 3-67 I2CM 读数据流程图

3.35.4 软件设计—硬件方式 (i2cm_demo2)

数据写入 EEPROM 以后, 需等待读取查询写入周期完成后, 再继续操作 EEPROM。

波特率配置的计算公式如下:

$$FSCL = Fosc / ((IICBRR + 1) \times 8)。$$

例如: 主频 $Fosc = 16\text{MHz}$, 当传输速率 $FSCL=400\text{KHz}$ 时, 得出 $IICBRR = 4$ 。

3.35.4.1 模块函数

模块函数定义同上。

3.35.4.2 流程图

同上。

3. 36 FLASH（内部存储器）

3. 36. 1 模块说明

该模块主要演示内部程序或数据 Flash 读写功能。

FLASH 模拟 256 字节数据存储。

例如：GDB-7P194 子板，地址范围 0x1F00~0x1FFF，第 62~63 页。

按键选择地址或读写操作，LED 显示结果（1 字节地址+1 字节数据）。

- （1）单击 K3：地址高半字节+1，同时读出当前数据；
- （2）单击 K4：地址低半字节+1，同时读出当前数据；
- （3）单击 K2：写入数据=当前地址；
- （4）单击 K1：写入数据=0x00；
- （5）长按 K5：全片擦除=0xFF。

注：由于 GDB-7P170 不支持 LED 数码管显示，因此通过 LD1~LD4 指示灯来表示读出数据的低 4 位。

3. 36. 2 软件设计（flash_demo）

3. 36. 2. 1 模块函数

（1）FLASH 写数据

/******

函数名： void FlashWriteData(uint addr, uchar *buf, uchar n)

描 述： Flash 写数据，逐个字节编程

输入值： addr—地址

 buf—发送数据缓存

 n—发送数据个数

输出值： 无

返回值： 无

*****/

（2）FLASH 读数据

/******

函数名： void FlashReadData(uint addr, uchar *buf, uchar n)

描 述： Flash 读数据，逐个字节读出

输入值： addr—地址

 n—接收数据个数

输出值: buf—接收数据缓存

返回值: 无

*****/

(3) FLASH 擦除

/******

函数名: uchar FlashErasePage(uint addr)

描述: Flash 擦除页 (填充 0xFF)

输入值: addr—地址

输出值: 无

返回值: 0—TRUE

1—FALSE

*****/

3.36.2.2 流程图

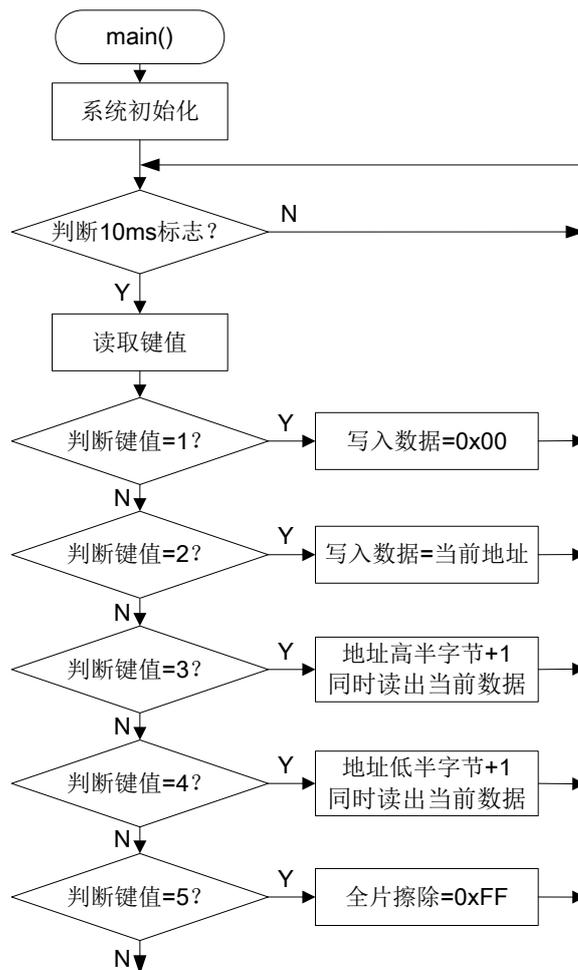


图 3-68 Flash 主程序流程图

3. 37 I2CS（从动）

3. 37. 1 模块说明

该模块主要演示 I2C 串行通讯从机的功能。

由辅助 AMCU 作为主机，周期读取从机的数据，并回送给从机，同时显示在 LED 数码管上。

按键修改从机要发送的数据（1 个字节），显示在 LED 左侧字节，然后从机接收主机写入的相同数据，显示在 LED 右侧字节。

(1) 单击 K3: 发送数据的高半字节+1

(2) 单击 K4: 发送数据的低半字节+1

基本协议:

(1) 主机读取数据: 从机地址 (0x5A) + 片内地址 (0x01) + 单字节数据。

(2) 主机写入数据: 从机地址 (0x5A) + 片内地址 (0x01) + 单字节数据。

采用中断方式接收并处理通讯协议。

注: 由于 GDB-7P170 不支持 LED 数码管显示, 因此通过 LD1~LD4 指示灯来表示接收数据的低 4 位。

3. 37. 2 硬件设计

SCL 时钟线必须采用 PINT 中断功能作为输入端口, 可以上升沿和下降沿触发。

例如: GDB-7P194 子板, PB0/PINT0 作为 SCL 时钟线, PB1/PINT1 作为 SDA 数据线。

3. 37. 3 软件设计—模拟方式 (i2cs_demo1)

3. 37. 3. 1 模块函数

(1) I2CS 初始化

```
/******
```

函数名: void I2CSlaverInit(void)

描述: I2C 从机初始化子程序

输入值: 无

输出值: 无

返回值: 无

```
*****/
```

3.37.3.2 流程图

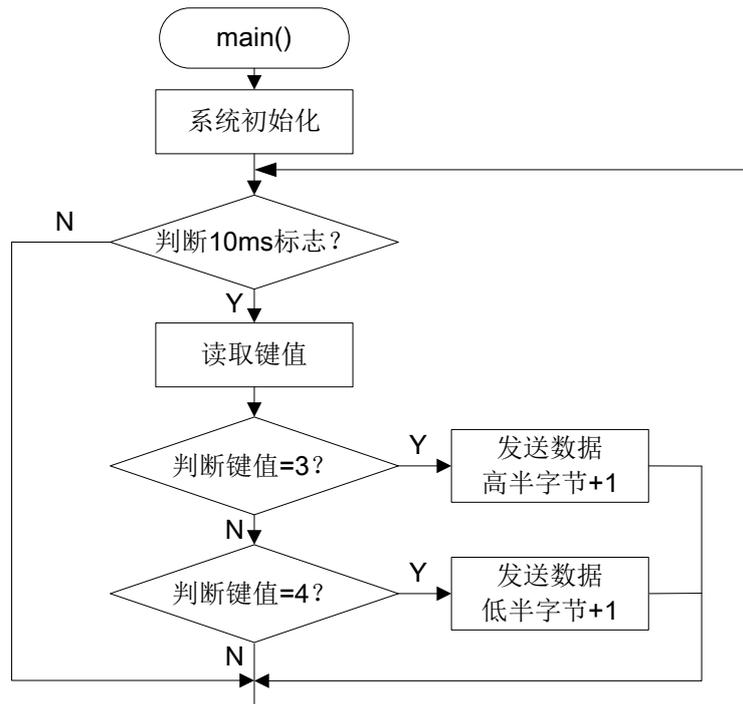


图 3-69 I2CS 主程序流程图

3.37.4 软件设计—硬件方式 (i2cs_demo2)

3.37.4.1 模块函数

模块函数定义同上。

3.37.4.2 流程图

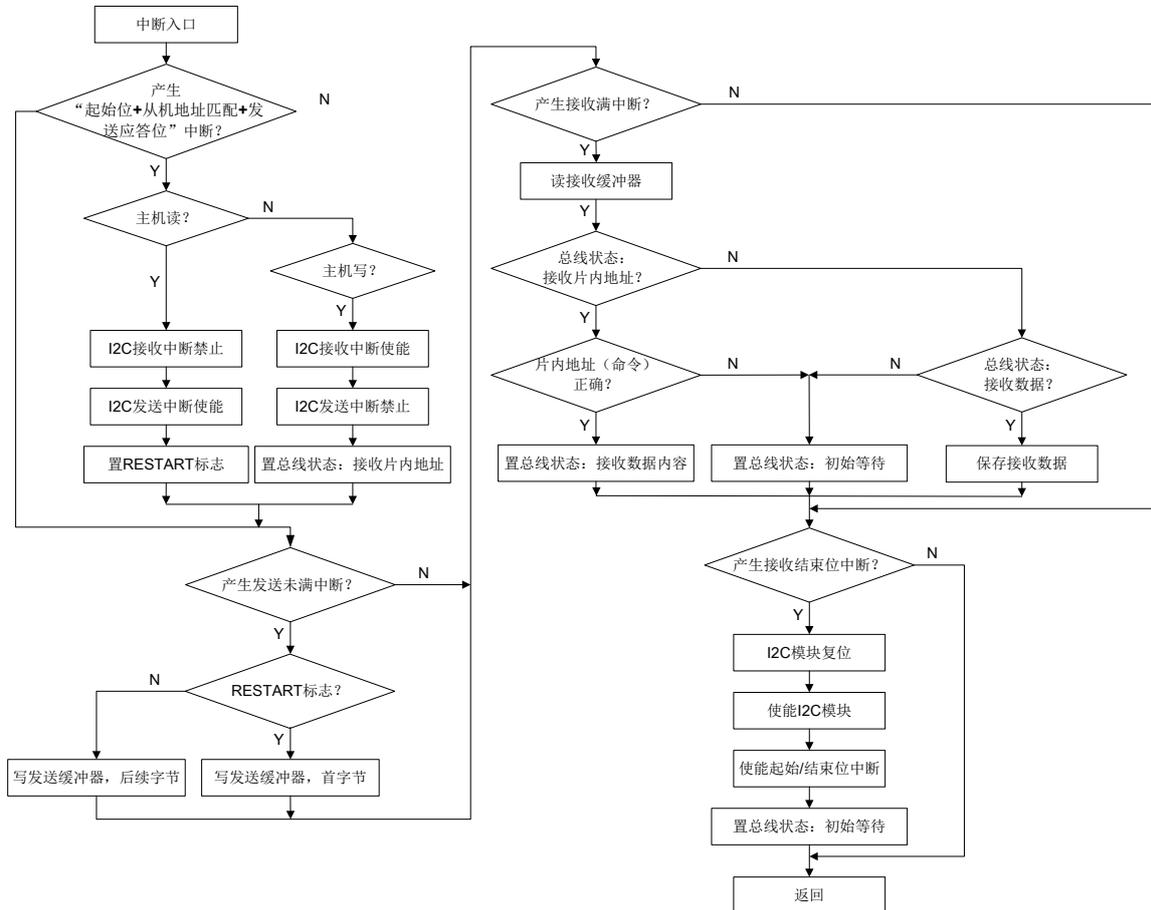


图 3-71 I2CS (硬件) 服务程序流程图

3. 38 UART（串口）

3. 38. 1 模块说明

该模块主要演示 UART 异步串行通讯功能和协议设计方法。

定义了串口通信主机请求/从机应答的通讯协议。

通讯协议如下：

- ◆ 数据帧头 1（固定） 0x55
- ◆ 数据帧头 2（固定） 0xAA
- ◆ 设备地址（Addr） 0x01
- ◆ 功能代码（Func） 1~255
- ◆ 数据字节（N 个） DataN
- ◆ 校验码（异或和） = 0x55 ^ 0xAA ^ Addr ^ Func ^ Data1 ^ ... ^ DataN
- ◆ 数据帧尾 1（固定） 0x5A
- ◆ 数据帧尾 2（固定） 0xA5

数据帧举例：“0x55 0xAA 0x01 0x41 0x11 0x12 0x13 0xAF 0x5A 0xA5”

“0x55 0xAA 0x01 0x99 0x22 0x45 0x5A 0xA5”

“0x55 0xAA 0x01 0x22 0xDC 0x5A 0xA5”

按照协议，HRSDK 接收上位机发送的数据帧，应答同样的数据帧。

LED 数码管显示数据帧中的设备地址和功能代码。

注：由于 GDB-7P170 不支持 LED 数码管显示，因此通过 LD1~LD4 指示灯来表示接收数据（功能代码）的低 4 位。

将 PC 主机与 HRSDK 开发板用串口线互连，打开超级终端或串口调试软件，发送数据帧到 HRSDK，HRSDK 将收到的数据帧回送给 PC 主机，在超级终端窗口上显示出来。

通讯设置：波特率 9600；数据位 8；停止位 1；奇偶校验位 无。

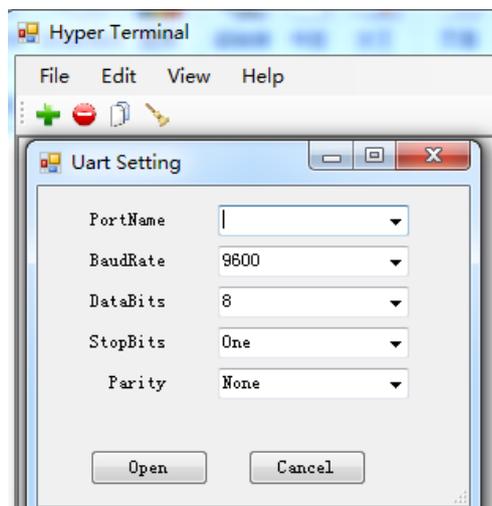


图 3-72 超级终端窗口图

3. 38. 2 硬件设计

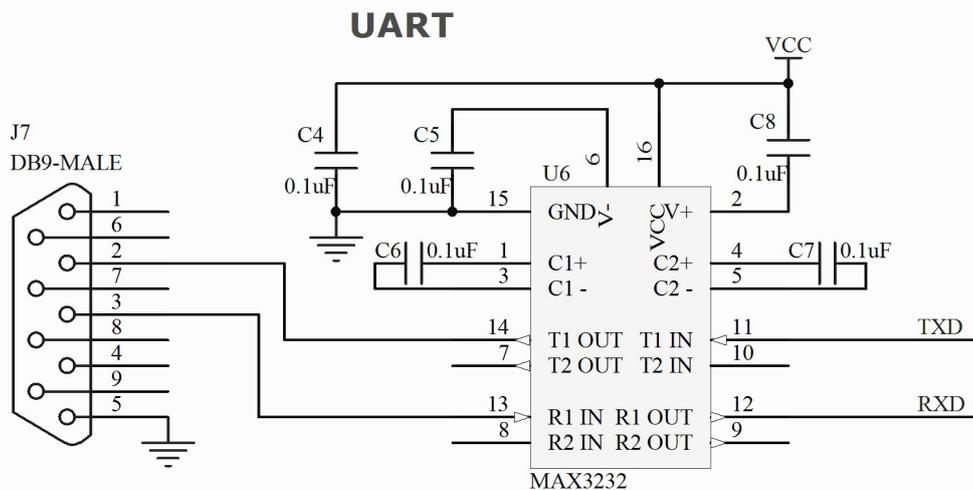


图 3-73 UART 原理图

3. 38. 3 软件设计 (uart_demo)

硬件控制采用串口中断程序，实现发送和接收。

高速波特率模式 BRGH=1，波特率的计算公式如下：

$$\text{Baud rate} = \text{Fosc} / (16 \times (\text{BRR} + 1))$$

例如：主频 Fosc = 16MHz，当波特率=9600bps 时，得出：BRR =104 (0x68)；

为了保证通讯可靠，滤除干扰信号，协议还限定了帧间的最小间隔时间，和帧内字节间的最大间隔时间，程序通过判断间隔时间，来分辨整帧并提取数据信息。

3. 38. 3. 1 模块函数

(1) UART 初始化

```
/******
```

函数名: void UARTInit(void)

描述: UART 初始化子程序

输入值: 无

输出值: 无

返回值: 无

```
*****/
```

(2) UART 发送

/******

函数名: void UARTTxData(uchar *buf, uchar n)

描 述: UART 发送数据整帧打包

输入值: buf—发送数据缓存

n—发送数据个数

输出值: 无

返回值: 无

*****/

(3) UART 接收

/******

函数名: uchar UARTRxData(uchar *buf, uchar n)

描 述: UART 接收数据整帧解包

输入值: 无

输出值: buf—接收数据缓存

n—接收数据个数

返回值: 0—TRUE

1—FALSE

*****/

3.38.3.2 流程图

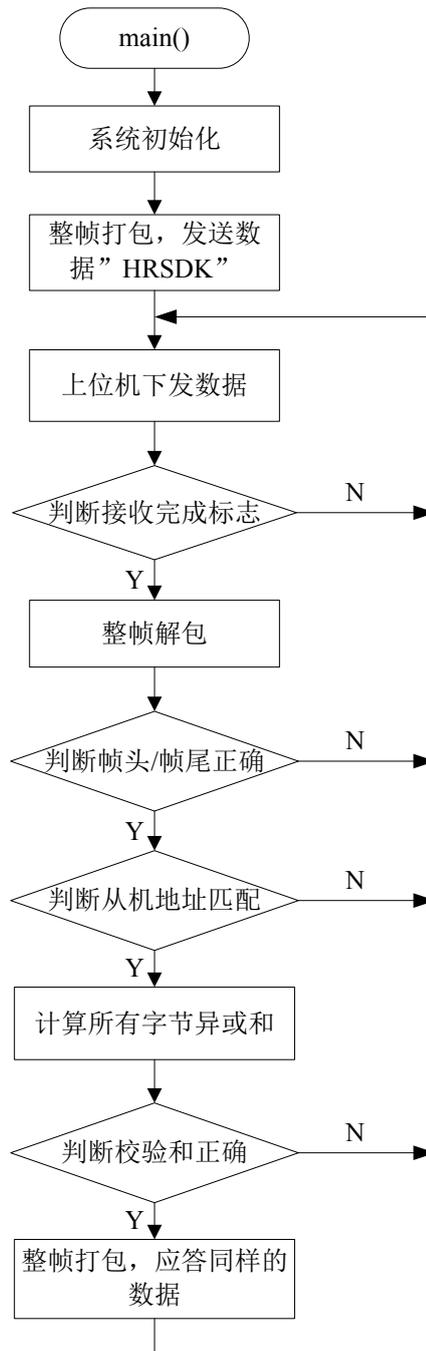


图 3-74 UART 主程序流程图

3. 39 CARD (IC卡)

3. 39. 1 模块说明

该模块主要演示 EUART 串行通讯和 ISO7816 协议中的复位应答功能。

当 RST 复位电平由低置高后，在 400~40000 个时钟周期内，IC 卡应答 ATR 复位序列，LED 显示 ATR 序列的前 2 个字节内容，即初始化字符 TS 和格式字符 T0。

3. 39. 2 硬件设计

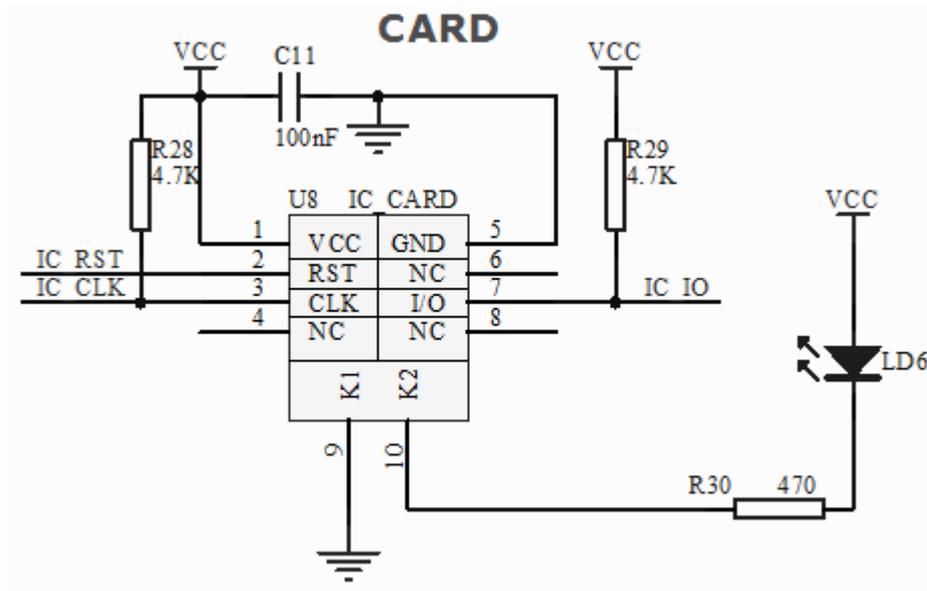


图 3-75 CARD 原理图

3. 40 RM（红外收发）

3. 40. 1 模块说明

该模块包括红外发射（RMT）和红外接收（RMR），主要演示 RM 红外收发和 PWM 载波调制功能。

通讯载波频率为 38KHz，数据由 MCU 调制，然后通过红外发射管发射信号。

采用自发自收方式，按键发送数据，并接收数据，同时显示在 LED 数码管上。

按键修改要发送的数据（1 个字节），显示在 LED 左侧字节，然后接收相同数据，显示在 LED 右侧字节。

（1）单击 K3：发送数据的高半字节+1，然后发送 1 次数据

（2）单击 K4：发送数据的低半字节+1，然后发送 1 次数据

采用中断方式接收并处理通讯协议。

注：如果内部晶振不能满足通讯的精度要求，则考虑采用外部晶振。

3. 40. 2 硬件设计

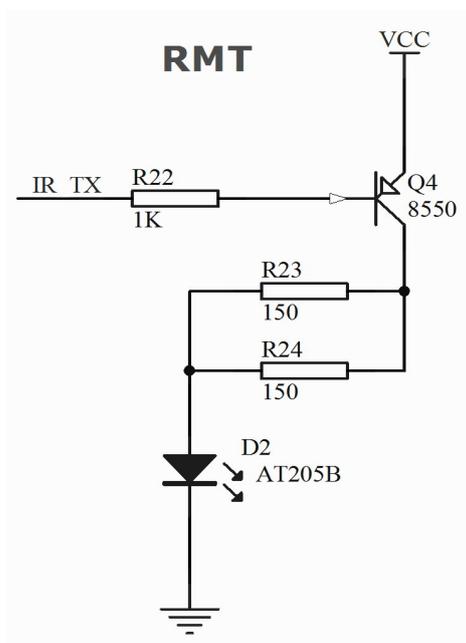


图 3-76 红外发射（RMT）原理图

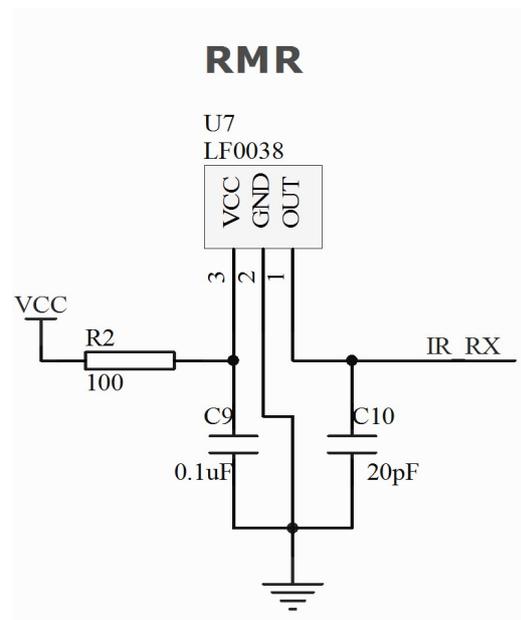


图 3-77 红外接收（RMR）原理图

3. 40. 3 软件设计 (rm_demo)

3. 40. 3. 1 模块函数

(1) RM 发送初始化 (PWM)

/******

函数名: void PWMInit(void)

描 述: RM 发送 (PWM) 初始化子程序

输入值: 无

输出值: 无

返回值: 无

*****/

(2) RM 接收初始化

/******

函数名: void RMRInit(void)

描 述: RM 接收初始化子程序

输入值: 无

输出值: 无

返回值: 无

*****/

(3) 开启 PWM 载波调制输出

/******

函数名: void RMSetCarrierOn(uchar t)

描 述: 控制红外发射二极管发送 38kHz 的脉冲

输入值: t—开启时间

输出值: 无

返回值: 无

*****/

(4) 关闭 PWM 载波调制输出

/******

函数名: void RMSetCarrierOff(uchar t)

描 述: 控制红外发射二极管处于停止状态

输入值: t—关闭时间

输出值: 无

返回值: 无

*****/

(5) 发送 1 字节数据

/*****

函数名: void RMsSendByte(uchar byte)

描述: 控制红外发送 1 字节数据

输入值: byte—发送的 8 位数据

输出值: 无

返回值: 无

*****/

(6) 发送完整的编码数据

/*****

函数名: void RMsSendData(uchar blrAddr, uchar blrData)

描述: 控制红外发射二极管发送一组数据

输入值: blrAddr—地址码

blrData—操作码

输出值: 无

返回值: 无

*****/

(7) 处理接收到的地址码

/*****

函数名: uchar RMRAddrProcess(ulong ldata)

描述: 处理 RM 接收到的 2 字节地址码

输入值: ldata—RM 接收到的 4 字节数据

输出值: 无

返回值: addr1—接收到的 8 位地址码

*****/

(8) 处理接收到的操作码

/*****

函数名: uchar RMRDataProcess(ulong ldata)

描述: 处理 RM 接收到的 2 字节操作码

输入值: ldata—RM 接收到的 4 字节数据

输出值: 无

返回值: data1 - 接收到的 8 位操作码

*****/

3.40.3.2 流程图

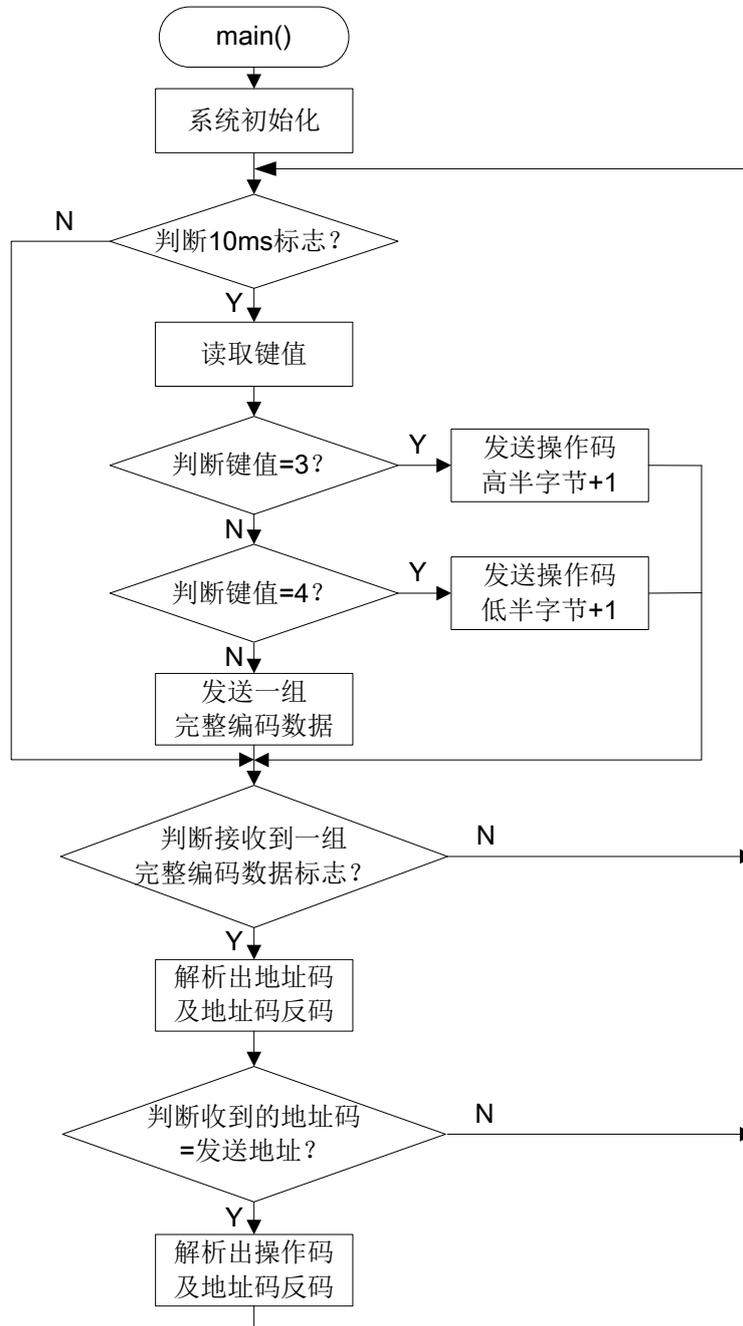


图 3-78 RM 主程序流程图

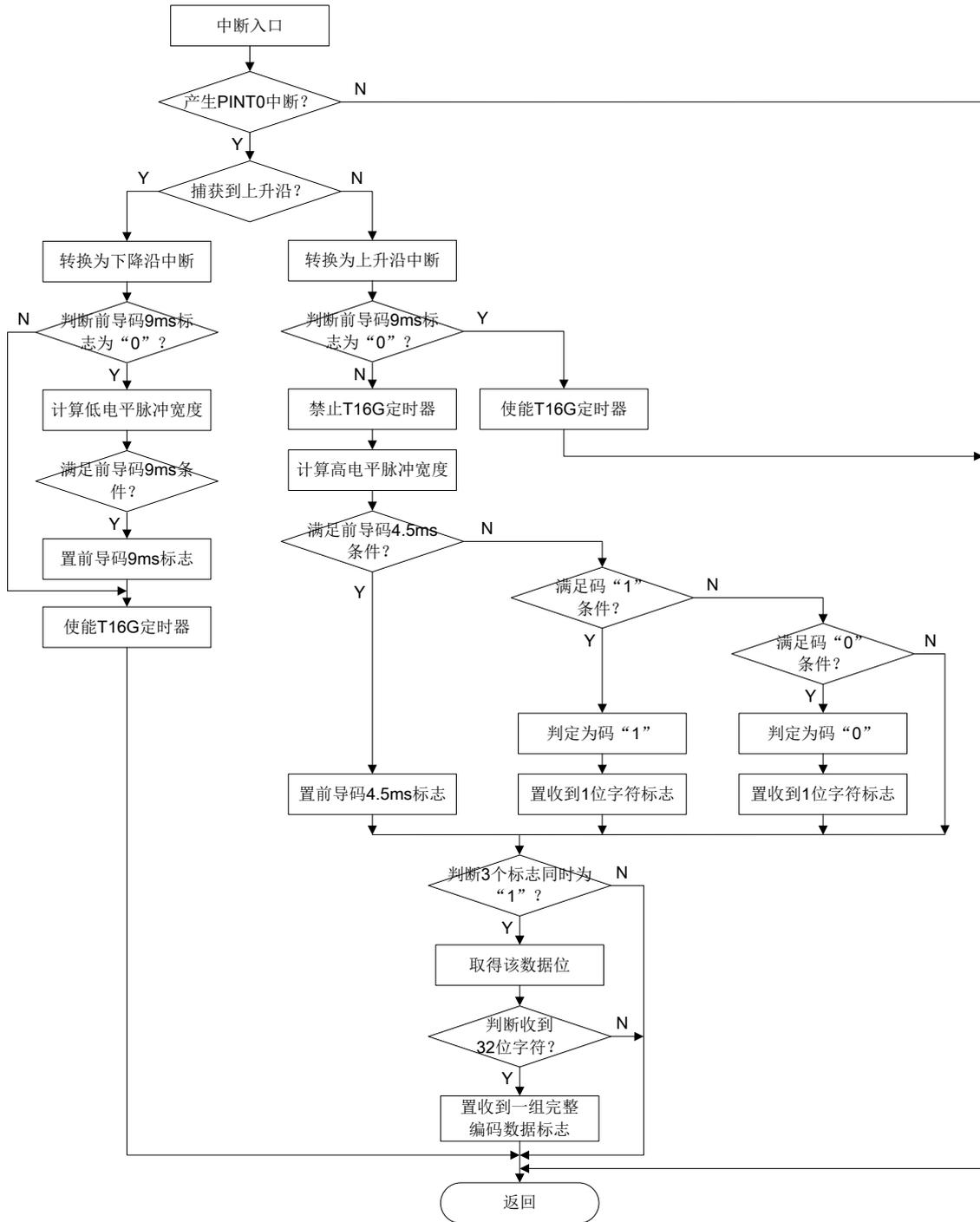


图 3-79 RM 中断服务程序流程图

3.41 AMCU（辅助MCU）

3.41.1 模块功能

AMCU 采用 HR7P169FGS4 芯片实现，通过 4 位拨码开关 S2 来选择功能。

- (1) 模拟过零检测用的方波信号输出，可选 50Hz 或 60Hz。
- (2) 用于 I2C 测试主机，与核心 MCU 从机进行通讯。

| 序号 | S2-1 (PB7) | S2-2 (PB6) | S2-3 (PB5) | S2-4 (PB4) | 功能 | LD7 指示灯 |
|----|---------------|---------------|---------------|---------------|---------------------------------|------------|
| 1 | OFF | OFF | N/A | OFF | 无 | 熄灭 |
| 2 | ON | OFF | N/A | X | PA1 输出 50Hz 方波，50%占空比 | 快闪（亮） |
| 3 | ON | ON | N/A | X | PA1 输出 60Hz 方波，50%占空比 | 快闪（亮） |
| 4 | X | X | N/A | ON | I2C 主机通讯 (PA4-SSCL、PA5-SSDA) | 慢闪 |

表 3-2 拨码开关选择功能表

其中，I2C 主机通讯功能为每秒定时工作，先读取核心 MCU 从机的数据（1 个字节），然后回写该数据。从机地址 0x5A，读取片内地址 0x01，写入片内地址 0x01。

AMCU 的程序下载调试接口为 5 芯 ICD 插座 J8。

3.41.2 硬件设计

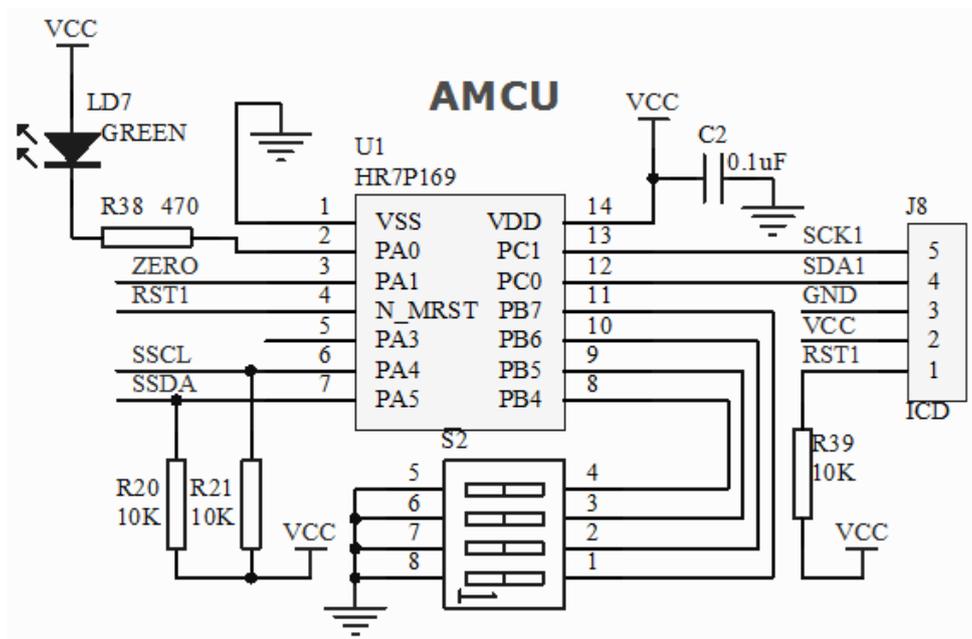


图 3-80 AMCU 原理图

3.41.3 软件设计 (amcu_demo)

3.41.3.1 模块函数

- (1) 50Hz/60Hz 方波信号，无模块函数，根据定时时间对输出端口进行电平翻转。
- (2) I2C 主机通讯，模块函数同 I2CM 模拟方式 (i2cm_demo1)。

3.41.3.2 流程图

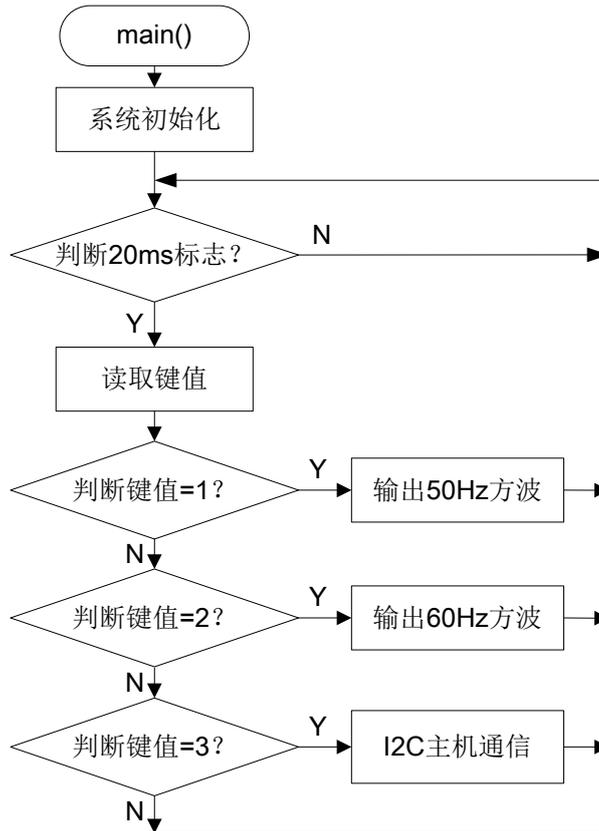


图 3-81 AMCU 主程序流程图

3.42 TK（触摸按键）

3.42.1 模块说明

该模块主要演示触摸按键控制功能。

3 个触摸按键分别表示：

- (1) 按键 S1，LED 数码管显示数字（曲目）+1
- (2) 按键 S2，LED 数码管显示数字（曲目）-1
- (3) 按键 S3，播放/停止当前曲目

3.42.2 硬件设计

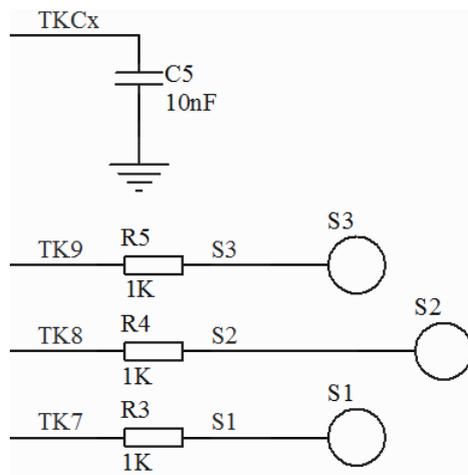


图 3-82 TK 原理图

3.42.3 软件设计（tk_demo）

3.42.3.1 模块函数

(1) TK 初始化

```
/******
```

函数名: uchar TKInit(void *data_ptr, uchar *key_ptr); //TK 初始化

描述: 初始化触摸按键，动态分配存储空间

输入值: *data_ptr — 按键数据结构指针

*key_ptr — 按键处理结构指针（驱动函数库内使用）

输出值: 无

返回值: 1 (TURE) — 空间分配成功

0 (FALSE) — 空间分配失败

```
*****/
```

(2) TK 周期启动

/*****

函数名: void TKScan (void)

描 述: 切换通道并启动 TK 按键扫描, 周期扫描

输入值: 无

输出值: 无

返回值: 无

*****/

(3) TK 读键值

/*****

函数名: void TKRead(void)

描 述: 触摸按键读取键值子程序, 判键并输出键值

输入值: 无

输出值: 键值存放全局变量 KeyPressValue

返回值: 无

*****/

3.42.3.2 流程图

无流程图。

3.42.3.3 程序说明

按键判别算法采用封装库函数, 文件名 tkm.hrlib。

用户可通过 tkm_config.h 和 tkm_config.c 配置相关硬件和寄存器初始化参数。

常用的配置参数为:

/* TK 软件配置 */

const uchar ChannelSum = KeySum; //实际按键数 KeySum=3 (tkm_config.h 中修改)

const uchar ChannelNum[ChannelSize] = {7, 8, 9}; //按键通道序号选择

const uchar FingerThreshold[ChannelSize] = {10, 10, 10}; //按键通道触摸门限

const uchar DebounceTimes = 10; //触摸去抖时间

/* TK 硬件配置 */

const uchar REG_VRC1 = 0x82; //使能 ADCVREF 2.6V, VREF1=1.5V

const uchar REG_ACPC4 = 0x11; //使能比较器 C4, 内部 VREF1

```
const uchar REG_TKMODL = 0xFF; //设置放大系数寄存器=0x1FFFFFFF
const uchar REG_TKMODM = 0xFF;
const uchar REG_TKMODH = 0x01;
const uchar REG_TKMODU = 0x00; //放大系数高 4 位=0
const uchar REG_TKTUN = 0x30; //采样次数 1 次，比较器输出滤波时间 8*Tosc，Cx 放电
时间 32*Tosc，TK 未触摸为低电平
const uchar REG_TKSEL = 0x40; //设定感应按键通道，充放电占空比 1:2，扫描频率 fosc/4。
（低半字节必须设为 0）
```

第4章 其他硬件

4.1 电源设计

4.1.1 模块说明

外部输入的 5V 电源接口，采用电源端子 J4 或 Mini USB 插座 J5，按压开关 S1 进行选择。

过流保护采用了快恢复保险丝 F1，限流 0.5A，过压保护采用了 5V 的 TVS 稳压管 D1。

电源输出为 5V 和 3.3V，由跳线开关 S3 选择。

通电后点亮电源指示灯 LD5。

4.1.2 硬件设计

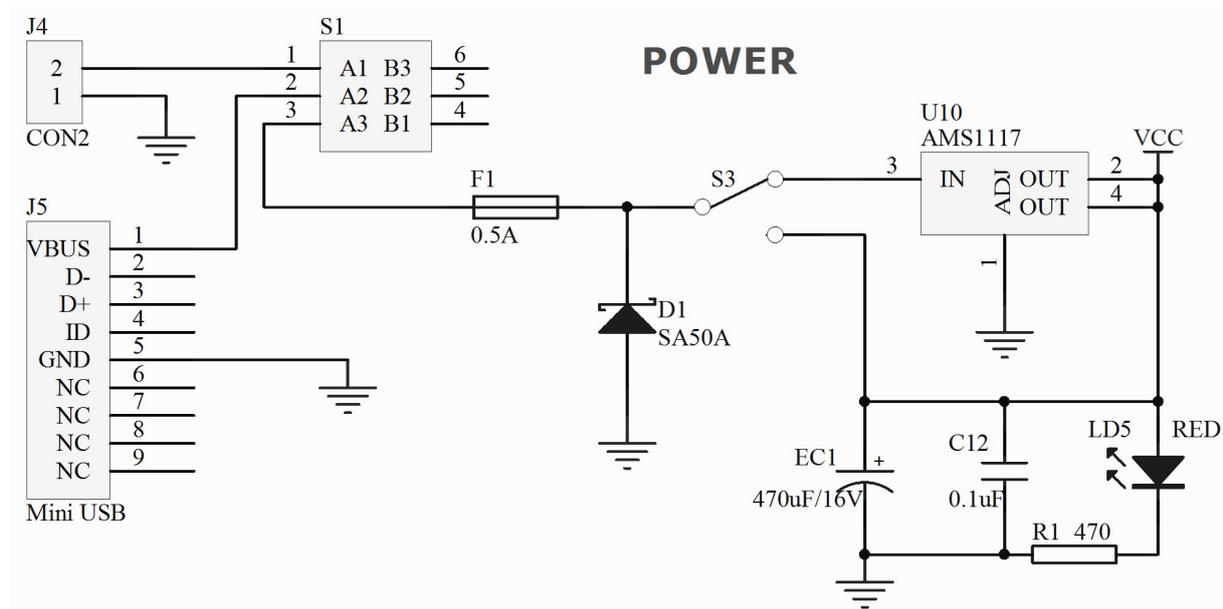


图 4-1 POWER 原理图

4.2 ICD（调试端口）

4.2.1 模块说明

调试端口位于核心 MCU 子板上（5 线接口），支持开发工具 ICD 或 HR10M 的仿真调试。

ISP_SDA 和 ISP_SCK 在仿真器内部上拉。

4.2.2 硬件设计

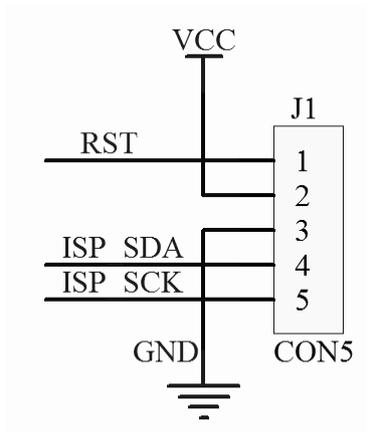


图 4-2 ICD 端口原理图

4.3 外部复位

4.3.1 模块说明

复位电路位于核心 MCU 子板上，提供外部上电复位和按键复位电路。

采用简单的阻容方式，根据不同的芯片，设计有高电平或低电平复位电路。

4.3.2 硬件设计

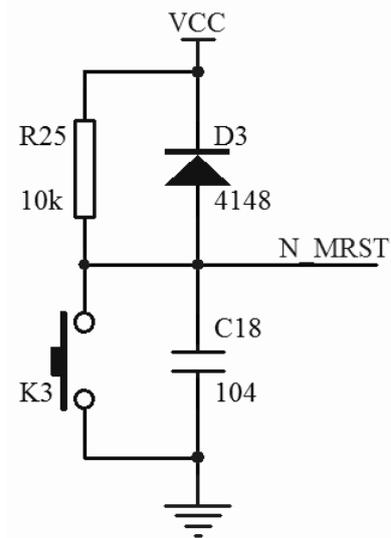


图 4-3 低电平复位原理图

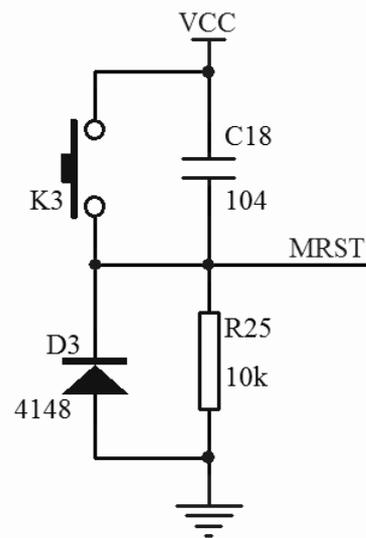


图 4-4 高电平复位原理图

4.4 外部时钟

4.4.1 模块说明

外部时钟电路位于核心 MCU 子板上，提供外部时钟，晶振频率 16MHz，负载电容 15pF，HS 高速方式。

4.4.2 硬件设计

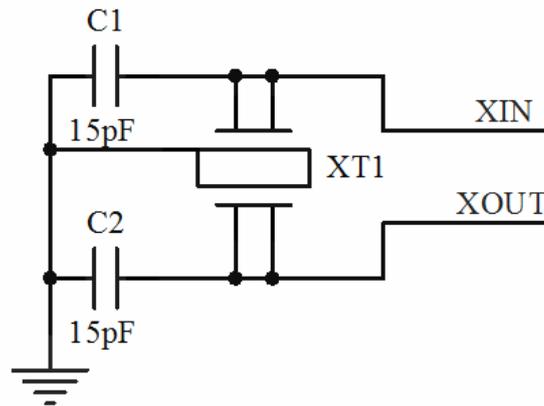


图 4-5 外部时钟原理图

4.5 连接插座

4.5.1 模块说明

子板与母板的连接插座采用 2 排 DIN32 插座。

4.5.2 硬件设计

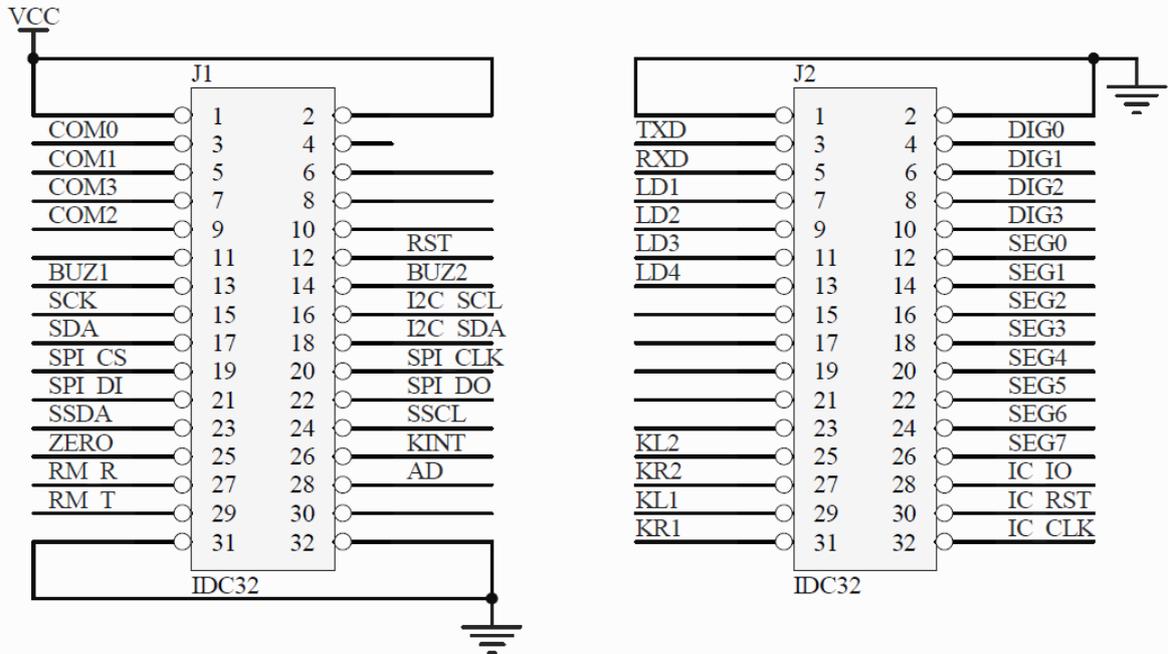


图 4-6 连接插座（DIN32）原理图