

# AVR\_SCLib manual

Schummacher S.J.F

2017 年 5 月 27 日

# Contents

<b>1</b>	<b>综述</b>	<b>3</b>
<b>2</b>	<b>ADC</b>	<b>3</b>
2.1	ADC_Date . . . . .	3
2.2	eg. . . . .	3
<b>3</b>	<b>EEPROM</b>	<b>3</b>
3.1	EEPROM_write . . . . .	3
3.2	EEPROM_write . . . . .	3
3.3	eg. . . . .	3

## 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。

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 L<sup>A</sup>T<sub>E</sub>X

## 2 ADC

### 2.1 ADC\_Date

```
unsigned int ADC_Date(unsigned char i);
```

其中参量 i 是 PA 组的 IO 口数, 返回左对齐的 10bit 采样

### 2.2 eg.

我们要采 PA1 的样，并将 ADC 采样值付给变量 ADC\_temp:

```
ADC_temp = ADC_Date(1);
```

## 3 EEPROM

### 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

### 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

### 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);
```