Arduino 寄存器（ADC）

原创 WxxMaster 2021-02-22 19:00:38 137 收藏 版权
分类专栏： Arduino寄存器编程 文章标签： arduino

> 说明：Arduino Nano（ATMEGA328P）；10位ADC（0~1023）
> 附：Arduino Nano 内置温度测量（ADC8）

文章目录

1.ADC转换时间

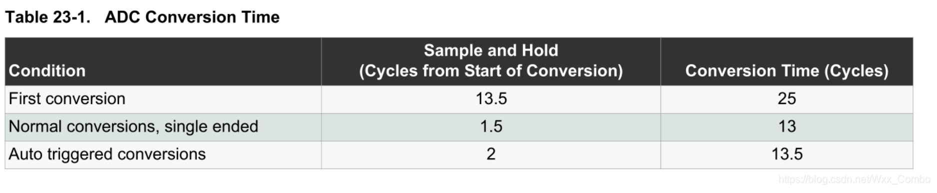
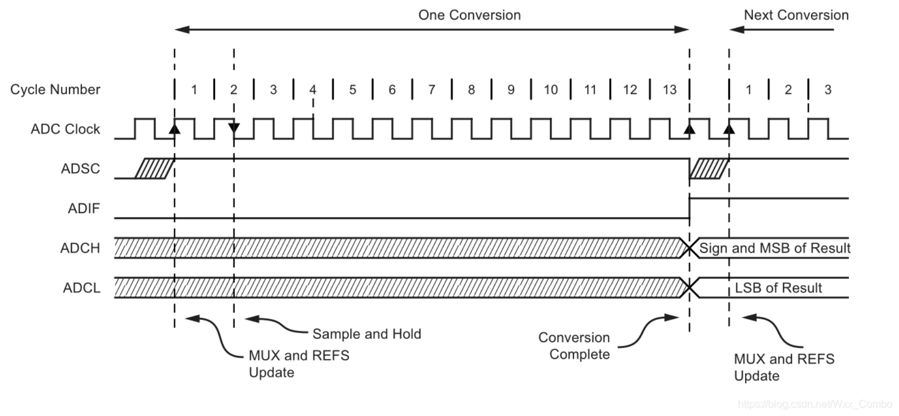


Table 23-1. ADC Conversion Time

Condition	Sample and Hold (Cycles from Start of Conversion)	Conversion Time (Cycles)
First conversion	13.5	25
Normal conversions, single ended	1.5	13
Auto triggered conversions	2	13.5

2.转换过程（举例）

Figure 23-5. ADC Timing Diagram, Single Conversion

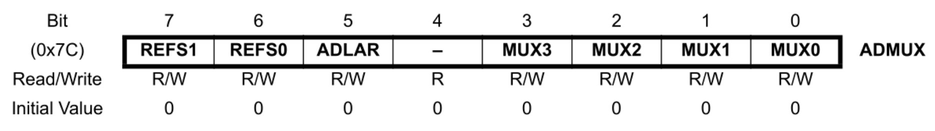


1.开始转换（ADSC置1）
2.MUX和REFS更新

3.样本保存（保存期间不会改变）
4.ADC转换
5.更新MUX和REFS

> 另外还有两种，相比差距不大

3.多路复用器选择寄存器（Multiplexer Selection Register）

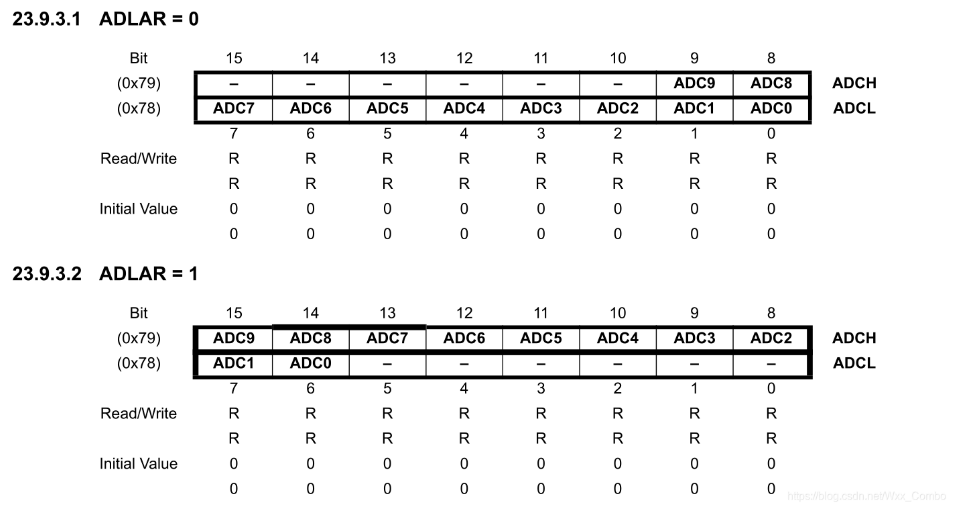


Bit	7	6	5	4	3	2	1	0	
(0x7C)	REFS1	REFS0	ADLAR	–	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

1.前两位（REFS1和REFS0）用来选着参考电压

REFS1	REFS0	基准电压
0	0	AREF
0	1	AVCC
1	0	保留
1	1	内部1.1V基准电压

2.第3位用来选择数据保存方式



23.9.3.1 ADLAR = 0

Bit	15	14	13	12	11	10	9	8	
(0x79)	–	–	–	–	–	–	ADC9	ADC8	ADCH
(0x78)	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

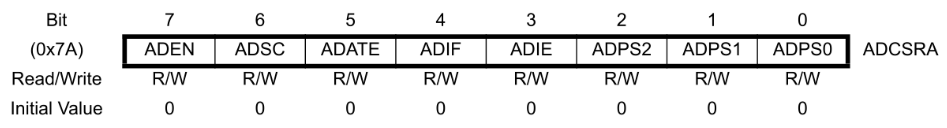
23.9.3.2 ADLAR = 1

Bit	15	14	13	12	11	10	9	8	
(0x79)	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADCH
(0x78)	ADC1	ADC0	–	–	–	–	–	–	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

3.后四位选择ADC通道

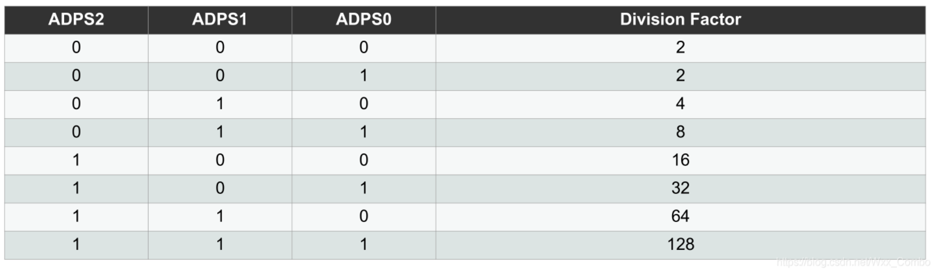
MUX3、2、1、0	ADC通道
0000	ADC0
0001	ADC1
0010	ADC2
0011	ADC3
0100	ADC4
0101	ADC5
0110	ADC6
0111	ADC7
1000	ADC8(内部温度测量)

4.控制状态寄存器 A（Control and Status Register A）



Bit	7	6	5	4	3	2	1	0	
(0x7A)	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

1.第一位进行ADC使能
2.第二位表示开始转换
3.第三位进行ADC触发使能
4.第四位被置1，表示转换完成
5.第四位被置1，表示触发转换完成
6.后三位选择预分频系数



ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

5.相关其他寄存器

- 控制和状态寄存器 B（用作选择触发模式和比较器）
- 数字输入禁用寄存器 0

> 暂时没有用到，后期用到再补充

> 以上寄存器讲完，开始代码：

6.程序示例

示例1

开启ADC2，使用AVCC基准源，8倍预分频

```
void setup(){
  Serial.begin(9600);  // 设置串口通讯

  // 设置使用AVCC基准电压源；设置测量ADC2；
  ADMUX = _BV(REFS0) | _BV(MUX1);
  // ADC使能；设置ADC预分频器为8；
  ADCSRA = _BV(ADEN) | _BV(ADPS1) | _BV(ADPS0);
  }

void loop(){
  ADCSRA |= _BV(ADSC);  // 开始转换
  Serial.println(ADC);
  delay(100);  // 延时100MS
  }
```

示例2

开启ADC7，使用内置标准电压源，64分频

```
void setup(){
  Serial.begin(9600);  // 设置串口通讯

  // 设置使用AVCC基准电压源；设置测量ADC2；
  ADMUX = _BV(REFS1) | _BV(REFS0) | _BV(MUX2) | _BV(MUX1) | _BV(MUX0);
  // ADC使能；设置ADC预分频器为8；
  ADCSRA = _BV(ADEN) | _BV(ADPS2) | _BV(ADPS1);
  }

void loop(){
  ADCSRA |= _BV(ADSC);  // 开始转换
  Serial.println(ADC);
  delay(100);  // 延时100MS
  }
```

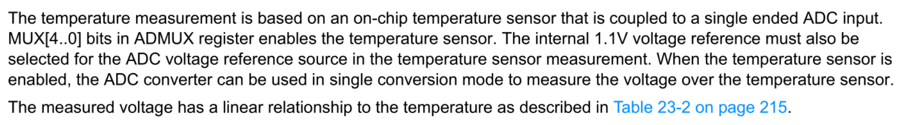
7.Arduino Nano 内置温度测量（ADC8）

先上代码

```
float T;

void setup(){
  Serial.begin(9600);  // 设置串口通讯

  // 设置使用内置1.1V基准电压源；设置测量ADC8；
  ADMUX = _BV(REFS1) | _BV(REFS0) | _BV(MUX3);
  // ADC使能；设置ADC预分频器为16；
  ADCSRA = _BV(ADEN) | _BV(ADPS2);
  }

void loop(){
  ADCSRA |= _BV(ADSC);  // 开始转换

  T = (ADC-320) / 1.28;  // 转化为摄氏度

  Serial.print(T); // 输出ADC值
  Serial.println(" 摄氏度");
  delay(100);  // 延时100MS
  }
```

ATMEGA328P温度测量说明

- 该芯片内部温度传感器连接ADC8（未输出引脚）
- 测量温度时需使用内置1.1V基准电压源


The temperature measurement is based on an on-chip temperature sensor that is coupled to a single ended ADC input. MUX[4..0] bits in ADMUX register enables the temperature sensor. The internal 1.1V voltage reference must also be selected for the ADC voltage reference source in the temperature sensor measurement. When the temperature sensor is enabled, the ADC converter can be used in single conversion mode to measure the voltage over the temperature sensor.

The measured voltage has a linear relationship to the temperature as described in Table 23-2 on page 215.

- ADC转换温度

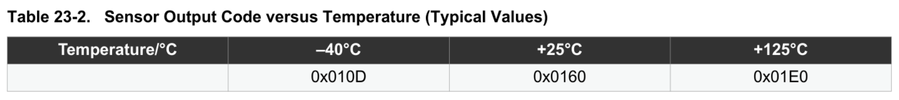
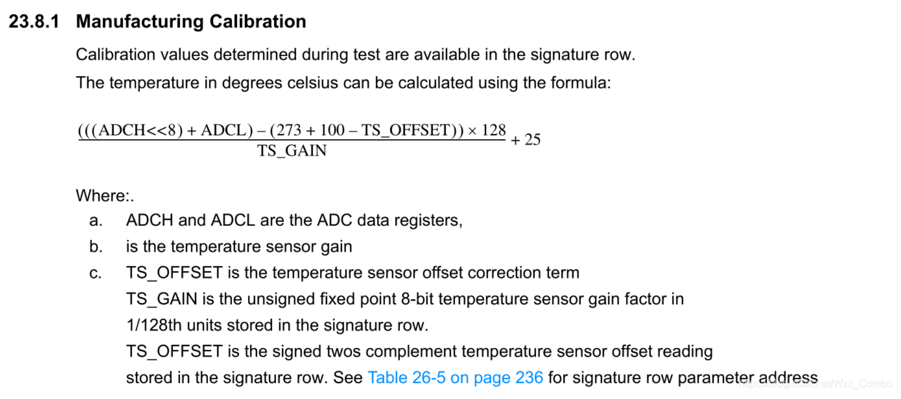


Table 23-2. Sensor Output Code versus Temperature (Typical Values)

Temperature/°C	–40°C	+25°C	+125°C
	0x010D	0x0160	0x01E0

具体转化都懂，就不写了

- 另外还提供了校准，但是没有理解，，，有大佬懂的可以解释一下



23.8.1 Manufacturing Calibration

Calibration values determined during test are available in the signature row.

The temperature in degrees celsius can be calculated using the formula:

$$\frac{(((ADCH<<8)+ADCL)-(273+100-TS\_OFFSET))\times 128}{TS\_GAIN}+25$$

Where:.

a. ADCH and ADCL are the ADC data registers,

b. is the temperature sensor gain

c. TS_OFFSET is the temperature sensor offset correction term

TS_GAIN is the unsigned fixed point 8-bit temperature sensor gain factor in 1/128th units stored in the signature row.

TS_OFFSET is the signed twos complement temperature sensor offset reading stored in the signature row. See Table 26-5 on page 236 for signature row parameter address

附通过寄存器地址读取寄存器代码：

```
volatile uint8_t *PA=(volatile uint8_t *)0x103;

void setup(){
  Serial.begin(9600);
}

void loop(){
  Serial.println(*PA);
  delay(1000);
}
```

写在最后：

1.此内容还不全，更详细的内容（ADC中断等）需查看ATMEGA328P数据手册

2.此部分内容若存在理解错误，还请指出，本文章代码均已验证

显示推荐内容

点赞 评论 分享 收藏 打赏 举报 关注 一键三连

优质评论可以帮助作者获得更高权重 评论