Spider做网关实现协议转化

现实项目中会遇到需要将不同协议的设备接入的情况，一般有两种做法：一种是在上层系统内部直接做不同协议适配器；另一种是通过类似的网关的设备，将不同协议的设备先接入网关，然后由网关接入到上层系统。两种做法各有优缺点，我们这里不做讨论；我们只讨论怎么用Spider这个软件作为基础，让你快速实现一个具有上述功能的网关。

Spider是一个开放性的数据采集平台，它通过驱动的方式来实现对不同设备采集接入的扩展。设备通讯协议目前已经支持Modbus、OPC UA|DA ，将要支持各种常见PLC设备以及104规约等；网络通讯协议已经支持MQTT、Coap、WebApi等。Spider有两种特殊的驱动（Calculate\Custom），这两种驱动运行我们嵌入C#脚本，我们可以通过嵌入的脚本，来达到我们协议转化的目的。下面我们通过“将一个只支持Modbus协议的设备通过MQTT 协议的形式将设备的数据发送出去” 这个实例，来描述整个实现的流程。

1. 协议定义。由于MQTT协议只是一个网络通讯协议，具体怎么和上层系统进行交互，还需要定义一个基于MQTT网络通讯协议的设备通讯协议，这里我们的设备通讯协议采用JSON的形式，具体定义如下：

{

“Command”:”upload”, //指令的类型，upload 表示主动上传数据

“Datetime”:”2022-08-31 9:54:00”, //上传的时间

“Fun”:”devicevalue” //数据的种类，devicevalue表示设备的参数

“Values”: //值的集合

[

{“Name”:”温度”, “Value”:10,”Quality”:0},

{“Name”:”速度”, “Value”:20,”Quality”:0},

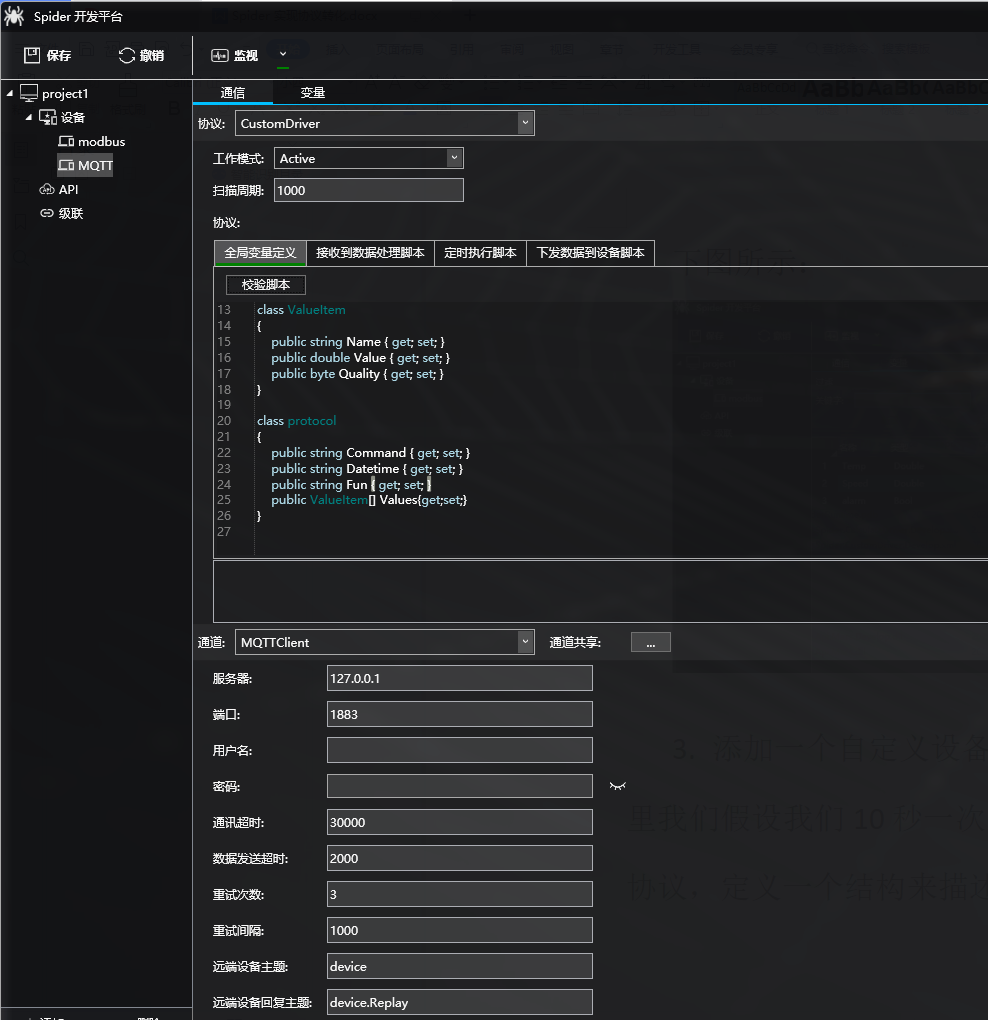
]

}

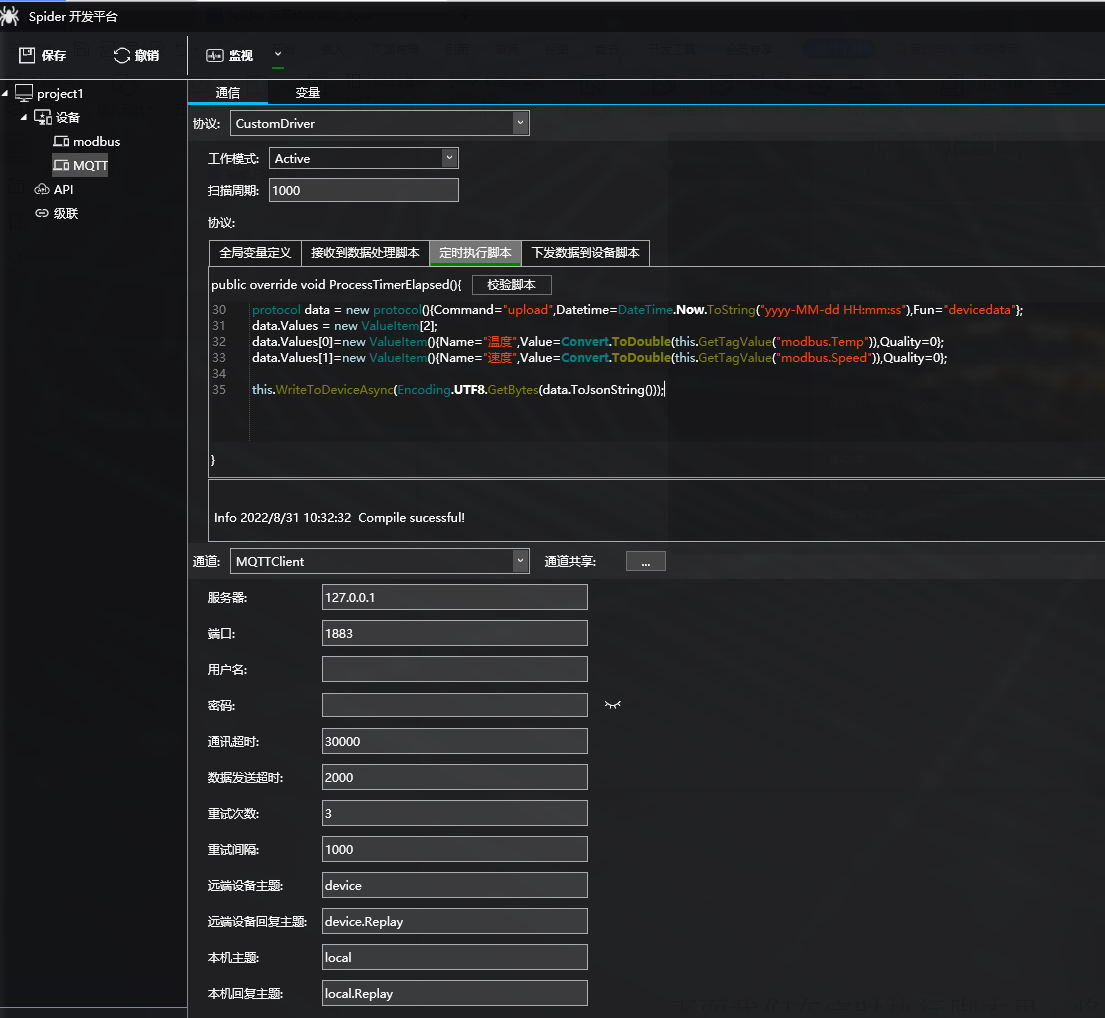
1. 添加一个Modbus 设备，以及添加设备中需要采集的参数。如 下图所示：



1. 添加一个自定义设备，该设备的通讯通道使用MQTT协议。这里我们假设我们1秒一次主动上传设备。首先根据我们上面定义的协议，定义一个结构来描述这个协议，如下图：



下面我们在定时执行脚本里，将数据发送出去。



如果需要响应外部系统的请求指令，则需要在接收到数据处理脚本中，写入数据解析的代码。例如我们这里响应外部系统给设备下发数据，如下如：

