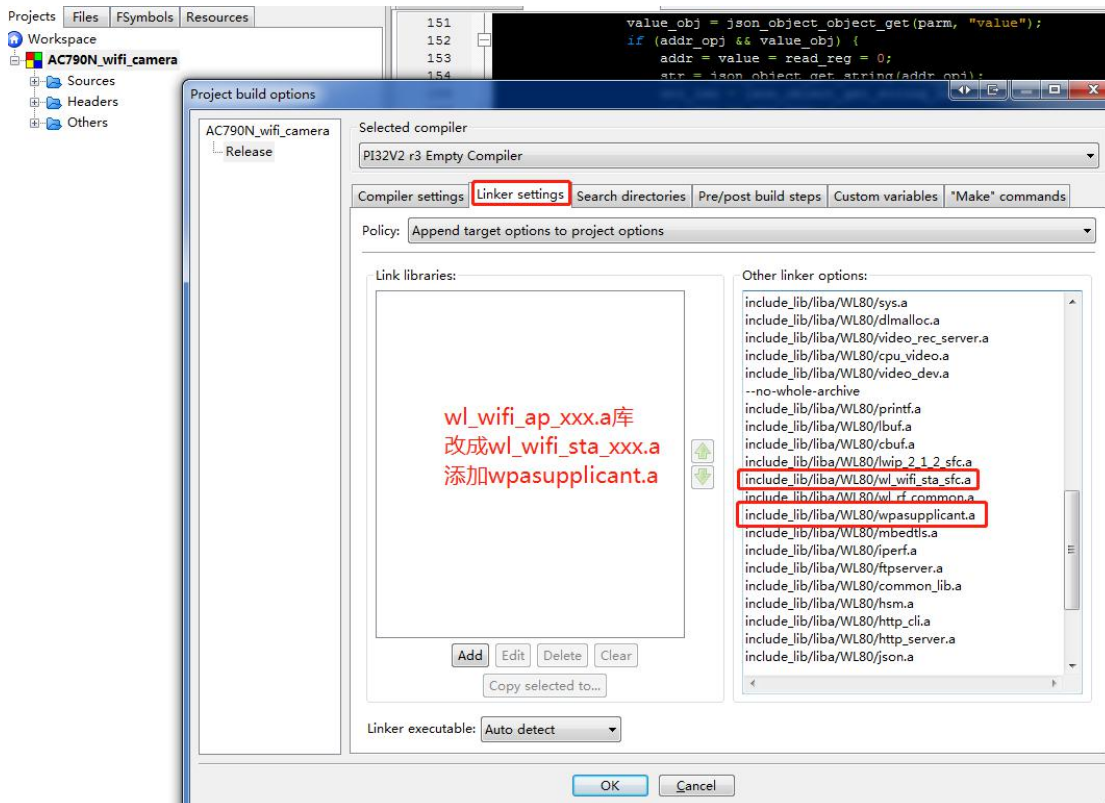


一、SDK 修改说明

(1) 开启量产测试，填写路由器名称和密码

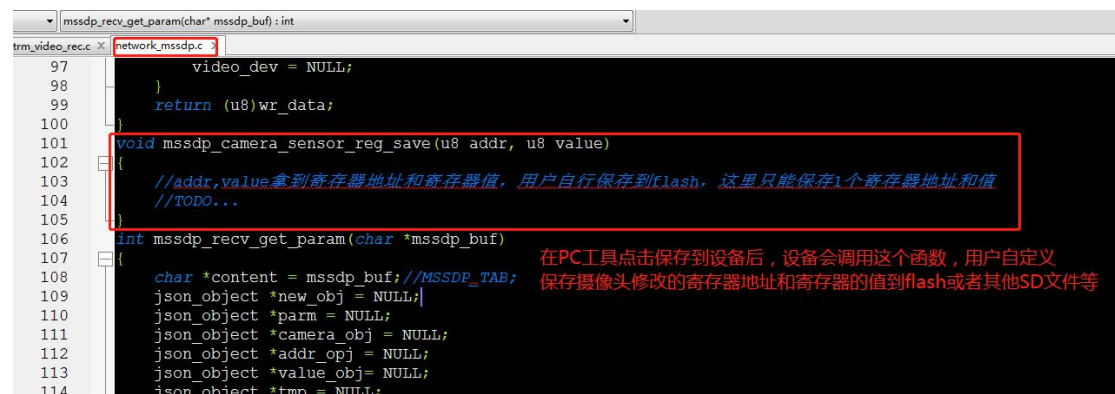
```
127 #ifndef CONFIG_NET_ENABLE
128 #define CONFIG_WIFI_ENABLE /* 无线WiFi */
129 #define CONFIG_SAVE_MAC_ADDR_IN_VM /* 决定flash保存在vm还是单独4K空间来保存 */
130 #define CONFIG_LOCAL_MACADDR_ENABLE /* 使用flash uid+随机数分配MAC地址 */
131 // #define CONFIG_STATIC_IPADDR_ENABLE /* 记忆路由分配的IP,下次直接使用记忆IP节省DHCP时间 */
132 // #define CONFIG_IPERF_ENABLE /* iperf测试 */
133 // #define CONFIG_DUMA_SDK_ENABLE
134 #define CONFIG_MASS_PRODUCTION_ENABLE // 启用量产模式
135 #endif
136
137 /* #define CONFIG_OSD_ENABLE */
138 #define CONFIG_VIDEO_REC_PPBUF_MODE
139 /* #define CONFIG_VIDEO_SPEC_DOUBLE_REC_MODE */
140
141 #ifndef CONFIG_MASS_PRODUCTION_ENABLE
142 #define ROUTER_SSID "genrui" // 路由器名称
143 #define ROUTER_PWD "12345678" // 路由器密码
144 // #define CONFIG_PRODUCTION_IO_PORT IO_PORTB_01 // 配置进入量产模式的IO
145 // #define CONFIG_PRODUCTION_IO_STATE 0 // 配置进入量产模式的IO状态: 0低电平, 1高电平
146 #endif
147
148 #ifndef CONFIG_SD0_ENABLE
149 #define CONFIG_STORAGE_PATH "storage/sd0"
150 #define SDX_DEV "sd0"
```

(2) 工程添加和修改库，原工程是 wl_wifi_ap_XXX.a 库则需要替换 wl_wifi_sta_XXX.a，如果是 wl_wifi.a 或 wl_wifi_sfc.a 说明支持 STA 模式，则不需要替换。没有 wpasupplicant.a 库加上 wpasupplicant.a 库。



(3) 用户自定义保存摄像头调试寄存器地址和值到 flash 或 SD 文件等接口。void mssdp_camera_sensor_reg_save(u8 addr, u8 value)，在 PC 点击保存到设备后设备会调用该函数，用户自行加代码进行保存，当然也可以在 PC 另存保存文件。

设备保存寄存器 API 接口



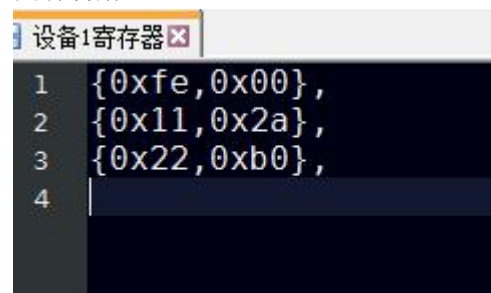
```
mssdp_rcv_get_param(char* mssdp_buf): int
network_mssdp.c
97     video_dev = NULL;
98 }
99     return (u8)wr_data;
100 }
101 void mssdp_camera_sensor_reg_save(u8 addr, u8 value)
102 {
103     //addr,value拿到寄存器地址和寄存器值。用户自行保存到flash, 这里只能保存1个寄存器地址和值
104     //TODO...
105 }
106 int mssdp_rcv_get_param(char *mssdp_buf)
107 {
108     char *content = mssdp_buf; //MSSDP_TAB;
109     json_object *new_obj = NULL;
110     json_object *parm = NULL;
111     json_object *camera_obj = NULL;
112     json_object *addr_obj = NULL;
113     json_object *value_obj = NULL;
114     json_object *tmp = NULL;
```

在PC工具点击保存到设备后，设备会调用这个函数，用户自定义保存摄像头修改的寄存器地址和寄存器的值到flash或者其他SD文件等

PC 保存寄存器按钮（后面有讲解）



PC 另存保存文件打开后效果，直接可以复制这些寄存器和值加到到 SDK 摄像头驱动里（后面有讲解）



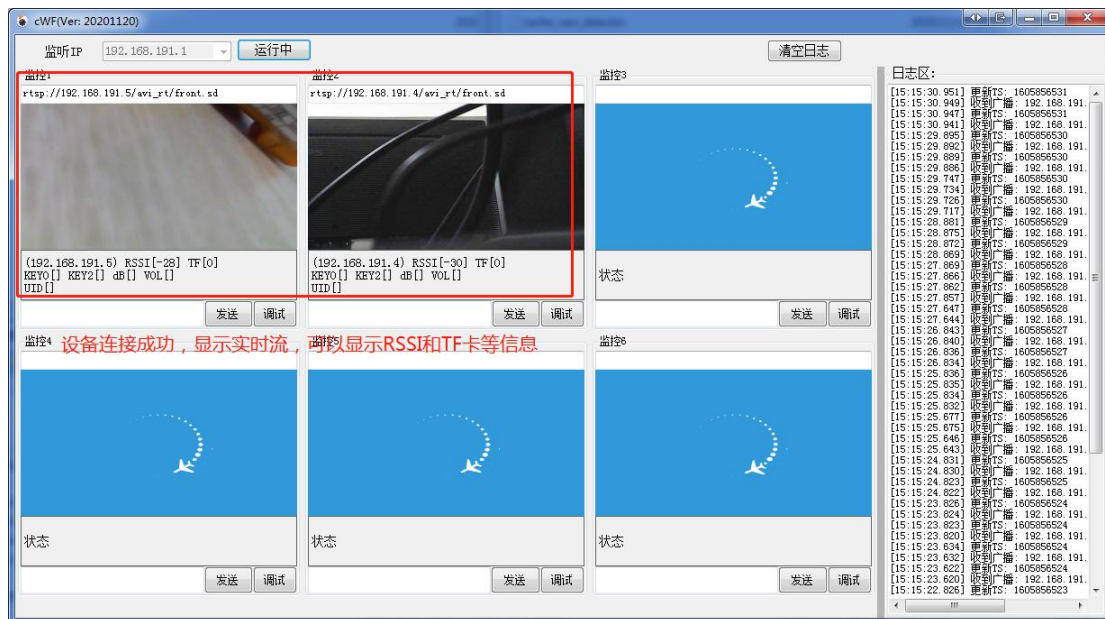
（4）改动完成，全编译下载到设备

二、PC 量产工具使用说明

(1) 打开 SDK\工具\量产测试工具\cFW-V2.0\cFW.exe，开启电脑 PC 的量产工具监听功能



(2) 设备成功连接如下图



(3) 需要调试摄像头（不调试摄像头跳过该步骤）

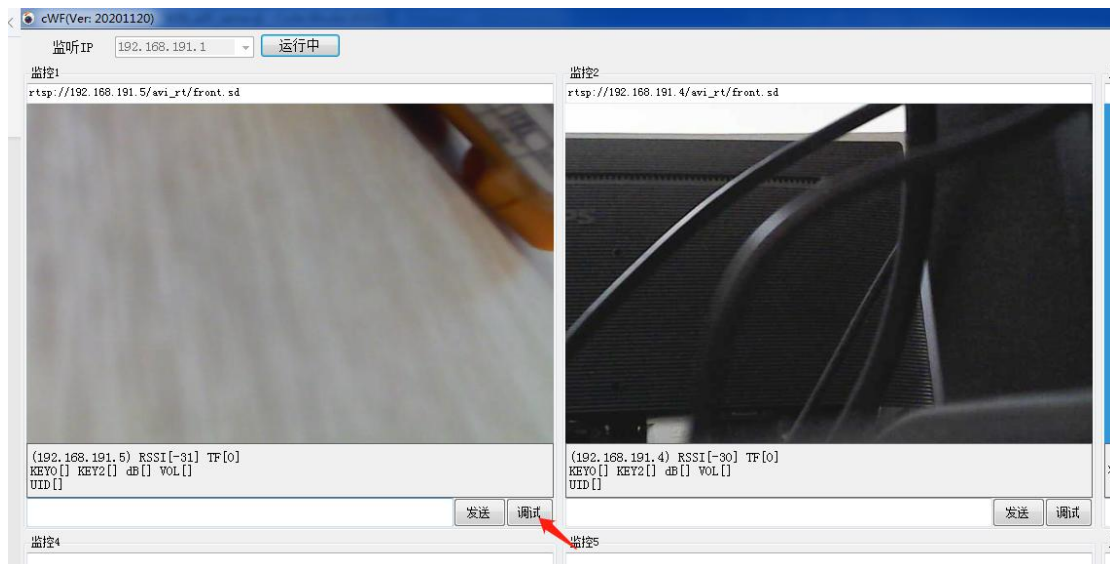
在 PC 量产工具对摄像头进行调试步骤：

①首先在 SDK 的 strm_video_rec_get_abr 函数调高码率以便效果更佳（**调高码率，下载固件到设备，量产模式时候码率需要调回低于 2000 以提高量产速度**）

```
strm_video_rec.c  network_mssdp.c
37  /*码率控制, 根据具体分辨率设置*/
38  static int strm_video_rec_get_abr(u32 width)
39  {
40      if (width <= 384) {
41          return 1000;
42      } else if (width <= 640) {
43          return 5000; //2000;
44      } else if (width <= 1280) {
45          return 5000;
46          /* return 10000; */
47      } else if (width <= 1920) {
48          return 3000;
49      } else {
50          return 3000;
51      }
52  }
53  int strm_video_rec_get_fps(void)
54  {
```

调试摄像头码率：5000-55000
调试摄像头需要保存稳定，才能看出效果
量产时码率：0-2000即可

②点击调试按钮，进入摄像头寄存器调试界面



调试摄像头可以双击显示区放大显示，方便看效果



进入界面：



一般摄像头调试都是改变寄存器的值，所以调试工具只提供寄存器读写，不同摄像头对应地址不一样，需要用户自行和摄像头原厂联系如何调试摄像头效果

③比如读取 id 号对应寄存器地址，则在寄存器区域写对应寄存器值(GC032 为 0xf0)，点击读取，如下图：



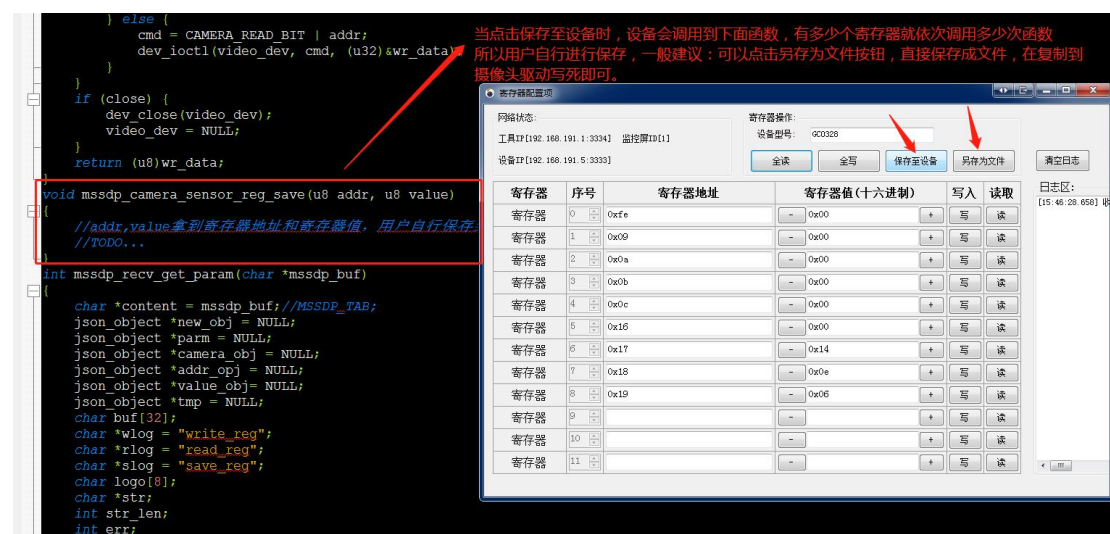
④比如需要写很多个寄存器，一般先选择寄存器页数（一般为 0xfe），然后再继续写其他寄存器，按照 PC 工具寄存器序号顺序填写寄存器地址和寄存器值，写完后可以通过点击右侧的读写按钮进行单次手动读写，也可以点击上面全读写按钮进行依次全部寄存器的读写。调试摄像头时，寄存器值可以手动输入，也可以输入第一个值后点击两侧加减 (+) 按钮进行寄存器值的加减，方便调试。



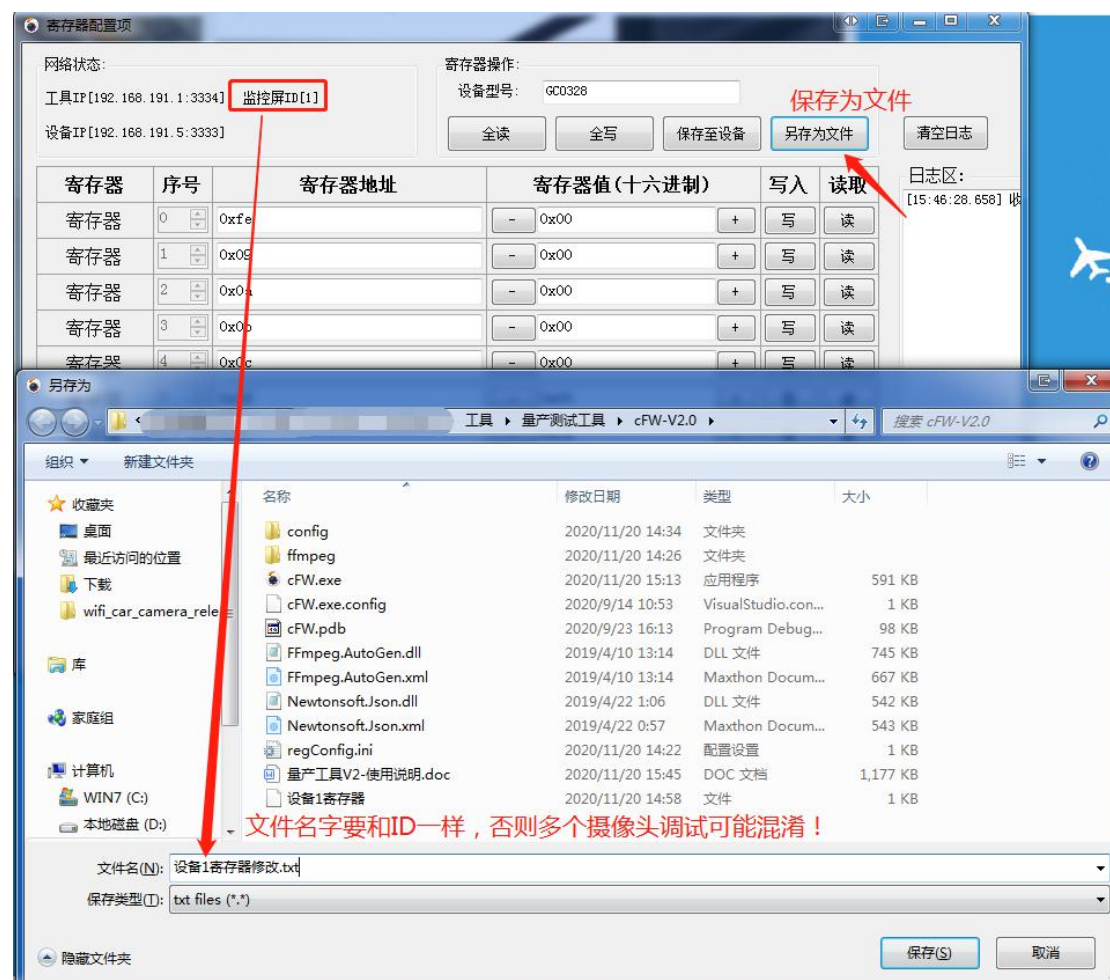
单击读写按钮后，可能由于网络延迟和丢帧等原因，未能及时完成读写，看不到效果则需要多次点击读写按钮（或者全读写按钮），当然调试时接上设备打印就可以看出设备是否收到是否完成摄像头读写操作，也可以看日志收到数据时间和电脑系统时间进行判断，如下图。



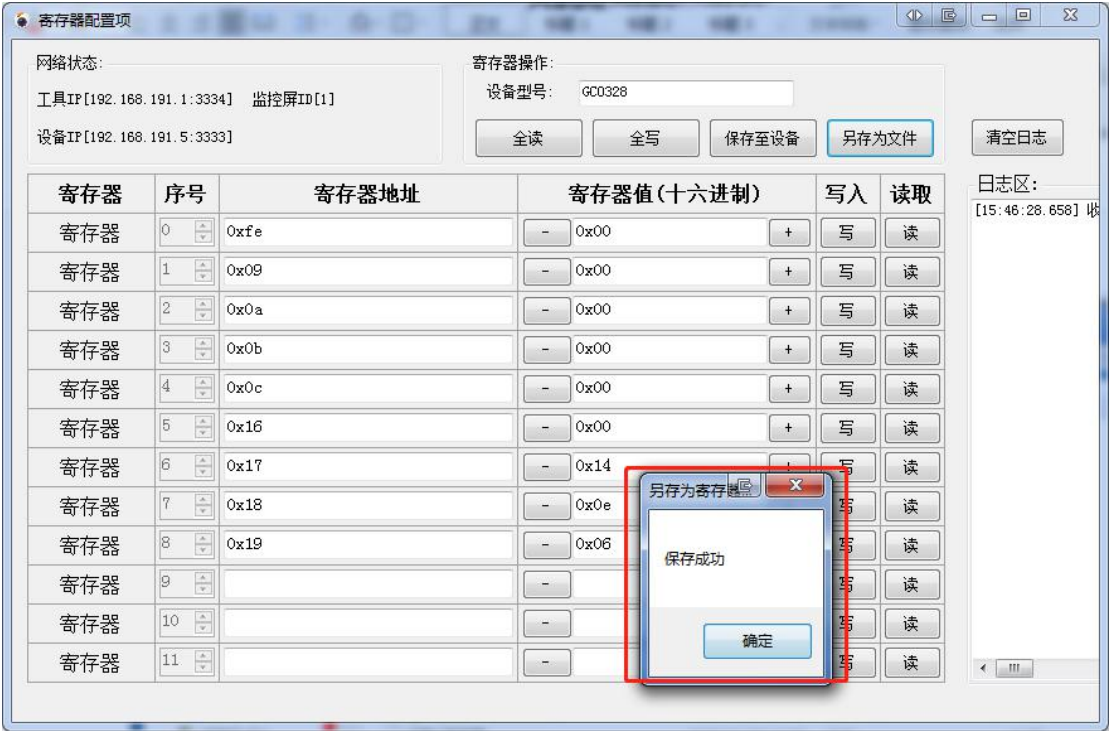
⑤保存寄存器，可以选择保存到设备，保存到设备需要用户自行保存，也可以保存成文件。



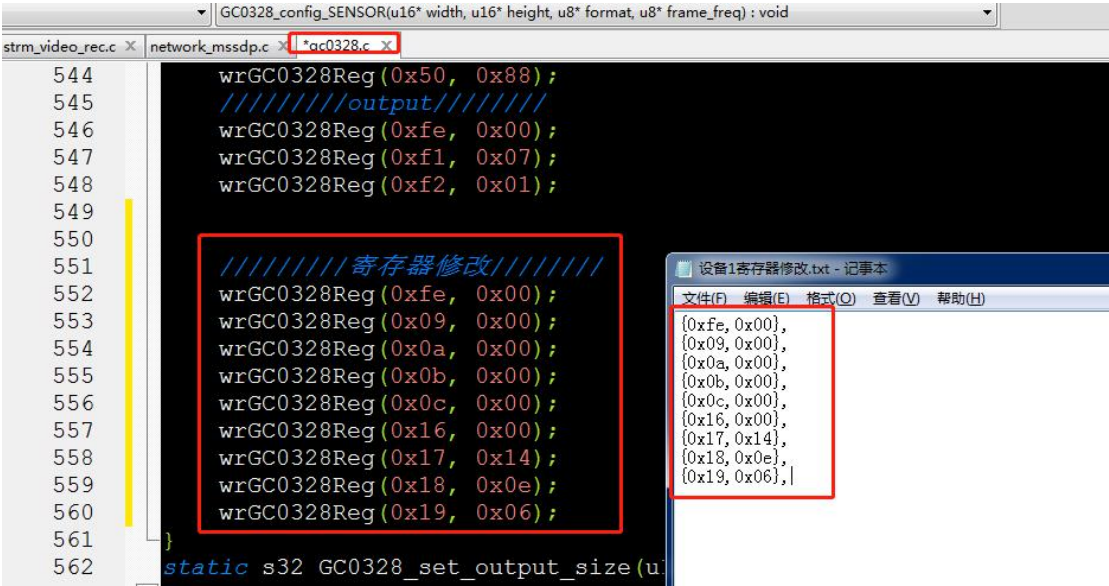
建议：点击另存为文件，保存成文件，文件名字要和监控屏 ID 有关联，否则多个摄像头调试时候保存文件容易混淆。



保存成功:



⑥得到效果之后改动的寄存器添加到摄像头的驱动，例如



⑦摄像头效果调试完毕，把修改的寄存器值和添加的寄存器添加到摄像头对应驱动里，同时把 `strm_video_rec_get_abr` 函数的码率降低。


```
strm video rec.c
37  /*码率控制, 根据具体分辨率设置*/
38  static int strm_video_rec_get_abr(u32 width)
39  {
40      if (width <= 384) {
41          return 1000;
42      } else if (width <= 640) {
43          return 1000;//2000;
44      } else if (width <= 1280) {
45          return 1000;
46          /* return 10000; */
47      } else if (width <= 1920) {
48          return 3000;
49      } else {
50          return 3000;
51      }
52  }
53  int strm_video_rec_get_fps(void)
```

⑧编译下载到设备，完成!!!