My Study for 600 Product Department

# Overview

Product Manager

George

Mentor

Chippy

Team Member

Rick, Hugo, Join in recent April

Azure

HR

May, Judith, Krist

Recommend by Chippy

Miles, Darren, George

Experts

# Products

## 5.8GHz 多路径识别产品系列



5.8GHz CPC复合卡

5.8GHz 自由流天线

MTC 车道读卡器

CPC 卡自动发行机

## ETC车道天线系列



ETC车道二代天线

ETC车道U+型天线

ETC车道并驰天线

ETC车道并驰U+天线

ETC车道一代天线

ETC停车场天线

流量信息检测天线

智能手持机

### ETC车载充值系列



蓝牙互联网OBU

普通OBU

双模OBU

行车记录仪OBU

后视镜OBU

蓝牙充值宝

## OBU Reference

<https://wenku.baidu.com/view/6040f1e7ccbff121dc3683ac.html>

<https://wenku.baidu.com/view/cae4ae30783e0912a3162a24.html>

<https://wenku.baidu.com/view/d8007603f7ec4afe05a1df11.html?sxts=1536572308244>

<https://bbs.pcauto.com.cn/topic-4985741.html>



自由流

<http://www.tranbbs.com/application/Intercity/application_148905.shtml>

## OBU summary

高速公路上设立在出入口的收费站有两种。

MTC人工收费站

ETC收费站

1 MTC人工收费站

业务示例：

车主开车进入高速，通过MTC人工收费通道，向入口车站人员领取CPC卡（CPC卡支持5.8G DSRC 和 13.56MHz NFC），该卡记录了现在的入口为起点。车辆开出高速，在MTC人工收费通道出口，交还CPC卡，出口车站人员读取CPC路径信息(方式：13.56MHz NFC)，路径为入口为起点，该出口为终点，据此向车主人工收费。

2 ETC收费站

业务示例：

车辆装有OBU电子标签且插卡，贴在车前窗上，OBU上的液晶屏显示有足够余额，进入高速，通过ETC收费通道，OBU与RSU路侧单元通信（方式：5.8GHz DSRC），记录当前点为起点；待出口时通过ETC通道，OBU与出口的RSU路测单元通信，记录路径，该出口为终点，对OBU上的余额卡扣费。

另外，OBU电子标签也可在MTC人工收费通道使用，方式为通过5.8GHz DSRC对OBU标记路径，据此再人工收费。

3 多车道自由流路径识别系统

现状：高速路上采用自由流系统主要用于路径识别，用于在高速路中设立标记点，以便准确记录车辆的具体在某一条高速路的路径点；路径信息写入CPC卡(方式 5.8G DSRC)，电子标签OBU（方式 5.8G DSRC）；在出口收费站（MTC或ETC收费站）进行收费。

自由流路径识别与OBU的交互

ETC收费站与OBU的交互

二者在物理层技术，交互过程是类似的。只是自由流只标记路径，不扣费。而ETC收费站标记路径，并进行收费。

高速路自由流收费技术，在自由流路径识别的基础上进一步发展收费业务，

这是目前开展中的项目阶段。

### 参考建议

ETC方案设计可参考：

ETC标准化专用车道设计方案（PC-RSU接口）V3.1

自由流路径识别可参考：

收费公路多义性路径识别技术要求（出版社V0.1）(1).pdf

# RSU数据域定义

## -F0

C:\Users\Administrator\Documents\学习资料

收费公路联网电子不停车收费技术要求.doc

## -C0

参考

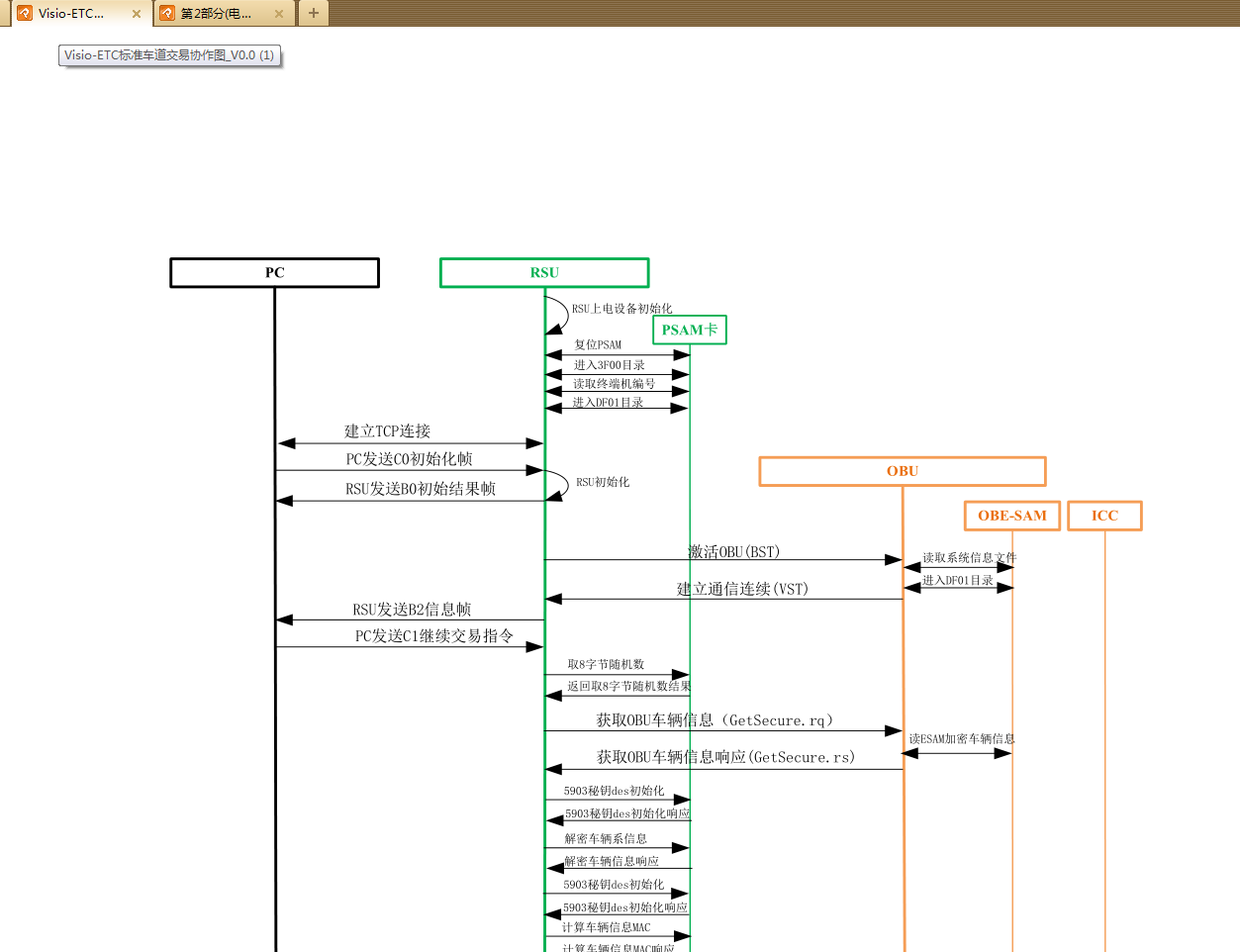
C:\Users\Administrator\Documents\部中心ETC标准化（2017）\软件标准化\接口协议

ETC标准化专用车道设计方案（PC-RSU接口）V2.0.doc

标准化ETC专用车道设计方案v2.3水印.pdf

ETC标准化专用车道设计方案（PC-RSU接口）V3.1.docx

# Business Diagram



# Git learn

Reference:

<https://www.cnblogs.com/zhangxiaoyong/p/6000084.html>

When SSH:

login github.com

ztmian BASE@gi

Account->Settings->SSH

Download

<https://git-scm.com/download/win>

Git init



git config --global user.name “ztmian”

git config --global user.email [“ztmian@qq.com”](mailto:“ztmian@qq.com”)

ssh-keygen

## Second use

cd C:\Users\Administrator\Documents\0\_Git\_Project

$ ln -s /c/Users/Administrator/Documents/0\_Git\_Project /c/0\_Git\_Project

Administrator@USER MINGW32 ~/Documents/0\_Git\_Project/prj\_Excel\_VBA\_LogSelect (ma ster)

$ git init

Initialized empty Git repository in C:/Users/Administrator/Documents/0\_Git\_Proje ct/prj\_Excel\_VBA\_LogSelect/.git/

Administrator@USER MINGW32 ~/Documents/0\_Git\_Project/prj\_Excel\_VBA\_LogSelect (ma ster)

$ ls

LogSelect\_VBA.xlsm LogSelect\_VBA\_code.vb Readme.txt

Administrator@USER MINGW32 ~/Documents/0\_Git\_Project/prj\_Excel\_VBA\_LogSelect (ma ster)

$ git add LogSelect\_VBA.xlsm LogSelect\_VBA\_code.vb Readme.txt

Administrator@USER MINGW32 ~/Documents/0\_Git\_Project/prj\_Excel\_VBA\_LogSelect (ma ster)

$ git commit -m “First commit”

[master (root-commit) a3771d5] First Commit Excel\_VBA\_Code

3 files changed, 59 insertions(+)

create mode 100644 LogSelect\_VBA.xlsm

create mode 100644 LogSelect\_VBA\_code.vb

create mode 100644 Readme.txt

$ git status

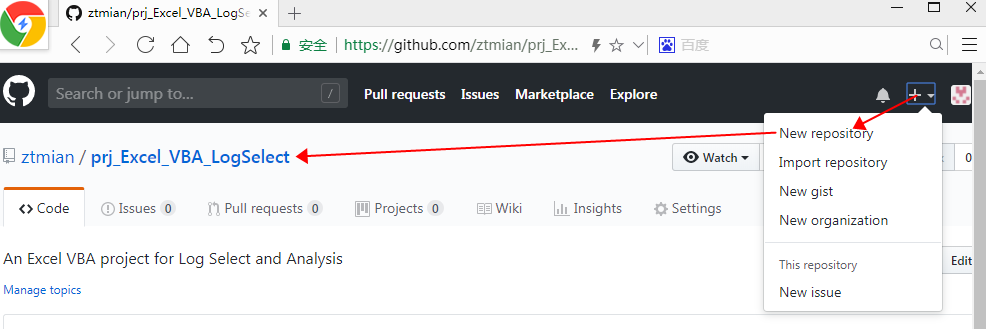
$ git commit Readme.txt -m “Update note”

Or use

$ git add Readme.txt

$ git commit -m “Update note”

$ git log



$ git remote add origin <https://github.com/ztmian/prj_Excel_VBA_LogSelect.git>

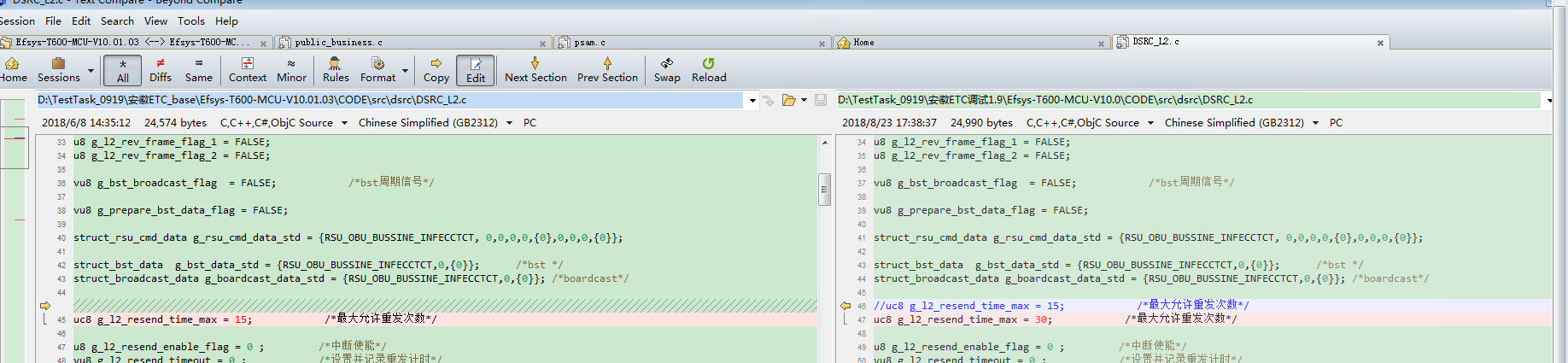
$ git push -u origin master

Second time use (after -u )

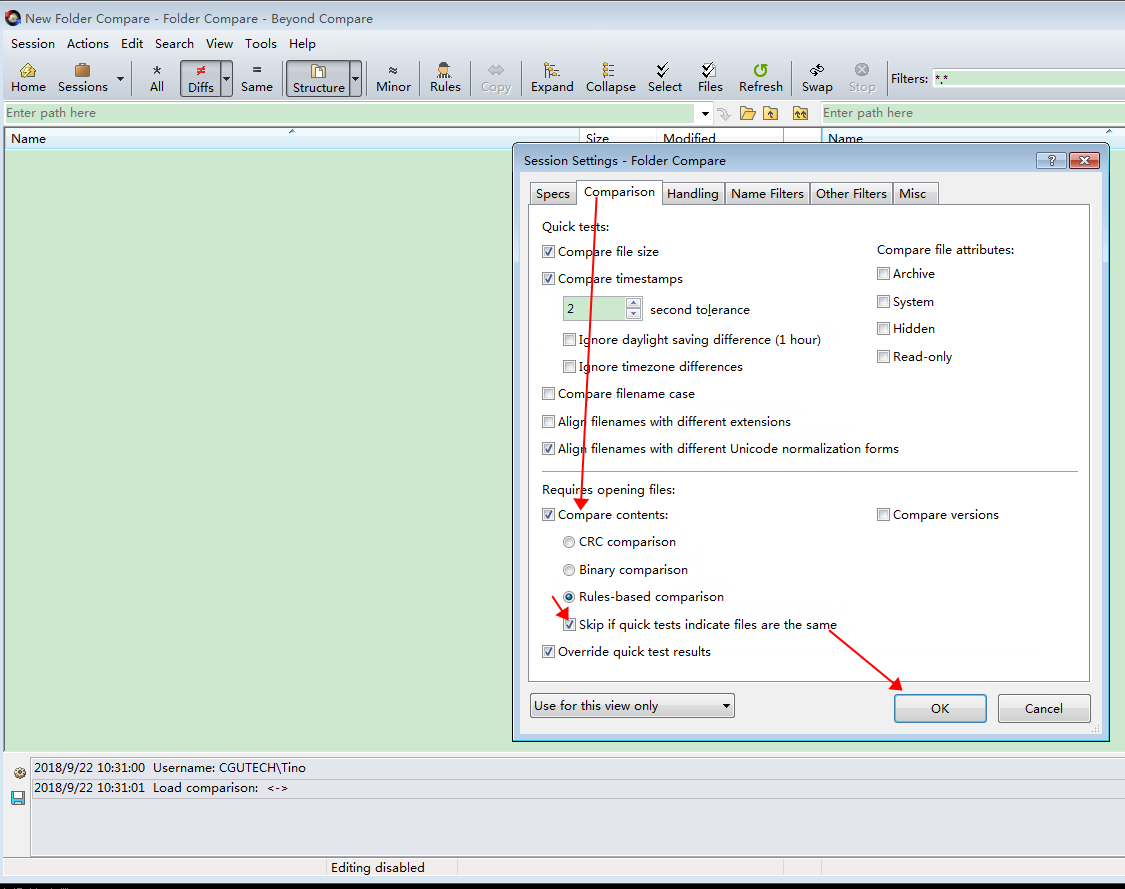
$ git push

## How to branch to master

# Compare 安徽Base , 安徽v1.9



# Beyond Compare

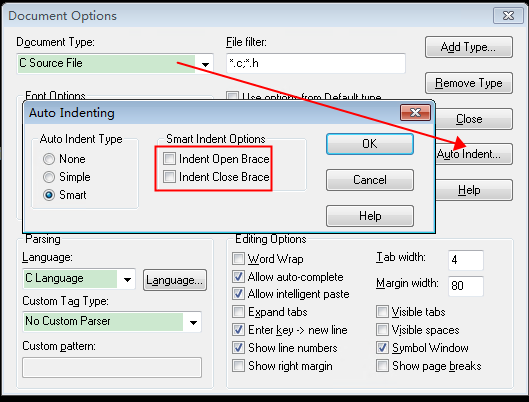


# Source insight

<https://blog.csdn.net/wfdtxz/article/details/8520629>

Tab to Spaces

自动缩进修改



### Source Insight Ctrl+F 会卡住

那是因为你在 Ctrl + f 的时候，你的光标当前位置是中文吧~~~

# Keil

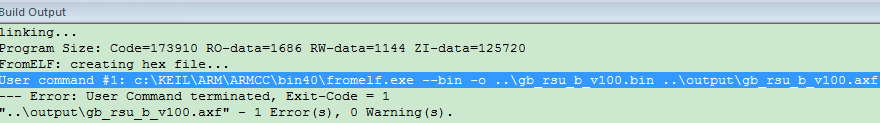
## Keil & Proteus Debug

refer

<https://blog.csdn.net/yagnruinihao/article/details/45843341>

## Build Output

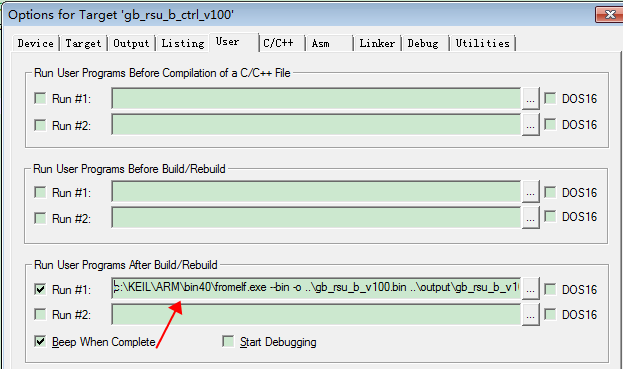
### In control program



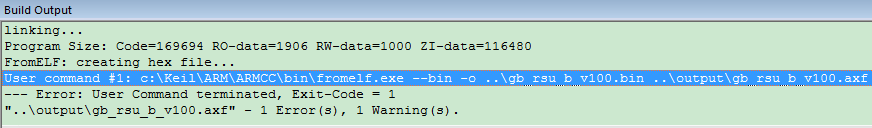
Check



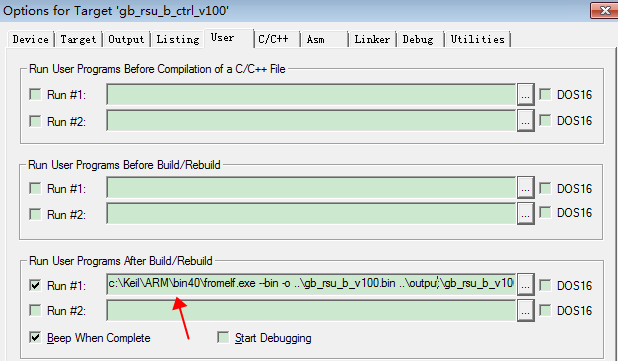
Fixed as below



### In antenna program

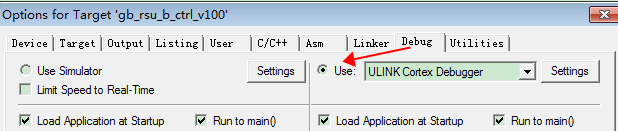


Fixed as below

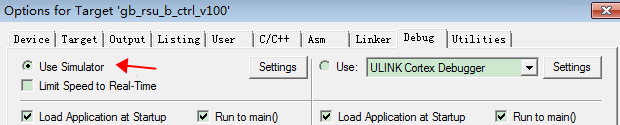


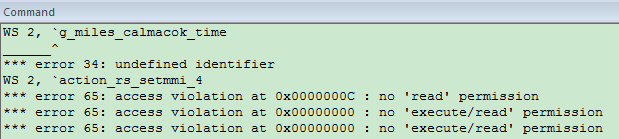
### Antenna program debug

Before



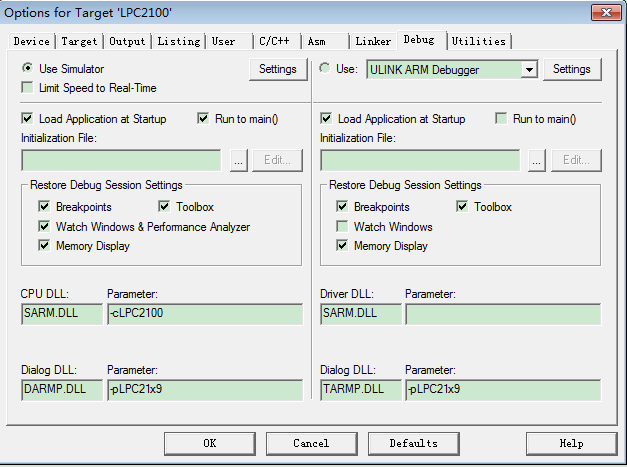
After



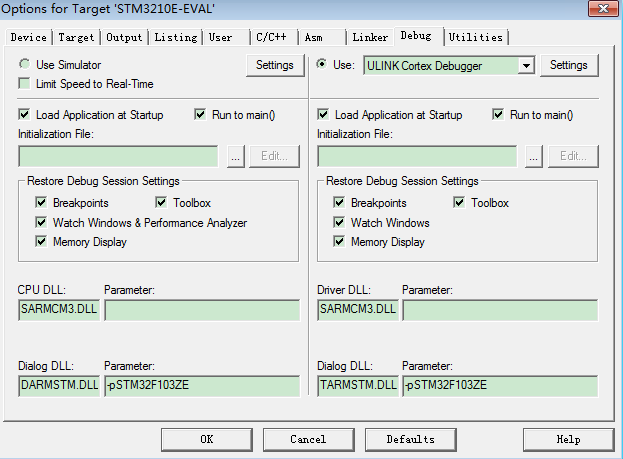


<https://blog.csdn.net/ben392797097/article/details/53184450>

#### keil example for debug

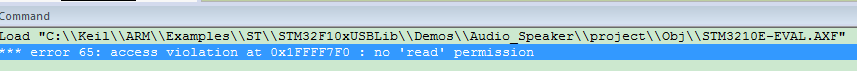


Different in Dialog DLL



### Keil examples debug in simulation error

a



b



c

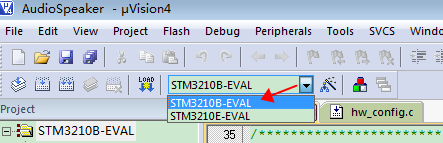


d



以上为默认E版

选择B版后



无报错，又好了？

奇怪。。。

### Share Software 600

远程后

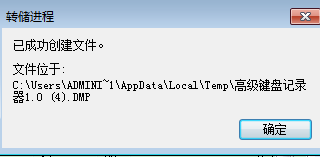
[\\192.168.11.12](\\\\192.168.11.12)

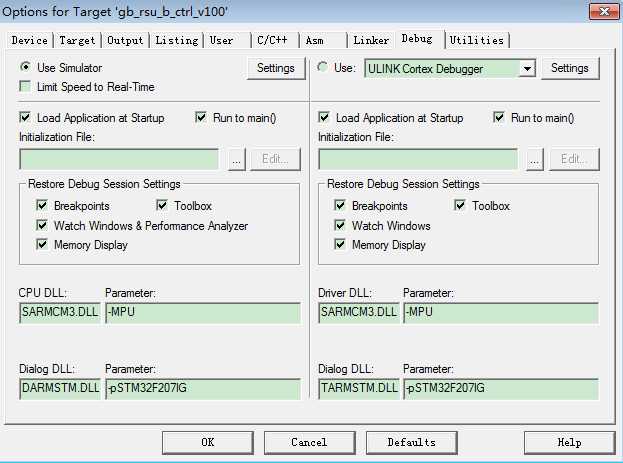
### Try

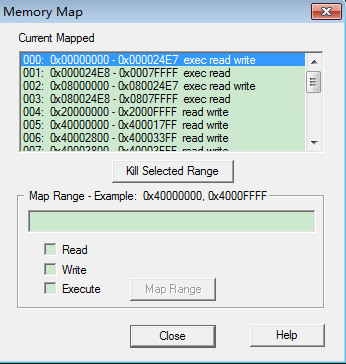
<https://blog.csdn.net/maple_leaf_15/article/details/51057991>

<https://pan.baidu.com/s/1pJrcqdd#list/path=%2Fpack%2FSTM32F2&parentPath=%2F>

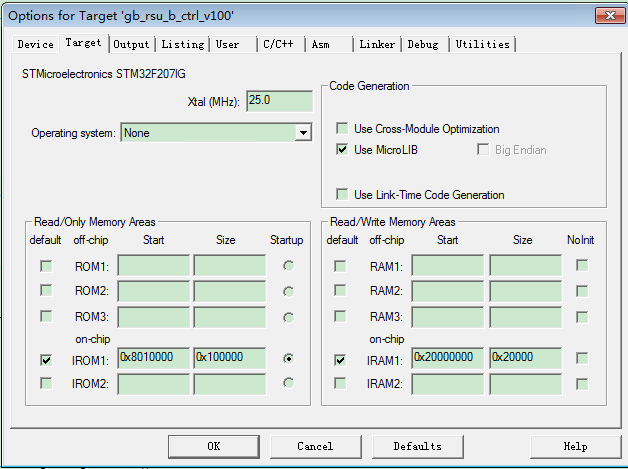
Before



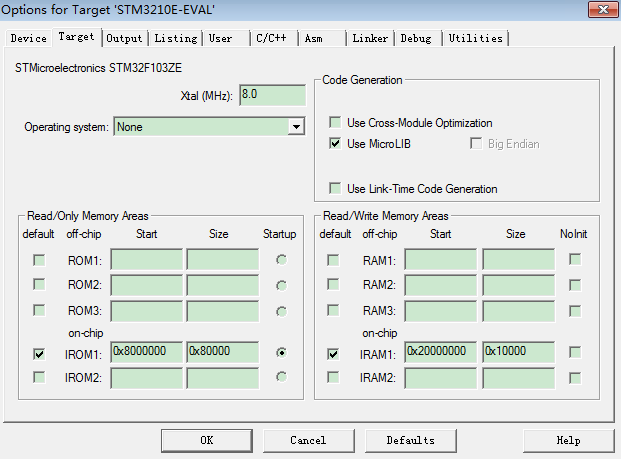




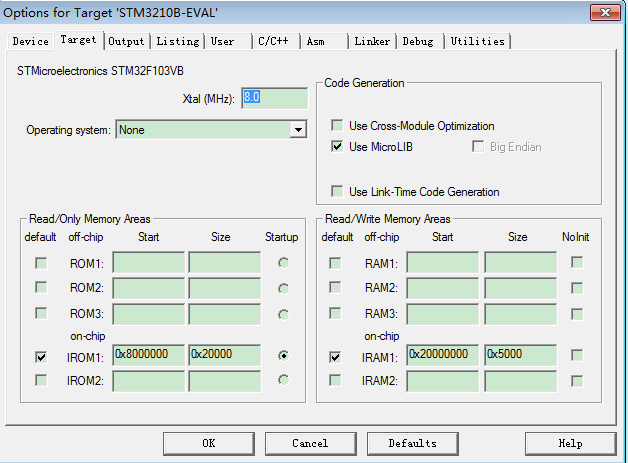
@cg



@E



@B



## Debug Simulation Antenna

Create my keil prj import antenna source code. Debug and view ASM code OK.

But couldnot continue, tr to change the debug option as below

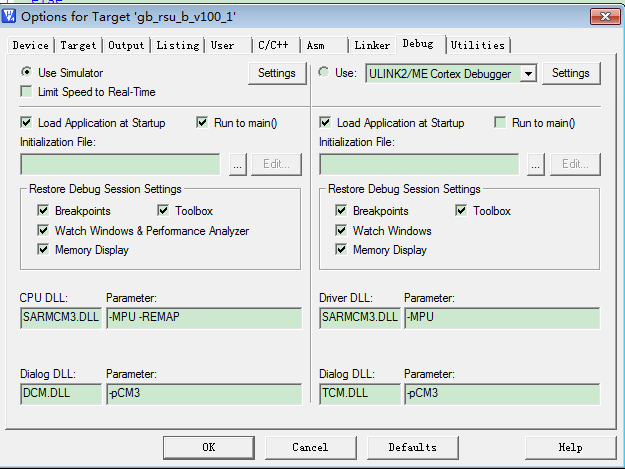
Refer to

<http://www.openedv.com/posts/list/26954.htm>

<https://blog.csdn.net/kinsno/article/details/43939759>

FF 80 F2 02 DB 3E 20 40 F7 91 05 01 00 14 80 01 00 00 15 BD 66 E5 0E 1D 5B EB 97 00 00 1D FF --读取OBU车辆信息

Before



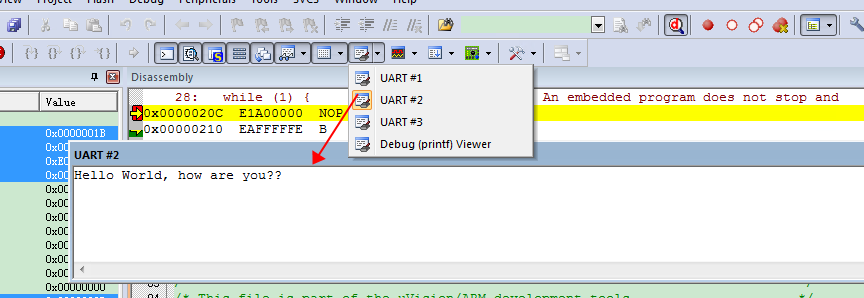
After

..

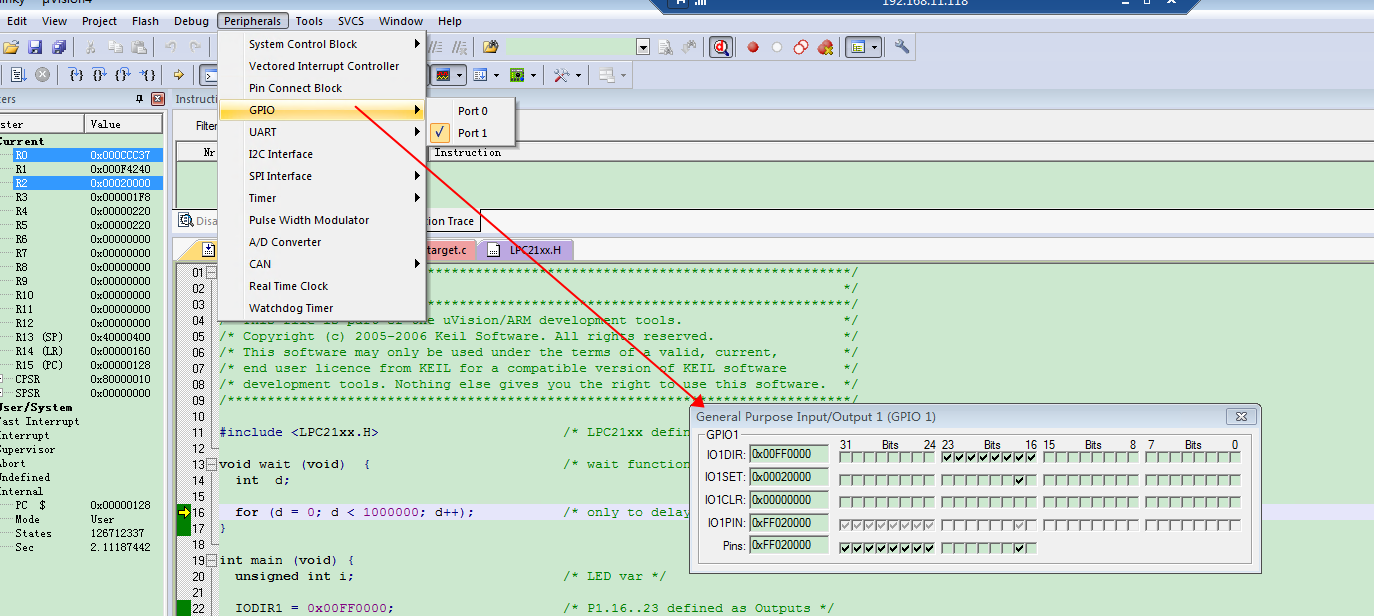
**Above Fail**

## Debug Simulation

Serial window, view re-target printf



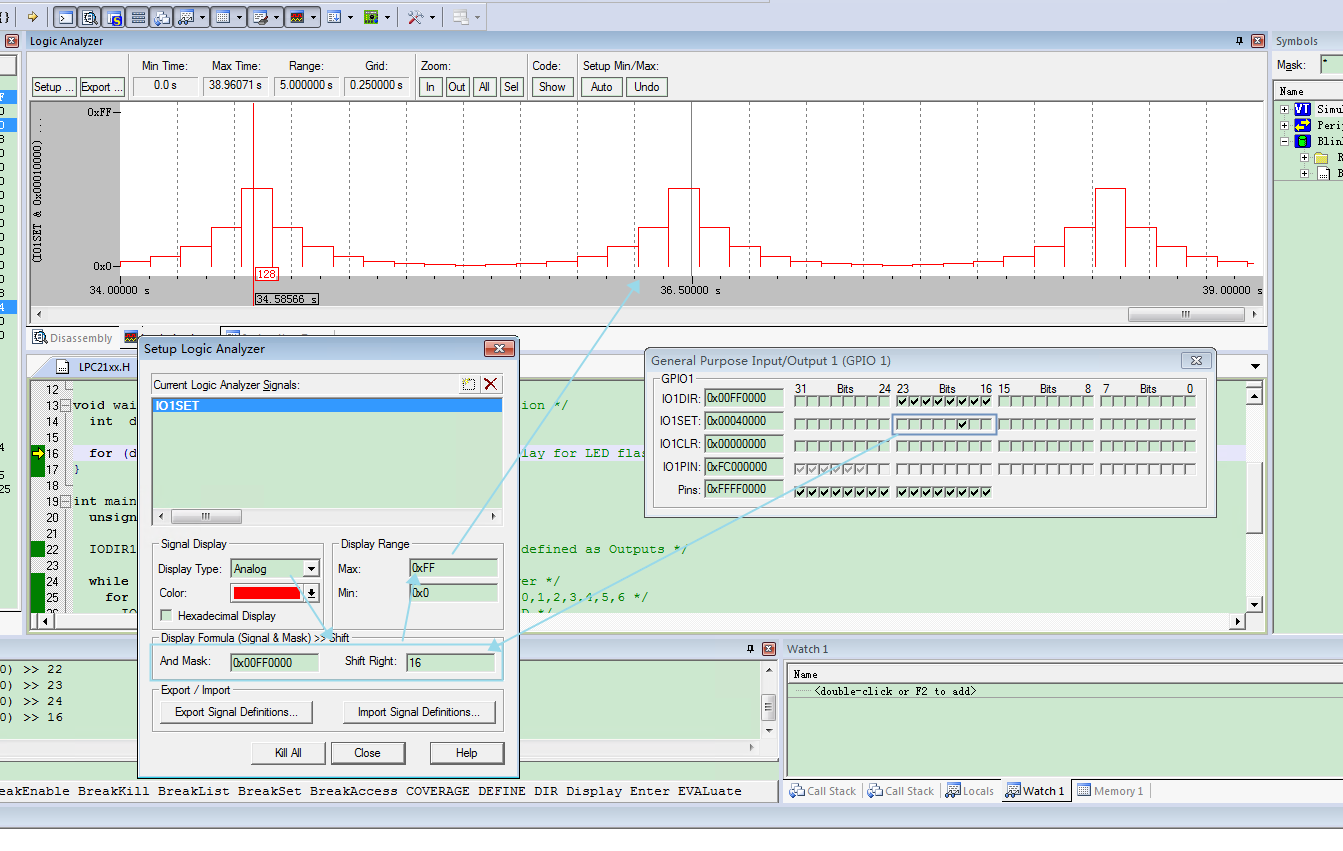
Debug->view GPIO



View IO1 blink in diagram,

Use IO1SET, And MASK 0x00FF0000, so that to get P.16 to P.23;

Shift right 16, and value range in [0x00~0xFF]



# My Reference

Antanna

傅里叶变换

<https://blog.csdn.net/wangchao712217/article/details/78731494>

CRC MD5校验

<https://blog.csdn.net/nodeathphoenix/article/details/8151750>

### P2P

<http://www.wangdai.info/2017/02/27/boluolicai/>

短信验证

<http://www.w6888.cn/login.aspx>

Ztmian 2730910

110108015521118

#### Qichacha

<https://www.qichacha.com/firm_b69e044dad995897b4a7bdd6b009df38.shtml>

# Notepad++正则

删除两行

.\*OBU编号.\*\r\n.\*OBU已过期，状态码：A2\r\n

车辆驶入1#线圈.\*\r\n.\*切换到【近端】交易区域.\*\r\n.\*未检测到标签

非贪婪匹配

车辆驶入1#线圈(.\*\r\n)\*?.\*未检测到

.\*车辆驶入1#线圈(.\*\r\n)\*?.\*未检测到.\*

.\*车辆驶入1#线圈(.\*\r\n)\*?.\*未检测到.\*(.\*\r\n)\*?.\*OBU编号.\*

.\*未检测到.\*(.\*\r\n)\*?.\*OBU编号.\*

# Excel VBA

<https://blog.csdn.net/christopherchen/article/details/53345695>

Like 通配符

<https://blog.csdn.net/narutohcl/article/details/7090353>

# C语言在线测试

<http://www.dooccn.com/c/>

### 

enum DAY { MON=1, TUE, WED, THU, FRI, SAT, SUN };

int main()

{

int my\_day = 0;

printf(“input a number”);

scanf(“%d”,&my\_day);

switch(my\_day)

{

case MON:

printf(“MON=%d”,my\_day);

break;

case TUE:

printf(“TUE=%d”,my\_day);

break;

case WED:

printf(“WED=%d”,my\_day);

break;

default:

break;

}

return 0;

}

/\*\*

Test of int static

\*\*/

#include <stdio.h>

int my\_count(int base)

{

//static int vi = 0; /\*vi will increase\*/

int vi = 0; /\*vi will always 0\*/

vi++;

printf("vi:%d\n",vi);

return (base+vi);

}

int main(void)

{

int range = 6;

int j;

scanf("%d",&range);

for(j = 0; j < range; j++)

{

printf("count:%d\n",my\_count(100));

}

return 0;

}

### 位域的使用

1）一个位域必须存储在同一个字节中，不能跨两个字节

2）位域可以无位域名，这时它只用来作填充或调整位置。无名的位域是不能使用的

3）使用无符号类型定义位域变量

//一个字节 8位

int 首位 为 符号位,

错误示例：  
typedef struct  
{  
     int bit1 : 1; // -1~0  
     int bit2 : 1;  
    // int bit3 : 1;

int bit3 : 2; // -2~1

} bit;  
  
    bit.bit1 = 1; // 0x1 saved as 0x1 = -1   
    bit.bit2 = 3; //0x11 save as 0x1 = -1  
    bit.bit3 = 6; //0x110 save as 0x10 = -1  
    printf("%d, %d, %d", bit.bit1, bit.bit2, bit.bit3 );  
              
            输出结果为：-1,-1, *-2*，不是: 1,3,6.

# End