

## 基于 MC9S08QG4 的烟雾传感器应用设计

作者：師英

飞思卡尔半导体公司8/16位应用工程师

在消防安全报警系统中，烟雾传感器设计的主要要求为：低功耗，低成本，高可靠性。飞思卡尔半导体推出的 MC9S08QG4 的具有高集成度、高性能，低功耗，高性价比和优异的可靠性，非常适合用于烟雾传感器的设计。

MC9S08QG4 是采用高性能、低功耗的 HCS08 内核的飞思卡尔 8 位微控制器系列中具有很高的集成度的器件。MC9S08QG MCU 集成了那些通常只有较大、较昂贵的元器件才具有的性能，包括背景调试系统以及可进行实时总线捕捉的内置在线仿真（ICE）功能，具有单线的调试及仿真接口。该产品的特色还包括一个可编程的 16 位定时器/脉冲宽度调制（PWM）模块（TPM），是同类产品中最灵活、又最经济的模块之一。

结构紧凑、高集成度的 MC9S08QG4 提供了飞思卡尔丰富的外设与 HCS08 内核卓越性能的多功能组合，包括更长的电池寿命（即使工作电压低至 1.8V，也能发挥最大效能）、业界领先的 FLASH 技术以及创新的开发支持。对于诸如无线通讯、手持式设备、小家电、基于简化型媒体访问控制器（SMAC）的应用，以及玩具等对功耗和尺寸要求较高的应用，MC9S08QG4 是出色的解决方案。

### 块状图

火警安全系统全系统块状图如图 1 中所示：

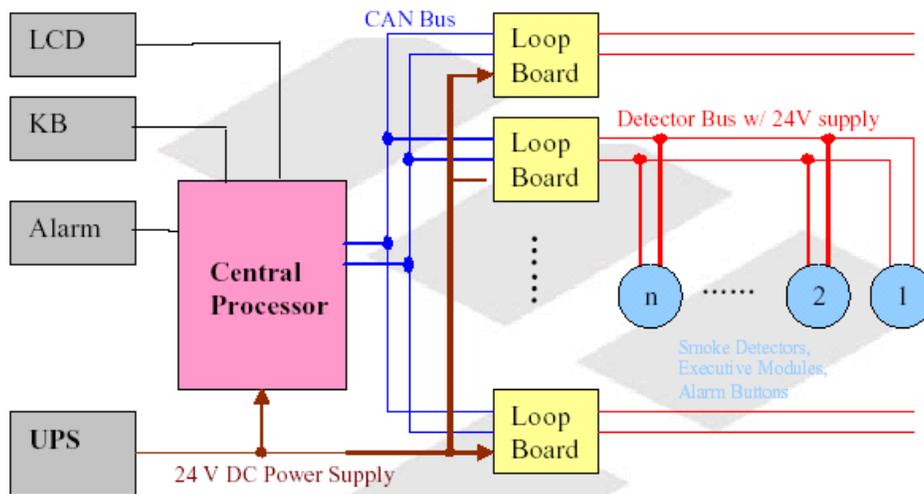


图 1：火灾控制系统的块状图

控制器联网 (CAN) 总线  
环行板  
带有 24V 电源的探测器总线  
中央处理器  
烟雾探测器  
执行模块  
告警按钮  
24V DC 电源  
液晶显示器  
键盘  
报警器  
不间断电源

- 中央处理器

中央处理器控制整个系统，定时询问每个烟雾探测器和告警按钮的状态，并通过环路板向每个执行模块发送命令。而且，它还提供带有TFT 液晶显示器和键盘的操作员接口，如果出现火情，就显示位置并报警。

- 环路板

通常在火警安全系统中有10个或更多的环路板。环路板与中央处理器连接。RS-485以前是环路板和中央处理器之间最常用的连接方法，但是控制器联网总线目前成为最现代的新型设计解决方案。

每个环路板都提供一个带有模块数据的探测器总线，并在两条线上提供24V 直流电源。根据不同生产商的产品不同，最多可以有240个烟雾探测器、告警按钮和执行模块连接在一个环路上。

- 执行模块和告警按钮

执行模块是执行设备，如遇到火灾时根据探测器总线传来的命令打开洒水器。一般来说，这些模块包括一个或多个由 MCU 多点控制单元的 GPIO 通用输

入/输出接口控制的中继。通常情况下，执行模块使用与烟雾探测器相同种类的 MCU 多点控制单元。

告警按钮安装在走廊上，一旦遇到火灾或紧急事件，人们可以打破盖板，按下按钮进行报警。有时为了降低成本，使用更简单的 MCU。

执行模块和烟雾探测器的数量比是 1: 2 或更少，系统中的告警按钮数量比执行模块数量少。

- 电源

整个火灾控制系统的电源是由 UPS（不间断电源）提供的。24V 直流电提供给中央处理器和每个环路板，然后环路板通过探测器总线将 24V 的直流电源提供给所有烟雾探测器、执行模块和告警按钮。通信数据可以在每根总线上调制。

- 探测器的总线

探测器总线是一种装有条线的特殊控制总线，带有 24V 直流电源和调制数据。图 2 显示了探测器总线上的常见波形，但没有更具体的波形，因为不同的生产商定义不同的通信协议。

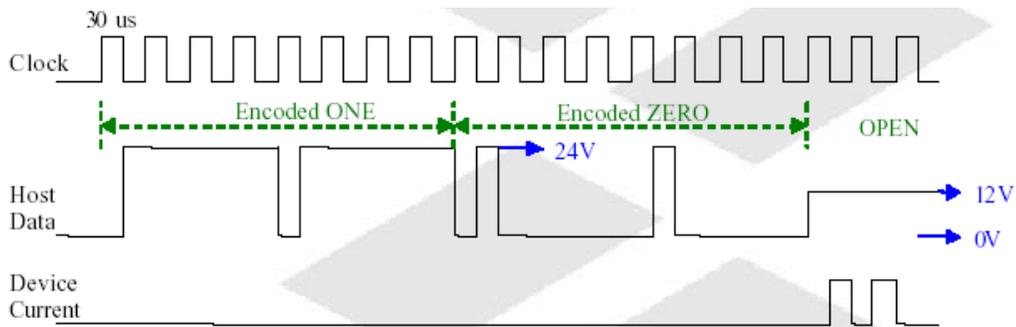


图 2: 探测器总线调时

时钟

托管数据

主机数据

设备电流

代码 1

代码 0

OPEN

调时决定了总线 MCU 频率的要求。

虽然它是个三态代码，但是只使用两种状态来传输总线数据，1 和 0（24V 和 0V）。第三种状态 OPEN 用来表示主机等待某一设备响应的状态。在这种状态下，总线被设置为一个恒定的中间电压，例如 12V，然后设备可从总线获得电流，总线此状态下作为一个电流回路。通过改变总线电流来代表逻辑 1 和 0 进行通信。

- 烟雾探测器

图 3 显示常用烟雾探测器的块状图

环路中的每个烟雾探测器都有独特的地址代码。询问命令定时通过总线发射（例如：每三秒钟）这种询问命令是一个广播，可以由所有的烟雾探测器接收的。如果一个探测器检测到火情，它会将信息发送给控制器进行报告。

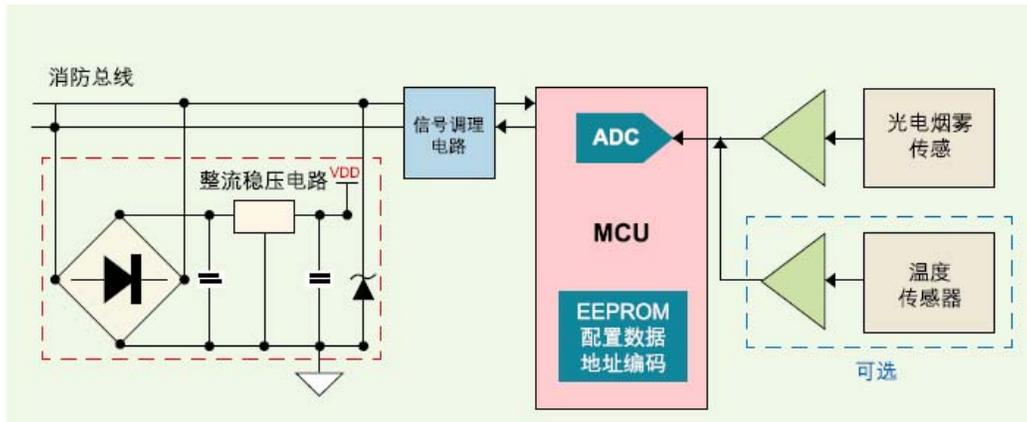


图 3：烟雾探测器块状图

### MCU 所需的资源和功能

以下是烟雾探测器应用所需的要求：

- 非常低的功耗，对整个电路板所需的功率不超过 400  $\mu$ A

- 一个信道模拟数字转换器 (ADC)
- 7 个 GPIO 通用输入/输出接口
- 一个用于定时功能的 8 位计时器
- 监控器、计时器和低电压检测，用于保证系统安全
- 16 位的 EEPROM 电可擦写可编程只读存储器用于存储配置数据如地址和配置
- 2K 的闪存，80 字节的内存

### 停止模式

QG4 有四种操作模式，激活、等待、停止 3、停止 2 和停止 1。

激活背景模式用于代码开发

- 等待模式：
  - CPU 停止操作来节约功率
  - 系统时钟运行
  - 保持全电压调节
- 停止模式：CPU 和总线时钟停止
  - 停止 1：内部电路完全停止，以最大限度地节约功率。
  - 停止 2：内部电路功率部分停止，保持内存内容。
  - 停止 3：所有内部电路可以快速恢复，保持内存和寄存器中的内容。

重启后，操作的正常模式是运行模式，并且在此模式中 CPU 被激活并且外围设备可使用。通过执行一个等待命令，MCU 进入等待模式。在等待模式中所需功率减小，因为 CPU 没有计时。为了进一步减少功率消耗，可以使用停止模式。当执行停止命令时，就输入三个停止模式中的一个。停止 1、停止 2 和停止 3，每一种都具有不同级别的操作可减少功率的损耗。表 1A 描述了停止模式的行为。一般来说，停止模式的电源供应 (3V 供应) 如下：

- 停止1模式供应电流, 475 nA
- 停止2模式供应电流, 600 nA

- 停止3模式供应电流, 750 nA
- RTI加法器在停止1、停止2或停止3模式下, 300 nA
- LVI加法器在停止3模式下 (LVDE = LVDSE = 1), 70 uA
- 振荡器加法器在停止3模式下, 5 uA

表 1B 描述了如何通过配置不同注册器的字节以及每个停止模式的结束条件来选择停止模式。

模式	CPU, 数字外围 是闪存 关闭	RAM	ICG	ATG 禁止	KBI	稳压器 或	I/O PIN 脚	RTI
停止 1	关闭	关闭	关闭	*	关闭	关闭	重设	关闭
停止 2	关闭	备用	关闭	禁止	关闭	备用	状态保持	可选择打 开
停止 3	备用	备用	备用**	禁止	可选择打 开	备用	状态保持	可选择打 开

\*ATD 停止模式或功率的减少都根据 ATDPU 的状态决定

\*\*晶体振荡器可配置运行停止 3。请见 ICG 注册器。

**表 1A: 停止模式行为**

STOPE	ENBDEM	LVDE	LVDSE	PDC	PPDC	停止模式	退出
0	x	x		x	x	禁用停止 模式	无
1	1	X		X	X	支持 BDM 的 停止模式 带有电压 调节器激 活的停止 3	IRQ、RST、 RTI、KBI
1	0	两个字节都必须 是 1		x	x		

1	0	任何字节都是 0	0	x	停止 3	
1	0	任何字节都是 0	1	1	停止 2	IRQ/RST/RTI
1	0	任何字节都是 0	1	0	停止 1	IRQ/RST

**表 1B: 停止模式的选择和退出条件**

当使用停止 2 时要考虑以下几个因素来确保正确操作。

- IRQ PIN 中断请求脚必须激活或从外部拉起
- LVD 低压差动传输模式必须在停止模式中禁用 (LVDSE = 0)
- 如果使用 RTI, 只有内部时钟源在停止 2 模式下生效。
- OSCSTEN 字节在停止 2 模式中无效, 此时钟参考将总是消耗电源
- 只有 RAM 保持功率, 所有其它的 I/O 寄存器将在唤醒状态下重新设置
- 在 I/O 引脚从停止 2 模式进入状态被修改前, PPDF 标记必须被清除。

从停止模式恢复需要时间。在 3V 电压时, 停止 1 和停止 2 的停止恢复时间大约为 50 us, 停止 3 的恢复时间根据时钟定义的不同而变化。如果使用内部参考时钟, 启动时间大约为 100 us, 而如果使用外部 32KHz 水晶, 如果振荡器在停止模式下激活 (设置为 OSCSTEN), 则是 2.4ms。如果振荡器在停止模式下禁用 (OSCSTEN 被清除), 则是 180 ms 到 300 ms。

不幸的是, 虽然使用停止模式可以大量节约功率, 但是许多烟雾探测器的设计师都不能接受停止模式, 因为它需要一直保持 MCU 工作, 以从总线获得响应。从停止模式恢复的时间使其很困难或甚至不可能。

#### 低功耗设计的 ICS 网络联机共享配置和操作模式

- **建议使用 ICS 配置和操作模式**

由一般总线调时确定, 至少需要 100KHZ 总线频率。

根据不同的ICS配置，QG4有六种操作模式：FEI、FEE、FBI、FBILP、FBE 和 FBELP。因为FBE模式有最低的功耗和最准确的时钟，所以建议用于烟雾探测器的应用。

遗憾的是，由于生产商使用不同的总线计时，所以有时可能无效。例如：有些总线计时需要1MHz的总线频率，QG4需要一个外部2MHz晶体振荡器用于FBE模式，许多客户不愿意支持额外的2个MHz晶体振荡器成本，有些总线需要很低的总线频率（如16Hz），然后可以选择FBI模式。QG4的内部32KHz时钟具有典型的0.2%的准确性和在整个操作温度范围有2%的误差。

非常普遍而且低成本、低功率的配置是使用一个455KHz震荡电路，用于FBELP模式。总线频率是227.5KHz（可以分开为113.75KHz）。如果供应电压是3V，227.5KHz频率下的(BDIV = 0)的QG4的电流供应就是240 μ A，而113.75 KHz（BDIV=1）的电流供应是200 μ A，28.4 KHz（BDIV= 4）的电流供应是170 UA。

表1显示所有适合烟雾探测器应用的操作模式，考虑到功率损耗和成本问题。而且考虑以下因素用于计算不同模式的总线频率。

- FEI:  $F_{bus} = F_{irc} * 512 / (2 * \text{总线分配器})$
- FBI:  $F_{bus} = F_{irc} / (2 * \text{总线分配器})$
- FBE:  $F_{bus} = F_{erc} / (2 * \text{总线分配器})$

模式	Fbus	优势	劣势	备注
FBELP	227.5 KHz	低成本，高准确度，低电流供应	成本高于内部参考时钟	建议采用的解决方案。振荡器的启动时间是5MS（大范围、低功耗，并且电流供应是5 μ A）
FBILP	16 KHz	最低的成本、	非常低的总线	只有总线时间接受内部晶体振荡

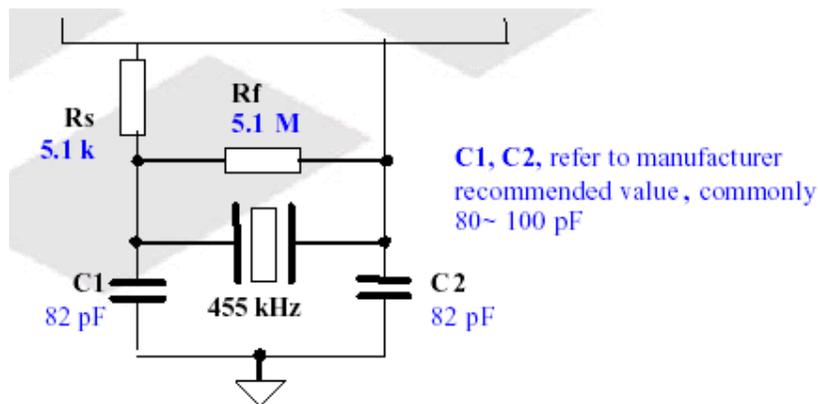
		一般0.2%，最差2%的准确度，快速启动	频率，可用于一些特殊设计	器的低Fbus为60 us才有用。
FEI	1MHz	最低的成本，一般0.2%，最差2%的准确度	3V的1MHz时使用2mA的电流	有些总线规定需要1MHz Fbus才能进行正常通信。虽然Fbus可通过旁路和禁止的转换到更低的频率，甚至当总线空闲时，MCU也可能被停止，恢复时间可能与应用不匹配，FLL的启动时间是0.5-1 ms。FLL本身使用电流为210~300 uA。

表2: 烟雾探测器所建议使用的操作模式

- 与外部455KHz振荡电路共同使用

当MCU与外部455KHz振荡电路共同使用，获得功率损耗、系统成本、时钟准确度和总线频率的最佳平衡时，必须考虑振荡电路组件的选择以使其可靠。

图4显示典型的455KHz振荡器电路。



C1, C2是指生产商的建议值，通常80到100 pF

图4: 455KHz的振荡器电路

在QG4的数据表上，规定了两种特点，低范围是32到38.4KHz，高范围超过1MHz。但是QG4可以与455KHz震荡电路共同使用；虽然在电路特点中没有具体说明。通过此配置，ICSC2注册器5位的范围应该设置为ICS的高范围模式。

- **高电流应该于POR**

QG4进行重新设置时的电源，ICS的默认模式总是FEI，在此模式中总线频率是4Hz，大约从电源消耗2mA的电流，不适用于烟雾探测器应用。建议在POR之后立即设置ICS到FBI模式。虽然高电流状态一直存在于POR，但是用户可以通过此方法缩短几个说明周期的时间。在Tcsth之后，晶体振荡器启动时间（5 ms用于高范围，低的振荡器功率模式），用户可以将ICS转换到最终工作模式，对FBE进行正确操作。在Tcsth的过程中，用户可以完成对系统初始工作，这对计时并不重要。

- **使用内部模块的电流供应**

每个内部模式在工作时都从电源获得电流，可以禁止某些未使用的模块以节约电源，但是已使用的模块的电流供应必须考虑系统的电源损耗，以下是QG4内部模块的电流供应列表。

- RTI: 300 nA
- LVD: 70 uA
- ACMP: 20 uA
- ADC: 120 uA（取样时间长，低电流模式）
- OSC: 5 uA（范围低、低电源模式）
- IRG: 100 uA
- FLL: 210 uA ~ 300 uA（根据频率和电压的供应决定）
- FLL: 210 uA ~ 300 uA（根据频率和电压的供应决定）

MCU的I/O 针脚必须正确设置来节约电源。所有未使用的I/O必须通过挂起或下拉的输入来设置，或简单的输出模式可输出禁止外部电路的状态。

## 带有闪存的EEPROM仿真

烟雾探测器使用EEPROM存储数据，如地址和系统配置。16位对于这些数据足够了。QG4的闪存可轻松用于模拟EEPROM。通过内部V<sub>pp</sub>充电泵，无需外部V<sub>pp</sub>。QG4的闪存可擦写/编程10000次，是HC08 Flash MCU的10倍还多。而且建议采用一些解决方案来延长闪存的生命周期。

- 闪存擦写/编程代码样本

如何消除/编程闪存的样本代码可在CW5.x的样本目录中找到。

```

if (FSTAT&0x10){ //Check to see if FACCERR is set
    FSTAT = FSTAT | 0x10; //write a 1 to FACCERR to clear
}
*((volatile unsigned char *)(Address)) = data; //write to somewhere in flash
    FSTAT = FSTAT | 0x80; //Put FCBEF at 1.
    _asm NOP; //Wait 4 cycles
    _asm NOP;
    _asm NOP;
    _asm NOP;
if (FSTAT&0x30){ //check to see if FACCERR or FVIOL
    return 0xFF; // are set if so, error.
}
while ((FSTAT&0x40)==0){ //else wait for command to complete
    ;
}
  
```

在V<sub>pp</sub>状态时，CPU使用不安全，那些闪存擦写/编程代码必须在RAM上执行，而不是在闪存上执行，为此用户可以限定一个阵列，包含上述C语言源代码所编辑的二位代码，并且称RAM中的这个阵列为子程序。

```

//Array of opcode instructions of the Erase/Program function in the HCS08 family
volatile unsigned char PGM[59] = {
0x87,0xC6,0x18,0x25,0xA5,0x10,0x27,0x08,0xC6,0x18,0x25,0xAA,0x10,0xC7,0x18,0x25,
0x9E,0xE6,0x01,0xF7,0xA6,0x20,0xC7,0x18,0x26,0x45,0x18,0x25,0xF6,0xAA,0x80,0xF7,
0x9D,0x9D,0x9D,0x9D,0x45,0x18,0x25,0xF6,0xF7,0xF6,0xA5,0x30,0x27,0x04,0xA6,0xFF,
0x20,0x07,0xC6,0x18,0x25,0xA5,0x40,0x27,0xF9,0x8A,0x81};
  
```

PGM[21]总是闪存操作命令。以下macros操作可以被定义以使RAM中的这些代码更容易做为子程序而被调用。

```
#define Page_Erase PGM[21]=0x40; temp = ((unsigned char*)(unsigned int))(PGM)
#define Program_Byte PGM[21]=0x20; temp = ((unsigned char*)(unsigned int, unsigned char))(PGM)
```

以下实例显示如何调用函数。

```
unsigned char temp, data;
data = 0x55;
temp = Page_Erase(0xF000);
if(temp==0xFF){
    // error
}
temp = Program_Byte(0xF000, data);
if(temp==0xFF){
    // error
}
```

在闪存可以正确擦写或编程前，FCLK必须设置在150KHz到200 KHz之间的正确值。

- 在**227.5 KHz FCLK** 下正确操作。

QG4的Ffclk特性是在150KHz到200 KHz之间。FCDIV的注册器必须被写入正确的值以设置正确的Ffclk。如果是PRDIV8，此寄存器的六位必须被清除。

- **fCLK = fBus / (DIV + 1)**

通过外部455 KHz FBE模式，在此模式中总线频率是227.5KHz。Ffclk 只能被设置为227.5 KHz（不分开）或113.75 KHz（两分），它最接近特定的Ffclk。

对于可靠的操作来说，建议采用带有113.75 KHz Ffclk的突发编程模式闪存。请注意突发编程与byte编程略有不同，所以源代码应该为突发模式重写。

另一方面，当闪存编程/擦写时电源消耗更大，FLL加法器的供应电流可以被忽略，所以另一个建议是使FLL获得更高的总线频率，然后在完成闪存操作后回到正常的设置。

- **模拟EEPROM的生命周期延长**

闪存只能被擦除为页。QG4闪存的页面大小是512个字节，由于只有少量字节用于烟雾探测器，所以可采用一些解决方案来延长模拟EEPROM的生命周期。解决方案之一就是存储记录数据在整个EEPROM上（参考图5）。每个记录有n个字节数据和1个字节标记组成，这个标记字节说明这n+1个字节地址是否被使用。如果没被使用，用户代码可将新的数据写入此位置；如果已被使用，那么将数据写入下一个位置。当阅读虚拟EEPROM时，用户代码应该首先检测这个标记，然后从最后一个已使用位置上读取数据，它包含最新的数据。当所有的地址(页面尺寸/记录尺寸)被使用时，下一个写操作应该从页面的第一个地址重新开始，结果闪存这一页的生命周期就被(页面尺寸/记录尺寸)成倍延长了。

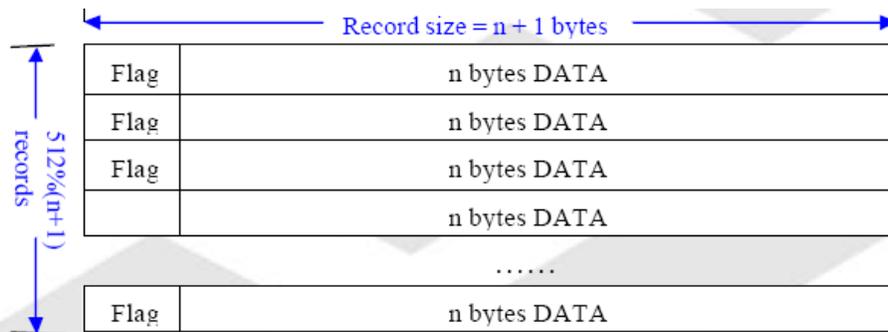


图5 存储配置数据作为记录

如果15个字节被要求存储地址代码和配置数据，那么要使用额外的一个字节作为标记，所以记录的尺寸是16个字节，一页可被使用 $512/16=32$ 次，其生命周期可延长至 $32*100,000=3,200,000$ 循环。

- **当闪存擦写/重新编程时的电流**

当删除/编程闪存时要考虑提供电流。当Vdd = 3.2V, Ffclk = 227.5 kHz, 擦除页使用5.78 mA, 并且花费4000Tfcyc, 也就是17.58 ms。

		特定的FCLK范围		客户的FCLK范围		单位
参数	FCLK循环	200	150	227.5	113.75	kHz
字节编程	9	45.00	60.00	39.56	79.12	μs
字节编程 (突发)	4	20.00	26.67	17.58	35.16	μs
页面擦除	4000	20.00	26.67	17.58	35.16	ms
大量擦除	20000	100.00	133.33	87.91	175.82	ms

**表3: 闪存编程/擦写时间**

- **闪存块保护**

由于Flash编程/删除代码包括在用户代码中, 所以闪存必须被块保护以避免由于程序运行错误所导致的意外内存操作。要块保护闪存, 用最后一个未被保护内存的地址来设置NVPROT寄存器。例如, 要通过0xFFFF保护0xFA00, 0xF9FF是最后一个未保护地址, 所以0xF8必须向NVPROT写为15位到9位的内存地址(8位到0位都是逻辑1)。

如果支持块保护, 确保向量是否被重新导向。如果向量被清除NVOPT寄存器内的FNORED位进行重新导向, 用户代码必须改变, 以从新的向量位置分配和接入向量。如果引导装入程序被在系统内编程或更新使用, 它就是块保护, 向量必须重新导向, 或者引导装入不能编辑默认的向量位置, 因为如果只支持块保护, 这些也被保护。

正常情况下, 中断服务的方式如下:

```

interrupt 3 int lvi_isr();
// The 2nd entry (number 2) gets the address of func g().
interrupt 4 int f(); // OK
// third entry in vector points to f()
    
```

3号或4号说明ISR函数的绝对向量数量，例如，QG4的向量3是低电压检测中断，向量位置是0xFFF8:0xFFF9。

但是如果支持向量的重新导向，这种情况就不生效——在被重新导向后，模拟比较器向量就转向地址0xFDF8:0xFDF9。仍旧从0xFFF8:0xFFF9输入中断服务将导致致命错误，然后中断服务的方法如下：

```
interrupt void lvi_isr(); // OK
// same as #pragma TRAP_PROC,
// please set the entry number in the prm-file

// In the PRM file, declare entry with its real address
VECTOR ADDRESS    0xFDD6        lvi_isr
```

宣布中断服务的另一种方法是在向量表起始的位置定义一个函数阵列。

```
/* Interrupt vector table */
void (* const _vect[])() @0xFFD0 = {
    UNASSIGNED_ISR, /* Int.no. 23 Vrti (at FFD0) */
    UNASSIGNED_ISR, /* Int.no. 22 Reserved2 (at FFD2) */
    UNASSIGNED_ISR, /* Int.no. 21 Reserved3 (at FFD4) */
    UNASSIGNED_ISR, /* Int.no. 20 Vacmp (at FFD6) */
    UNASSIGNED_ISR, /* Int.no. 19 Vadc (at FFD8) */
    UNASSIGNED_ISR, /* Int.no. 18 Vkeyboard (at FFDA) */
    UNASSIGNED_ISR, /* Int.no. 17 Viic (at FFDC) */
    UNASSIGNED_ISR, /* Int.no. 16 Vscitx (at FFDE)*/
    UNASSIGNED_ISR, /* Int.no. 15 Vscirx (at FFE0)*/
    UNASSIGNED_ISR, /* Int.no. 14 Vscierr (at FFE2)*/
    UNASSIGNED_ISR, /* Int.no. 13 Vspi (at FFE4) */
    UNASSIGNED_ISR, /* Int.no. 12 Vmtim (at FFE6) */
    UNASSIGNED_ISR, /* Int.no. 11 Reserved13 (at FFE8) */
    UNASSIGNED_ISR, /* Int.no. 10 Reserved14 (at FFEA) */
    UNASSIGNED_ISR, /* Int.no. 9 Reserved15 (at FFEC) */
    UNASSIGNED_ISR, /* Int.no. 8 Reserved16 (at FFEE) */
    UNASSIGNED_ISR, /* Int.no. 7 Vtpmovf (at FFF0) */
    UNASSIGNED_ISR, /* Int.no. 6 Vtpmchl (at FFF2) */
    UNASSIGNED_ISR, /* Int.no. 5 Vtpmch0 (at FFF4)*/
    UNASSIGNED_ISR, /* Int.no. 4 Reserved20 (at FFF6)*/
    lvi_isr, /* Int.no. 3 Vlvd (at FFF8) */
    UNASSIGNED_ISR, /* Int.no. 2 Virq (at FFFA) */
    UNASSIGNED_ISR, /* Int.no. 1 Vswi (at FFFC) */
    _startup /* Int.no. 0 Vreset (at FFFE)Reset vector*/
};
```

一旦向量被重新导向，就可以将函数阵列地址改变到新的位置。

```
void (* const _vect[])() @0xFDD0 = {
    UNASSIGNED_ISR, /* Int.no. 23 Vrti (at FFD0) */
    // .....
    lvi_isr,        /* Int.no. 3 Vlvd (at FFF8) */
    UNASSIGNED_ISR, /* Int.no. 2 Virq (at FFFA) */
    UNASSIGNED_ISR, /* Int.no. 1 Vswi (at FFFC) */
    UNASSIGNED_ISR, /* Int.no. 0 Vreset (at FFFE) Reset vector*/}
```

在任何情况下，重设向量都不被重新导向，它总是0xFFFE:0xFFFF。所以，启动函数地址不能改变。单独在PRM文件中的定义它的函数地址。

```
VECTOR_0_Startup /* Reset vector: this is the default entry point for an application. */
```

## 模数转换器（ADC）的配置

- 模数转换器（ADC）时钟选择

模数转换器（ADC）能够使用多点控制单元MCU总线时钟、一个总线时钟脉冲分为两次或在模块内的本地异步时钟进行转换操作。如果总线频率低于fADCK的频率，当支持短信号时，连续转化的准确信号时间就不能保证。如果总线频率低于fADCK频率的1/11，当支持长信号时，连续转化的准确信号时间就不能保证。

当455kHz振荡电路被使用，并且总线频率是227.5kHz时，异步时钟（ADACK）应该被选为模数转换器（ADC）的时钟源。

## 比较

表2 列出烟雾探测仪应用最常使用的MCU。

	MC9S08QG4		PIC16F676		MSP430F1122		TINY24	
	典型	最大	典型	最大	典型	最大	典型	最大
Supply Current@3.0V, 1 MHz	490uA*	1mA*	500uA**	780uA**	300uA***	350 uA***	400 uA	无

Supply Current@3.0V, 455 kHz	200uA <sup>^</sup>		<sup>^</sup> <sup>^</sup> 无	无	无	无	无	无
内部OSC的准确度	0.20%	2%	1%	无	无	5%	1.00%	3%
EEPROM	闪存模拟		内部128字节		闪存模拟		内部128字节	

\*这是为Vdd=3V、Fbus=1 MHz、FBE模式（带有外部2 MHz晶振）提供的电流。FLL使用大约220-310uA，ICR一般使用100uA，对于FEI模式，一般490+250+100=840uA；同时OSC在LP模式下使用5uA，带有32kHz外部表晶体振荡器的FEE模式大概是490+250+5=745uA。

\*\*这是为Vdd=3V、Fcpu=1 MHz，带有外部4 MHz RC振荡器提供的电流。

\*\*\*这是为Vdd=3V、Fmclk=1 MHz，带有32768表晶体振荡器提供的电流。因此包括DCO和振荡器的供应电流。

<sup>^</sup>FBE模式，Fbus=455(2\*2)=113.75kHz。

<sup>^^</sup>Fmclk=455/4=113.75kHz。

**表4: MCU的比较**

## 参考

飞思卡尔MC9S08QG4数据表

飞思卡尔AN2493、MC9S08GB/GT低功耗模式