ZigBee Attribute Report 机制分析

(shaozhong.liang@nxp.com)

ZigBee3.0 提供了 attribute report 机制，可以实现 attribute 属性的 report 功能，提供了一种服务端和客户端数据同步的机制。

下面以$sdk\JN-SW-4170 和 JN-AN-1220-Zigbee-3-0-Sensors\OccupancySensor 为例子，分析 report attribute 的实现过程，具体步骤如下：

文件 JN-AN-1220-Zigbee-3-0-Sensors\OccupancySensor\Source\app_zlo_sensor_node.c 的 APP_ZCL_vInitialise 函数初始化 Report Manager 管理器。

APP_ZCL_vInitialise
→eZCL_Initialise
→eZCL_CreateZCL
→eZCL_CreateOptionalManagers
→eZCL_CreateReportManager

```
#define ZCL_NUMBER_OF_REPORTS       3
#define ZLO_MIN_REPORT_INTERVAL     1
#define ZLO_MAX_REPORT_INTERVAL     0x3d
```

$sdk\JN-SW-4170\Components\ZCIF\Source\zcl_library_options.c

```
PUBLIC teZCL_Status eZCL_CreateOptionalManagers(
                uint8           u8NumberOfReports,
                uint16          u16SystemMinimumReportingInterval,
                uint16          u16SystemMaximumReportingInterval)
{
    teZCL_Status eStatus = E_ZCL_SUCCESS;
#ifdef ZCL_ATTRIBUTE_REPORTING_SERVER_SUPPORTED
    // Create Reports
    eStatus    = eZCL_CreateReportManager(u8NumberOfReports, u16SystemMinimumReportingInterval, u16SystemMaximumReportingInterval);
    if(eStatus != E_ZCL_SUCCESS)
    {
        return eStatus;
    }
#endif
    return eStatus;
 }

PUBLIC    teZCL_Status eZCL_CreateReportManager(
                    uint8           u8NumberOfReports,
                    uint16          u16SystemMinimumReportingInterval,
                    uint16          u16SystemMaximumReportingInterval)
{
    int i;
    // create report record array
    vZCL_HeapAlloc(psZCL_Common->psReportRecord,
                    tsZCL_ReportRecord,
                    (sizeof(tsZCL_ReportRecord))*u8NumberOfReports,
                    TRUE,
                    "Report_Record");

    if (psZCL_Common->psReportRecord == NULL)
    {
        return E_ZCL_ERR_HEAP_FAIL;
    }

    /* initialise lists */
    vDLISTinitialise(&psZCL_Common->lReportAllocList);
    vDLISTinitialise(&psZCL_Common->lReportDeAllocList);

    psZCL_Common->u8NumberOfReports = u8NumberOfReports;
    psZCL_Common->u16SystemMinimumReportingInterval = u16SystemMinimumReportingInterval;
    psZCL_Common->u16SystemMaximumReportingInterval = u16SystemMaximumReportingInterval;
```

```
    // add timer click function to ZCL
    if(eZCL_TimerRegister(E_ZCL_TIMER_CLICK_MS, 0, vReportTimerClickCallback)!= E_ZCL_SUCCESS)
    {
        return(E_ZCL_FAIL);
    }

    // initialise structure
    for(i=0;i<u8NumberOfReports;i++)
    {
        /* add all header slots to the to free list */
        vDLISTaddToHead(&psZCL_Common->lReportDeAllocList, (DNODE *)&psZCL_Common->psReportRecord[i]);
        // initialise
    }
    return(E_ZCL_SUCCESS);
}
```

在函数 eZCL_CreateReportManager 中根据 ZCL_NUMBER_OF_REPORTS 宏定义值，分配 lReportAllocList 链表空间，并注册定时器回调函数 vReportTimerClickCallback。这个 Timer 回调函数将周期检查 lReportAllocList 链表是否需要 report attribute 属性信息。

vReportTimerClickCallback

→vReportSchedulerUpdate

sdk\JN-SW-4170\Components\ZCIF\Source\zcl_reportScheduler.c

```
PRIVATE void vReportSchedulerUpdate(uint32 u32UTCTime)
{
    ……..
    psHeadRecord = (tsZCL_ReportRecord *)psDLISTgetHead(&psZCL_Common->lReportAllocList);
    if(psHeadRecord==NULL)
    {
        return;
    }
    while(psHeadRecord)
     {
        ……
        / * report attribute information to gateway */
        /* go the next record */
        // get next list member
        psHeadRecord = (tsZCL_ReportRecord *)psDLISTgetNext((DNODE *)psHeadRecord);
    }
}
```

在 OccupancySensor 应用代码中，有一个 default Report 的配置数组，在系统启动后，将会恢复默认的 report attribute 属性。

vMakeSupportedAttributesReportable

→eZCL_CreateLocalReport

→eZCLAddReport

→vDLISTaddToTail(&psZCL_Common->lReportAllocList, (DNODE *)psHeadReportRecord);

```
/* define the default reports */
tsReports asDefaultReports[ZCL_NUMBER_OF_REPORTS] = \
{\
    {MEASUREMENT_AND_SENSING_CLUSTER_ID_OCCUPANCY_SENSING,{0, E_ZCL_BMAP8, E_CLD_OS_ATTR_ID_OCCUPANCY, ZLO_MIN_REPORT_INTERVAL, ZLO_MAX_REPORT_INTERVAL, 0, {0}}},\
};
```

```
/*****************************************************************************/
PUBLIC void APP_vInitialiseNode(void)
{
.....
.....
    /* Restore any report data that is previously saved to flash */
    eStatusReportReload = eRestoreReports();
    uint16 u16ByteRead;
    PDM_eReadDataFromRecord(PDM_ID_APP_SENSOR,
                            &sDeviceDesc,
                            sizeof(tsDeviceDesc),
                            &u16ByteRead);

    /* Set security state */
    ZPS_vDefaultKeyInit();

    /* Initialize ZBPro stack */
    ZPS_eAplAfInit();

    DBG_vPrintf(TRACE_SENSOR_NODE, "\nAPP Sensor Node: ZPS_eAplAfInit");

    APP_ZCL_vInitialise();

    /* Set end device age out time to 11 days 9 hours & 4 mins */
    ZPS_bAplAfSetEndDeviceTimeout(ZED_TIMEOUT_16384_MIN);

    /*Load the reports from the PDM or the default ones depending on the PDM load record status*/
    if(eStatusReportReload !=PDM_E_STATUS_OK )
    {
        /*Load Defaults if the data was not correct*/
        vLoadDefaultConfigForReportable();
    }
    /*Make the reportable attributes */
    vMakeSupportedAttributesReportable();
```

当应用的收到网关下发的 Configure Reporting 命令后，将会 vZCL_HandleConfigureReportingCommand 函数，增加 report attribute 节点到链表中。

vZCL_HandleConfigureReportingCommand

→eZCLAddReport

→vDLISTaddToTail(&psZCL_Common->lReportAllocList, (DNODE *)psHeadReportRecord);

Stack	Layer	Packet Information	MAC Src.	MAC D
ZigBee	MAC	Data Request	0xD99A	0x000(
ZigBee	MAC	Acknowledgement		
ZigBee	ZCL	Power Configuration: Configure Reporting	0x0000	0xD99A
ZigBee	MAC	Acknowledgement		
ZigBee	APS	Acknowledgement	0xD99A	0x000(
ZigBee	MAC	Acknowledgement		
ZigBee	ZCL	Power Configuration: Configure Reporting…	0xD99A	0x000(
ZigBee	MAC	Acknowledgement		
ZigBee	MAC	Data Request	0xD99A	0x000(
ZigBee	MAC	Acknowledgement		
ZigBee	ZCL	Power Configuration: Report Attributes	0xD99A	0x000(
ZigBee	MAC	Acknowledgement		
ZigBee	MAC	Data Request	0xD99A	0x000(
ZigBee	MAC	Acknowledgement		

```
▷ Frame Information: (57 bytes)
▷ MAC Header: (9 bytes)
◢ MAC Payload: (46 bytes)
    ▷ NWK Header: 0x361E0000D99A0248
    ▷ NWK Aux Header: (14 bytes)
    ◢ NWK Payload: (20 bytes)
        ▷ APS Header: 0xCF01010400010140
        ◢ APS Payload: (12 bytes)
            ▷ ZCL Header: 0x061D00
            ◢ ZCL Payload: (9 bytes)
                ◢ Attributes 0: (9 bytes)
                      Direction: [0x00] Reported
                      Attribute ID: [0x0021] Battery Percentage Remaining
                      Attribute Data Type: [0x20] Unsigned 8-bit Integer
                      Minimum Reporting Interval: 10
                      Maximum Reporting Interval: 7200
                      Reportable Change: 0x01
```

更详细的细节请参考 JN-UG-3115 ZigBee Cluster Library(ZigBee3.0)附件 B. Attribute Reporting 相关章节。