

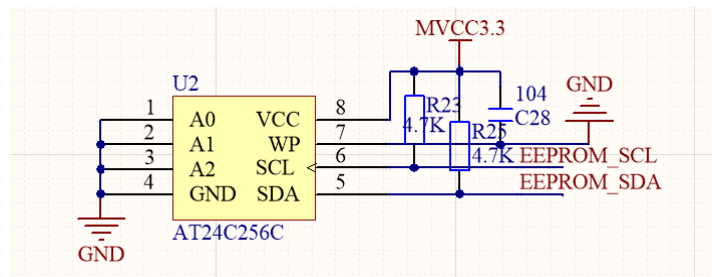
# AT24CXXX系列EEPROM驱动

WTR——欧阳俊源 2020/01/31

原理：使用I2C通信接口，向芯片内部写入和读取信息。

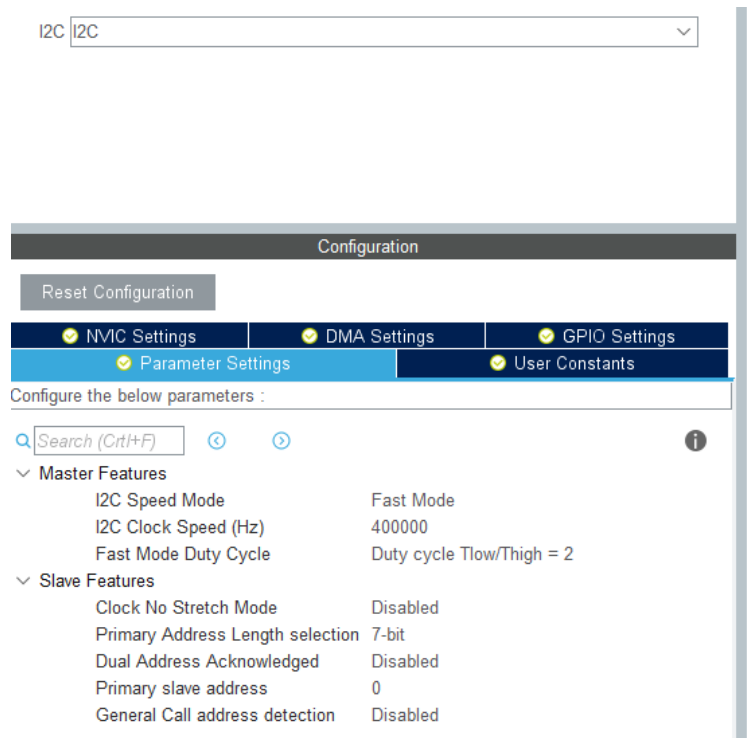
## 硬件原理图

- 2个I2C上拉4.7K电阻
- WP为写保护引脚
- A0A1A2为I2C硬件地址



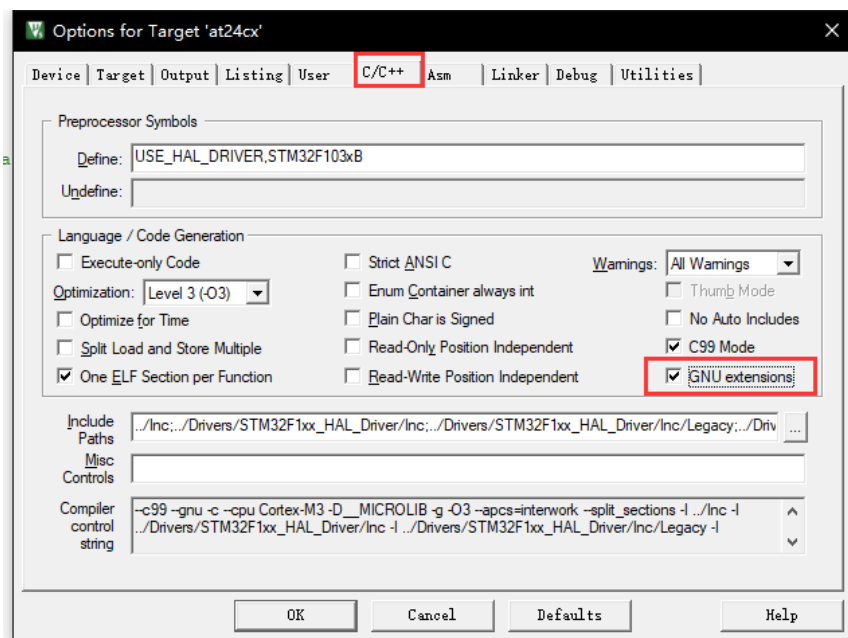
## 软件配置

### CubeMx



### Code

勾选keil中的这个选项



在at24c256.h中配置

- STORAGE\_I2C: 配置使用hal的i2c句柄
- MAX\_NUM\_STORAGE\_ONE: 配置最大存储的变量数。即调用AT24Cx\_REGISTER()的次数
- MEMORY\_SIZE: 该值为at24c256的bytes数。不同的芯片可以修改。

```

1 #ifndef AT24C256_H_OUYJY
2
3 #include "i2c.h"
4 #include <stdlib.h>
5 #include "string.h"
6
7
8 #define STORAGE_I2C hi2c1
9 #define MAX_NUM_STORAGE_ONE 20
10 #define MEMORY_SIZE (256*1024/8)
11
12 typedef struct
13 {
14     __IO void *data;
15     int size;
16 }

```

使用方法:

- AT24Cx\_REGISTER(变量名) 进行注册存储变量。
- AT24Cx\_STORAGE(变量名) 存储改变量在此语句的当前值。
- AT24Cx\_RECOVER(变量名) 恢复出该变量名存储的值。

## 例子

```

/* USER CODE BEGIN 0 */
typedef struct
{
    int a;
    float b;
}my;
my strmy={.a=2,.b=5};

```

```

AT24Cx_REGISTER(strmy);

AT24Cx_STORAGE(strmy);
if(strmy.a==2)
{
    strmy.a=0,strmy.b=0;
}
AT24Cx_RECOVER(strmy);
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{

```

恢复前把strmy的a和b都置0

wnload: Bank 0 @ 0x08000000: Skipped. Contents already match  
after reset via DEMCR.VC CORERESET.

Name	Value	Type
strmy	0x20000000 &strmy	struct <untagged>
a	0x00000000	int
b	0	float

恢复完毕

wnload: Bank 0 @ 0x08000000: Skipped. Contents already match  
after reset via DEMCR.VC CORERESET.  
ice via AIRCR.SYSRESETEQ.

Name	Value	Type
strmy	0x20000000 &strmy	struct <untagged>
a	0x00000002	int
b	5	float

1