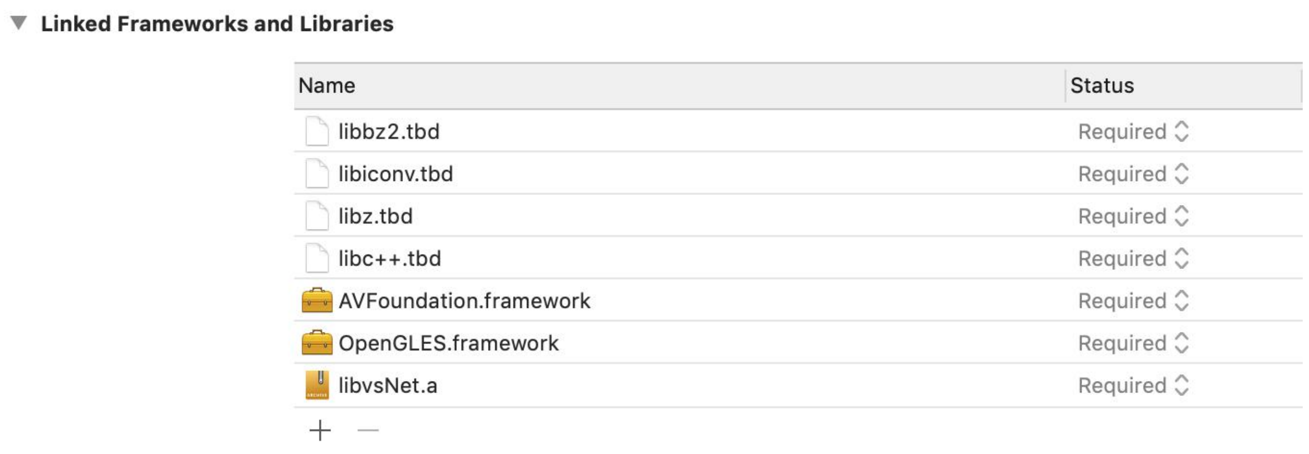
IOS VSNet 库使用说明

1.初化库

1）XCODE: Enable Bitcode yes 改为 no

Enable Testability yes 改为 no

2）依赖库

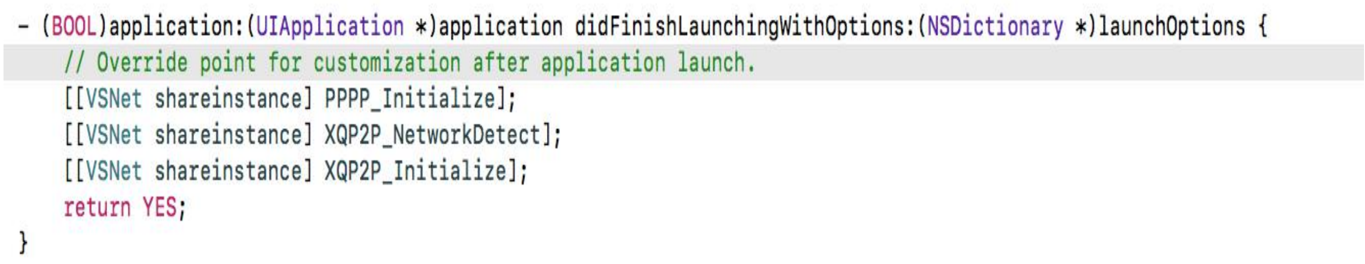


▼ Linked Frameworks and Libraries

Name	Status
libbz2.tbd	Required
libiconv.tbd	Required
libz.tbd	Required
libc++.tbd	Required
AVFoundation.framework	Required
OpenGLES.framework	Required
libvsNet.a	Required

注意：XCode10 以上需要把依赖库 libstbc++.tbd 替换成 libc++.tbd.

3）初始化库



```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
    // Override point for customization after application launch.
    [[VSNet shareinstance] PPPP_Initialize];
    [[VSNet shareinstance] XQP2P_NetworkDetect];
    [[VSNet shareinstance] XQP2P_Initialize];
    return YES;
}
```

2.设备管理

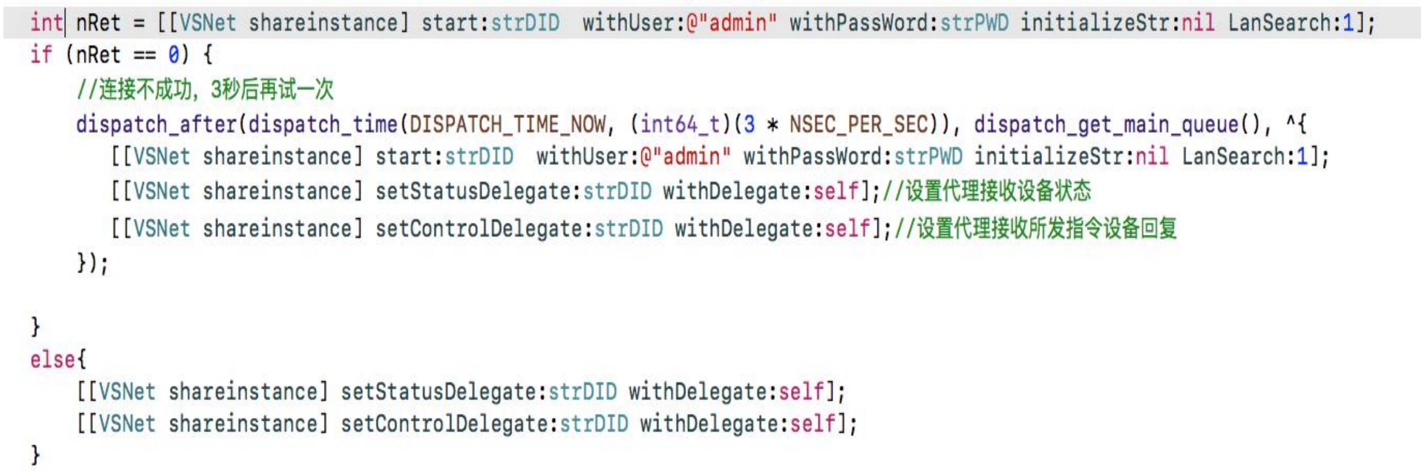
1）连接设备

Vuid 和 uid 的区别（特别注意）

中间数字大于 6 位（不做上限限制），且整串的头部和尾部均带字母的（只是带字母即可，不做位数判断），即为 VUID，否则为 uid，不同的 id 使用不同的连接函数。

判断 uid 是否是 VUID 通过–(BOOL) IsVUID:(NSString*) strID;接口

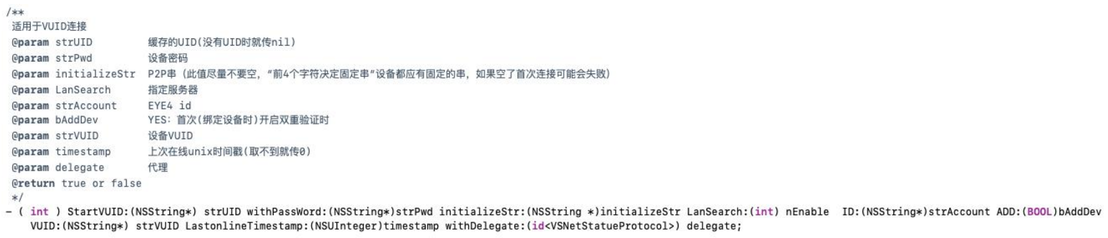
如果非 vuid，就调下面方法



```
int nRet = [[VSNet shareinstance] start:strDID  withUser:@"admin" withPassWord:strPWD initializeStr:nil LanSearch:1];
if (nRet == 0) {
    //连接不成功，3秒后再试一次
    dispatch_after(dispatch_time(DISPATCH_TIME_NOW, (int64_t)(3 * NSEC_PER_SEC)), dispatch_get_main_queue(), ^{
       [[VSNet shareinstance] start:strDID  withUser:@"admin" withPassWord:strPWD initializeStr:nil LanSearch:1];
       [[VSNet shareinstance] setStatusDelegate:strDID withDelegate:self];//设置代理接收设备状态
       [[VSNet shareinstance] setControlDelegate:strDID withDelegate:self];//设置代理接收所发指令设备回复
    });

}
else{
    [[VSNet shareinstance] setStatusDelegate:strDID withDelegate:self];
    [[VSNet shareinstance] setControlDelegate:strDID withDelegate:self];
}
```

如果是 vuid，就调下面的 StartVUID 方法开始连接

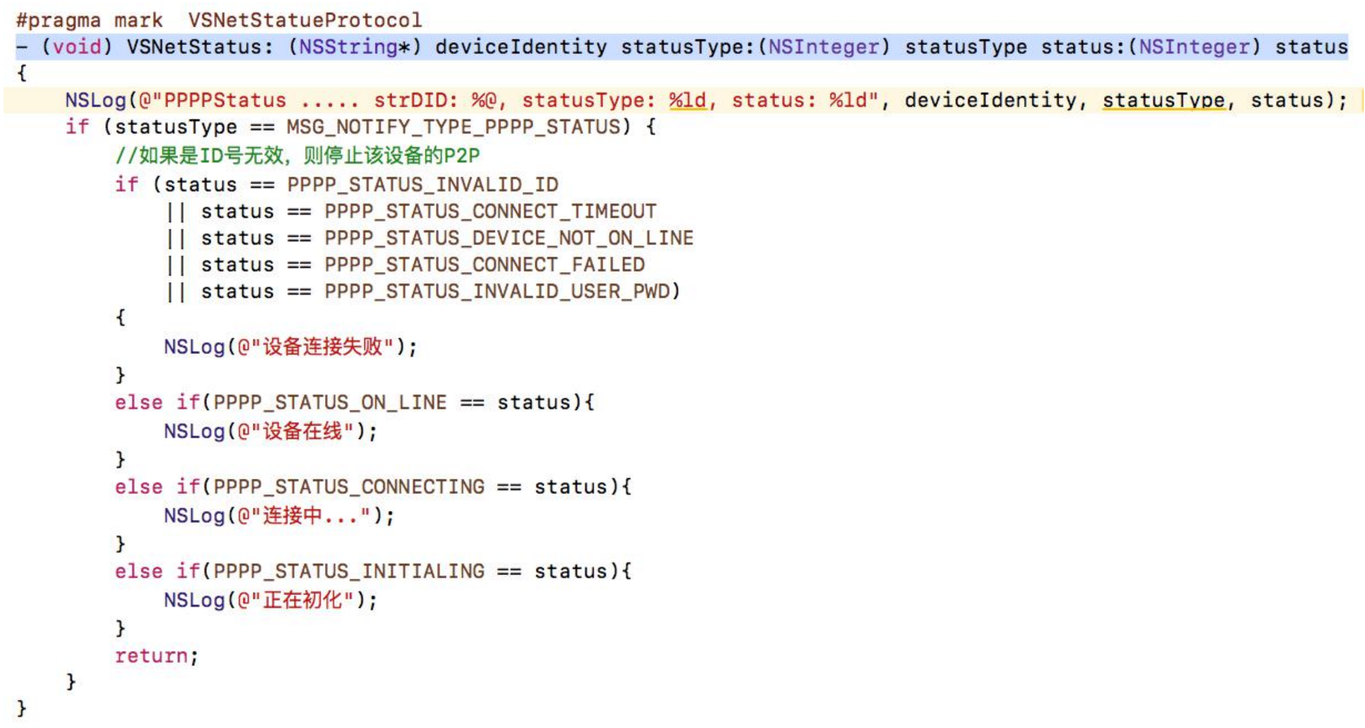


```
/**
 适用于VUID连接
 @param strUID          缓存的UID(没有UID时就传nil)
 @param strPwd          设备密码
 @param initializeStr   P2P串（此值尽量不要空，“前4个字符决定固定串”设备都应有固定的串，如果空了首次连接可能会失败）
 @param LanSearch       指定服务器
 @param strAccount      EYE4 id
 @param bAddDev         YES：首次(绑定设备时)开启双重验证时
 @param strVUID         设备VUID
 @param timestamp       上次在线unix时间戳(取不到就传0)
 @param delegate        代理
 @return true or false
 */
- ( int ) StartVUID:(NSString*) strUID withPassWord:(NSString*)strPwd initializeStr:(NSString *)initializeStr LanSearch:(int) nEnable  ID:(NSString*)strAccount ADD:(BOOL)bAddDev
    VUID:(NSString*) strVUID LastonlineTimestamp:(NSUInteger)timestamp withDelegate:(id<VSNetStatueProtocol>) delegate;
```

2） 断开连接

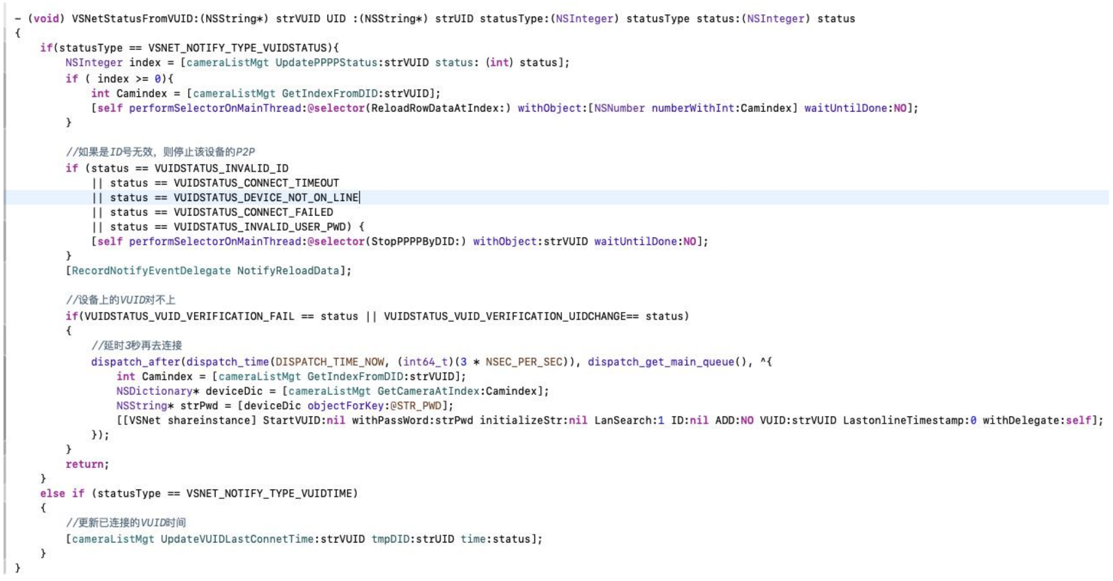
```
[[VSNet shareinstance] stop:strDID];
```

3） uid 连接状态接收



```
#pragma mark  VSNetStatueProtocol
- (void) VSNetStatus: (NSString*) deviceIdentity statusType:(NSInteger) statusType status:(NSInteger) status
{
    NSLog(@"PPPPStatus ..... strDID: %@, statusType: %ld, status: %ld", deviceIdentity, statusType, status);
    if (statusType == MSG_NOTIFY_TYPE_PPPP_STATUS) {
        //如果是ID号无效，则停止该设备的P2P
        if (status == PPPP_STATUS_INVALID_ID
            || status == PPPP_STATUS_CONNECT_TIMEOUT
            || status == PPPP_STATUS_DEVICE_NOT_ON_LINE
            || status == PPPP_STATUS_CONNECT_FAILED
            || status == PPPP_STATUS_INVALID_USER_PWD)
        {
            NSLog(@"设备连接失败");
        }
        else if(PPPP_STATUS_ON_LINE == status){
            NSLog(@"设备在线");
        }
        else if(PPPP_STATUS_CONNECTING == status){
            NSLog(@"连接中...");
        }
        else if(PPPP_STATUS_INITIALING == status){
            NSLog(@"正在初化");
        }
        return;
    }
}
```

VUID 连接过程中的状态返回



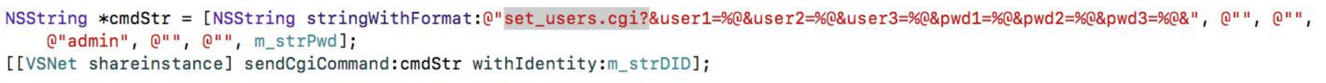
```
- (void) VSNetStatusFromVUID:(NSString*) strVUID UID :(NSString*) strUID statusType:(NSInteger) statusType status:(NSInteger) status
{
    if(statusType == VSNET_NOTIFY_TYPE_VUIDSTATUS){
        NSInteger index = [cameraListMgt UpdatePPPPStatus:strVUID status: (int) status];
        if ( index >= 0){
            int Camindex = [cameraListMgt GetIndexFromDID:strVUID];
            [self performSelectorOnMainThread:@selector(ReloadRowDataAtIndex:) withObject:[NSNumber numberWithInt:Camindex] waitUntilDone:NO];
        }

        //如果是ID号无效，则停止该设备的P2P
        if (status == VUIDSTATUS_INVALID_ID
            || status == VUIDSTATUS_CONNECT_TIMEOUT
            || status == VUIDSTATUS_DEVICE_NOT_ON_LINE
            || status == VUIDSTATUS_CONNECT_FAILED
            || status == VUIDSTATUS_INVALID_USER_PWD) {
            [self performSelectorOnMainThread:@selector(StopPPPPByDID:) withObject:strVUID waitUntilDone:NO];
        }
        [RecordNotifyEventDelegate NotifyReloadData];

        //设备上的VUID对不上
        if(VUIDSTATUS_VUID_VERIFICATION_FAIL == status || VUIDSTATUS_VUID_VERIFICATION_UIDCHANGE== status)
        {
            //延时3秒再去连接
            dispatch_after(dispatch_time(DISPATCH_TIME_NOW, (int64_t)(3 * NSEC_PER_SEC)), dispatch_get_main_queue(), ^{
                int Camindex = [cameraListMgt GetIndexFromDID:strVUID];
                NSDictionary* deviceDic = [cameraListMgt GetCameraAtIndex:Camindex];
                NSString* strPwd = [deviceDic objectForKey:@STR_PWD];
                [[VSNet shareinstance] StartVUID:nil withPassWord:strPwd initializeStr:nil LanSearch:1 ID:nil ADD:NO VUID:strVUID LastonlineTimestamp:0 withDelegate:self];
            });
        }
        return;
    }
    else if (statusType == VSNET_NOTIFY_TYPE_VUIDTIME)
    {
        //更新已连接的VUID时间
        [cameraListMgt UpdateVUIDLastConnetTime:strVUID tmpDID:strUID time:status];
    }
}
```

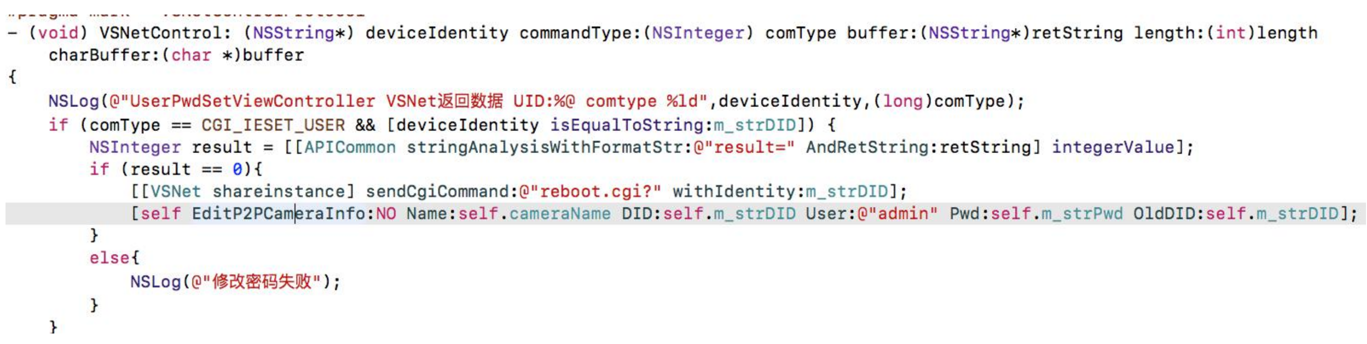
4）设备密码管理

(4.1) 重置设备密码



```
NSString *cmdStr = [NSString stringWithFormat:@"set_users.cgi?&user1=%@&user2=%@&user3=%@&pwd1=%@&pwd2=%@&pwd3=%@&", @"", @"",
    @"admin", @"", @"", m_strPwd];
[[VSNet shareinstance] sendCgiCommand:cmdStr withIdentity:m_strDID];
```

(4.2) 重置设备密码返回



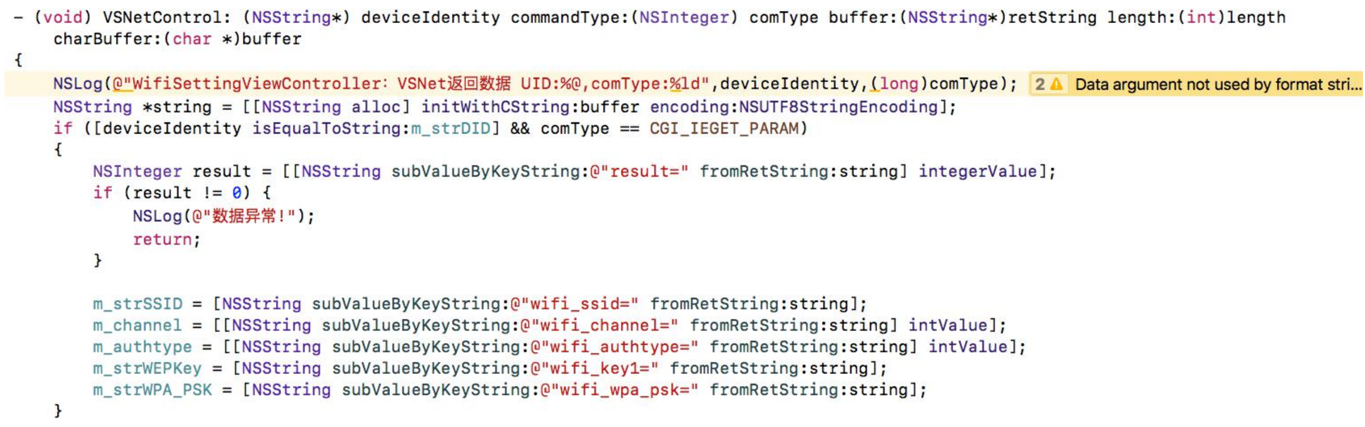
```
- (void) VSNetControl: (NSString*) deviceIdentity commandType:(NSInteger) comType buffer:(NSString*)retString length:(int)length
    charBuffer:(char *)buffer
{
    NSLog(@"UserPwdSetViewController VSNet返回数据 UID:%@ comtype %ld",deviceIdentity,(long)comType);
    if (comType == CGI_IESET_USER && [deviceIdentity isEqualToString:m_strDID]) {
        NSInteger result = [[APICommon stringAnalysisWithFormatStr:@"result=" AndRetString:retString] integerValue];
        if (result == 0){
            [[VSNet shareinstance] sendCgiCommand:@"reboot.cgi?" withIdentity:m_strDID];
            [self EditP2PCameraInfo:NO Name:self.cameraName DID:self.m_strDID User:@"admin" Pwd:self.m_strPwd OldDID:self.m_strDID];
        }
        else{
            NSLog(@"修改密码失败");
        }
    }
```

5）设备 wifi 管理

(5.1) 获取当前设备 WIFI

```
[[VSNet shareinstance] sendCgiCommand:@"get_params.cgi?"withIdentity:self.m_strDID];
```

(5.2)获取当前设备 WIFI 返回



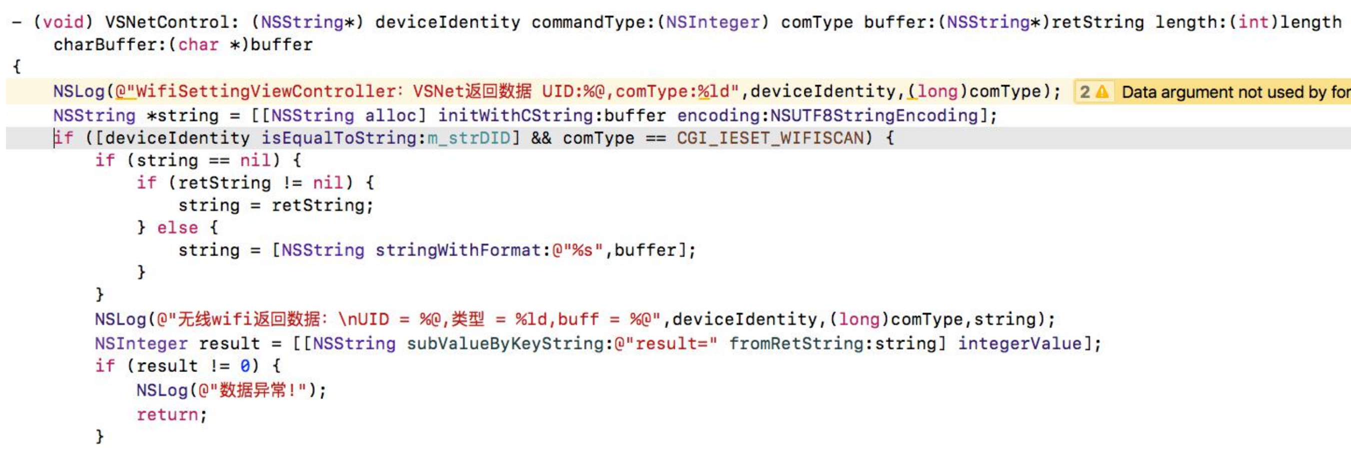
```
- (void) VSNetControl: (NSString*) deviceIdentity commandType:(NSInteger) comType buffer:(NSString*)retString length:(int)length
    charBuffer:(char *)buffer
{
    NSLog(@"WifiSettingViewController: VSNet返回数据 UID:%@,comType:%ld",deviceIdentity,(long)comType);  2 ⚠ Data argument not used by format stri...
    NSString *string = [[NSString alloc] initWithCString:buffer encoding:NSUTF8StringEncoding];
    if ([deviceIdentity isEqualToString:m_strDID] && comType == CGI_IEGET_PARAM)
    {
        NSInteger result = [[NSString subValueByKeyString:@"result=" fromRetString:string] integerValue];
        if (result != 0) {
            NSLog(@"数据异常!");
            return;
        }

        m_strSSID = [NSString subValueByKeyString:@"wifi_ssid=" fromRetString:string];
        m_channel = [[NSString subValueByKeyString:@"wifi_channel=" fromRetString:string] intValue];
        m_authtype = [[NSString subValueByKeyString:@"wifi_authtype=" fromRetString:string] intValue];
        m_strWEPKey = [NSString subValueByKeyString:@"wifi_key1=" fromRetString:string];
        m_strWPA_PSK = [NSString subValueByKeyString:@"wifi_wpa_psk=" fromRetString:string];
    }
```

(5.3) 获取设备 WIFI 列表

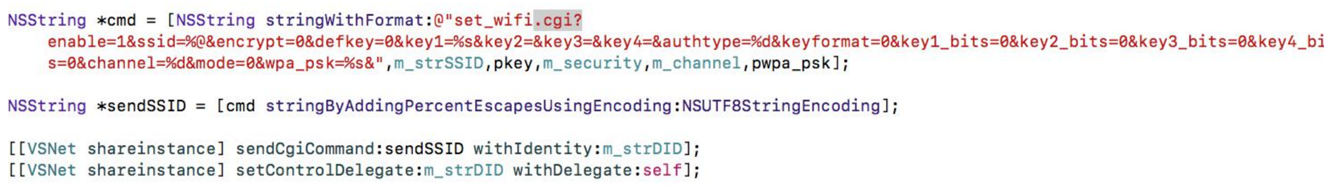
```
[[VSNet shareinstance] sendCgiCommand:@"wifi_scan.cgi?"
withIdentity:self.m_strDID];
[[VSNet shareinstance] setControlDelegate:self.m_strDID withDelegate:self];
```

(5.4)获取设备 WIFI 列表返回



```
- (void) VSNetControl: (NSString*) deviceIdentity commandType:(NSInteger) comType buffer:(NSString*)retString length:(int)length
    charBuffer:(char *)buffer
{
    NSLog(@"WifiSettingViewController: VSNet返回数据 UID:%@,comType:%ld",deviceIdentity,(long)comType);  2 ⚠ Data argument not used by for
    NSString *string = [[NSString alloc] initWithCString:buffer encoding:NSUTF8StringEncoding];
    if ([deviceIdentity isEqualToString:m_strDID] && comType == CGI_IESET_WIFISCAN) {
        if (string == nil) {
            if (retString != nil) {
                string = retString;
            } else {
                string = [NSString stringWithFormat:@"%s",buffer];
            }
        }
        NSLog(@"无线wifi返回数据: \nUID = %@,类型 = %ld,buff = %@",deviceIdentity,(long)comType,string);
        NSInteger result = [[NSString subValueByKeyString:@"result=" fromRetString:string] integerValue];
        if (result != 0) {
            NSLog(@"数据异常!");
            return;
        }
```

(5.5) 设置设备 WIFI

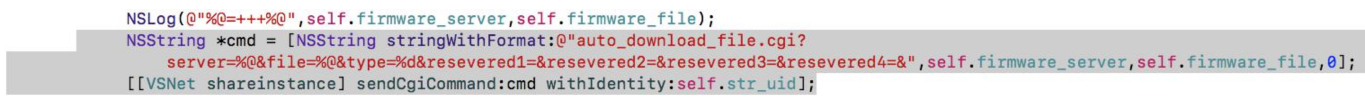


```
NSString *cmd = [NSString stringWithFormat:@"set_wifi.cgi?
    enable=1&ssid=%@&encrypt=0&defkey=0&key1=%s&key2=&key3=&key4=&authtype=%d&keyformat=0&key1_bits=0&key2_bits=0&key3_bits=0&key4_bi
    s=0&channel=%d&mode=0&wpa_psk=%s&",m_strSSID,pkey,m_security,m_channel,pwpa_psk];

NSString *sendSSID = [cmd stringByAddingPercentEscapesUsingEncoding:NSUTF8StringEncoding];

[[VSNet shareinstance] sendCgiCommand:sendSSID withIdentity:m_strDID];
[[VSNet shareinstance] setControlDelegate:m_strDID withDelegate:self];
```

6) 设备固件升级



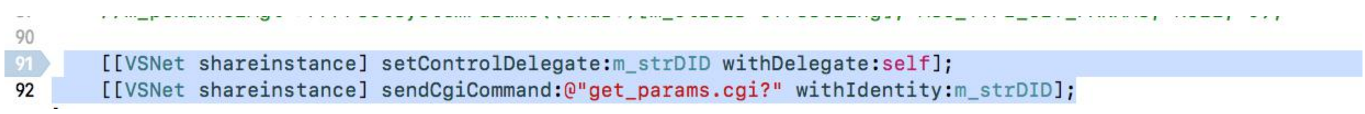
```
    NSLog(@"%@=+++%@",self.firmware_server,self.firmware_file);
    NSString *cmd = [NSString stringWithFormat:@"auto_download_file.cgi?
        server=%@&file=%@&type=%d&resevered1=&resevered2=&resevered3=&resevered4=&",self.firmware_server,self.firmware_file,0];
    [[VSNet shareinstance] sendCgiCommand:cmd withIdentity:self.str_uid];
```

7) 重启设备

```
[[VSNet shareinstance] sendCgiCommand:@"reboot.cgi?" withIdentity:strUID];
```

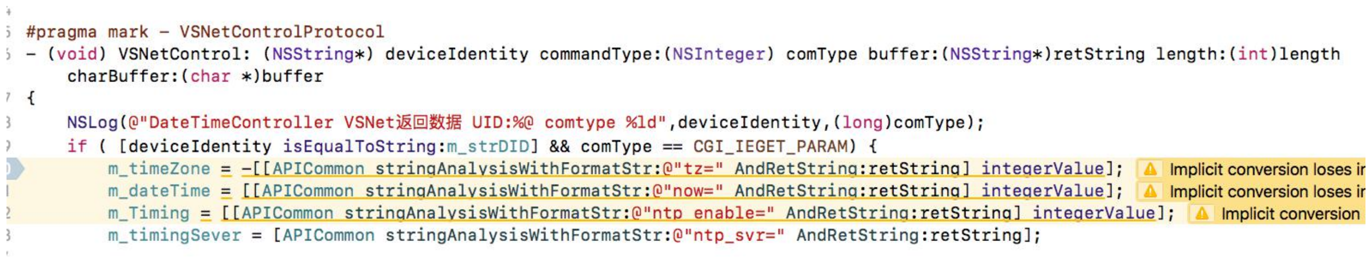
8) 设备参数

(8.1) 获取设备参数



```
    [[VSNet shareinstance] setControlDelegate:m_strDID withDelegate:self];
    [[VSNet shareinstance] sendCgiCommand:@"get_params.cgi?" withIdentity:m_strDID];
```

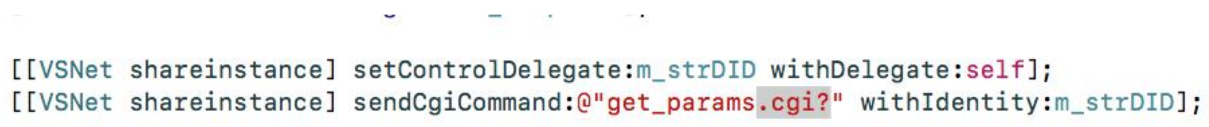
(8.2) 获取设备参数返回



```
#pragma mark - VSNetControlProtocol
- (void) VSNetControl: (NSString*) deviceIdentity commandType:(NSInteger) comType buffer:(NSString*)retString length:(int)length
    charBuffer:(char *)buffer
{
    NSLog(@"DateTimeController VSNet返回数据 UID:%@ comtype %ld",deviceIdentity,(long)comType);
    if ( [deviceIdentity isEqualToString:m_strDID] && comType == CGI_IEGET_PARAM) {
        m_timeZone = -[[APICommon stringAnalysisWithFormatStr:@"tz=" AndRetString:retString] integerValue];  ⚠ Implicit conversion loses i
        m_dateTime = [[APICommon stringAnalysisWithFormatStr:@"now=" AndRetString:retString] integerValue];  ⚠ Implicit conversion loses i
        m_Timing = [[APICommon stringAnalysisWithFormatStr:@"ntp_enable=" AndRetString:retString] integerValue];  ⚠ Implicit conversion
        m_timingSever = [APICommon stringAnalysisWithFormatStr:@"ntp_svr=" AndRetString:retString];
```

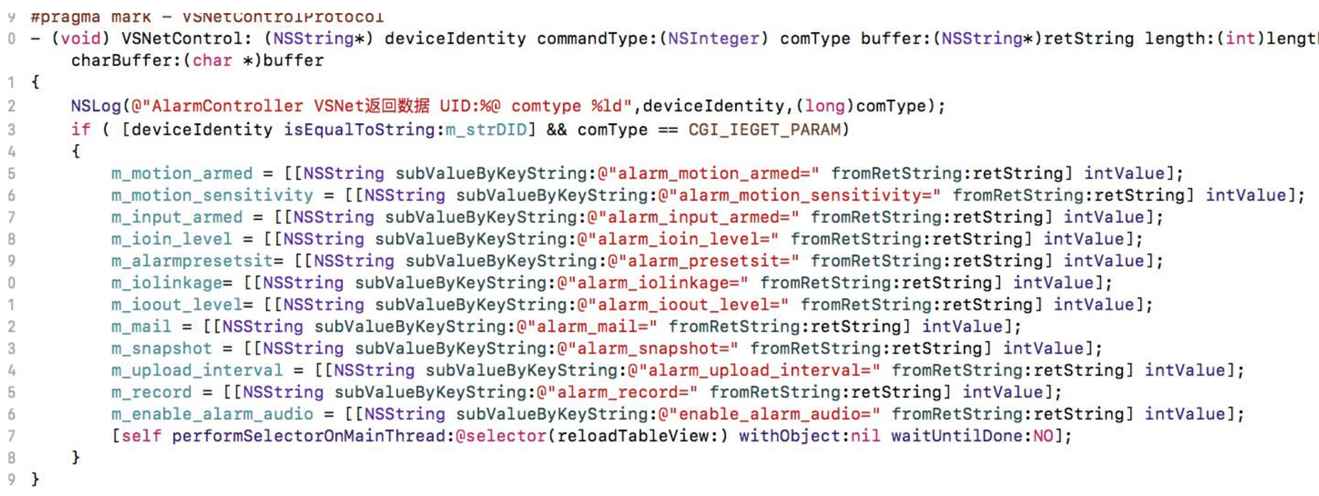
9）设备报警

(9.1）获取报警参数



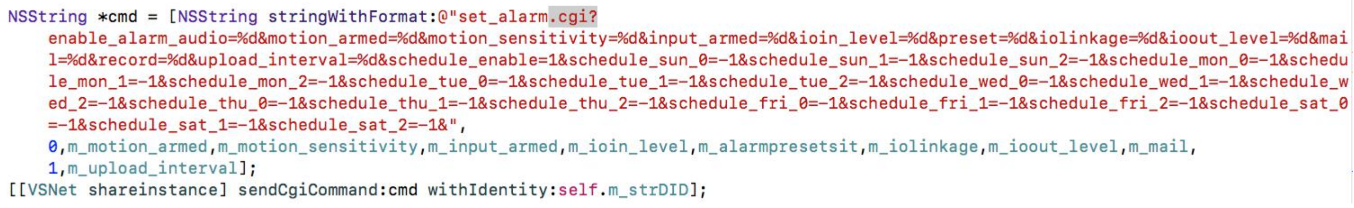
```
[[VSNet shareinstance] setControlDelegate:m_strDID withDelegate:self];
[[VSNet shareinstance] sendCgiCommand:@"get_params.cgi?" withIdentity:m_strDID];
```

(9.2）返回获取报警参数



```
#pragma mark - VSNetControlProtocol
- (void) VSNetControl: (NSString*) deviceIdentity commandType:(NSInteger) comType buffer:(NSString*)retString length:(int)length
    charBuffer:(char *)buffer
{
    NSLog(@"AlarmController VSNet返回数据 UID:%@ comtype %ld",deviceIdentity,(long)comType);
    if ( [deviceIdentity isEqualToString:m_strDID] && comType == CGI_IEGET_PARAM)
    {
        m_motion_armed = [[NSString subValueByKeyString:@"alarm_motion_armed=" fromRetString:retString] intValue];
        m_motion_sensitivity = [[NSString subValueByKeyString:@"alarm_motion_sensitivity=" fromRetString:retString] intValue];
        m_input_armed = [[NSString subValueByKeyString:@"alarm_input_armed=" fromRetString:retString] intValue];
        m_ioin_level = [[NSString subValueByKeyString:@"alarm_ioin_level=" fromRetString:retString] intValue];
        m_alarmpresetsit= [[NSString subValueByKeyString:@"alarm_presetsit=" fromRetString:retString] intValue];
        m_iolinkage= [[NSString subValueByKeyString:@"alarm_iolinkage=" fromRetString:retString] intValue];
        m_ioout_level= [[NSString subValueByKeyString:@"alarm_ioout_level=" fromRetString:retString] intValue];
        m_mail = [[NSString subValueByKeyString:@"alarm_mail=" fromRetString:retString] intValue];
        m_snapshot = [[NSString subValueByKeyString:@"alarm_snapshot=" fromRetString:retString] intValue];
        m_upload_interval = [[NSString subValueByKeyString:@"alarm_upload_interval=" fromRetString:retString] intValue];
        m_record = [[NSString subValueByKeyString:@"alarm_record=" fromRetString:retString] intValue];
        m_enable_alarm_audio = [[NSString subValueByKeyString:@"enable_alarm_audio=" fromRetString:retString] intValue];
        [self performSelectorOnMainThread:@selector(reloadTableView:) withObject:nil waitUntilDone:NO];
    }
}
```

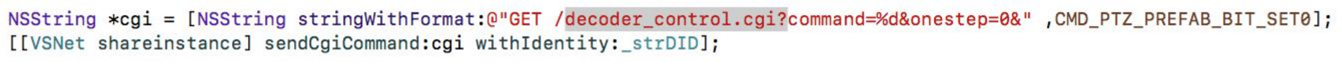
(9.3）设置报警参数



```
NSString *cmd = [NSString stringWithFormat:@"set_alarm.cgi?
    enable_alarm_audio=%d&motion_armed=%d&motion_sensitivity=%d&input_armed=%d&ioin_level=%d&preset=%d&iolinkage=%d&ioout_level=%d&mai
    l=%d&record=%d&upload_interval=%d&schedule_enable=1&schedule_sun_0=-1&schedule_sun_1=-1&schedule_sun_2=-1&schedule_mon_0=-1&schedu
    le_mon_1=-1&schedule_mon_2=-1&schedule_tue_0=-1&schedule_tue_1=-1&schedule_tue_2=-1&schedule_wed_0=-1&schedule_wed_1=-1&schedule_w
    ed_2=-1&schedule_thu_0=-1&schedule_thu_1=-1&schedule_thu_2=-1&schedule_fri_0=-1&schedule_fri_1=-1&schedule_fri_2=-1&schedule_sat_0
    =-1&schedule_sat_1=-1&schedule_sat_2=-1&",
    0,m_motion_armed,m_motion_sensitivity,m_input_armed,m_ioin_level,m_alarmpresetsit,m_iolinkage,m_ioout_level,m_mail,
    1,m_upload_interval];
[[VSNet shareinstance] sendCgiCommand:cmd withIdentity:self.m_strDID];
```

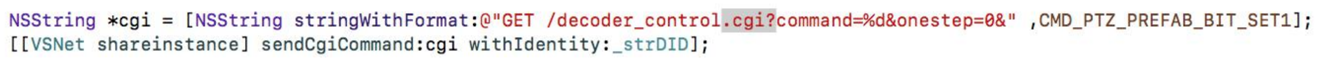
10）设备预置位

(10.1）设置设备预置位 0



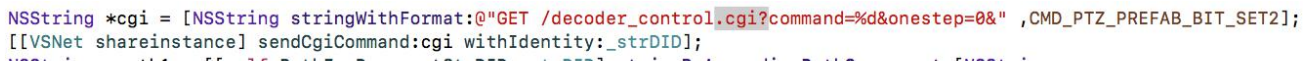
```
NSString *cgi = [NSString stringWithFormat:@"GET /decoder_control.cgi?command=%d&onestep=0&" ,CMD_PTZ_PREFAB_BIT_SET0];
[[VSNet shareinstance] sendCgiCommand:cgi withIdentity:_strDID];
```

(10.2）设置设备预置位 1



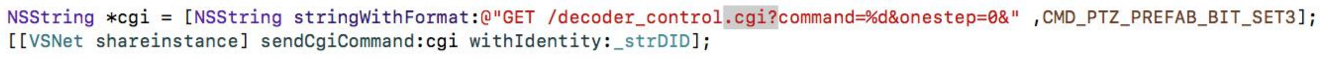
```
NSString *cgi = [NSString stringWithFormat:@"GET /decoder_control.cgi?command=%d&onestep=0&" ,CMD_PTZ_PREFAB_BIT_SET1];
[[VSNet shareinstance] sendCgiCommand:cgi withIdentity:_strDID];
```

(10.3）设置设备预置位 2



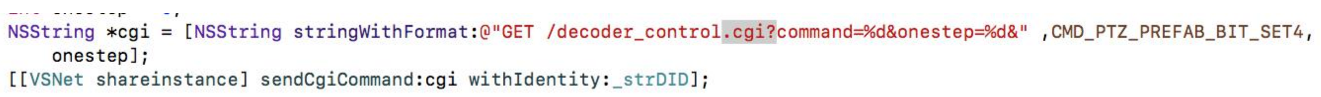
```
NSString *cgi = [NSString stringWithFormat:@"GET /decoder_control.cgi?command=%d&onestep=0&" ,CMD_PTZ_PREFAB_BIT_SET2];
[[VSNet shareinstance] sendCgiCommand:cgi withIdentity:_strDID];
```

(10.4）设置设备预置位 3



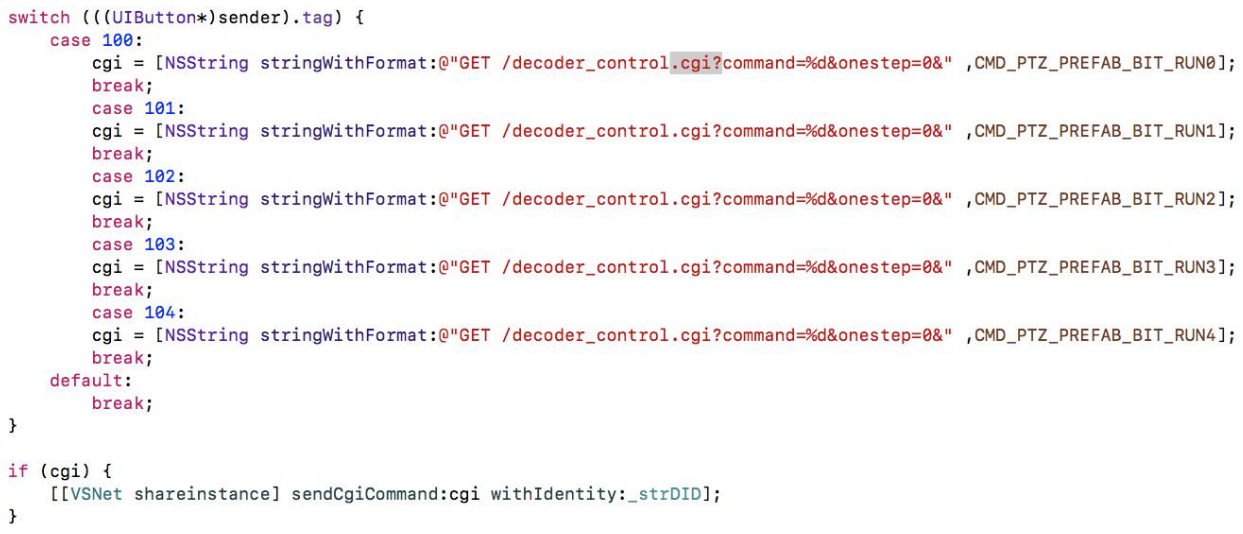
```
NSString *cgi = [NSString stringWithFormat:@"GET /decoder_control.cgi?command=%d&onestep=0&" ,CMD_PTZ_PREFAB_BIT_SET3];
[[VSNet shareinstance] sendCgiCommand:cgi withIdentity:_strDID];
```

(10.5）设置设备预置位 4



```
NSString *cgi = [NSString stringWithFormat:@"GET /decoder_control.cgi?command=%d&onestep=%d&" ,CMD_PTZ_PREFAB_BIT_SET4,
    onestep];
[[VSNet shareinstance] sendCgiCommand:cgi withIdentity:_strDID];
```

(10.6）调用设备预置位



```
switch (((UIButton*)sender).tag) {
    case 100:
        cgi = [NSString stringWithFormat:@"GET /decoder_control.cgi?command=%d&onestep=0&" ,CMD_PTZ_PREFAB_BIT_RUN0];
        break;
        case 101:
        cgi = [NSString stringWithFormat:@"GET /decoder_control.cgi?command=%d&onestep=0&" ,CMD_PTZ_PREFAB_BIT_RUN1];
        break;
        case 102:
        cgi = [NSString stringWithFormat:@"GET /decoder_control.cgi?command=%d&onestep=0&" ,CMD_PTZ_PREFAB_BIT_RUN2];
        break;
        case 103:
        cgi = [NSString stringWithFormat:@"GET /decoder_control.cgi?command=%d&onestep=0&" ,CMD_PTZ_PREFAB_BIT_RUN3];
        break;
        case 104:
        cgi = [NSString stringWithFormat:@"GET /decoder_control.cgi?command=%d&onestep=0&" ,CMD_PTZ_PREFAB_BIT_RUN4];
        break;
    default:
        break;
}

if (cgi) {
    [[VSNet shareinstance] sendCgiCommand:cgi withIdentity:_strDID];
}
```

11）云台操作

(11.1）上下巡航

```
- (IBAction) btnUpDown:(id)sender
{
    if (m_bPtzIsUpDown) {
        int onestep = 0;
        NSString *cgi = [NSString stringWithFormat:@"GET /decoder_control.cgi?command=%d&onestep=%d&" ,CMD_PTZ_UP_DOWN_STOP, onestep];
        [[VSNet shareinstance] sendCgiCommand:cgi withIdentity:strDID];

        btnUpDown.style = UIBarButtonItemStyleBordered;
        [_upDownBtn setImage:_arrowUpDownImg forState:UIControlStateNormal];
    }else {
        int onestep = 0;
        NSString *cgi = [NSString stringWithFormat:@"GET /decoder_control.cgi?command=%d&onestep=%d&" ,CMD_PTZ_UP_DOWN, onestep];
        [[VSNet shareinstance] sendCgiCommand:cgi withIdentity:strDID];
        btnUpDown.style = UIBarButtonItemStyleDone;
        [_upDownBtn setImage:_arrowUpDownImgOn forState:UIControlStateNormal];
    }
```

(11.2）左右巡航

```
- (IBAction) btnLeftRight:(id)sender
{
    if (m_bPtzIsLeftRight) {
        int onestep = 0;
        NSString *cgi = [NSString stringWithFormat:@"GET /decoder_control.cgi?command=%d&onestep=%d&" ,CMD_PTZ_LEFT_RIGHT_STOP, onestep];
        [[VSNet shareinstance] sendCgiCommand:cgi withIdentity:strDID];

        btnLeftRight.style = UIBarButtonItemStyleBordered;
        [_leftRightBtn setImage:_arrowLeftRightImg forState:UIControlStateNormal];
    }else {
        int onestep = 0;
        NSString *cgi = [NSString stringWithFormat:@"GET /decoder_control.cgi?command=%d&onestep=%d&" ,CMD_PTZ_LEFT_RIGHT, onestep];
        [[VSNet shareinstance] sendCgiCommand:cgi withIdentity:strDID];

        btnLeftRight.style = UIBarButtonItemStyleDone;
        [_leftRightBtn setImage:_arrowLeftRightImgOn forState:UIControlStateNormal];
    }
```

12）图像传感器参数控制

(12.1)翻转与镜像

```
NSString *cmd = [NSString stringWithFormat:@"camera_control.cgi?param=5&value=%d&",value];
[[VSNet shareinstance] sendCgiCommand:cmd withIdentity:strDID];
```

(12.2)亮度

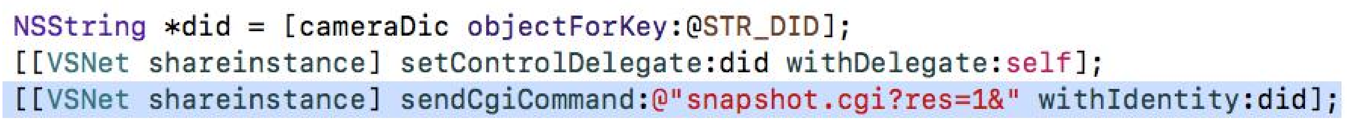
```
int f = sliderBrightness.value;
NSString *cmd = [NSString stringWithFormat:@"camera_control.cgi?param=1&value=%d&",f];
[[VSNet shareinstance] sendCgiCommand:cmd withIdentity:strDID];
```

(12.3)对比度

```
int f = sliderContrast.value;
NSString *cmd = [NSString stringWithFormat:@"camera_control.cgi?param=2&value=%d&",f];
[[VSNet shareinstance] sendCgiCommand:cmd withIdentity:strDID];
```

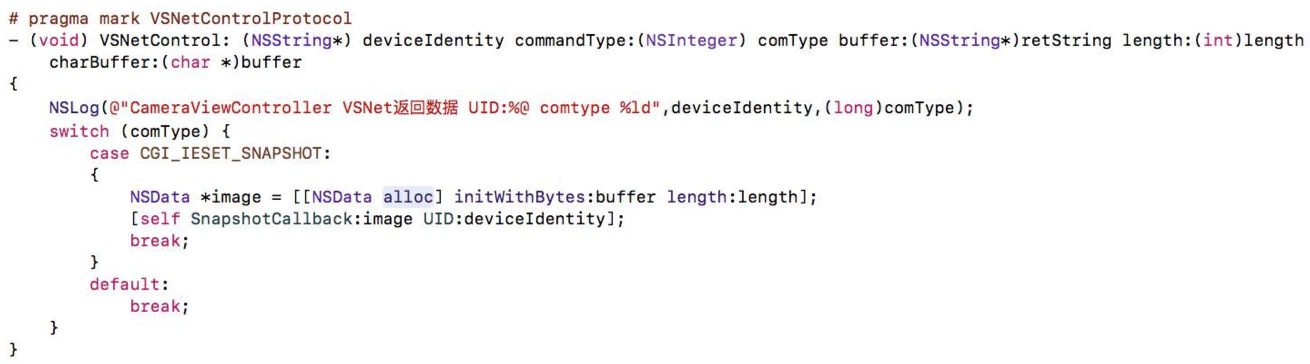
13）设备截图

(13.1）获取设备截图



```
NSString *did = [cameraDic objectForKey:@STR_DID];
[[VSNet shareinstance] setControlDelegate:did withDelegate:self];
[[VSNet shareinstance] sendCgiCommand:@"snapshot.cgi?res=1&" withIdentity:did];
```

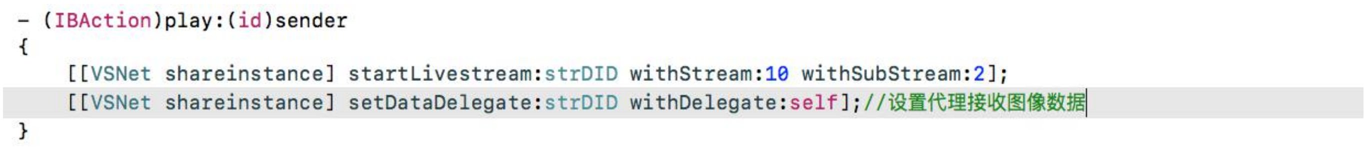
（13.2）获取设备截图返回



```
# pragma mark VSNetControlProtocol
- (void) VSNetControl: (NSString*) deviceIdentity commandType:(NSInteger) comType buffer:(NSString*)retString length:(int)length
    charBuffer:(char *)buffer
{
    NSLog(@"CameraViewController VSNet返回数据 UID:%@ comtype %ld",deviceIdentity,(long)comType);
    switch (comType) {
        case CGI_IESET_SNAPSHOT:
        {
            NSData *image = [[NSData alloc] initWithBytes:buffer length:length];
            [self SnapshotCallback:image UID:deviceIdentity];
            break;
        }
        default:
            break;
    }
}
```

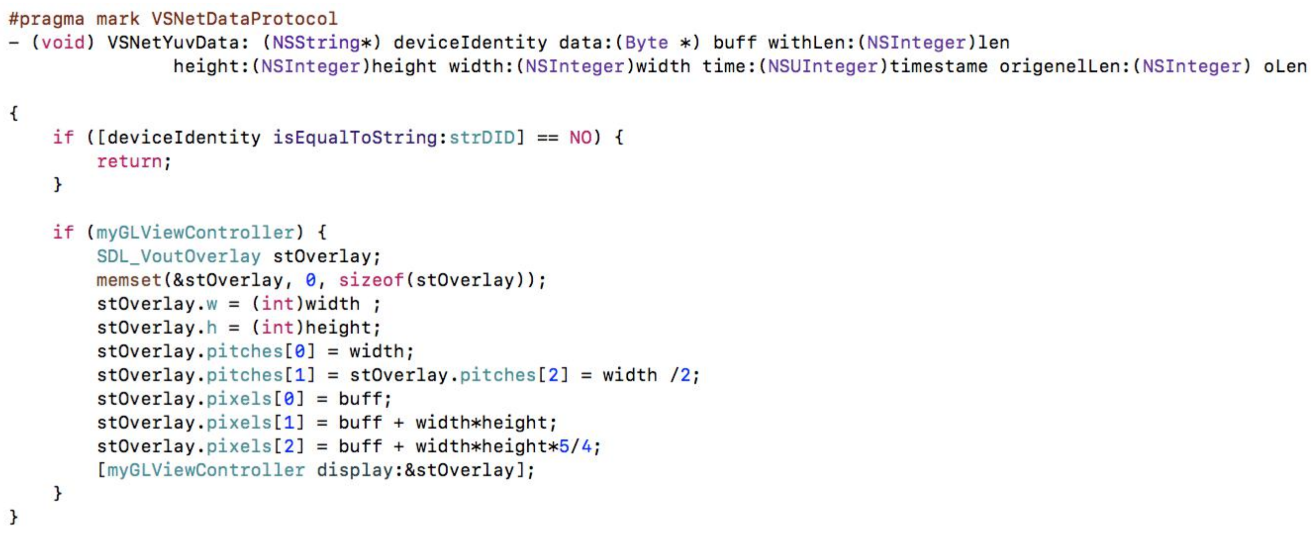
3.预览视频

1)开启预览视频



```
- (IBAction)play:(id)sender
{
    [[VSNet shareinstance] startLivestream:strDID withStream:10 withSubStream:2];
    [[VSNet shareinstance] setDataDelegate:strDID withDelegate:self];//设置代理接收图像数据
}
```

2)开启预览视频

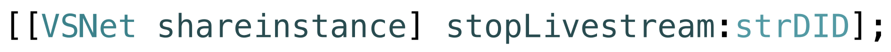


```
#pragma mark VSNetDataProtocol
- (void) VSNetYuvData: (NSString*) deviceIdentity data:(Byte *) buff withLen:(NSInteger)len
                height:(NSInteger)height width:(NSInteger)width time:(NSUInteger)timestame origenelLen:(NSInteger) oLen

{
    if ([deviceIdentity isEqualToString:strDID] == NO) {
        return;
    }

    if (myGLViewController) {
        SDL_VoutOverlay stOverlay;
        memset(&stOverlay, 0, sizeof(stOverlay));
        stOverlay.w = (int)width ;
        stOverlay.h = (int)height;
        stOverlay.pitches[0] = width;
        stOverlay.pitches[1] = stOverlay.pitches[2] = width /2;
        stOverlay.pixels[0] = buff;
        stOverlay.pixels[1] = buff + width*height;
        stOverlay.pixels[2] = buff + width*height*5/4;
        [myGLViewController display:&stOverlay];
    }
}
```

3)关闭预览视频



```
[[VSNet shareinstance] stopLivestream:strDID];
```

4.监听声音

1）开启监听

```
[[VSNet shareinstance] startAudio:strDID withEchoCancellationVer:NO];
```

2）停止监听

```
[[VSNet shareinstance] stopAudio:strDID];
```

5.对讲

1）开启对讲

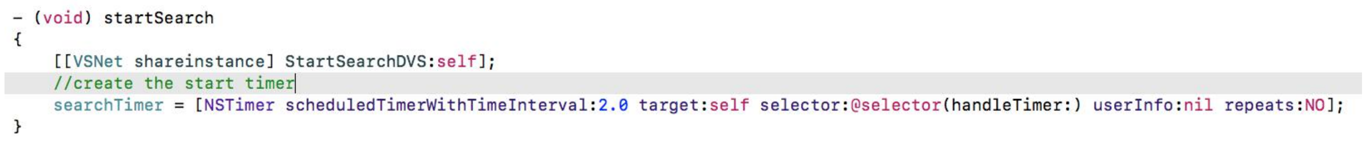
```
[[VSNet shareinstance] startTalk:strDID withEchoCancellationVer:NO];
```

2）停止对讲

```
[[VSNet shareinstance] stopTalk:strDID];
```

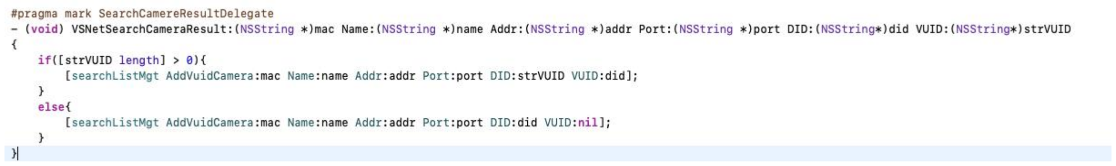
6.局域网内搜索在线设备

1）开始搜索



```
- (void) startSearch
{
    [[VSNet shareinstance] StartSearchDVS:self];
    //create the start timer
    searchTimer = [NSTimer scheduledTimerWithTimeInterval:2.0 target:self selector:@selector(handleTimer:) userInfo:nil repeats:NO];
}
```

2）搜索到设备回调　（20191201 更新方法）



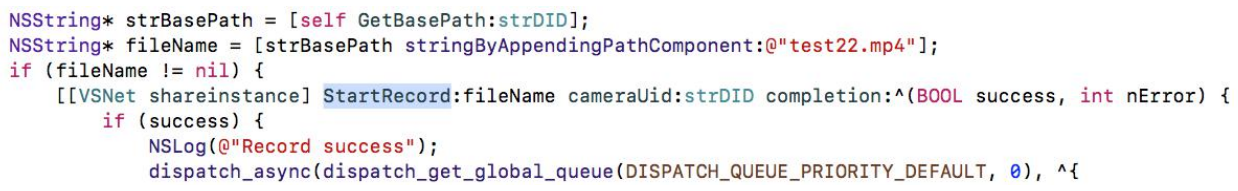
```
#pragma mark SearchCamereResultDelegate
- (void) VSNetSearchCameraResult:(NSString *)mac Name:(NSString *)name Addr:(NSString *)addr Port:(NSString *)port DID:(NSString*)did VUID:(NSString*)strVUID
{
    if([strVUID length] > 0){
        [searchListMgt AddVuidCamera:mac Name:name Addr:addr Port:port DID:strVUID VUID:did];
    }
    else{
        [searchListMgt AddVuidCamera:mac Name:name Addr:addr Port:port DID:did VUID:nil];
    }
}
```

3）停止搜索

```
[[VSNet shareinstance] StopSearchDVS];
```

7.录制预览视频

1）开始录制预览视频



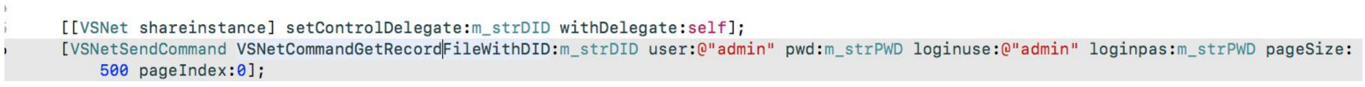
```
NSString* strBasePath = [self GetBasePath:strDID];
NSString* fileName = [strBasePath stringByAppendingPathComponent:@"test22.mp4"];
if (fileName != nil) {
    [[VSNet shareinstance] StartRecord:fileName cameraUid:strDID completion:^(BOOL success, int nError) {
        if (success) {
            NSLog(@"Record success");
            dispatch_async(dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0), ^{
```

2）停止录制视频

```
[[VSNet shareinstance] StopCameraUid:strDID];
```

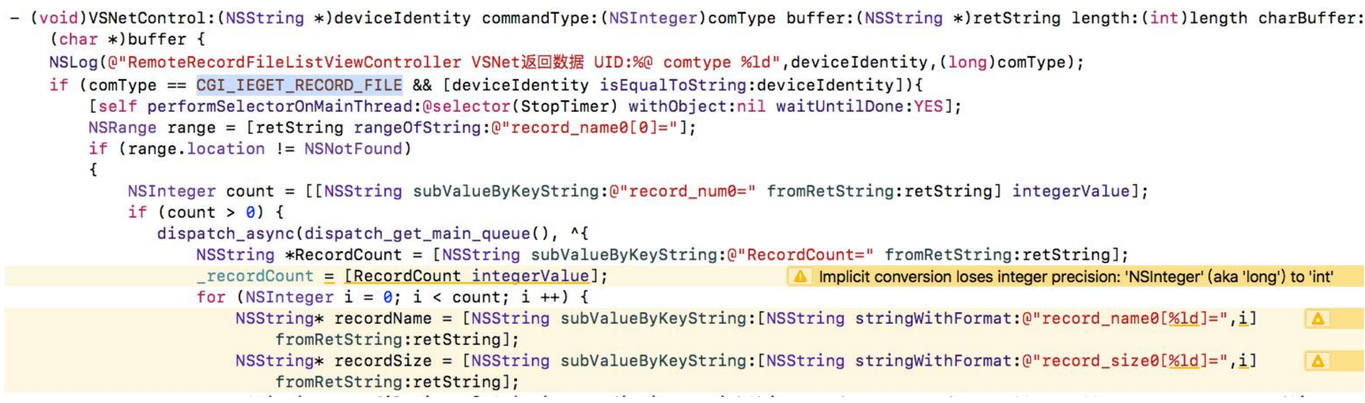
8.SD 卡录像

1）获取 SD 卡录像文件列表



```
    [[VSNet shareinstance] setControlDelegate:m_strDID withDelegate:self];
    [VSNetSendCommand VSNetCommandGetRecordFileWithDID:m_strDID user:@"admin" pwd:m_strPWD loginuse:@"admin" loginpas:m_strPWD pageSize:
        500 pageIndex:0];
```

2）返回获取 SD 卡录像文件列表



```
- (void)VSNetControl:(NSString *)deviceIdentity commandType:(NSInteger)comType buffer:(NSString *)retString length:(int)length charBuffer:
    (char *)buffer {
    NSLog(@"RemoteRecordFileListViewController VSNet返回数据 UID:%@ comtype %ld",deviceIdentity,(long)comType);
    if (comType == CGI_IEGET_RECORD_FILE && [deviceIdentity isEqualToString:deviceIdentity]){
        [self performSelectorOnMainThread:@selector(StopTimer) withObject:nil waitUntilDone:YES];
        NSRange range = [retString rangeOfString:@"record_name0[0]="];
        if (range.location != NSNotFound)
        {
            NSInteger count = [[NSString subValueByKeyString:@"record_num0=" fromRetString:retString] integerValue];
            if (count > 0) {
                dispatch_async(dispatch_get_main_queue(), ^{
                    NSString *RecordCount = [NSString subValueByKeyString:@"RecordCount=" fromRetString:retString];
                    _recordCount = [RecordCount integerValue];        ⚠ Implicit conversion loses integer precision: 'NSInteger' (aka 'long') to 'int'
                    for (NSInteger i = 0; i < count; i ++) {
                        NSString* recordName = [NSString subValueByKeyString:[NSString stringWithFormat:@"record_name0[%ld]=",i]     ⚠
                            fromRetString:retString];
                        NSString* recordSize = [NSString subValueByKeyString:[NSString stringWithFormat:@"record_size0[%ld]=",i]     ⚠
                            fromRetString:retString];
```

3）播放 SD 卡录像文件

```
[[VSNet shareinstance] startPlayBack:strDID fileName:m_strFileName withOffset:0 fileSize:_record_Size delegate:self SupportHD:1];
```

4）停止播放 SD 卡录像文件

```
[[VSNet shareinstance] stopPlayBack:strDID];
```

9.H265 转 mp4

1.StratMergeH265File 合并下载文件接口

```
/**
 * 开始合并视频文件 H265
 * @param deviceIdentity 设备 id
 * @param strInPath       输入文件(单个文件)
 * @param strOutPath      输出文件路径
 *@param nCount         输入文件个数(总文件个数),第一个文件用 strInPath 参数传路径,如有其它文件使用 PutFile
来传。
 * @param delegate       进度回调代理
 * @return 1 成功, 0 失败
 */
-(int) StratMergeH265File:(NSString*)deviceIdentity InputPath:(NSString*)strInPath
OutPath:(NSString*)strOutPath FileCount:(int)nCount Delegate:(id <MergerVideoProtocol>)
delegate;
```

2.PutFile 往合并文件队列中添加文件

```
/**
 * 往合并视频列表中添加文件(如果是只有一个文件合并的此接口就不用调用了, StratMergeH265 中已经传了)
 * @param deviceIdentity 设备 id
 * @param strInPath       输入文件(单个文件)
 * @return 1 成功, 0 失败
 */
-(int) PutFile:(NSString*)deviceIdentity InputPath:(NSString*)strInPath;
```

3.GetMergeMP4FilePos 获取进度

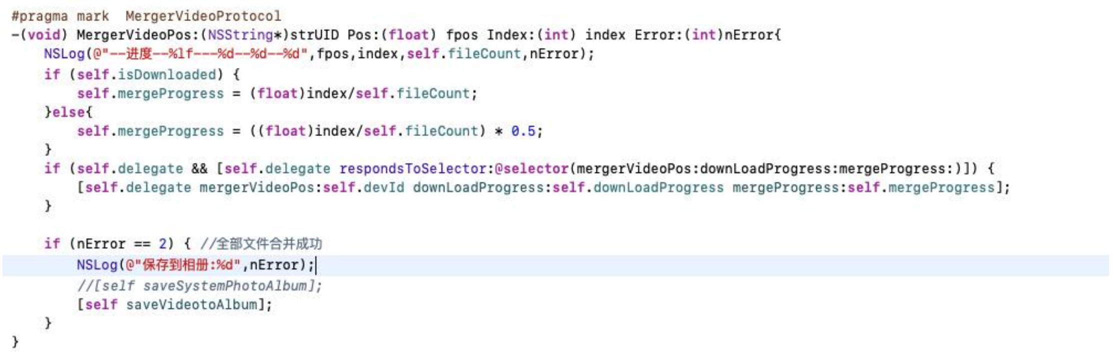
```
/**
 * 获取合并文件的合并文件进度
 * @param deviceIdentity 设备 id
 * @return 0~1.0 进度  -1 代表错误
 */
-(float) GetMergeMP4FilePos:(NSString*)deviceIdentity;
```

4.StopMergeMP4File 中断或停合并下载

```
/**
 * 中断合并
 * @param deviceIdentity 设备 id
 * @return 1 成功, 0 失败
 */
-(int) StopMergeMP4File:(NSString*)deviceIdentity;
```

5.合并下载结果与进度返回

```
/**
 合并视频回调进度
 @param strUID: 设备 UID
 @param fpos:   0~1.0 进度(某个文件的进度)
 @param index   文件索引(第几个文件)
 @param nError  0:有错误 1:无错误 2:全部文件合并成功
 **/
-(void) MergerVideoPos:(NSString*)strUID Pos:(float) fpos Index:(int) index Error:(int)nError;
```



```objc
#pragma mark  MergerVideoProtocol
-(void) MergerVideoPos:(NSString*)strUID Pos:(float) fpos Index:(int) index Error:(int)nError{
    NSLog(@"--进度--%lf---%d--%d--%d",fpos,index,self.fileCount,nError);
    if (self.isDownloaded) {
        self.mergeProgress = (float)index/self.fileCount;
    }else{
        self.mergeProgress = ((float)index/self.fileCount) * 0.5;
    }
    if (self.delegate && [self.delegate respondsToSelector:@selector(mergerVideoPos:downLoadProgress:mergeProgress:)]) {
        [self.delegate mergerVideoPos:self.devId downLoadProgress:self.downLoadProgress mergeProgress:self.mergeProgress];
    }

    if (nError == 2) { //全部文件合并成功
        NSLog(@"保存到相册:%d",nError);
        //[self saveSystemPhotoAlbum];
        [self saveVideotoAlbum];
    }
}
```