

32 位微控制器

HC32F460 系列的集成电路总线 I²C

适用对象

F 系列	HC32F460
------	----------

目 录

1	摘要	3
2	I²C 简介.....	3
3	HC32F460 系列的 I²C.....	4
3.1	系统框图	5
3.2	硬件连接	6
3.3	I2C 协议	7
3.4	I2C 模块可配置功能.....	7
3.4.1	滤波功能配置	7
3.4.2	波特率配置.....	7
3.4.3	从机地址配置	7
3.4.4	广播地址匹配功能	8
3.4.5	SMBus 主机地址匹配使能.....	8
3.4.6	SMBus 超时测量.....	8
3.4.7	快速 ACK 配置	8
3.5	中断源及事件信号输出	9
3.6	寄存器描述.....	11
3.7	初始化流程.....	12
4	样例代码	13
4.1	代码介绍	13
4.2	代码运行	15
5	总结	16
6	版本信息 & 联系方式	17

1 摘要

本篇应用笔记主要介绍 HC32F460 系列芯片的集成电路总线（Inter-Integrated Circuit, I2C）模块，并通过展示对 E2PROM 读写样例代码简要说明如何使用 I2C 模块。

2 I²C 简介

I2C 协议为 Philips 公司提出的一种双线制串行通信协议，支持半双工数据通信，已经广泛应用于电子电路产品中。常见的采用 I2C 通信协议的设备有 LCD 显示屏驱动器、存储器 E2PROM、音频解码器 CODEC 等等。

SMBus 总线基于 I2C 协议进行优化，为系统和电源控制相关的管理任务提供控制总线，因应用于控制系统，所以 SMBus 与 I2C 协议相比更侧重于通信过程中的实时响应。

3 HC32F460 系列的 I²C

HC32F460 系列的集成电路总线（I2C）模块支持 I2C 标准协议，还可配置为支持 SMBus Ver2.0 总线协议。本系列芯片拥有 3 个独立的 I2C 外设模块，I2C1~I2C3。

I2C 模块基本功能如下：

- 支持多主多从
- 支持标准模式最大 100Kbps，快速模式最大 400Kbps。
- 可以设定 2 个从机地址，可以设定 7 位地址格式和 10 位地址格式，能检测到广播呼叫地址。
- I2C 协议握手功能、仲裁功能、SCL 时钟同步功能。
- 输入信号数字滤波及模拟滤波功能
- 2 种复位方式
- 4 种中断和事件输出
- 可实现 DMA 方式通信

针对 SMBus 总线模式可支持功能如下：

- 数据传输速度支持 10Kbps~100Kbps
- SCL 电平超时测量
- 快速 ACK/NACK 功能

3.1 系统框图

I2C 模块挂载于芯片外设总线 APB4，采用外设时钟 PCLK3 进行分频产生波特率时钟，单个通道 I2C 模块系统框图如图 1 所示。

模块内部为数据端口 SDA 和时钟端口 SCL 提供输出控制和输入滤波功能；数据输出时，输出寄存器 I2CDTR 通过移位寄存器 I2CDSR 传送数据；数据输入时，输入寄存器 I2CDRR 从移位寄存器 I2CDSR 中得到输入数据；模块内还包含 ACK 控制单元、仲裁单元、波特率控制单元、超时控制单元、中断控制单元与发送接收控制单元一起实现 I2C 通信功能。

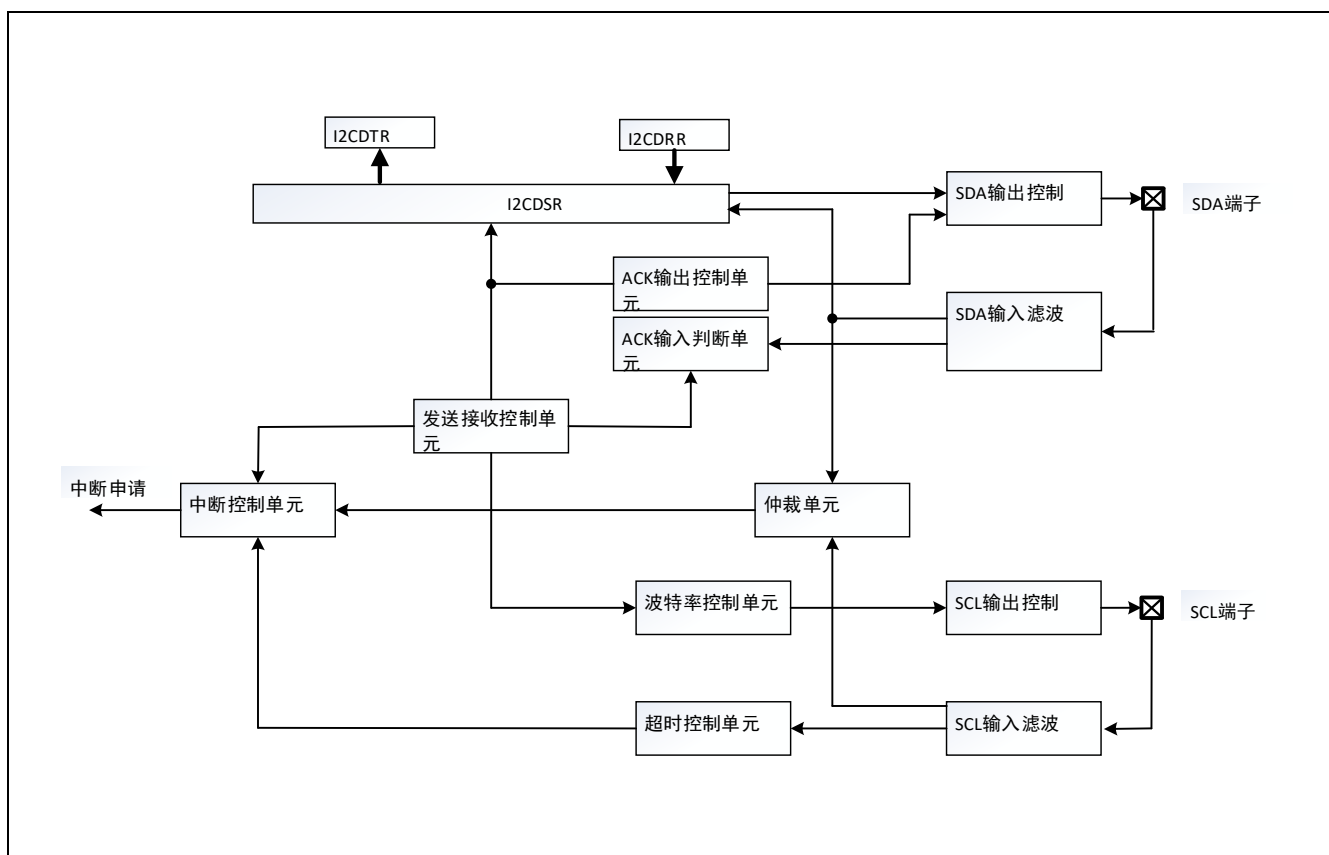


图 1 I2C 系统框图

3.2 硬件连接

本系列 I2C 模块支持多主多从的总线协议，推荐的总线结构如图 2 所示。总线的上拉电阻阻值由系统设计决定。

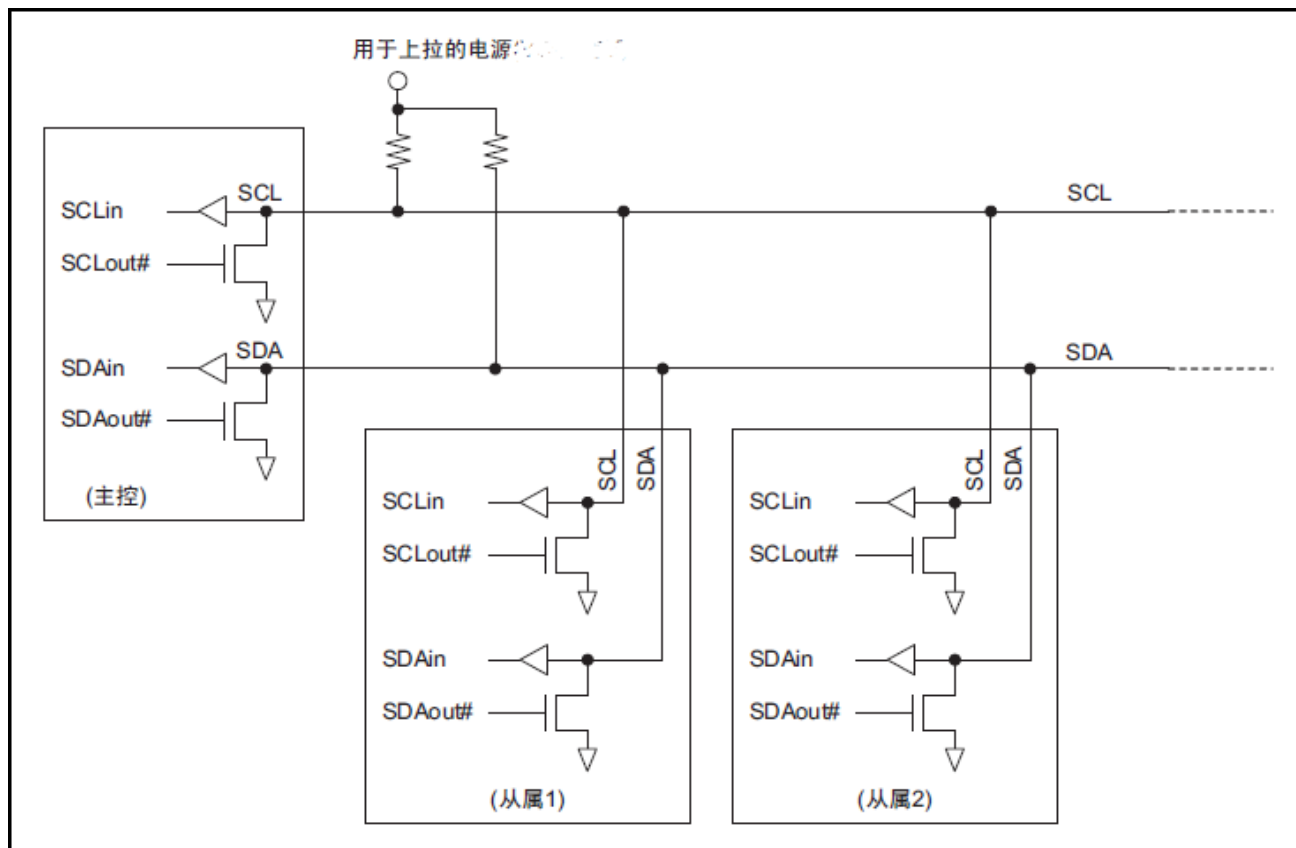


图 2 I2C 总线结构图

芯片 I2C 功能引脚资源包含 3 个通道的 SDA 和 SCL 端口，本系列芯片的 I2C 端口可以分配在功能组内任意端口。端口配置原理及方法请参考本系列芯片数据手册的引脚配置及功能章节，也可以参考厂商提供样例和驱动库进行配置。

I2C 通道	功能端口	方向
I2C1	I2C1_SDA I2C1_SCL	I/O
I2C2	I2C2_SDA I2C2_SCL	I/O
I2C3	I2C3_SDA I2C3_SCL	I/O

3.3 I2C 协议

I2C 总线协议请参考本系列芯片用户手册章节 27.3 的详细介绍，包括 I2C 协议的开始条件，地址传送，数据传送，停止条件，重新开始条件，SCL 时钟同步，仲裁、握手等。

3.4 I2C 模块可配置功能

以下 3.4.1~3.4.4 为 I2C 模块的基本配置功能：

3.4.1 滤波功能配置

接收数据滤波功能分为数字滤波功能和模拟滤波功能。

I2C 模块的滤波功能默认为禁止状态，需要使用时请在 I2C 功能使能前通过寄存器 I2C_FLTR 进行配置。

3.4.2 波特率配置

根据数字滤波功能和 I2C 波特率分频时钟设置的不同，各种情况波特率计算方法如下所示：

DNFE	FREQ	Baudrate
0	000	$\frac{1}{\frac{(\text{SHIGHW} + 3) + (\text{SLOWW} + 3)}{\Phi\text{I2C}} + \text{SCL 上升时间} + \text{SCL 下降时间}}$
1	000	$\frac{1}{\frac{(\text{SHIGHW} + 3 + \text{滤波能力}) + (\text{SLOWW} + 3 + \text{滤波能力})}{\Phi\text{I2C}} + \text{SCL 上升时间} + \text{SCL 下降时间}}$
0	非 000	$\frac{1}{\frac{(\text{SHIGHW} + 2) + (\text{SLOWW} + 2)}{\Phi\text{I2C}} + \text{SCL 上升时间} + \text{SCL 下降时间}}$
1	非 000	$\frac{1}{\frac{(\text{SHIGHW} + 2 + \text{滤波能力}) + (\text{SLOWW} + 2 + \text{滤波能力})}{\Phi\text{I2C}} + \text{SCL 上升时间} + \text{SCL 下降时间}}$

ΦI2C : I2C 波特率时钟频率，由 PCLK3 分频后得到。

公式中 SCL 上升时间和 SCL 下降时间与系统设计相关，可通过正确代入实际的 SCL 上升时间和下降时间修正输出波特率。

3.4.3 从机地址配置

本系列芯片有个两个从机地址寄存器 I2C_SLR0 和 I2C_SLR1 可以同时支持两个从机地址，可以分别支持 7 位或 10 位从机地址。

3.4.4 广播地址匹配功能

通过寄存器标志位 I2C_CR1.GCEN 可以使能广播地址匹配功能，使能后从机的 I2C 设备可以接收识别到地址为 0 的广播命令并将 I2C_SR.GCF 标志位置位。

以下 3.4.5~3.4.7 为 SMBus 模式时可配置的功能。

3.4.5 SMBus 主机地址匹配使能

SMBus 模式使能在从机接收模式时可以配置主机地址配置使能、报警响应地址匹配使能、默认地址匹配使能。详细说明请参考本系列芯片用户手册“27.3.2 地址匹配”章节。

3.4.6 SMBus 超时测量

SMBus 的超时测量包括 SCL 电平超时测量、从机的超时测量、主机的超时测量。本 I2C 模块硬件支持 SCL 电平超时测量功能，通过 TMOUTEN、HTMOUT 和 LTMOUT 寄存器可以配置超时功能，详细说明请参考本系列芯片用户手册“27.3.2.3 SMBus 主机地址匹配”章节。从机超时测量和主机超时测量功能可以通过软件实现。

3.4.7 快速 ACK 配置

在 SMBus 接收模式时，需要在第 8 个时钟下降沿将 SCL 保持低电平，待 CPU 处理完数据报错误码（PEC）的寄存后，根据计算结果软件写入 ACK 位来解除 SCL 低电平状态。可以通过 I2C_CR3.FACKEN 标志位使能此功能。

3.5 中断源及事件信号输出

中断源输出如下所示。

名称	中断源	中断标志	中断条件
I2C_EEI	通信错误/通信事件	ARLOF	ARLOF=1&ARLOIE=1
		SLADDR0F	SLADDR0F=1& SLADDR0IE=1
		SLADDR1F	SLADDR1F=1& SLADDR1IE=1
		SMBALRTF	SMBALRTF =1& SMBALRTIE=1
		SMBHOSTF	SMBHOSTF =1& SMBHOSTFIE=1
		SMBDEFAULTF	SMBDEFAULTF =1& SMBDEFAULTIE=1
		GENCALLF	GENCALLF =1& GENCALLIE=1
		NACKF	NACKF=1&NACKIE=1
		TMOUTF	TMOUTF=1&TMOUTIE=1
		STARTF	STARTF=1&STARTIE=1
		STOPF	STOPF=1&STOPIE=1
I2C_RXI	接收数据满	RFULLF	RFULLF=1&RFULLIE=1
I2C_TXI	发送数据空	EMPTYF	EMPTYF=1&EMPTYIE=1
I2C_TEI	发送结束	TENDF	TENDF=1&TENDIE=1

事件信号输入如下所示，通过事件信号输出标志可以使能 DMA 数据接收及发送功能，使用方法请参考本系列芯片 DMA 功能的应用说明文档。

名称	事件源	事件条件
I2C_EEI	通信错误/通信时间	ARLOF=1
		SLADDR0F=1
		SLADDR1F=1
		SMBALRTF=1
		SMBHOSTF=1
		SMBDEFAULTF=1
		GENCALLF=1
		NACKF=1
		TMOUTF=1
		STARTF=1
		STOPF=1
I2C_RXI	接收数据满	RFULLF=1
I2C_TXI	发送数据空	EMPTYF=1
I2C_TEI	发送结束	TENDF=1

3.6 寄存器描述

I2C1 基准地址: 0x4004E000

I2C2 基准地址: 0x4004E400

I2C3 基准地址: 0x4004E800

寄存器名	符号	偏移地址	位宽	复位值
I2C控制寄存器1	I2C_CR1	0x00	32	0x0000 0040
I2C控制寄存器2	I2C_CR2	0x04	32	0x0000 0000
I2C控制寄存器3	I2C_CR3	0x08	32	0x0000 0006
I2C从机地址寄存器0	I2C_SLR0	0x10	32	0x0000 1000
I2C从机地址寄存器1	I2C_SLR1	0x14	32	0x0000 0000
I2C状态寄存器	I2C_SLTR	0x18	32	0xFFFF FFFF
I2C状态寄存器	I2C_SR	0x1C	32	0x0000 0000
I2C状态寄存器	I2C_CLR	0x20	32	0x0000 0000
I2C数据发送寄存器	I2C_DTR	0x24	8	0xFF
I2C数据接收寄存器	I2C_DRR	0x28	8	0x00
I2C波特率控制寄存器	I2C_CCR	0x2C	32	0x0000 1F1F
I2C波特率控制寄存器	I2C_FLTR	0x30	32	0x0000 0010

以上是本系列芯片 I2C 模块 3 个独立通道的寄存器分布说明，详细寄存器描述请查看用户手册相关章节。

3.7 初始化流程

I2C 模块推荐初始化流程如下：

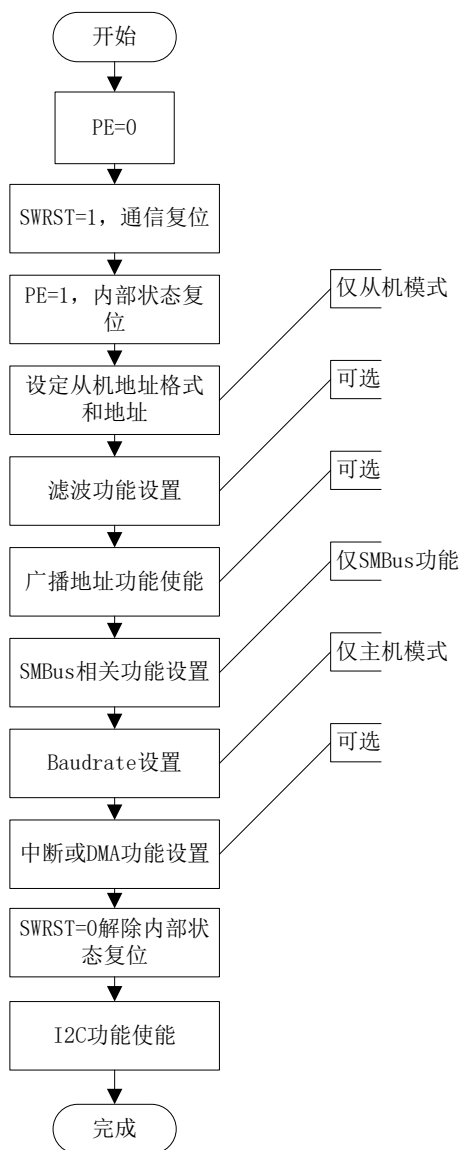


图 3 I2C 模块初始化流程

4 样例代码

4.1 代码介绍

用户可根据上述的工作流程编写自己的代码来学习验证该模块，也可以直接通过华大半导体的网站下载到设备驱动库（Device Driver Library, DDL）的样例代码并使用其中的 I2C 的样例进行验证。

以下部分简要介绍基于 DDL 的 I2C 模块的 E2PROM 读写样例。

1) 测试 buffer、测试 IO 初始化及系统时钟初始化

```
uint8_t u8TxBuf[PAGE_LEN];
uint8_t u8RxBuf[PAGE_LEN];
uint32_t i;
uint8_t u8Ret = I2C_RET_OK;
stc_port_init_t stcPortInit;

for(i=0; i<PAGE_LEN; i++)
{
    u8TxBuf[i] = i+1;
}
memset(u8RxBuf, 0x00, PAGE_LEN);

/* Initialize system clock*/
SysClkIni();

/*initialize LED port*/
MEM_ZERO_STRUCT(stcPortInit);
stcPortInit.enPinMode = Pin_Mode_Out;
stcPortInit.enExInt = Enable;
stcPortInit.enPullUp = Enable;

/* LED0 Port/Pin initialization */
PORT_Init(LED0_PORT, LED0_PIN, &stcPortInit);
/* LED1 Port/Pin initialization */
PORT_Init(LED1_PORT, LED1_PIN, &stcPortInit);
```

2) I2C 端口配置及 I2C 外设门控使能

```
/* Initialize I2C port*/
PORT_SetFunc(I2C1_SCL_PORT, I2C1_SCL_PIN, Func_I2c1_Scl, Disable);
PORT_SetFunc(I2C1_SDA_PORT, I2C1_SDA_PIN, Func_I2c1_Sda, Disable);

/* Enable I2C Peripheral*/
PWC_Fcg1PeriphClockCmd(PWC_FCG1_PERIPH_I2C1, Enable);
```

3) I2C 功能初始化配置

```
/* Initialize I2C peripheral and enable function*/  
E2_Initialize();
```

4) E2PROM 写一个字节

```
/* E2prom byte write*/  
u8Ret = E2_StartOrRestart(GENERATE_START);  
u8Ret = E2_SendAdr((E2_ADDRESS<<1)|E2_ADDRESS_W);  
JudgeResult(u8Ret);  
u8Ret = E2_SendAdr(DATA_TEST_ADDR);  
JudgeResult(u8Ret);  
u8Ret = E2_WriteData(u8TxBuf, 1);  
JudgeResult(u8Ret);  
u8Ret = E2_Stop();  
JudgeResult(u8Ret);
```

5) 5mS 的延时及 E2PROM 读一个字节

```
/* 5mS delay for e2prom*/  
Ddl_Delay1ms(5);  
  
/* E2prom random read*/  
u8Ret = E2_StartOrRestart(GENERATE_START);  
JudgeResult(u8Ret);  
u8Ret = E2_SendAdr((E2_ADDRESS<<1)|E2_ADDRESS_W);  
JudgeResult(u8Ret);  
u8Ret = E2_SendAdr(DATA_TEST_ADDR);  
JudgeResult(u8Ret);  
  
u8Ret = E2_StartOrRestart(GENERATE_RESTART);  
JudgeResult(u8Ret);  
u8Ret = E2_SendAdrRevData((E2_ADDRESS<<1)|E2_ADDRESS_R, u8RxBuf, 1);  
JudgeResult(u8Ret);  
u8Ret = E2_Stop();  
JudgeResult(u8Ret);
```

6) 比较读写数据，通过 LED 反馈测试结果

```
/* Compare the data */  
if(0x01 != u8RxBuf[0])  
{  
    /* e2prom byte write error*/  
    while(1)  
    {  
        LED0_TOGGLE();  
        Ddl_Delay1ms(500);  
    }  
}
```

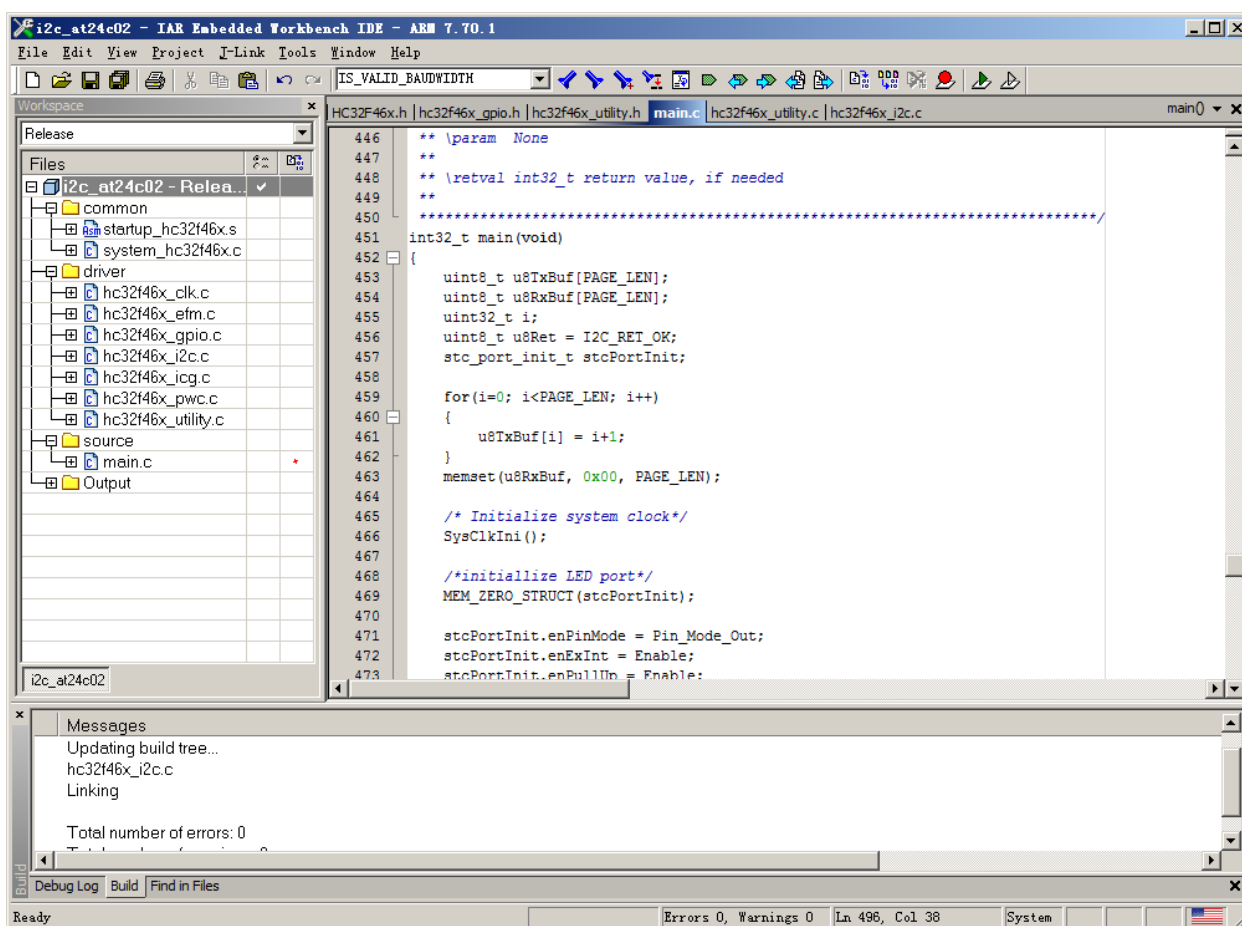
7) E2PROM 页操作读写及 LED 反馈结果（略）



4.2 代码运行

用户可以通过华大半导体的网站下载到 HC32F460 的 DDL 的样例代码（i2c_at24c02），并配合评估用板（EV-HC32F460-LQFP100-050-V1.1）运行相关代码学习使用 I2C 模块。

以下部分主要介绍如何在评估板上运行样例代码并观察结果：

- 一 确认安装正确的 IAR EWARM v7.7 工具（请从 IAR 官方网站下载相应的安装包，并参考用户手册进行安装）。
- 一 从华大半导体网站下载 HC32F460 DDL 代码。
- 一 下载并运行 i2c_at24c02\中的工程文件：
 - 1) 打开 i2c_at24c02\工程，并打开 ‘main.c’ 如下视图：



- 2) 点击  重新编译整个项目。
- 3) 点击  将代码下载到评估板上，全速运行。
- 4) 观察 LED 的亮灭变化，来确定 E2PROM 读写是否成功。运行结束后绿灯闪烁表示读写及数据比较成功，红灯闪烁表示 I2C 通信失败。

5 总结

以上章节简要介绍了 HC32F460 系列的 I2C，说明了 I2C 模块的寄存器及初始化操作流程，并且演示了如何使用样例代码，在实际开发中用户可以根据自己的需要配置和使用 I2C 模块。

6 版本信息 & 联系方式

日期	版本	修改记录
2019/3/15	Rev1.0	初版发布
2020/8/26	Rev1.1	更新 3.5 表格中“名称”；更新支持型号



如果您在购买与使用过程中有任何意见或建议，请随时与我们联系。

Email: mcu@hdsc.com.cn

网址: <http://www.hdsc.com.cn/mcu.htm>

通信地址: 上海市浦东新区中科路 1867 号 A 座 10 层

邮编: 201203

