科睿特物联网云平台方案

版本管理

序号	版本号	更改内容
1	V0.1	初始版本
2	V0.2	细化对接协议部分，增加数据加密章节，增加版本管理
3	V0.3	修改 CMD 01 错误，更改为 CMD 1，新增 MODBUS 协议命令
4	V0.4	增加企业轻应用 JSON 数据格式说明
5	V0.5	增加平台下发命令后的回复，以明确机器当前状态
6	V0.6	增加报警状态位的指示说明
7	V0.7	修正部分名称与应用平台不一致问题
8	V0.8	3.1.4 增加应用服务器增加自定义字段功能，方便对接设备。

目录

1.简介

物联网是当前最具发展潜力的技术潮流，到 2020 年全球将有 200 亿—2000 亿物联网设备（ Gartner 预测 260 亿，ABI 预测 300 亿，Oracle 预测 500 亿，Intel 预测 2000 亿）。管理众多的联网设备产生的（动态）数据，相比管理传统档案（静态）数据要复杂很多 —— 最重要的难点是“数据量”大幅提升和“处理实时性”要求显著加强。

- **数据量**：数据量提升包含数据总量和数据产生速度两个方面。物联网应用系统演化过程中，传感器数量不断增多；数据采样频率不断提升；数据积累时间也越来越久，因此产生的数据量非常大（动即十亿、百亿、千亿存储规模），而且数据产生速度也非常快（动即每秒十万、百万纪录）。
- **实时性**：传感器时序数据很多时候用于异常预警、趋势预测等目的，要求能根据数据立刻做出反应。因此数据必须能实时查询、实时分析。

数据量变和实时性要求提升的大前提下，面向于 IOT 的数据库朴素需求细化：

读写特性

- 写操作多于读操作，但读写都要求高速
- 追加为主，但应允许少量更新操作
- 顺序追加为主，但应允许乱序入库
- 可按时间段删除记录，但应允许删除给定纪录
- 支持（优化）给定时间段查询，允许给定字段的精确、模糊、前缀等查询
- 读写并发要求较高，尤其读并发
- 海量存储支持（T - P 级别）

实时性

数据入库即可用（ingest realtime），即任意数据入（input）库后就可立即被第三方应用使用数据 Adhoc 查询 / 分析。

检索和分析特性

- 支持过滤投影
- 支持聚合分析
- 支持关联分析（Join 分析）
- 支持数据挖掘

高可扩展性

- 可按需在线水平扩展

高可用性

- 7*24 小时在线

2.物联网云平台系统架构

物联网云平台系统架构主要包含四大组件：

- 设备接入
- 设备管理
- 设备数据存储与解析
- 规则引擎

2.1 设备接入

物接入（IoT Hub）是一个全托管的云服务，帮助建立设备与云端之间安全可靠的双向连接，以支撑海量设备的数据收集、监控、故障预测等各种物联网场景。

设备接入功能：

- 安全可靠的双向连接
- 认证与授权
- 支持主流物联网协议
- 设备影子

2.1.1 安全可靠的双向连接

物接入服务是全托管的服务，用户可以快速创建物联网服务的实例并安全可靠地连接设备与云端并而不用为运维操心。

提供不同网络的设备接入方案，例如 2/3/4G、NB-IoT、LoRa 等，解决企业异构网络设备接入管理的痛点

2.1.2 认证与授权

提供设备级别的认证，以及基于策略的授权，允许控制设备对特定主题的读写等权限，保障物联网应用的安全。

提供一机一密的设备认证机制，降低设备被攻破的安全风险。

提供设备权限管理机制，保障设备与云端安全通信。

2.1.3 支持主流物联网协议

需要支持 MQTT,COAP,HTTP 等物联网协议，以支持不同的物联网设备。

MQTT 是标准物联网协议，用户可以使用丰富的 MQTT 客户端，使用熟悉的编程语言以及设备平台开发物联网项目。

Coap（Constrained Application Protocol）是一种在物联网世界的类 web 协议，它的详细规范定义在 RFC 7252。COAP 名字翻译来就是“受限应用协议”，顾名思义，使用在资源受限的物联网设备上。物联网设备的 RAM，ROM 都通常非常小，运行 TCP 和 HTTP 是不可以接受的。

2.1.4 设备影子

设备在云端的影子，实时反应设备的当前状态，实现监控、告警、可视化及设备反控等场景。

提供设备影子缓存机制，将设备与应用解耦，解决在无线网络不稳定情况下的通信不可靠痛点。

平台应提供虚拟设备，使用虚拟设备可以模拟实际设备发送数据到云端，亦可以虚拟设备命令发送到物联网设备，方便开发人员的调试与测试。

2.2 设备管理

物联网设备管理提供方便快捷的设备管理能力，在海量设备中快速检索到指定设备，您可以定义设备

的属性、事件、服务，基于定义的物模型对设备进行远程调试、远程监控、远程维护等操作。

提供设备和网关的注册、在线离线状态、数据上报、控制设备、禁用删除等基础设备管理功能。

其功能应包含以下模块：

- 设备注册

  可以自定义设备唯一标识 ID 进行单个或者批量注册需要连接的设备

- 设备产品分组

  可以为某些设备创建分组，基于不同产品进行分组管理搜索

- 设备标签

  可以为设备创建标签，定义设备编号并且可以基于标签搜索管理您的设备

  可以为设备创建设备拥有人，设备地址信息，设备铺设日期，设备详情等基础信息选项，丰富设备信息。

- 设备状态

  可以实时从平台获取设备设备的状态变更信息，例如设备的上下线

- 设备数据采集

  可以基于物模型上报设备数据到云端，并且会帮您将设备物模型数据结构化存储下来，您可以随时查询设备历史数据

- 设备禁用删除

  可以对可疑设备进行远程禁用或者删除，避免可疑设备造成不必要的损失

2.3 设备数据存储与解析

云平台应具备设备数据的分类存储与解析功能，对设备端上传的数据进行在线监控与存储，支持 SQL 等大数据分析平台，必要时能够进行数据可视化图表管理应用。其数据存储应存储副本，采取分布式部署保证数据的存储可靠性。提供设备全链路日志监控，实时知晓设备当前状态，监控设备并排除问题。

远程控制能力，对单个设备或者海量设备下发指令控制。

提供固件升级能力，对大规模设备进行远程升级。

2.4 规则引擎

规则引擎帮助用户灵活地转发和处理设备消息，用户可通过 SQL 的形式设定规则，对消息数据筛选、变型、转发，根据不同场景将数据无缝转发至不同的数据目的地，如时序数据库、物接入主题、机器学习、流式处理、对象存储和关系型存储等。

支持 MQTT,COAP,MODBUS 等终端设备协议解析能力，能对设备原始数据进行数据解析与可视化应用。

提供开放标准的 API，可通过调用 API 实现控制台操作，方便第三方应用快速集成云端服务。

3.设备接入协议

3.1 自定义接入协议

此协议用于设备与服务器通讯，通讯基于 TCP 链接实现，采用 JSON 数据格式进行数据交互,其设备认证采用密钥进行认证。

云平台应提供服务器 IP 地址及域名 ,端口号用于设备连接。例如 :yuapi.xinyuegogo.com ,40000。

其设备接入流程如下：

设备登录→设备鉴权→设备数据接收→设备心跳维持→设备上下线管理。

3.1.1 设备登录指令

设备发送：

```
{
"CMD":1,
"ID":"865933034485942",
"SIM":"898607B9191791469403",
"KEY":"123456abcdefg"
"HW":"V0001",
"FW":"V0001",
"NUM":10,
}
```

云台回复：

```
{
"CMD":1,
"UTC":"1554102064",
"REPLY":1,
}
```

"CMD":用于识别设备命令，其命令标识从 01~99 代表不同的设备命令。其标识格式见下表 3.1。

命令区域	命 令 符	命令含义	备注
1-10	1	设备登录指令，用于设备登录鉴权	02~10 备用，为设备登录类命令
	2	待定	
11-80	11	设备心跳，用于设备心跳发送数据	12~80 备用，为设备心跳传输数据类命令
	12	待定	
81-90	81	设备升级指令，指示设备升级开始	83~90 备用，为设备升级类命令
	82	设备升级数据包，用于发送设备升级数据包	
91-99	91	设备复位指令，将设备复位	92~99 备用，为设备控制类命令
	92	待定	

表 3.1　设备命令符含义表

"ID":设备唯一标识码，为全球唯一标识码，其一般由设备端的 NB-IOT、2G、4G 模组 IMEI 构成，用于标识唯一设备。

其由数字或字母组合而成，可以为纯数字编码或纯字母编码，亦可以由两者组合而成。

其数据长度一般为 15 位，但不限于 15 位，可低于或者高于 15 位，但长度不超过 50 位。

"SIM":设备使用无线通讯时使用到的 SIM 卡号，其由 20 位数字或者字母组成，为全球唯一标识码。

"KEY":设备认证秘钥,由云平台采用 MD5 或者 Password Hashing 加密算法生成（具体加密方式由云平台开发人员决定，但生成的密钥长度应尽量不超过 50 字节，以节省物端的数据流量费用），此密钥应与设备唯一标识码 ID 相关联，平台收到设备登录命令应对设备密钥鉴权，确认其合法性。防止非法设备入侵。

"HW":设备硬件版本号，方便云平台管理，为设备提供升级服务。

"FW":设备软件版本号，方便云平台管理，为设备提供升级服务。

"UTC":北京时间当前 UNIX 时间戳。

"REPLY"：回复 1 表示登录 OK,回复 0 表示登录失败，回复 2 表示设备密钥错误。

"NUM"：设备端登录次数，每登录一次数值加 1。

3.1.2 MODBUS 协议指令

➢ 协议指令 CMD20

服务器端发送：

```
{
"CMD":20,
"ID":"865933034485942",
"AGREEMENT":"MODBUS",
"MODE":"RTU",
"BAUD":9600,
}
```

机器端回复：

```
{
"CMD":20,
"ID":"865933034485942",
"AGREEMENT":"MODBUS",
"MODE":"RTU",
"BAUD":9600,
"REPLY":1,
}
```

"AGREEMENT"：协议类型，在未收到 CMD：20 前设备默认为 JSON 格式的普通协议，无专用协议通道。

可能的协议类型："MODBUS"，"PROFIBUS"，"PROFIBUS"，"MPI"…等。

"MODE":MODBUS 协议模式，"RTU"或者"ASCII"。

"BAUD":协议 RS485 接口通讯波特率，为 1200、2400、4800、9600、19200 等。

"REPLY":是否设置成功，回复 0：失败，1：成功

➢ 命令指令 CMD21

服务器发送 MODBUS 包：

```
{
"CMD":21,
"ID":"865933034485942",
"AGREEMENT":"MODBUS",
"FUNCTION":"0X01",
"M_DATA":[3,3,16,100,0,26,128,252],
}
```

"M_DATA":主机 MODBUS 完整帧节数据，完整的 ADU,包含地址域、PDU、CRC 校验值。

"FUNCTION":MODBUS 协议功能码，其具体功能码与协议一致，见下图：

	数据类型	功能描述	功能码	功能码（十六进制）	异常功能码
比特访问	物理离散量输入	读输入离散量	02	0x02	0x82
	内部比特或者物理线圈	读线圈	01	0x01	0x81
		写单个线圈	05	0x05	0x85
		写多个线圈	15	0x0F	0x8F
16比特访问	输入存储器	读输入寄存器	04	0x04	0x84
	内部存储器或物理输出存储器（保持寄存器）	读多个寄存器	03	0x03	0x83
		写单个寄存器	06	0x06	0x86
		写多个寄存器	16	0x10	0x90
		读/写多个寄存器	23	0x17	0x97
		屏蔽写寄存器	22	0x16	0x96
文件记录访问		读文件记录	20	0x14	
		写文件记录	21	0x15	

设备回复 MODBUS 包：

```
{
"CMD":21,
"ID":"865933034485942",
"AGREEMENT":"MODBUS",
"FUNCTION":"0X01"
"S_DATA":[3,3,16,100,0,26,128,252],
"STATUS": "OK"
}
```

"S_DATA":从机 MODBUS 完整回复帧节数据，完整的 ADU,包含地址域、PDU、CRC 校验值。

"STATUS":回复状态，其分为以下几种：

i. "OK"：收到从机答复且数据格式正确

ii. "ERROR_CRC"：收到从机答复，但 CRC 校验错误，其完整回复数据包仍然在"S_DATA"中

iii. "ERROR_OT"：从机回复超时，可能无此从机或 RS485 物理接线断开

3.1.3 心跳指令

设备端心跳指令：

```
{
"CMD":11,
"ID":"865933034485942",
"CSQ":"-51",
}
```

"CSQ": 设备信号强度，用于服务器建立数据模型分析当前设备信号强度

心跳包，云端无需回复。

设备连线后服务器定时检查是否有数据包，如 5 分钟无数据包则主动把设备 Socket 链接踢掉。

注：数据包包含心跳包及其它指令数据包，是否有心跳包取决于设备类型发送数据量，是否需要心跳包维持链接，一般情况下心跳包每 2 分钟会上报一次。例如终端设备数据量巨大，设备无时间窗口发送心跳包，则可能设备无心跳包。

3.1.4 其它指令

云端收到其它指令时，只需要将数据存储到当前 Socket 链接映射设备号，将对应的数据存储到对应设备影子。应用服务器通过调用设备存储数据，获取设备状态信息。

在应用平台，需要增加自定义的命令字与消息，以方便设备与应用平台对接。

自定义命令字规则如下：

1.命令字固定为 JSON "CMD"，其内容为 Int 型数据，数据范围为表 3.1 规定的 11~80。

2.命令字的取值可由开发人员在取值范围内自定义，单个产品无命令字冲突即可。

自定义消息规则如下：

1.自定义消息取名只能包含大小写字母，数字，下划线等符号，例如: Temp_1 为合法消息，Temp>1 为非法消息名。

2.自定义的消息内容可为 JSON 规定的字符串，数字，数组等。

例如：在应用服务器下创建命令字：21 及消息名 EPC_DATA,消息类型定义为字符串。

则设备端发送消息如下：

{"CMD":21,"ID":"865192040645037","EPC_DATA":"13000101010CE2000017001400481770 60CBAB2F"}

服务器根据对应的 CMD，EPC_DATA 对数据进行解析。

3.2 MQTT 接入协议

暂无

3.3 COAP 接入协议

暂无

3.4 数据加密

3.4.1 TCP 通道

目前采用此种方式，采用 TCP 链接方式，数据本身不加密，安全级别低。

3.4.2 TCP 通道+对称加密

采用 TCP 链接方式，数据本身采用设备密钥做对称加密。

协议待定

3.4.3 TCP 通道+TLS 协议

采用 TCP 链接方式，数据采用 TLS 协议加密。

协议待定

参考文档：

D:\MODBUS\MODBUS 协议中文版(高清版).pdf

4.企业云平台轻应用

4.1　JSON 数据说明

设备上报生产数据：

{
"CMD":11,
"ID":"865933034485942",
"MACHINE":"0X33",

"DATA":[863 , 316 , 1237 , 256 , 255 , 51 , 502 , 1236 , 16 , 123 , 265 , 354,4254 ,

256,256,256,256,256,256,256,256,256,256,256,256,256] //总共 26 组数据

}

此数据用于应用平台分析，无需回复。

平台下发生产命令：

{
"CMD":12,
"ID":"865933034485942",
"MACHINE":"0X33",

"ORDER":[863，316，1237，256，255，51，502，1236，16，123，265，354,4254，256,256,256],//

总共 16 组数据

}

回复：

{
"CMD":12,
"ID":"865933034485942",
"MACHINE":"0X33",

"STATUS": 1　//0：失败 ， 1：成功 ， 2：超时 3：CRC 错误

}

释义：

"MACHINE":"0X33" 服务器定义的机台号，此项定义需要开发人员填写机台号与从机号对应关系表，

举例如：

序号	机台号	MACHINE
1	机台 1	"0X03"
2	机台 2	"0X34"
3	机台 3	"0X35"

"DATA":机台汇报的数据，其数组对应数据关系见下表：

序号	数组	对应数据	单位：	备注
1	DATA[0]	湿度	%RH*10	如 863 则代表实际湿度：86.3 %RH
2	DATA[1]	温度	℃*10	如 316 则代表实际温度 36.1℃
3	DATA[2]	称重重量	Kg*100	如 1237 则代表称重重量 12.37 Kg
4	DATA[3]	质检重量	g*10	如 3271 则代表 327.1 g
5	DATA[4]	变频频率	HZ*10	如 256 则代表 25.6HZ
6	DATA[5]	电机频率	HZ*10	如 256 则代表 25.6HZ
7	DATA[6]	当前长度	米*10(M*10)	如 51 则代表 5.1M
8	DATA[7]	上卷长度	米*10(M*10)	如 502 则代表 5.02M
9	DATA[8]	剩余次数		
10	DATA[9]	运行状态		
11	DATA[10]	卷长序号		
12	DATA[11]	称重序号		
13	DATA[12]	质检序号		

14	DATA[13]	当班产量	平方米	
15	DATA[14]	前班产量	平方米	
16	DATA[15]	通信匹配		
17	DATA[16]	当班班次		
18	DATA[17]	前班班次		
19	DATA[18]	主机手		
20	DATA[19]	下副手		
21	DATA[20]	放卷工		
22	DATA[21]	加米数	米(M)	
23	DATA[22]	班单产量	平方米	
24	DATA[23]	保留		
25	DATA[24]	保留		
26	DATA[25]	保留		

其中，DATA[9]对应的故障状态见下表：

序号	DATA[9]对应位数	异常	备注
1	0	运行状态	DATA[9] = 0X00，表示停止；0X01 表示运行
2	1	计数到停	数据有上传，不解析
3	2	任务停	数据有上传，不解析
4	3	变频故障	0 为正常状态，1 为报警状态，以下同理
5	4	烤温报警	
6	5	长度报警	
7	6	称重超重	
8	7	称重偏轻	
9	8	质检超重	
10	9	质检偏轻	
11	10	湿度报警	
12	11	保留	
13	12	保留	
14	13	保留	
15	14	保留	
16	15	保留	

"ORDER"：下达到机台的命令，其数组对应数据关系如下：

序号	数组	对应数据	单位：	备注
1	ORDER[0]	生产模式		
2	ORDER [1]	任务卷长	米*10(M*10)	如 51 则代表 5.1M
3	ORDER [2]	任务次数		
4	ORDER [3]	坯布幅宽		

5	ORDER［4］	成品幅宽		
6	ORDER［5］	任务称重	Kg*100	如 1237 则代表称重重量 12.37 Kg
7	ORDER［6］	任务质检	g*10	如 3271 则代表 327.1 g
8	ORDER［7］	保留		
9	ORDER［8］	保留		
10	ORDER［9］	保留		
11	ORDER［10］	保留		
12	ORDER［11］	保留		
13	ORDER［12］	保留		
14	ORDER［13］	远程启停		0=停止，1=启动
15	ORDER［14］	远程类型		0=不操作，1=远程启停，2=远程下单
16	ORDER［15］	通信匹配		起验证码作用，下达指令时自增 1

远程启动：ORDER［13］=1，ORDER［14］=1，ORDER［15］在上传的通信匹配值的基础上自增 1，然后数组 16 个数据下发给设备

远程停止：ORDER［13］=0，ORDER［14］=1，ORDER［15］在上传的通信匹配值的基础上自增 1，然后数组 16 个数据下发给设备

远程下单：ORDER［14］=2，ORDER［15］在上传的通信匹配值的基础上自增 1，ORDER[0]-［6］的值为应用端人为填入的任务数据，然后数组 16 个数据下发给设备

设备上报监测数据：

```
{
"CMD":13,
"ID":"865933034485942",
"MACHINE":"0X33",
"PARAMETER":[0，30，0，0，0，0，0，0，0，0，0，0,40，0,0,0],//总共 16 组数据
}
```

此数据用于应用平台分析，无需回复。

"PARAMETER":机台运行参数及状态

序号	数组	对应数据	单位：	备注
1	PARAMETER [0]	1#电流参数	A	
2	PARAMETER [1]	1#振动参数	mm/s	
3	PARAMETER [2]	1#启停状态		
4	PARAMETER [3]	1#故障状态		
5	PARAMETER [4]	1#电流阀值	A	
6	PARAMETER [5]	1#振动阀值	mm/s	
7	PARAMETER [6]	保留		
8	PARAMETER [7]	保留		
9	PARAMETER [8]	保留		
10	PARAMETER [9]	保留		
11	PARAMETER [10]	2#电流参数	A	
12	PARAMETER [11]	2#振动参数	mm/s	
13	PARAMETER [12]	2#启停状态		
14	PARAMETER [13]	2#故障状态		
15	PARAMETER [14]	2#电流阀值	A	
16	PARAMETER [15]	2#振动阀值	mm/s	

注：1#，2#为对应机台 1，机台 2.此对应关系需另行说明。

平台下发监测命令：

{
"CMD":14,
"ID":"865933034485942",
"MACHINE":"0X33",

"MONITOR":[0，30，0，0，0，0，0，0，0，0，0，0],//总共 12 组数据

}

回复：

{
"CMD":13,
"ID":"865933034485942",
"MACHINE":"0X33",

"STATUS": 1 //0：失败 ， 1：成功 ， 2：超时 3：CRC 错误

}

"MONITOR":设置的机台监测参数

序号	数组	对应数据	单位：	备注
1	MONITOR [0]	1#电流阀值	A	

2	MONITOR [1]	1#振动阀值	mm/s	
3	MONITOR [2]	保留		
4	MONITOR [3]	保留		
5	MONITOR [4]	保留		
6	MONITOR [5]	保留		
7	MONITOR [6]	保留		
8	MONITOR [7]	保留		
9	MONITOR [8]	保留		
10	MONITOR [9]	保留		
11	MONITOR [10]	2#电流阀值	A	
12	MONITOR [11]	2#振动阀值	mm/s	