

南京立超电子科技有限公司

N79A8211 之 ADC 应用笔记

2009 年 2 月 28

中国南京市和燕路 251 号金港大厦 A 幢 2406 室

Room 2406, Tower A, Jingang mansion, 251 Heyan Road, Nanjing 210028, P. R. China

Tel: 0086-25-83306839/83310926 Fax: 0086-25-83737785

[Http://www.dycmcu.com](http://www.dycmcu.com)

版权申明

立超电子科技股份有限公司保留对此文件修改之权利且不另行通知。立超电子科技有限公司所提供之信息相信为正确且可靠之信息，但并不保证本文件中绝无错误。请于向立超电子科技股份有限公司提出订单前，自行确定所使用之相关技术文件及规格为最新之版本。若因贵公司使用本公司之文件或产品，而涉及第三人之专利或著作权等智能财产权之应用及配合时，则应由贵公司负责取得同意及授权，本公司仅单纯贩售产品，上述关于同意及授权，非属本公司应为保证之责任。又未经立超电子科技股份有限公司之正式书面许可，本公司之所有产品不得用于医疗器材，维持生命系统及飞航等相关设备。

修改记录

| 版本 | 日期 | 作者 | 修订内容 | 对应页码 |
|------|-----------|----|------|------|
| V1.0 | 2/28/2009 | 宋娴 | 初始版本 | |

目 录

| | |
|-------------------------|----|
| 1、文件概要 | 1 |
| 2、ADC | 2 |
| 2.1、ADC的概述 | 2 |
| 2.2、ADC的解析度和模拟电源 | 3 |
| 2.3、ADC的技术指标 | 3 |
| 3、ADC的相关寄存器 | 4 |
| 3.1、辅助功能寄存器 | 4 |
| 3.2、ADC转换结果高位寄存器 | 5 |
| 3.3、ADC 控制寄存器 | 5 |
| 3.4、ADC控制寄存器 1 | 6 |
| 3.5、端口 ADC 数字输入禁止 | 7 |
| 4、ADC的使用 | 8 |
| 5、案例及说明 | 12 |
| 5.1、电路图 | 12 |
| 5.2、流程图 | 13 |
| 5.3、程序 | 13 |
| 编后说明 | 16 |
| 参考资料 | 17 |

1、文件概要

该内容主要针对N79A8231的ADC的使用作相应的应用说明。

下文将用图文分别说明各项功能。

注：以下文件中Fosc是指晶振、RC或外部输入的时钟，F_{sys}为系统时钟。

2.2、ADC的解析度和模拟电源

ADC有自己的电源输入脚 (AVDD and AVSS) 和一个参考电压输入脚 (Vref+) 连接到DAC的各自电阻阶梯网络。该阶梯网络由电阻分成1023个相等的阶梯；第一个阶梯是AVSS上的0.5XR，最后一个阶梯是Vref+下的0.5XR。总共有1024XR电阻阶梯。该结构确定DAC的单调变化并误差均匀。

N79A8211系列的AD中 $V_{ref+} = AVDD = VDD$, $AVSS = VSS$ 。见图2-1ADC框图。

输入电压在AVSS和 $[(V_{ref+}) + \frac{1}{2} \text{LSB}]$ 之间，10-位ADC结果会是00 0000 0000 b = 000H；输入电压在 $[(V_{ref+}) - \frac{3}{2} \text{LSB}]$ 和Vref+之间，10-位ADC结果会是11 1111 1111B = 3FFH。AVref+和AVSS可以在AVDD + 0.2V和AVSS - 0.2V之间。Avref+应该是比AVSS的电位高，并且输入电压 (Vin) 应该在AVref+和AVSS之间。

下面是计算结果方程式：

$$\text{Result} = 1024 * \frac{V_{ref}}{V_{in+}} \text{ or } \text{Result} = 1024 * \frac{V_{in}}{VDD}$$

2.3、ADC的技术指标

(1)、分辨率：是指数字量变化一个最小量时模拟信号的变化量，定义为满刻度与 2^n 的比值。通常以数字信号的位数来表示。N79A8211 系列是一个十位分辨率的A/D转换器。

(2)、转换速率：指完成一次 A/D 转换所需时间的倒数。N79A8211 系列是逐次比较型 ADC，是微妙极的。转换的时间为 50 个机器周期。

(3)、量化误差：通常是一个或半个最小数字量的模拟变化量。N79A8211 系列是 $\frac{3}{2}$ LSB。

3、ADC 的相关寄存器

3.1、辅助功能寄存器

助记符: AUXR1

地址: A2h

| | | | | | | | | |
|------|-----|-----|-----|-------|------|-------|------|---|
| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 名称 | EDF | BOD | BOI | LPBOV | SRST | ADCEN | BUZE | - |

| 位 | 名称 | 功能 |
|---|-------|--|
| 7 | EDF | 触发中断标志位: 1: 使能P1.0-1.2 边沿触发功能 (可以配置为上升沿/下降沿触发). 必须由软件清零.. |
| 6 | BOD | 欠压检测: 0: 使能欠压检测. 1: 禁止欠压检测. |
| 5 | BOI | 欠压检测中断: 0: 禁止欠压检测中断功能 1: 允许欠压检测功能中断. |
| 4 | LPBOV | 电源欠压检测控制: 0: 当BOD被允许, 无论在正常模式还是在掉电模式, 欠压检测功能一直处于打开状态。 1: 当BOD被允许, 欠压检测工作在掉电模式下, 可以节省 15/16 欠压电流。 当系统 在掉电模式下, 电源欠压启用内部 RC OSC (600KHz+/- 50%) |
| 3 | SRST | 软件复位: 1: 硬件复位芯片. |
| 2 | ADCEN | 0: 禁止 ADC 电路. 1: 允许ADC 电路. |
| 1 | BUZE | 方波使能位: 1: 方波输出提供 BUZ (P1.0). 0 - 保留. |

3.2、ADC转换结果高位寄存器

助记符: ADCH

地址: E2h

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|--------|--------|--------|--------|--------|--------|--------|--------|
| 名称 | ADC. 9 | ADC. 8 | ADC. 7 | ADC. 6 | ADC. 5 | ADC. 4 | ADC. 3 | ADC. 2 |

| 位 | 名称 | 功能 |
|-----|------------|----------|
| 7-0 | ADC. [9:2] | ADC转换结果. |

3.3、ADC 控制寄存器

助记符: ADCCON

地址: E1h

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|--------|--------|-------|------|------|-------|-------|-------|
| 名称 | ADC. 1 | ADC. 0 | ADCEX | ADCI | ADCS | RCCLK | AADR1 | AADRO |

| 位 | 名称 | 功能 |
|-----|----------|--|
| 7-6 | ADC. 1-0 | 10位ADC结果的低2位. |
| 5 | ADCEX | = '0' , 只有软件把ADCS 位置 '1' 才开始ADC转换。 = '1' , 软件把ADCS 位置 '1' 或外部STADC (1.4) 上的下降沿开始ADC转换。 |
| 4 | ADCI | ADC中断标志位。 当ADC转换结束, 转换结果可以读时, ADCI标志位置 '1'。若ADC中断使能, 就可以进入ADC中断, 进入中断后该标志位清 '0' (也可以软件清 '0' 但不能软件置 '1')。若该标志位为 '1' 时, 就无法开始新的ADC转换。 |
| 3 | ADCS | ADC开始和状态标志位。 把ADCS置 '1' 开始一次ADC转换, 可以由软件或外部的STADC信号置 '1', 当ADC忙时ADCS= '1'。转换结束, ADCI置位后ADCS= '0'。如果ADCS= '1' 或 ADCI = '1' 时无法开始新的ADC转换. 如果软件在设ADCI=0的时候, ADCS=1, 那么在同一通道上的新的A/D转换马上开始. 但推荐先ADCI=0, 再ADCS=1。 |
| 2 | RCCLK | 0: CPU时钟作为ADC时钟。 1: 内部RC时钟作为ADC时钟。 |
| 1 | AADR1 | 见下表. |
| 0 | AADRO | 见下表. |

ADC 状态:

| ADCI | ADCS | ADC状态 |
|------|------|-------------------------|
| 0 | 0 | ADC空闲, 可以开始一个新的转换 |
| 0 | 1 | ADC忙, 转换中 |
| 1 | 0 | 转换结束, 开始新的转换时, 要求ADCI=0 |
| 1 | 1 | 转换结束, 开始新的转换时, 要求ADCI=0 |

AADR2, AADR1, AADR0: 输入通道选择位:

ADC 仿真输入通道选择位。只有当ADCI=0 和ADCS=0 时才可以改变这些位。

| AADR2 | AADR1 | AADR0 | 选择输入通道 |
|-------|-------|-------|-------------|
| 0 | 0 | 0 | AD0 (P0. 3) |
| 0 | 0 | 1 | AD1 (P0. 4) |
| 0 | 1 | 0 | AD2 (P0. 5) |
| 0 | 1 | 1 | AD3 (P0. 6) |
| 1 | 0 | 0 | AD4 (P0. 2) |
| 1 | 0 | 1 | AD5 (P0. 1) |
| 1 | 1 | 0 | AD6 (P0. 0) |
| 1 | 1 | 1 | AD7 (P0. 7) |

3.4、ADC控制寄存器 1

助记符: ADCCON1

地址: E3h

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|----------|----------|---|---|---|-------|---|---|
| 名称 | ADCLK. 1 | ADCLK. 0 | - | - | - | AADR2 | - | - |

| 位 | 名称 | 功能 | | | | | | | | | | | | | | | |
|----------|------------|---|----------|----------|---------|---|---|----------|---|---|----------|---|---|--------------|---|---|----------|
| 7-6 | ADCLK. 1~0 | ADC 时钟选择: 10位 ADC 时钟源满足 200KHz ~ 5MHz. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>ADCLK. 1</th> <th>ADCLK. 0</th> <th>ADC时钟频率</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>ADCCLK/1</td> </tr> <tr> <td>0</td> <td>1</td> <td>ADCCLK/2</td> </tr> <tr> <td>1</td> <td>0</td> <td>ADCCLK/4(默认)</td> </tr> <tr> <td>1</td> <td>1</td> <td>ADCCLK/8</td> </tr> </tbody> </table> <p><i>Note:</i> 必须清ADCCEN (ADCCEN = 0) 当重新配置 ADC 时钟时.</p> | ADCLK. 1 | ADCLK. 0 | ADC时钟频率 | 0 | 0 | ADCCLK/1 | 0 | 1 | ADCCLK/2 | 1 | 0 | ADCCLK/4(默认) | 1 | 1 | ADCCLK/8 |
| ADCLK. 1 | ADCLK. 0 | ADC时钟频率 | | | | | | | | | | | | | | | |
| 0 | 0 | ADCCLK/1 | | | | | | | | | | | | | | | |
| 0 | 1 | ADCCLK/2 | | | | | | | | | | | | | | | |
| 1 | 0 | ADCCLK/4(默认) | | | | | | | | | | | | | | | |
| 1 | 1 | ADCCLK/8 | | | | | | | | | | | | | | | |
| 5-3 | - | 保留. | | | | | | | | | | | | | | | |

| | | |
|-----|-------|----------|
| 2 | AADR2 | ADC输入选择. |
| 1-0 | - | 保留. |

3.5、端口 ADC 数字输入禁止

助记符: PADIDS

地址: F6h

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 名称 | PADIDS. 7 | PADIDS. 6 | PADIDS. 5 | PADIDS. 4 | PADIDS. 3 | PADIDS. 2 | PADIDS. 1 | PADIDS. 0 |

| 位 | 名称 | 功能 |
|---|-----------|---|
| 7 | PADIDS. 7 | P0. 7数字输入禁止位. 0: 默认. 1: 禁止ADC通道7数字输入. |
| 6 | PADIDS. 6 | P0. 6数字输入禁止位. 0: 默认. 1: 禁止ADC通道3数字输入. |
| 5 | PADIDS. 5 | P0. 5数字输入禁止位. 0: 默认. 1: 禁止ADC通道2数字输入. |
| 4 | PADIDS. 4 | P0. 4数字输入禁止位. 0: 默认. 1: 禁止ADC通道1数字输入. |
| 3 | PADIDS. 3 | P0. 3数字输入禁止位. 0: 默认. 1: 禁止ADC通道0数字输入. |
| 2 | PADIDS. 2 | P0. 2数字输入禁止位. 0: 默认. 1: 禁止ADC通道4数字输入. |
| 1 | PADIDS. 1 | P0. 1数字输入禁止位. 0: 默认. 1: 禁止ADC通道5数字输入. |
| 0 | PADIDS. 0 | P0. 0数字输入禁止位. 0: 默认. 1: 禁止ADC通道6数字输入. |

4、ADC 的使用

ADC的使用可以按照以下步骤:

(1)、打开(关闭)ADC电路:

AUXR1.2 (ADCEN) =1打开ADC 的电路, AUXR1.2 (ADCEN) =0关闭ADC 的电路。

```
如:      void EnableADC()
        {
            AUXR1|=0x04; //打开ADC 的电路, ADC 的电路开始工作
        }

        void DisableADC()
        {
            AUXR1&=0xFB; // 关闭ADC 的电路, ADC 的电路停止工作
        }
```

(2)、选择通道(如通道6)

参照 AADR2, AADR1, AADR0: 输入通道选择位:

ADC 仿真输入通道选择位。只有当ADCI=0 和ADCS=0 时才可以改变这些位。

| AADR2 | AAADR1 | AAADR0 | 选择输入通道 |
|-------|--------|--------|-------------|
| 0 | 0 | 0 | AD0 (P0. 3) |
| 0 | 0 | 1 | AD1 (P0. 4) |
| 0 | 1 | 0 | AD2 (P0. 5) |
| 0 | 1 | 1 | AD3 (P0. 6) |
| 1 | 0 | 0 | AD4 (P0. 2) |
| 1 | 0 | 1 | AD5 (P0. 1) |
| 1 | 1 | 0 | AD6 (P0. 0) |
| 1 | 1 | 1 | AD7 (P0. 7) |

助记符: ADCCON

地址: E1h

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|--------|--------|-------|------|------|-------|--------|--------|
| 名称 | ADC. 1 | ADC. 0 | ADCEX | ADCI | ADCS | RCCLK | AAADR1 | AAADR0 |

助记符: ADCCON1

地址: E3h

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|----------|----------|---|---|---|--------|---|---|
| 名称 | ADCLK. 1 | ADCLK. 0 | - | - | - | AAADR2 | - | - |

/**/

** 函数名称: AD_sel

** 函数描述: 初始化AD

** 输入参数: AD通道号AD_Channel

** 输出参数: 无

*****/

```
void AD_Sel(uint8 AD_Channel)
{
    switch(AD_Channel)
    {
        case 0:
            P0|=0x04;
            PADIDS |=0x08;
            ADCCON1&=0xfb;
            ADCCON &=0xfd;
            ADCCON &=0xfe;           //ADDR2=0;ADDR1=0;ADDR0=0;选择AD0;
            break;
        case 1:
            P0|=0x10;
            PADIDS |=0x10;
            ADCCON1&=0xfb;
            ADCCON &=0xfd;
            ADCCON |=0x01;         //ADDR2=0;ADDR1=0;ADDR0=1;选择AD1;
            break;
        case 2:
            P0|=0x20;
            PADIDS |=0x20;
            ADCCON1&=0xfb;
            ADCCON |=0x02;
            ADCCON &=0xfe;         //ADDR2=0;ADDR1=1;ADDR0=0;选择AD2;
            break;
        case 3:
            P0|=0x40;
            PADIDS |=0x40;
            ADCCON1&=0xfb;
            ADCCON |=0x02;
            ADCCON |=0x01;         //ADDR2=0;ADDR1=1;ADDR0=1;选择AD3;
```

```
        break;
    case 4:
        P0|=0x04;
        PADIDS |=0x04;
        ADCCON1|=0x04;
        ADCCON &=0xfd;
        ADCCON &=0xfe;           //ADDR2=1;ADDR1=0;ADDR0=0;选择AD4;
        break;
    case 5:
        P0|=0x02;
        PADIDS |=0x02;
        ADCCON1|=0x04;
        ADCCON &=0xfd;
        ADCCON |=0x01;           //ADDR2=1;ADDR1=0;ADDR0=1;选择AD5;
        break;
    case 6:
        P0|=0x01;
        PADIDS |=0x01;
        ADCCON1|=0x04;
        ADCCON |=0x02;
        ADCCON &=0xfe;           //ADDR1=1;ADDR0=1;ADDR2=0;选择AD6;
        break;
    case 7:
        P0|=0x80;
        PADIDS |=0x80;
        ADCCON1|=0x04;
        ADCCON |=0x02;
        ADCCON |=0x01;           //ADDR1=1;ADDR0=1;ADDR2=1;选择AD7;
        break;
    }
}
```

(3)、选择软件启动或者硬件启动

设置：1)、若ADCCON.5(ADCEX)=0；把ADCCON.3(ADCS)置'1'，就开始一次新的A-D转换。

如：

```
void StartADC()
{
    ADCCON|=0x08;
}
```

- 2)、若ADCCON.5(ADCEX)=1, 软件把ADCS 位置 ‘1’ 或STADC(P1.4)上的上升沿, 就开始一次新的 A-D转换。

助记符: ADCCON

地址: E1h

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|-------|-------|-------|------|------|-------|-------|-------|
| 名称 | ADC.1 | ADC.0 | ADCEX | ADCI | ADCS | RCCLK | AADR1 | AADRO |

- (4)、转换结束, 可以判断ADCI是否为“1”, 或者采取中断方式

查询方式: 设置: `while((ADCCON&0x10) !=0x10);`

中断方式: 设置:

- 1)、打开ADC中断。

```
void EnableADC_INT()
{
    EADC=1;
    EA=1;
}
```

- 2)、编写ADC中断服务程序。

```
void ADC_ISR() interrupt 11
{
    .....//
}
```

- (5)、转换结果

控制位 ADCCON.4 (ADCI)置 ‘0’ 标志10-位转换的结束。

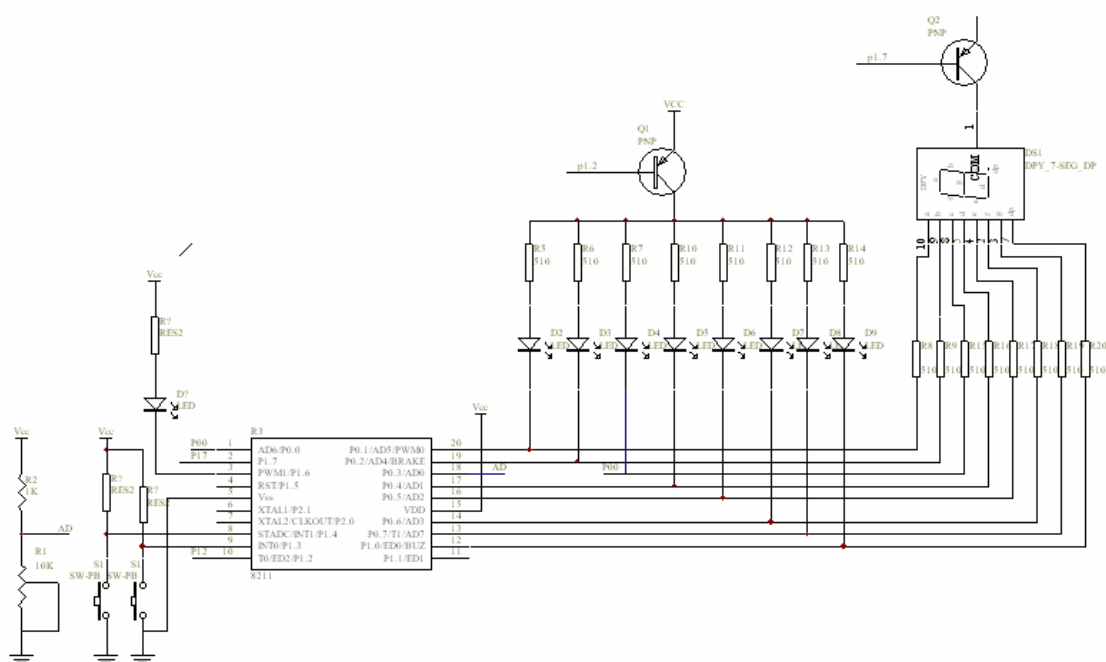
转换结果的高8位存放在特殊功能寄存器ADCH中, 剩下的两位存放在ADCCON.7 (ADC.1) ADCCON.6 (ADC.0) 中。

5、案例及说明

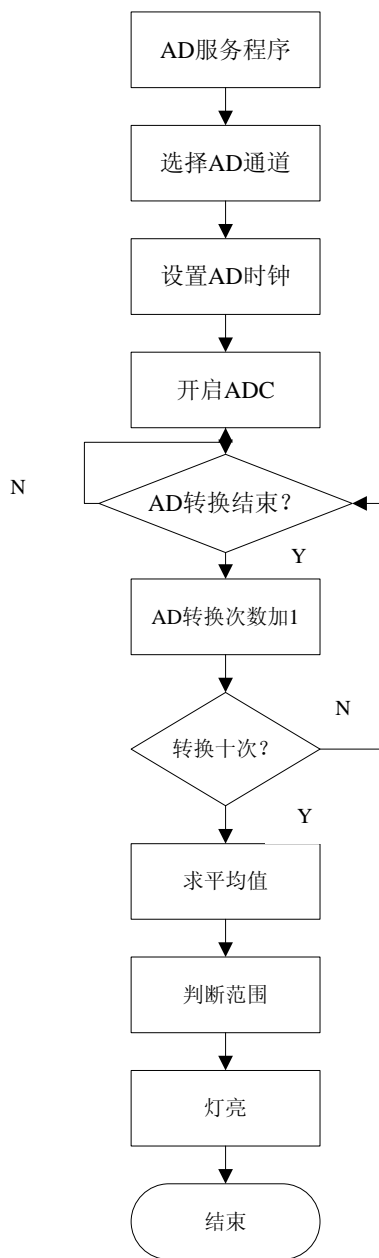
下面是以N79A8211 为例编写程序：（下文仅列出AD部分程序）

- 功能：
- （1）、按键一：复位。按键二：按一次数码显示加一。
 - （2）、数码管显示“1”：八个LED依次循环点亮。
 - （3）、数码管显示“2”：八个LED显示AD转换数据。
 - （4）、数码管显示“3”：一个LED从亮渐暗直到灭，再渐亮直到最亮。
 - （5）、数码管显示“4”：写如NVM值。

5.1、电路图



5.2、流程图



5.3、程序

```

    /*****
    *****/
    ** 函数名称: Start_AD
    ** 函数描述:
    ** 输入参数: 无
    ** 输出参数: 无
    *****/
    *****/
    
```

```
void Start_AD()
{
    AUXR1|=0x04;           //打开 ADC 的电路,ADC 的电路开始工作
    ADCCON1&=0x3f;        //设置 ADC 时钟, 1/1

    AD_Sel(1);            //选择 AD1;
    ADCCON=Bin(00001101); //采用内部 RC; ADCEX=0;ADCS=1;开 ADC
}

/*****
*****/
** 函数名称: AD_Sub
** 函数描述:
** 输入参数: 无
** 输出参数: 无
*****/
*****/
void AD_Sub()
{
    uint16 tmpval1,tmpval2;
    AD_cnt=10;
    AD_data_sum=0;
    for(;AD_cnt>0;AD_cnt--)
    {

        ADCCON&=0xef;
        ADCCON|=0x08;
        while((ADCCON&0x10)!=0x10);

        tmpval1=ADCH;
        tmpval1<<=2;
        tmpval1|=(ADCCON&0xC0)>>6;
        tmpval2=tmpval1;
        AD_data_sum+=tmpval2;
    }
    AD_data=AD_data_sum/10;
    if(AD_data==0)
        led_val=0;
    else if(AD_data>0&AD_data<=128)
        led_val=9;
    else if(AD_data>128&AD_data<=256)
        led_val=10;
    else if(AD_data>256&AD_data<=384)
```

```
        led_val=11;
    else if(AD_data>384&AD_data<=512)
        led_val=12;
    else if(AD_data>512&AD_data<=640)
        led_val=13;
    else if(AD_data>640&AD_data<=768)
        led_val=14;
    else if(AD_data>768&AD_data<=896)
        led_val=15;
    else if(AD_data>896&AD_data<=1024)
        led_val=16;
    else
        led_val=0;

}
/*****
** 函数名称: main
** 函数描述: 程序主函数
** 输入参数: 无
** 输出参数: 无
*****/
void main(void)
{
    Init_IO();
    Init_RAM();
    Start_AD();
    AD_Sub();
}
```

编后说明

NA79A8211 系列单片机ADC使用指南,是立超电子为帮助用户迅速掌握NA79A8211单片机而编写的,限于水平,难免有错误和不妥之处,恳请广大读者批评指正。请将您的建议和批评发至E-Mail:xian.song@dycmcu.com,我们深表感谢!

继本指南之后,我们已经或即将编写以下使用指南、应用设计和范例程序等,欢迎及时访问我们的网页:
www.dycmcu.com,或通过电话、E-Mail 索取更多的应用指南及内部资料等。

使用指南:

NA79A8211 系列单片机A/D 转换器使用指南

NA79A8211 系列单片机PWM 使用指南

NA79A8211 系列单片机内置WDT 使用指南

NA79A8211 系列单片机中断使用指南

NA79A8211 系列单片机的低功耗设计方法

.....

应用设计:

SH69PXX 系列单片机与E2PROM 接口及程序设计

SH69PXX 系列单片机I2C 总线模拟程序包

.....

其它:

NA79A8211 系列单片机特殊功能寄存器定义库文件

NA79A8211 系列单片机特殊功能寄存器定义库文件

NA79A8211 系列单片机特殊功能寄存器定义库文件

本应用指南欢迎各相关电子网站转载,为了尊重我们付出的劳动,请您注明出处来自站点: www.dycmcu.com

立超电子 技术支持部

2009 年 2 月 28 日 星期三

编写单位

南京立超电子科技有限公司

参考资料

- 1、N79A8211 规格书 单片机规格书。