六、AD转换步骤

1.使能IO和ADC时钟

RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_ADC1 | RCC\_APB2Periph\_GPIOC, ENABLE);

2、 配置GPIO引脚为AIN

GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_1;

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_AIN;

GPIO\_Init(GPIOA, &GPIO\_InitStructure);

3、配置ADC结构体

ADC\_InitTypeDef ADC\_InitStructure;

ADC\_InitStructure.ADC\_Mode = ADC\_Mode\_Independent;

ADC\_InitStructure.ADC\_ScanConvMode = ENABLE;

ADC\_InitStructure.ADC\_ContinuousConvMode = ENABLE;

ADC\_InitStructure.ADC\_ExternalTrigConv = ADC\_ExternalTrigConv\_None;

ADC\_InitStructure.ADC\_DataAlign = ADC\_DataAlign\_Right;

ADC\_InitStructure.ADC\_NbrOfChannel = 1;

ADC\_Init(**ADC1**, &ADC\_InitStructure);

4、时钟分频 RCC\_ADCCLKConfig(RCC\_PCLK2\_Div8);

可以提前

只能是1、2、4、8

5、ADC通道选择和使能

ADC\_RegularChannelConfig(ADC1, ADC\_Channel\_11, 1, ADC\_SampleTime\_55Cycles5);

ADC\_Cmd(**ADC1**, ENABLE);

6、校准 (不变)

ADC\_ResetCalibration(ADC1);

while(ADC\_GetReset**Cal**ibrationStatus(ADC1));

ADC\_StartCalibration(ADC1);

while(ADC\_GetCalibrationStatus(ADC1));

7、真正开始

ADC\_SoftwareStartConvCmd(ADC1, ENABLE);

value =ADC1->DR;

ADC\_IT\_EOC ADC\_FLAG\_EOC

ADC\_GetFlagStatus(ADC1, , ENABLE )

8. 如果使用中断

ADC\_ITConfig(ADC1, ADC\_IT\_EOC, ENABLE);

void ADC1\_2\_IRQHandler(void)

{ADC\_Temp= ADC\_GetConversionValue(ADC1);

ADC\_Temp= ADC1->DR;

Vsense=(ADC\_Temp\*3.3)/4096;

ADC\_ClearITPendingBit(ADC1, ADC\_**IT**\_EOC)