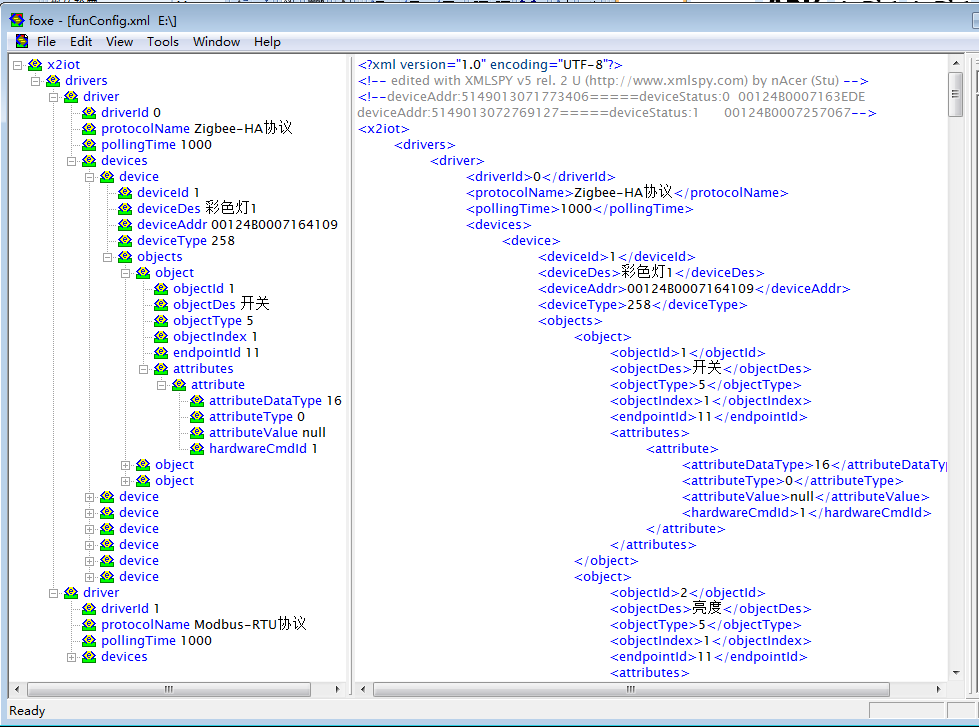
网关中“设备单品的功能”配置文件 说明



如图左侧：

1. x2iot

配置的最外层，里面包含了各种类型的设备

1. drivers

注：所有的驱动，里面包含了多种类型的驱动，如zgb，modbus等，一种驱动一个driver节点来概括，有多个driver则表示多个驱动，每一种驱动的区分看driver节点里的各个参数

1. driver

driverId：驱动id，网关用来区分这是那种驱动

|  |  |
| --- | --- |
| driverId | 驱动 |
| 0 | Zgb |
| 1 | Modbus |
| 2 | Bacnet |
|  |  |
|  |  |

1. protocolName

驱动的名字（该类型驱动的文字说明）

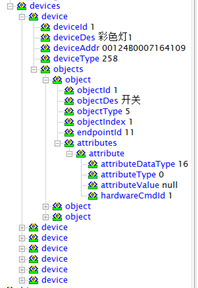
1. pollingTime

暂没用上,无说明

三. devices

注：在对应的驱动下，里面包含了该驱动类型的所有设备，如外层的driverId为0时，这里面将包含所有的zgb设备

多个设备视图 单个设备视图

1. deviceId

设备id，每个设备有一个，且在同一类型驱动下不能重复，如果重复了，网关只会识别最后那个,一个设备只需建立一个设备id,该设备的功能就是device里的object

2. deviceDes

设备说明，这个会在工艺图和网关入库界面上出现，同类型的设备，名字最好也不要一样，比如区别开来就像：彩色灯1，彩色灯2

3. deviceAddr

设备物理地址，这个是唯一的，每个硬件的外壳上，都应该贴上唯一的物理地址，如果这里留空，配置的这个设备将作废

4. deviceType

设备类型

|  |  |
| --- | --- |
| 设备 | DeviceType |
| di /do， ai/ao | 6（两种都是6，所以deviceDes设备名字的时候要谨慎） |
| 中继器 | 8 |
| 电表 | 83 |
| 温度传感器 | 770 |
| 压力传感器 | 773 |
|  |  |

四. Objects

注：包含了一种设备的所有功能，一个功能为一个<object></object>，如电表有多个功能（频率，a相电压，b相电压，c相电压等，则每个功能就是一个object）



1. objectId

id，每个功能对象的id都是唯一的，且在同一个设备下不能重复，如果重复了，网关只会识别最后那个

2. objectDes

功能对象说明，就好比这个功能的名字，如果随便起名，比如起重复了，在显示界面上如果区分不开，会让使用者混淆

3. objectType

功能类型：

|  |  |  |
| --- | --- | --- |
| 类型（名字） | 数值 | 说明 |
| OBJECT\_AI | 1 | 模拟输入 （只读） 可以是任何数值 |
| OBJECT\_AO | 2 | 模拟输出 （可读写） 可以是任何数值 |
| OBJECT\_AV | 3 | 模拟数值 （可读写）可以是任何数值 |
| OBJECT\_BI | 4 | 数字输入 （只读） 0 、1 |
| OBJECT\_BO | 5 | 数字输出 （可读写） 0 、1 |
| OBJECT\_BV | 6 | 数字数值 （可读写） 0 、1 |
| OBJECT\_FILE | 7 | 文件 |
| OBJECT\_UNKNOWN | 255 | 无效 |

客户端和网关会根据这个类型去判断那个是可读，还是可读可写，输入的数值类型是什么，比如di是只读且为数字输入（则这个值填4），do是可读可写且为数字输出（填5）

4. deviceIndex

5. endpointid

端口id，这个是读写（控制）这个功能需要的变量，不同功能的endpointid不一样

|  |  |  |
| --- | --- | --- |
| 设备 | 功能 | 端点 |
| 电表 | 所有的功能 | 8 |
| 温度床感器 | 环境温度 | 8 |
| 压力传感器 | 环境压力 | 8 |
| ai/ao模块 | Ai1 | 9 |
|  | Ai2 | 10 |
|  | Ao1 | 11 |
|  |  |  |
| di/do | Di1 | 9 |
|  | Di2 | 10 |
|  | Di3 | 11 |
|  | Di4 | 12 |
|  | Di5 | 13 |
|  | Di6 | 14 |
|  | Do1 | 15 |
|  | Do2 | 16 |
|  | Do3 | 17 |
|  |  |  |

五. Attributes

注：包含了设备某个功能里的所有属性，一个属性为一个<attribute></attribute>

1.<attribute></attribute>

表示一个属性节点

2. attributeDataType

属性数据类型，如是布尔值类型（0、1），浮点类型（0.01），都有一个代号，目前使用的有

1）整形

|  |  |  |
| --- | --- | --- |
| 类型名称 | 代号 | 使用 |
| DATATYPE\_UINT8 | 0x20 = 32 |  |

2）单精度 浮点

|  |  |  |
| --- | --- | --- |
| 类型名称 | 代号 | 使用 |
| DATATYPE\_SINGLE\_PREC | 0x39 = 57 | 压力传感器：环境压力 |
|  |  | 温度传感器：环境温度 |
|  |  | Modbus：频率 |
|  |  | Modbus：总的有功电能 |
|  |  | Bacnet：电压、电流等 |

，

3）无符号16位整形

|  |  |  |
| --- | --- | --- |
| 类型名称 | 代号 | 使用 |
| DATATYPE\_INT16 | 0x29 = 41 | ai/ao模块：ai（模拟输入） |
|  |  | ai/ao模块：ao（模拟输入、输出） |

4）布尔 只有0 或 1

|  |  |  |
| --- | --- | --- |
| 类型名称 | 代号 | 使用 |
| DATATYPE\_ BOOLEAN | 0x10 = 16 | di/do模块：di （数字输入） |
|  |  | di/do模块：do（数字输入、输出） |
|  |  |  |

3． attributeType

当前的属性值 的 类型，，现在基本都填0，“0” 表示这个属性的当前值，暂时还没有其他的类型，以后可以扩展

1. attributeValue

不需要填值，但这个 < attributeValue>< /attributeValue>必须有

1. hardwareCmdId

只作用于zgb 设备

当前属性要执行的命令，具体查看命令功能表(173\_device.xml),在表里找到对应的设备类型,找到对应的功能,其中appCmdId值就是hardwareCmdId要绑定的值

1. attributeValueMinMax

这个属性是否有最大最小值，表示只有abjectType 为2 、3、5、6（表示在界面上可控）的才有这个值，表现为< attributeValueMinMax>0,100</ attributeValueMinMax>，即最小输入值为0，最大输入值为100

1. 用于modbus设备

< registerAddr>4001</ registerAddr > 寄存器地址

< registerType>3</ registerType > 寄存器类型

寄存器类型，registerType 四种：0,1,3,4。

1. 对应 “线圈状态”（readCoils ）

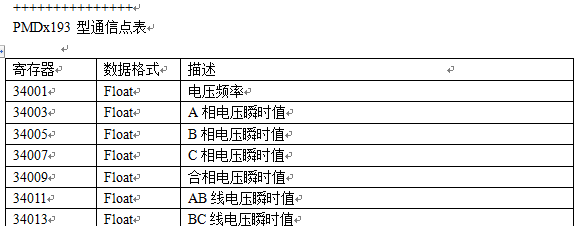
1 对应 “离散量输入”（ readDiscrInps）

3 对应 “输入寄存器”（readInpRegs）

4 对应 “保持寄存器“（readHoldRegs）

四种类型都可读,而其中0,3 类型为可写可读

在厂家提供的 点表(可以以文件夹下的PMDX193.DOC为例子) 里可以看到比如：



比如第一个34001，3表示寄存器类型，4001表示寄存地址，读（或写）的数据类型为（float）浮点型，下面为一个配置“电压频率”的例子：比如有个电表的设备地址为 1，

在点表中找到 电压频率的相关数据 ：34001 float(57表示浮点型，看上面的attributeDataType)说明

所以<attributeDataType>57</attributeDataType>  
 <registerAddr>4001</registerAddr>  
 <registerType>3</registerType>

完整的如下

<driver>  
 <driverId>1</driverId>  
 <protocolName>Modbus-RTU协议</protocolName>  
 <pollingTime>1000</pollingTime>  
 <devices>  
 <device>  
 <deviceId>1</deviceId>  
 <deviceAddr>1</deviceAddr>  
 <deviceDes>MODBUS智能电表1</deviceDes>  
 <objects>  
 <object>  
 <objectId>1</objectId>  
 <objectDes>频率</objectDes>  
 <objectType>1</objectType>  
 <objectIndex>1</objectIndex>  
 <attributes>  
 <attribute>  
 <attributeDataType>57</attributeDataType>  
 <attributeType>0</attributeType>  
 <attributeValue>null</attributeValue>  
 <registerAddr>4001</registerAddr>  
 <registerType>3</registerType>  
 </attribute>  
 </attributes>  
 </object>  
 </objects>  
 </device>  
 </devices>  
</driver>

1. 用于bacnet设备

< instanceType >1</ instanceType > 实例类型   
< instanceIndex>25</ instanceIndex >实例索引

instanceType 类型 有6大类

|  |  |  |
| --- | --- | --- |
| 类型（名字） | 数值 | 说明 |
| AI | 1 | 模拟输入 （只读） 可以是任何数值 |
| AO | 2 | 模拟输出 （可读写） 可以是任何数值 |
| AV | 3 | 模拟数值 （可读写）可以是任何数值 |
| BI | 4 | 数字输入 （只读） 0 、1 |
| BO | 5 | 数字输出 （可读写） 0 、1 |
| BV | 6 | 数字数值 （可读写） 0 、1 |

instanceIndex 实例索引 0 – 65535,这是根据厂家提供的标准

比如0绑定 哪种数据,1表示哪种数据,下面是一个用0绑定了一个电压的数据

