

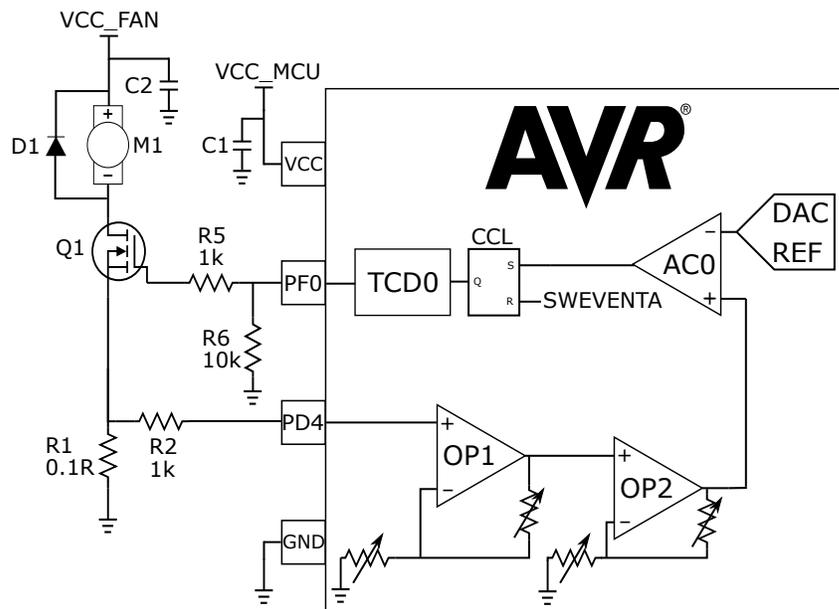
**使用 AVR® DB 系列单片机的集成运放实现过流保护**

**简介**

作者: Johan Vaarlid 和 Martin Mostad, Microchip Technology Inc.

AVR® DB 上的 D 型定时器/计数器 (Timer/Counter type D, TCD) 具有异步事件输入, 可用于在故障条件下关闭输出。本应用笔记将演示并介绍如何检测过流, 具体方法是使用可配置内部 OPAMP (模拟信号调理) 和模拟比较器 (Analog Comparator, AC) 检测风扇电机故障, 如果检测到故障, 则以信号形式发送到 TCD 的异步事件输入, 从而关闭负载。图 1 概述了工作原理:

**图 1. 过流保护电路原理图**



在正常工作模式下, TCD 输出引脚通过晶体管栅极控制风扇的速度。模拟比较器 (AC) 通过使用单片机上集成的运算放大器实现的电流检测放大器监视风扇电流。

R1 是电流检测电阻, 流经电机的电流通过该电阻时会产生电压。当 R1 上出现高压尖峰时, R2 会限制流入单片机的电流。

AC 配置为在风扇电流升至选定阈值以上后立即发送事件。造成此类电流尖峰的原因可能是机械障碍导致的风扇停转。器件的事件系统配置为将事件从 AC 发送到可配置定制逻辑 (Configurable Custom Logic, CCL) 中用于停止 TCD 输出的 RS 锁存器。

当风扇出现故障时, 上升电流会触发 AC 向 RS 锁存器的 S 端口发送一个事件, 从而使输出 Q 为高电平状态。该信号随后发送到 TCD 故障事件。TCD 将根据其异步事件输入中止驱动风扇并关闭电源电流。

在风扇电源被 TCD 中断后，可以执行应用特定的操作，具体取决于应用。一些可能的操作包括尝试重启、中止操作或发送故障信号。在本例中，我们已通过按下 AVR128DB48 Curiosity Nano 上的按钮启动简单的重启算法。

本应用是实现功能安全的一个示例。风扇电机会出现电流尖峰的一种情况是电机承受应力时，例如手指或物体卡在或压在风扇叶片上时。在此类事件中，可使用自动关断来保护最终用户免受伤害并降低损坏几率。

可从 GitHub 获取示例代码，以重现本应用笔记中所述的结果。



有关 AC、TCD、CCL、OPAMP、器件性能和常规配置的更多详细信息，请参见器件数据手册。

## 特性

- 根据各种正常负载电流消耗自动校准 AC
- 将运放与内部梯形电阻网络相集成来实现可调增益，使解决方案更具适应性
- 降低 BOM 并减小 PCB 面积
- 通过 MPLAB® Mindi™ 模型和原理图简化开发与调整过程

---

## 目录

---

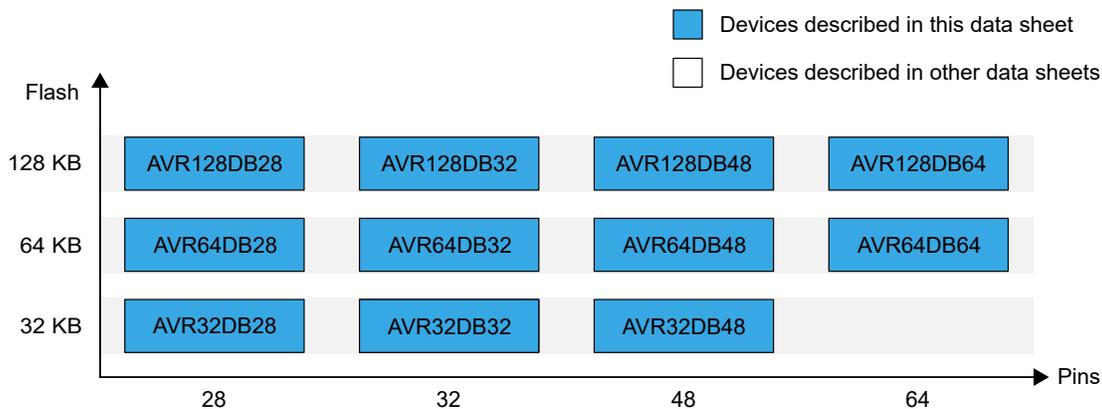
简介.....	1
特性.....	2
1. 相关器件.....	4
2. 概述.....	5
3. 硬件配置.....	6
4. 电流尖峰检测代码.....	7
4.1. 工作代码的结果.....	7
4.2. 初始化.....	7
4.3. 电机启动和 AC 校准.....	10
4.4. 检测和重启.....	11
5. MPLAB® Mindi™.....	12
5.1. 使用 MPLAB® Mindi™ 建模.....	12
5.2. AVR® DB 元件.....	13
5.3. 运行仿真.....	14
6. 参考资料.....	15
7. 版本历史.....	16
Microchip 网站.....	17
产品变更通知服务.....	17
客户支持.....	17
Microchip 器件代码保护功能.....	17
法律声明.....	17
商标.....	18
质量管理体系.....	18
全球销售及服务网点.....	19

## 1. 相关器件

本章列出了本文档的相关器件。下图给出了不同系列的器件之间的关系，并注明了不同的引脚数与存储器大小：

- 从下到上迁移无需修改代码，因为这些器件的引脚彼此兼容，后者可提供相同甚至更多的功能
- 水平向左移植会减少引脚数，进而减少可用的功能
- 具有不同闪存大小的器件通常也具有不同的 SRAM 和 EEPROM

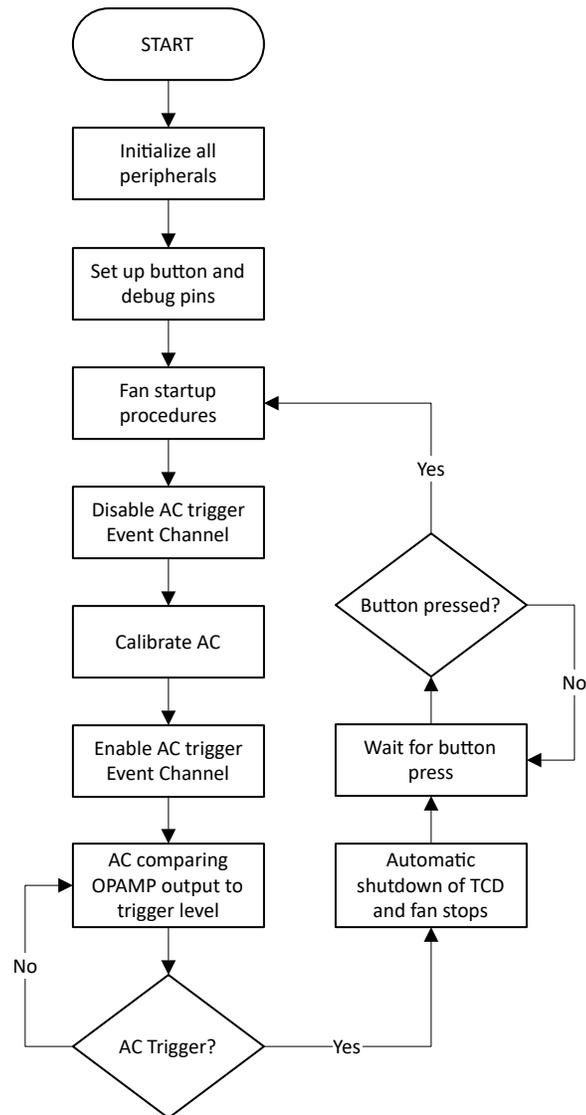
图 1-1. AVR® DB 系列概览



## 2. 概述

应用程序创建了一个电流尖峰检测器来独立于 CPU 连续监视电机电流。启动时，该器件将测量电流检测电阻两端的最大电压。为了避免误报，以及提供一些误差裕量，该电压限值增加了 50 mV。如果检测到尖峰，TCD 将自动关闭电机，代码将等待至按钮按下才会再次启动电机。

图 2-1. 代码流程图



### 3. 硬件配置

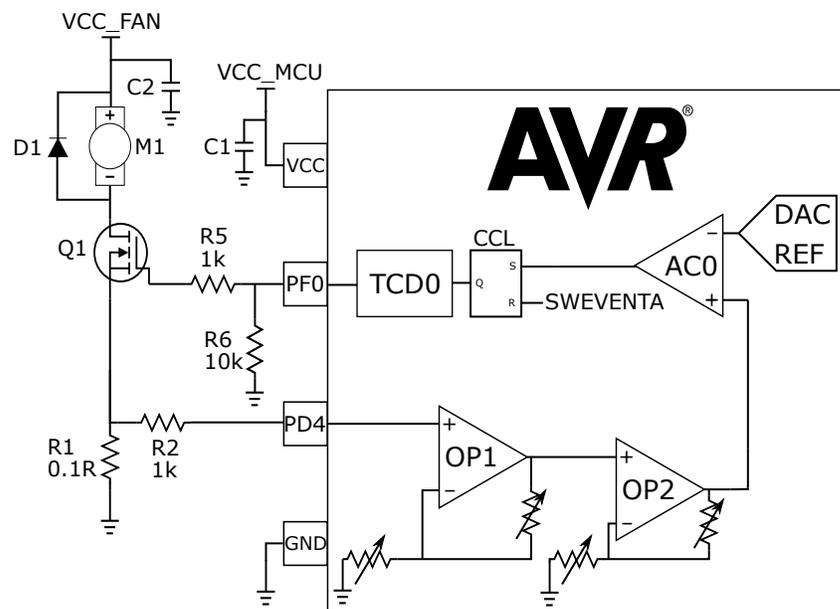
本应用笔记假定使用 AVR128DB48 Curiosity Nano (EV35L43A)。要设置该电路，可以将面包板或条状箔铜 PCB 与 AVR128DB48 Curiosity Nano 结合使用，另外需要以下元件：

- [AVR128DB48 Curiosity Nano 评估工具包](#)
- 小型 5V 计算机风扇或类似风扇 (M1)
- 2 个 1 k $\Omega$  电阻 (R2 和 R5)
- 1 个 10 k $\Omega$  电阻 (R6)
- 1 个 0.1 $\Omega$  电流检测电阻 (R1)
- 2 个 100 nF 电容 (C1 和 C2)
- 1 个 N 沟道 MOSFET 晶体管 (Q1)
- 1 个信号二极管 (D1)

注：AVR128DB48 Curiosity Nano 上已配有 C1，因此使用该工具包时无需再添加 C1。

上述元件必须按照以下原理图所示进行连接：

图 3-1. 过流保护电路原理图

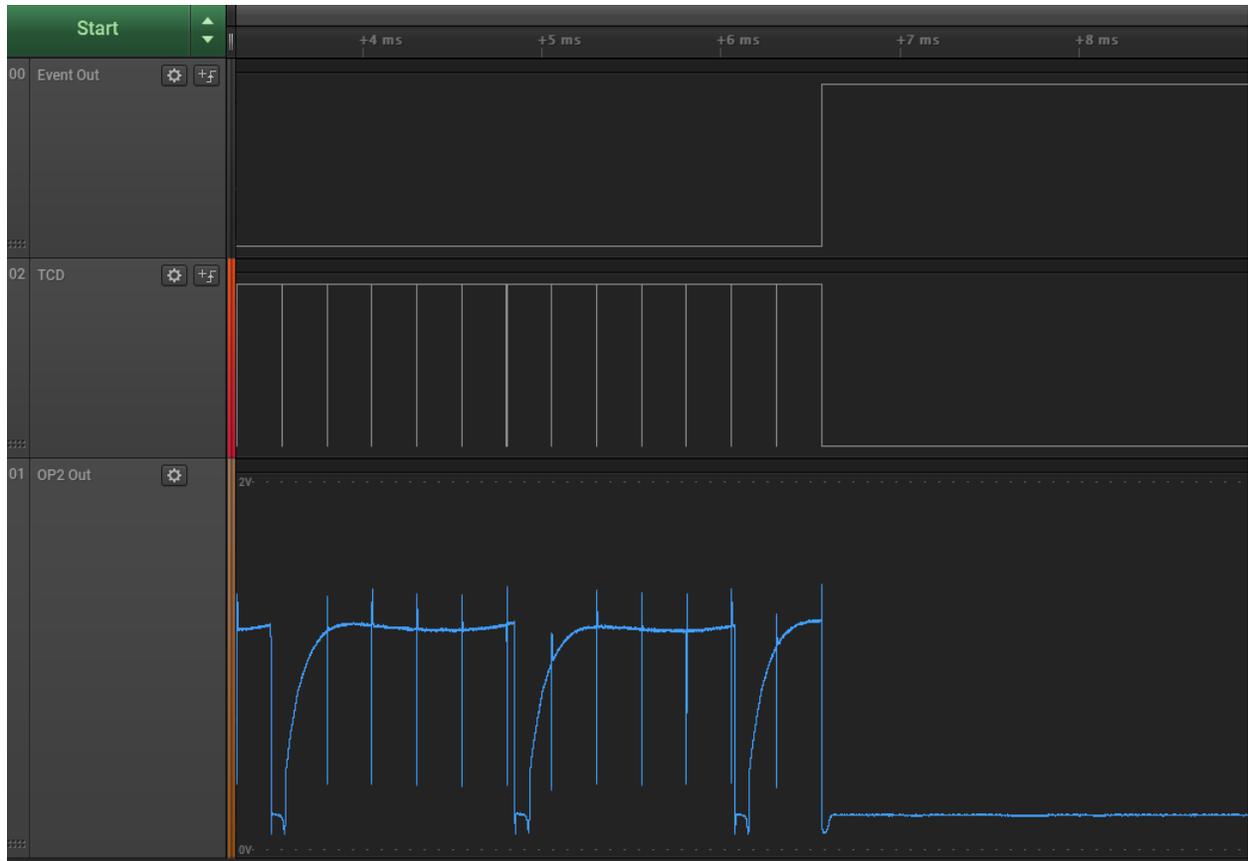


## 4. 电流尖峰检测代码

### 4.1 工作代码的结果

按照前一章所述设置硬件后，得到的 TCD 输出和电机电流消耗应如图 4-1 所示。电机不同，电流消耗曲线也可能会有所不同。“OP2 Out”（OP2 输出）中所示的尖峰由 TCD 翻转产生的噪声引起。在本例中，TCD 以最大占空比运行，即只有一个周期为低电平，其余时间均为高电平。

图 4-1. 电机故障事件（以 50 MHz 采样）



### 4.2 初始化

首先，初始化要使用的外设：TCD、CCL、AC 和 OPAMP。

先将 OPAMP 设置为两个级联的同相可编程增益放大器（Programmable Gain Amplifier, PGA），总增益为 60。该增益基于开发过程中使用的风扇电机的电流消耗所产生的输入信号来选择。可以根据电流消耗选择更高或更低的增益，从而满足应用需求。为了降低噪声，级联中的第一个运放设置为最大增益 16x，而第二个运放可根据应用需求调整，最好是可得到 1-2V 范围输出的增益，以便于检测。如果使用其他设置，请参见 [AVR128DB48 数据手册](#) 来计算增益。下表列出了 OPAMP.OP2RESMUX 的可能增益设置。

表 4-1. OPAMP.OP2RESMUX 的增益设置

组配置	得到的总增益
OPAMP_OP2RESMUX_MUXWIP_WIP0_gc	~17

..... (续)

组配置	得到的总增益
OPAMP_OP2RESMUX_MUXWIP_WIP1_gc	~18
OPAMP_OP2RESMUX_MUXWIP_WIP2_gc	~21
OPAMP_OP2RESMUX_MUXWIP_WIP3_gc	~32
OPAMP_OP2RESMUX_MUXWIP_WIP4_gc	~43
OPAMP_OP2RESMUX_MUXWIP_WIP5_gc	~64
OPAMP_OP2RESMUX_MUXWIP_WIP6_gc	~128
OPAMP_OP2RESMUX_MUXWIP_WIP7_gc	~256

```
void opamp_init(void)
{
    /* 禁止运放输出引脚上的输入 */
    PORTD.PIN5CTRL = PORT_ISC_INPUT_DISABLE_gc;
    PORTE.PIN1CTRL = PORT_ISC_INPUT_DISABLE_gc;

    /* 设置运放 */
    OPAMP.CTRLA = OPAMP_ENABLE_bm;
    OPAMP.TIMEBASE = (uint8_t) ceil(CLK_PER*0.000001)-1; /* 相当于 1 μs 的外设时钟周期数 */

    //OP2 setup
    OPAMP.OP2CTRLA = OPAMP_RUNSTBY_bm | OPAMP_ALWAYS_ON_bm | OPAMP_OP2CTRLA_OUTMODE_NORMAL_gc;
    OPAMP.OP2SETTLE = OPAMP_MAX_SETTLE_TIME; //由于稳定时间未知, 因此应设置最大值
    OPAMP.OP2INMUX = OPAMP_OP2INMUX_MUXNEG_WIP_gc | OPAMP_OP2INMUX_MUXPOS_LINKOUT_gc;
    OPAMP.OP2RESMUX = OPAMP_OP2RESMUX_MUXWIP_WIP5_gc | OPAMP_OP2RESMUX_MUXBOT_GND_gc |
    OPAMP_OP2RESMUX_MUXTOP_OUT_gc;

    //OP1 设置
    OPAMP.OP1CTRLA = OPAMP_RUNSTBY_bm | OPAMP_ALWAYS_ON_bm | OPAMP_OP1CTRLA_OUTMODE_NORMAL_gc;
    OPAMP.OP1SETTLE = OPAMP_MAX_SETTLE_TIME; //由于稳定时间未知, 因此应设置最大值
    OPAMP.OP1INMUX = OPAMP_OP1INMUX_MUXNEG_WIP_gc | OPAMP_OP1INMUX_MUXPOS_INP_gc;
    OPAMP.OP1RESMUX = OPAMP_OP1RESMUX_MUXWIP_WIP7_gc | OPAMP_OP1RESMUX_MUXBOT_INN_gc |
    OPAMP_OP1RESMUX_MUXTOP_OUT_gc;
}
```

然后, 将 TCD 设置为驱动 PF0 引脚并在发生故障事件时关闭。当前, TCD 未使能, 但将在稍后启动。

```
void tcd_init(void)
{
    PORTMUX.TCDROUTEA = PORTMUX_TCD0_ALT2_gc;
    PORTF.DIRSET=PIN0_bm;

    TCD0.CTRLB = TCD_WGMODE_TWORAMP_gc;
    TCD0.CMPASET = 0; /* 比较 A 置 1: 0 */
    TCD0.CMPACLAR = 1001; /* 比较 A 清零: 1001 */
    TCD0.CMPBSET = 0; /* 比较 B 置 1: 0 */
    TCD0.CMPBCLR = 1000; /* 比较 B 清零: 1000 */

    ccp_write_io((void*)&(TCD0.FAULTCTRL), 1 << TCD_CMPAEN_bp /* 比较 A 使能: 使能 */
    | 0 << TCD_CMPA_bp /* 比较 A 值: 禁止 */
    | 0 << TCD_CMPB_bp /* 比较 B 值: 禁止 */
    | 0 << TCD_CMPBEN_bp /* 比较 B 使能: 禁止 */
    | 0 << TCD_CMPC_bp /* 比较 C 值: 禁止 */
    | 0 << TCD_CMPCEN_bp /* 比较 C 使能: 禁止 */
    | 0 << TCD_CMPD_bp /* 比较 D 值: 禁止 */
    | 0 << TCD_CMPDEN_bp /* 比较 D 使能: 禁止 */);

    TCD0.EVCTRLA = TCD_CFG_ASYNC_gc /* 滤波器或异步事件均未使能 */
    | TCD_ACTION_FAULT_gc /* 事件触发故障 */
    | TCD_EDGE_RISE_HIGH_gc /* 事件的下降沿或低电平产生重新触发或故障操作 */
    | 1 << TCD_TRIGE1_bp; /* 触发事件使能: 使能 */
}
```

```

TCD0.INPUTCTRLA=TCD_INPUTMODE_WAITSW_gc; //等待复位命令
TCD0.INPUTCTRLB=TCD_INPUTMODE_WAITSW_gc; //等待复位命令

while ((TCD0.STATUS & TCD_ENRDY_bm) == 0); //等待使能就绪变为高电平

TCD0.CTRLA = 0 << TCD_ENABLE_bp      /* 使能: 禁止 */
| TCD_CLKSEL_OSCHF_gc      /* */
| TCD_CNTPRES_DIV1_gc      /* 同步时钟进行1分频 */
| TCD_SYNCPRES_DIV1_gc;

}

```

AC 需要设置为以适当电平触发。首先，将 AC 设置为以某个低电平（此处为 1.1V）触发。该电平随后将自动校准以满足目标电机需求。

```

void ac_init(void)
{
    PORTA.DIRSET=PIN7_bm;
    VREF.ACREF = VREF_REFSEL_4V096_gc;

    AC0.MUXCTRL = 0 << AC_INVERT_bp      /* 将 AC 输出反相: 禁止 */
    | AC_MUXNEG_DACREF_gc /* DAC 参考电压 */
    | AC_MUXPOS_AINP2_gc; /* 同相引脚0 */

    AC0.DACREF = ac_calculate_trigger_voltage(AC_TRIGGER_VOLTAGE_MV_INIT); /* DAC 参考电压: 0x64 */

    AC0.CTRLA = 1 << AC_ENABLE_bp /* 使能: 使能 */
    | AC_HYSMODE_NONE_gc /* 无滞后 */
    | AC_POWER_PROFILE0_gc /* 功率曲线 0, 最低功耗和最大响应时间。 */
    | 1 << AC_OUTEN_bp /* 输出缓冲器使能: 使能 */
    | 0 << AC_RUNSTDBY_bp; /* 在待机模式下运行: 禁止 */
}

```

辅助函数 `ac_calculate_trigger_voltage(uint16_t mV)` 用于设置 AC 触发信号，以及计算触发电平（mV）对应的 DACREF 值。

```

uint8_t ac_calculate_trigger_voltage(uint16_t mV)
{
    uint8_t triggerVoltage = (((uint32_t)mV*256)/VREF_AC_MV);
    return triggerVoltage;
}

```

为了使过流信号在复位前保持有效，CCL LUT0 和 LUT1 构成了一个 RS 锁存器，其中 AC0 输出用作 S 输入，软件事件 A 用作 R 输入，如图 3-1 所示。

```

void ccl_init(void)
{
    CCL.SEQCTRL0 = CCL_SEQSEL_RS_gc; // 构成一个 RS 锁存器
    /* 设置 LUT0 */
    CCL.LUT0CTRLB = CCL_INSEL0_AC0_gc; // AC0, 屏蔽所有其他位
    CCL.TRUTH0 = 0x02; // 当 AC 为高电平时, 输出为高电平, 否则输出为低电平
    CCL.LUT0CTRLA = CCL_ENABLE_bm | CCL_OUTEN_bm;
    PORTA.DIRSET = PIN3_bm;

    /* 设置 LUT1 */
    EVSYS.USERCCLLUT1A = EVSYS_USER_CHANNEL1_gc;
    CCL.LUT1CTRLB = CCL_INSEL0_EVENTA_gc; // 事件 A 用作输入, 屏蔽所有其他位
    CCL.TRUTH1 = 0x02; // 当事件 A 为高电平时, 输出为高电平, 否则输出为低电平
    CCL.LUT1CTRLA = CCL_ENABLE_bm;

    CCL.CTRLA = CCL_ENABLE_bm;
}

```

最后，设置用于显示 AC 事件电平的按钮和调试信号。

```
void button_init(void)
{
    PORTB.DIRCLR=PIN2_bm;
    PORTB.PIN2CTRL=PORT_ISC_FALLING_gc | PORT_PULLUPEN_bm;
    sei();
}
```

```
button_init();
PORTC.DIRSET = PIN4_bm | PIN5_bm;
PORTC.OUTCLR = PIN4_bm | PIN5_bm;
PORTD.DIRSET = PIN2_bm;
```

### 4.3 电机启动和 AC 校准

要启动风扇电机，应利用下面给出的 `fan_start()` 函数。该函数会调用一系列其他函数，这些函数将按照时间顺序进行说明。

```
void fan_start(void)
{
    ac_trig_event_disable();
    tcd_enable();
    ac_calibration();
    ac_trig_event_enable();
}
```

`ac_trig_event_disable()` 将 AC 触发事件与 CCLRS 锁存器断开连接，从而禁止电流尖峰检测器。

```
void ac_trig_event_disable(void)
{
    AC0.INTCTRL &= ~AC_CMP_bm;
    EVSYS.CHANNEL0=EVSYS_CHANNEL0_OFF_gc;
    EVSYS.USERTCDD0INPUTA = EVSYS_USER_OFF_gc;
    EVSYS.USEREVSYSSEVOUtd = EVSYS_USER_OFF_gc;
}
```

`tcd_enable()` 随后通过使能 TCD 外设来启动风扇电机。

```
void tcd_enable(void)
{
    TCD0.CTRLA = TCD_ENABLE_bm;
    TCD0.CTRLE = TCD_RESTART_bm;
}
```

`ac_calibration()` 以 50 mV 为单位增量增大 AC 的触发点，直到 100 ms 内未发生触发为止。此操作在每次启动时完成，以确保最优检测条件并避免误触发。一些电机的电流消耗受温度和其他物理条件影响。

```
void ac_calibration(void)
{
    PORTB.DIRSET = PIN3_bm;
    PORTB.OUTCLR = PIN3_bm;
    uint8_t calibrating = 1;
    uint16_t ac_trigger_voltage_mv=AC_TRIGGER_VOLTAGE_MV_INIT;
    AC0.INTCTRL = AC_INTMODE_NORMAL_POSEDGE_gc;
    while (calibrating)
    {
        ac_trigger_voltage_mv += 50;
        AC0.DACREF = ac_calculate_trigger_voltage(ac_trigger_voltage_mv);
        _delay_ms(100); // 留出一定时间将标志置 1
        if (!(AC0.STATUS & AC_CMPIF_bm))
        {
            calibrating=0;
            PORTB.OUTSET =PIN3_bm;
        }
        AC0.STATUS=AC_CMPIF_bm;
    }
}
```

```
}  
}
```

ac\_trig\_event\_enable() 将 AC 触发事件重新连接到 CCL，以便系统准备好检测电流尖峰事件。

```
void ac_trig_event_enable(void)  
{  
    AC0.INTCTRL = AC_INTMODE_NORMAL_POSEDGE_gc | AC_CMP_bm;  
    EVSYS.SWEVENTA = EVSYS_SWEVENTA_CH1_gc;  
    EVSYS.CHANNEL0=EVSYS_CHANNEL0_CCL_LUT0_gc; //RS 锁存器的输出用作触发信号  
    EVSYS.USERTCDOINPUTA=EVSYS_USER_CHANNEL0_gc;  
    EVSYS.USEREVSYSSEVOUTD = EVSYS_USER_CHANNEL0_gc;  
}
```

## 4.4 检测和重启

检测到电流尖峰后，硬件将使电机停转，无需 CPU 参与。按下 AVR128DB48 Curiosity Nano 上的按钮 SW0 将触发中断服务程序（Interrupt Service Routine, ISR）ISR(PORTB\_PORT\_vect)，该程序将调用 fan\_start() 程序来重启风扇电机。

```
ISR(PORTB_PORT_vect)  
{  
    fan_start();  
    PORTB.INTFLAGS=0xff;  
}
```

## 5. MPLAB® Mindi™

为方便根据不同的环境和风扇来开发与调整该应用程序，针对该应用程序创建了 MPLAB Mindi 仿真。仿真由两部分组成：第一部分是代表风扇的直流电机模型。第二部分是重现 AVR DB 的功能所需的所有元件。只有运放元件是 AVR DB 中的真实电路。其余元件均为仿真 AVR DB 行为所需的通用电路元件。

### 5.1 使用 MPLAB Mindi 建模

经典的直流电机建模方法是使用两个差分公式，分别为机械动态差分公式和电气动态差分公式，如下所示：

$$J\ddot{\theta} + b\dot{\theta} = Ki$$

$$L\dot{i} + Ri = V - K\dot{\theta}$$

MPLAB Mindi 只能仿真电路元件，因此机械公式应转换为等效的电路表示。具体实现方法是假定  $R_{los} = b/K^2$ 、 $L_l = J/K^2$  且  $i_{emf} = \dot{\theta}K$ ，得到的差分公式如下：

$$L_l \dot{i}_{emf} + R_{los} i_{emf} = i$$

$$L\dot{i} + Ri = V - i_{emf}$$

直流电机模型由三部分组成：[Motor\\_Electrical](#)、[Motor\\_Mechanical](#) 和 [Motor\\_Load](#)。

#### 5.1.1 电机电气

电机电气部分实现了上述的电气差分公式。该部分使用一个任意的电压源充当由 PWM 信号控制的直流电机的电源。此外，还有一个任意的电压源用于将机械部分的电流转换为电压，即代表电机的电动势。

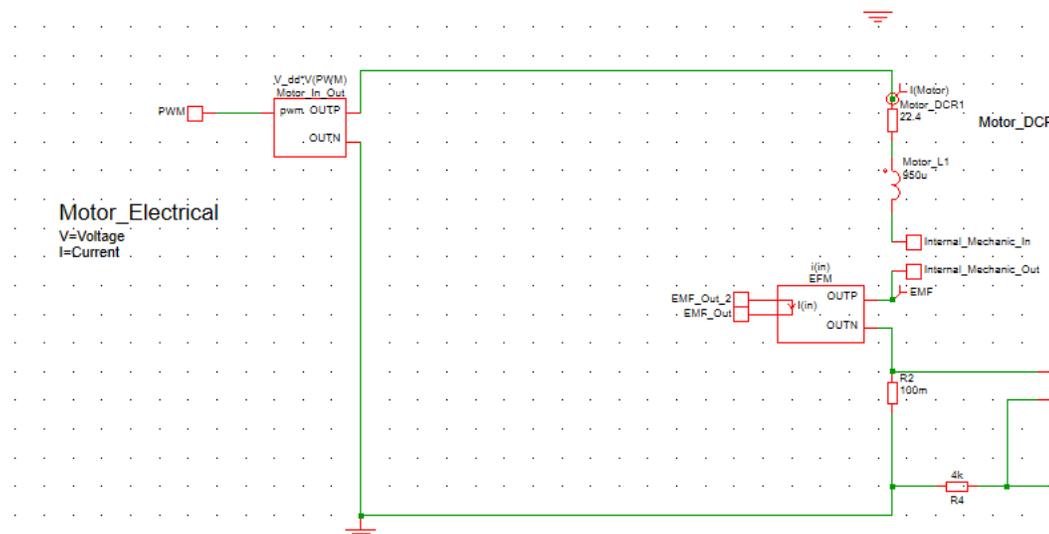
在电机电气部分中，R（表示为 [Motor\\_DCR1](#)）和 L（表示为 [Motor\\_L1](#)）这两个参数决定电机的行为，具体值可使用以下公式计算：

$$R = \frac{V_{in}}{i_{stall}}$$

$$L = R\tau_e$$

其中  $V_{in}$  为直流电机的输入电压， $i_{stall}$  为停转电流， $\tau_e$  为电气系统的时间常量。

图 5-1. 电机电气



### 5.1.2 电机机械

电机机械部分实现了上述转换后的机械差分公式。该部分包含两个任意的电压源，一个用于将电气部分的电流转换为电压，另一个用于代表负载引起的损耗。

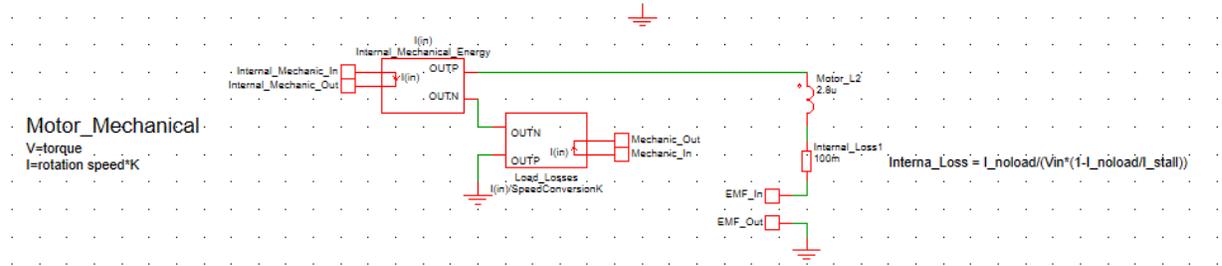
在电机机械部分中， $R_{loss}$  (Interla\_Loss1) 和  $L_i$  (Motor\_L2) 这两个参数决定电机的行为，具体值可使用以下公式计算：

$$R_{loss} = \frac{i_{no\_load}}{V_{in} \left( 1 - \frac{i_{no\_load}}{i_{stall}} \right)}$$

$$L = R_{loss} \tau_m$$

其中  $i_{no\_load}$  为无负载时电机的电流消耗， $i_{stall}$  为停转电流， $\tau_m$  为机械系统的时间常量。

图 5-2. 电机机械

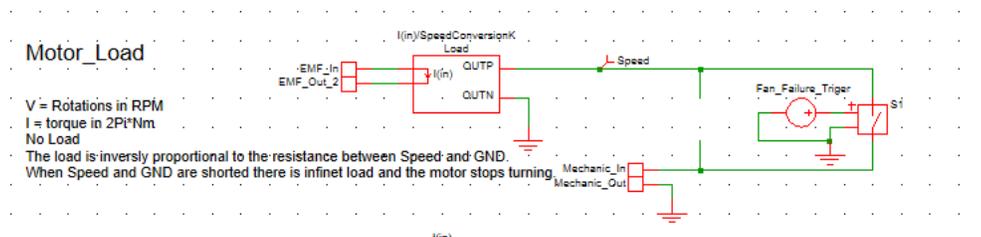


### 5.1.3 电机负载

电机负载部分代表负载引起的损耗，使用一个任意的电压源将机械部分中的电流转换为速度（RPM）。负载引起的损耗与电压源和地之间的电阻成反比。本例中使用的风扇永久连接到电机，因此风扇引起的损耗已计入机械部分。在正常工作模式下，电压源与地之间的电阻无限大（开路），即代表无负载。

为了仿真风扇故障（例如，某物在风扇中卡住），模型还配有一个开关，用于将电压源短接到地。该开关通过 Fan\_Failure\_Trigger 电压源发出的脉冲触发。

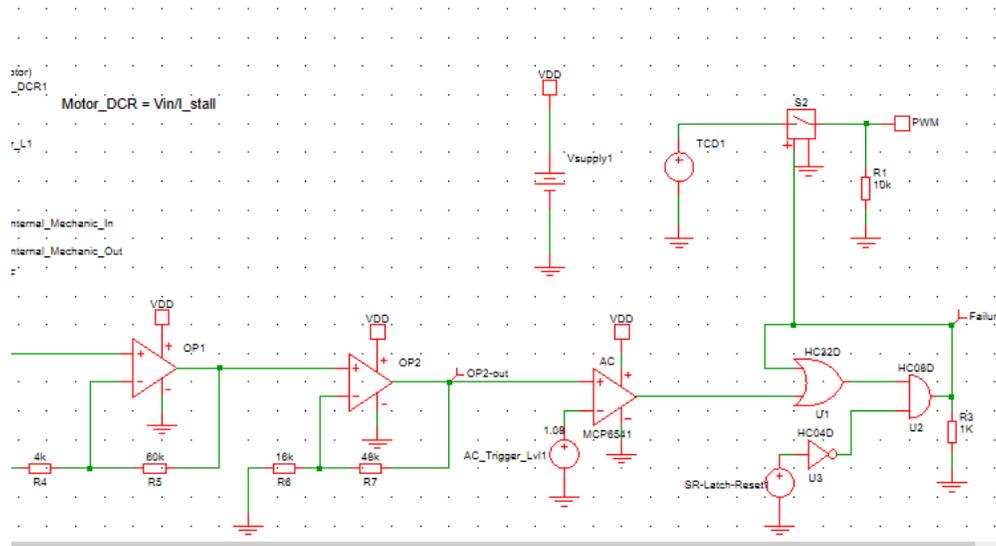
图 5-3. 电机负载



## 5.2 AVR® DB 元件

为了代表 AVR DB 功能，将两个 AVR DB 运放模型配置为级联同相 PGA，如同在软件中一样。OP2 的输出连接到通用模拟比较器（AC），后者代表 AVR DB 中的 AC。AC 还连接到电压源 AC\_Trigger\_Iv11，后者设置故障事件的触发电平。AC 的输出连接到由逻辑门和反相器构成的 RS 锁存器，如同在实际硬件中一样。RS 锁存器的输出连接到开关，后者将断开 PWM 信号与电机之间的连接，这表示 TCD 对故障事件作出反应并关闭。

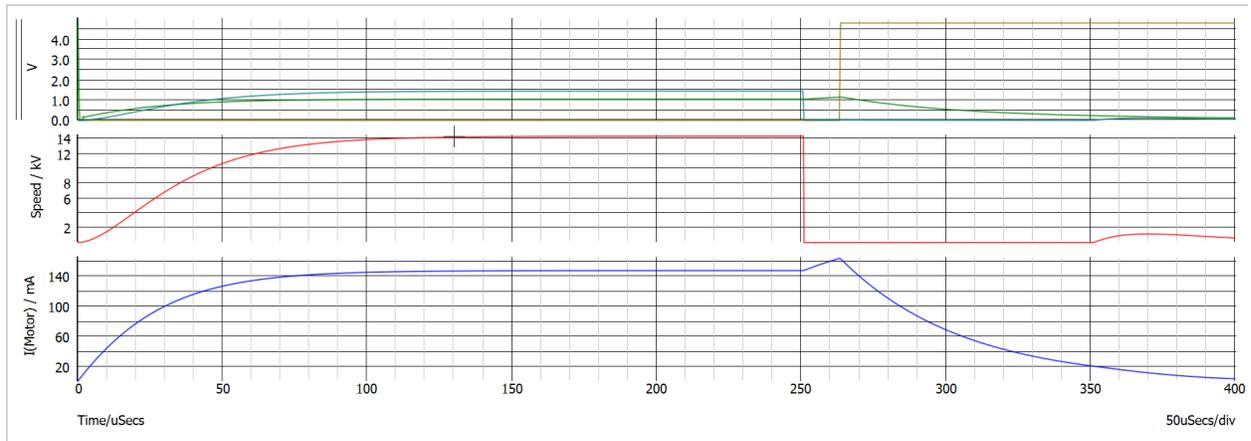
图 5-4. AVR® DB 元件



### 5.3 运行仿真

运行仿真之前，将上节所述的参数更改为满足目标直流电机需求的值。选择 **Simulator** → **Run Schematic**（仿真器 → 运行原理图）或按下功能键 **F9** 进行仿真。

图 5-5. 仿真



得到的输出图如上面的图 5-5 所示。蓝线代表电流消耗，红线代表电机速度（RPM），绿线代表运放的输出，蓝绿线代表电动势，黄线代表故障检测。可以看到，在大约 250  $\mu$ s 处，风扇被挡住而停转，产生电流尖峰，电流尖峰触发故障事件，最终电机关闭。

Mindi 模型的代码可从 GitHub 上获取：

**View Code Example on GitHub**

Click to browse repository

---

---

## 6. 参考资料

1. AVR128DB48 产品页面: [www.microchip.com/wwwproducts/en/AVR128DB48](http://www.microchip.com/wwwproducts/en/AVR128DB48)
2. AVR128DB48 数据手册: [www.microchip.com/DS40002247](http://www.microchip.com/DS40002247)
3. Curiosity Nano AVR128DB48 产品页面: [www.microchip.com/DevelopmentTools/ProductDetails/PartNO/EV35L43A](http://www.microchip.com/DevelopmentTools/ProductDetails/PartNO/EV35L43A)
4. AVR128DB48 Curiosity Nano 用户指南: [www.microchip.com/DS50003037](http://www.microchip.com/DS50003037)
5. Curiosity Nano AVR128DB48 原理图: [ww1.microchip.com/downloads/en/DeviceDoc/AVR128DB48\\_Curiosity\\_Nano\\_Schematics.pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/AVR128DB48_Curiosity_Nano_Schematics.pdf)

**7. 版本历史**

文档版本	日期	备注
B	2021 年 5 月	新增了原理图并更新了代码
A	2021 年 2 月	文档初始版本

---

## Microchip 网站

---

Microchip 网站 ([www.microchip.com](http://www.microchip.com)) 为客户提供在线支持。客户可通过该网站方便地获取文件和信息。我们的网站提供以下内容：

- **产品支持**——数据手册和勘误表、应用笔记和示例程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及归档软件
- **一般技术支持**——常见问题解答 (FAQ)、技术支持请求、在线讨论组以及 Microchip 设计伙伴计划成员名单
- **Microchip 业务**——产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排表、Microchip 销售办事处、代理商以及工厂代表列表

---

## 产品变更通知服务

---

Microchip 的产品变更通知服务有助于客户了解 Microchip 产品的最新信息。注册客户可在他们感兴趣的某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时，收到电子邮件通知。

欲注册，请访问 [www.microchip.com/pcn](http://www.microchip.com/pcn)，然后按照注册说明进行操作。

---

## 客户支持

---

Microchip 产品的用户可通过以下渠道获得帮助：

- 代理商或代表
- 当地销售办事处
- 应用工程师 (ESE)
- 技术支持

客户应联系其代理商、代表或 ESE 寻求支持。当地销售办事处也可为客户提供帮助。本文档后附有销售办事处的联系方式。

也可通过 [www.microchip.com/support](http://www.microchip.com/support) 获得网上技术支持。

---

## Microchip 器件代码保护功能

---

请注意以下有关 Microchip 产品代码保护功能的要点：

- Microchip 的产品均达到 Microchip 数据手册中所述的技术规范。
- Microchip 确信：在正常使用且符合工作规范的情况下，Microchip 系列产品非常安全。
- Microchip 注重并积极保护其知识产权。严禁任何试图破坏 Microchip 产品代码保护功能的行为，这种行为可能会违反《数字千年版权法案》(Digital Millennium Copyright Act)。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。代码保护功能处于持续发展中。Microchip 承诺将不断改进产品的代码保护功能。

---

## 法律声明

---

提供本文档的中文版本仅为了便于理解。请勿忽视文档中包含的英文部分，因为其中提供了有关 Microchip 产品性能和使用情况的有用信息。Microchip Technology Inc. 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc. 的英文原版文档。

本出版物及其提供的信息仅适用于 Microchip 产品，包括设计、测试以及将 Microchip 产品集成到您的应用中。以其他方式使用这些信息都将被视为违反条款。本出版物中的器件应用信息仅为您提供便利，将来可能会发生更新。如需额外的支持，请联系当地的 Microchip 销售办事处，或访问 <https://www.microchip.com/en-us/support/design-help/client-supportservices>。

Microchip “按原样”提供这些信息。Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对非侵权性、适销性和特定用途的适用性的暗示担保，或针对其使用情况、质量或性能的担保。

在任何情况下，对于因这些信息或使用这些信息而产生的任何间接的、特殊的、惩罚性的、偶然的或间接的损失、损害或任何类型的开销，Microchip 概不承担任何责任，即使 Microchip 已被告知可能发生损害或损害可以预见。在法律允许的最大范围内，对于因这些信息或使用这些信息而产生的所有索赔，Microchip 在任何情况下所承担的全部责任均不超出您为获得这些信息向 Microchip 直接支付的金额（如有）。如果将 Microchip 器件用于生命维持和/或生命安全应用，一切风险由买方自负。买方同意在由此引发任何一切损害、索赔、诉讼或费用时，会维护和保障 Microchip 免于承担法律责任。除非另外声明，在 Microchip 知识产权保护下，不得暗或以其他方式转让任何许可证。

## 商标

Microchip 的名称和徽标组合、Microchip 徽标、Adaptec、AnyRate、AVR、AVR 徽标、AVR Freaks、BesTime、BitCloud、chipKIT、chipKIT 徽标、CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、HELDO、IGLOO、JukeBlox、KeeLoq、Kleer、LANCheck、LinkMD、maXStylus、maXTouch、MediaLB、megaAVR、Microsemi、Microsemi 徽标、MOST、MOST 徽标、MPLAB、OptoLyzer、PacTime、PIC、picoPower、PICSTART、PIC32 徽标、PolarFire、Prochip Designer、QTouch、SAM-BA、SenGenuity、SpyNIC、SST、SST 徽标、SuperFlash、Symmetricom、SyncServer、Tachyon、TimeSource、tinyAVR、UNI/O、Vectron 和 XMEGA 均为 Microchip Technology Incorporated 在美国和其他国家或地区的注册商标。

AgileSwitch、APT、ClockWorks、The Embedded Control Solutions Company、EtherSynch、FlashTec、Hyper Speed Control、HyperLight Load、IntelliMOS、Libero、motorBench、mTouch、Powermite 3、Precision Edge、ProASIC、ProASIC Plus、ProASIC Plus 徽标、Quiet-Wire、SmartFusion、SyncWorld、Temux、TimeCesium、TimeHub、TimePictra、TimeProvider、WinPath 和 ZL 均为 Microchip Technology Incorporated 在美国的注册商标。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、Augmented Switching、BlueSky、BodyCom、CodeGuard、CryptoAuthentication、CryptoAutomotive、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、Espresso T1S、EtherGREEN、IdealBridge、In-Circuit Serial Programming、ICSP、INICnet、Intelligent Paralleling、Inter-Chip Connectivity、JitterBlocker、maxCrypto、maxView、memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certified 徽标、MPLIB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICtail、PowerSmart、PureSilicon、QMatrix、REAL ICE、Ripple Blocker、RTAX、RTG4、SAM-ICE、Serial Quad I/O、simpleMAP、SimpliPHY、SmartBuffer、SMART-I.S.、storClad、SQI、SuperSwitcher、SuperSwitcher II、Switchtec、SynchroPHY、Total Endurance、TSHARC、USBCheck、VariSense、VectorBlox、VeriPHY、ViewSpan、WiperLock、XpressConnect 和 ZENA 均为 Microchip Technology Incorporated 在美国和其他国家或地区的商标。

SQTP 是 Microchip Technology Incorporated 在美国的服务标记。

Adaptec 徽标、Frequency on Demand、Silicon Storage Technology 和 Symmcom 为 Microchip Technology Inc. 在其他国家或地区的注册商标。

GestIC 是 Microchip Technology Inc. 的子公司 Microchip Technology Germany II GmbH & Co. KG 在其他国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2024, Microchip Technology Incorporated 及其子公司。版权所有。

ISBN: 978-1-6683-3883-4

## 质量管理体系

有关 Microchip 的质量管理体系的信息，请访问 [www.microchip.com/quality](http://www.microchip.com/quality)。

## 全球销售及服务中心

美洲	亚太地区	亚太地区	欧洲
<b>公司总部</b> 2355 West Chandler Blvd. Chandler, AZ 85224-6199 电话: 480-792-7200 传真: 480-792-7277 技术支持: <a href="http://www.microchip.com/support">www.microchip.com/support</a> 网址: <a href="http://www.microchip.com">www.microchip.com</a>	<b>澳大利亚 - 悉尼</b> 电话: 61-2-9868-6733 <b>中国 - 北京</b> 电话: 86-10-8569-7000 <b>中国 - 成都</b> 电话: 86-28-8665-5511 <b>中国 - 重庆</b> 电话: 86-23-8980-9588 <b>中国 - 东莞</b> 电话: 86-769-8702-9880 <b>中国 - 广州</b> 电话: 86-20-8755-8029 <b>中国 - 杭州</b> 电话: 86-571-8792-8115 <b>中国 - 香港特别行政区</b> 电话: 852-2943-5100 <b>中国 - 南京</b> 电话: 86-25-8473-2460 <b>中国 - 青岛</b> 电话: 86-532-8502-7355 <b>中国 - 上海</b> 电话: 86-21-3326-8000 <b>中国 - 沈阳</b> 电话: 86-24-2334-2829 <b>中国 - 深圳</b> 电话: 86-755-8864-2200 <b>中国 - 苏州</b> 电话: 86-186-6233-1526 <b>中国 - 武汉</b> 电话: 86-27-5980-5300 <b>中国 - 西安</b> 电话: 86-29-8833-7252 <b>中国 - 厦门</b> 电话: 86-592-2388138 <b>中国 - 珠海</b> 电话: 86-756-3210040	<b>印度 - 班加罗尔</b> 电话: 91-80-3090-4444 <b>印度 - 新德里</b> 电话: 91-11-4160-8631 <b>印度 - 浦那</b> 电话: 91-20-4121-0141 <b>日本 - 大阪</b> 电话: 81-6-6152-7160 <b>日本 - 东京</b> 电话: 81-3-6880-3770 <b>韩国 - 大邱</b> 电话: 82-53-744-4301 <b>韩国 - 首尔</b> 电话: 82-2-554-7200 <b>马来西亚 - 吉隆坡</b> 电话: 60-3-7651-7906 <b>马来西亚 - 槟榔屿</b> 电话: 60-4-227-8870 <b>菲律宾 - 马尼拉</b> 电话: 63-2-634-9065 <b>新加坡</b> 电话: 65-6334-8870 <b>台湾地区 - 新竹</b> 电话: 886-3-577-8366 <b>台湾地区 - 高雄</b> 电话: 886-7-213-7830 <b>台湾地区 - 台北</b> 电话: 886-2-2508-8600 <b>泰国 - 曼谷</b> 电话: 66-2-694-1351 <b>越南 - 胡志明市</b> 电话: 84-28-5448-2100	<b>奥地利 - 韦尔斯</b> 电话: 43-7242-2244-39 传真: 43-7242-2244-393 <b>丹麦 - 哥本哈根</b> 电话: 45-4485-5910 传真: 45-4485-2829 <b>芬兰 - 埃斯波</b> 电话: 358-9-4520-820 <b>法国 - 巴黎</b> 电话: 33-1-69-53-63-20 传真: 33-1-69-30-90-79 <b>德国 - 加兴</b> 电话: 49-8931-9700 <b>德国 - 哈恩</b> 电话: 49-2129-3766400 <b>德国 - 海尔布隆</b> 电话: 49-7131-72400 <b>德国 - 卡尔斯鲁厄</b> 电话: 49-721-625370 <b>德国 - 慕尼黑</b> 电话: 49-89-627-144-0 传真: 49-89-627-144-44 <b>德国 - 罗森海姆</b> 电话: 49-8031-354-560 <b>以色列 - 若那那市</b> 电话: 972-9-744-7705 <b>意大利 - 米兰</b> 电话: 39-0331-742611 传真: 39-0331-466781 <b>意大利 - 帕多瓦</b> 电话: 39-049-7625286 <b>荷兰 - 德卢内市</b> 电话: 31-416-690399 传真: 31-416-690340 <b>挪威 - 特隆赫姆</b> 电话: 47-72884388 <b>波兰 - 华沙</b> 电话: 48-22-3325737 <b>罗马尼亚 - 布加勒斯特</b> 电话: 40-21-407-87-50 <b>西班牙 - 马德里</b> 电话: 34-91-708-08-90 传真: 34-91-708-08-91 <b>瑞典 - 哥德堡</b> 电话: 46-31-704-60-40 <b>瑞典 - 斯德哥尔摩</b> 电话: 46-8-5090-4654 <b>英国 - 沃金厄姆</b> 电话: 44-118-921-5800 传真: 44-118-921-5820
<b>亚特兰大</b> 德卢斯, 佐治亚州 电话: 678-957-9614 传真: 678-957-1455 <b>奥斯汀, 德克萨斯州</b> 电话: 512-257-3370 <b>波士顿</b> 韦斯特伯鲁, 马萨诸塞州 电话: 774-760-0087 传真: 774-760-0088 <b>芝加哥</b> 艾塔斯卡, 伊利诺伊州 电话: 630-285-0071 传真: 630-285-0075 <b>达拉斯</b> 阿迪森, 德克萨斯州 电话: 972-818-7423 传真: 972-818-2924 <b>底特律</b> 诺维, 密歇根州 电话: 248-848-4000 <b>休斯顿, 德克萨斯州</b> 电话: 281-894-5983 <b>印第安纳波利斯</b> 诺布尔斯特维尔, 印第安纳州 电话: 317-773-8323 传真: 317-773-5453 电话: 317-536-2380 <b>洛杉矶</b> 米慎维荷, 加利福尼亚州 电话: 949-462-9523 传真: 949-462-9608 电话: 951-273-7800 <b>罗利, 北卡罗来纳州</b> 电话: 919-844-7510 <b>纽约, 纽约州</b> 电话: 631-435-6000 <b>圣何塞, 加利福尼亚州</b> 电话: 408-735-9110 电话: 408-436-4270 <b>加拿大 - 多伦多</b> 电话: 905-695-1980 传真: 905-695-2078			