

# AVR\_SCLib manual

Schummacher S.J.F

2017 年 7 月 28 日

# Contents

<b>1</b>	<b>综述</b>	<b>2</b>
<b>2</b>	<b>函数库使用</b>	<b>3</b>
2.1	ADC . . . . .	3
2.1.1	ADC_Date . . . . .	3
2.1.2	ADC_Average_Get . . . . .	3
2.1.3	eg. . . . .	3
2.2	Devich . . . . .	3
2.3	EEPROM . . . . .	4
2.3.1	EEPROM_write . . . . .	4
2.3.2	EEPROM_write . . . . .	4
2.3.3	eg. . . . .	4

# Chapter 1

## 综述

作者本未想做一套轮子，但使用中深感不便，每次实现一些重复性强的功能都要重新写一遍，实在是不胜其烦就打算写一些自己用且简单粗暴的函数，但写着写着函数也就慢慢多了起来，渐渐的整理出来一些库，现在把他开源出来，希望用 AVR 的人可以少造一些轮子。

ADC 为 ADC 转换库。Cmp 是复数库。Device 类似于 main.h。EEPROM 是内部存储器库。I2C 是 I2C/TWI 库 LCD 是 1602 液晶显示器的基础库。MyLCD 是基于 LCD 制作带数位分离加显示库。Sort 是排序库。USART 是串口通信库。

至于为什么不用默认的 README.md 和 Wiki 嘛，我只能坦白的说我不会 Markdown 和 Wiki 只会  $\text{\LaTeX}$

PS: 此函数库均基于 ATmega16A

## Chapter 2

# 函数库使用

### 2.1 ADC

#### 2.1.1 ADC\_Date

```
uint8_t ADC_Date(uint8_t i);
```

其中参量 i 是 PA 组的 IO 口数, 返回左对齐的 10bit 采样

#### 2.1.2 ADC\_Average\_Get

```
uint16_t ADC_Average_Get(uint8_t pin);
```

进行 8 次 AD 采样然后去平均值, 返回无符号 16 位采样信号, 参量 i 是 PA 所对应的端口号

#### 2.1.3 eg.

我们要采 PA1 的样, 并将 ADC 采样值付给变量 ADC\_temp:

```
ADC_temp_1 = ADC_Date(1);  
ADC_temp_2 = ADC_Average_Get(1);
```

### 2.2 Devich

Device.h 里面的文件有点类似于很多人写的 main.h 但是我在这里还增加了一些单片机常用的一些宏定义

```

#define LSL(x, y) x=x<<y
#define LSR(x, y) x=x>>y

```

把 x 左/右移 y 位并将结果付给 x

```

#define MAX(a, b) a>b?a:b
#define MIN(a, b) a<b?a:b

```

返回 a, b 中最大/小的数

```

#define SEI asm("sei");
#define CLI asm("cli");

```

全局中断使能/清除

## 2.3 EEPROM

### 2.3.1 EEPROM\_write

```
void EEPROM_write(unsigned int uiAddress, unsigned char ucData);
```

uiAddress 是数据需要储存的地址, ucData 是数据

### 2.3.2 EEPROM\_read

```
unsigned char EEPROM_read(unsigned int uiAddress);
```

uiAddress 是数据需要读出的地址, 返回值是读出的数据

### 2.3.3 eg.

我们将变量 i 储存到地址 0x00, 然后将地址为 0x01 的数据读出到 j

```
EEPROM_write(0x00, i);
j = EEPROM_read(0x01);
```