**HMI**

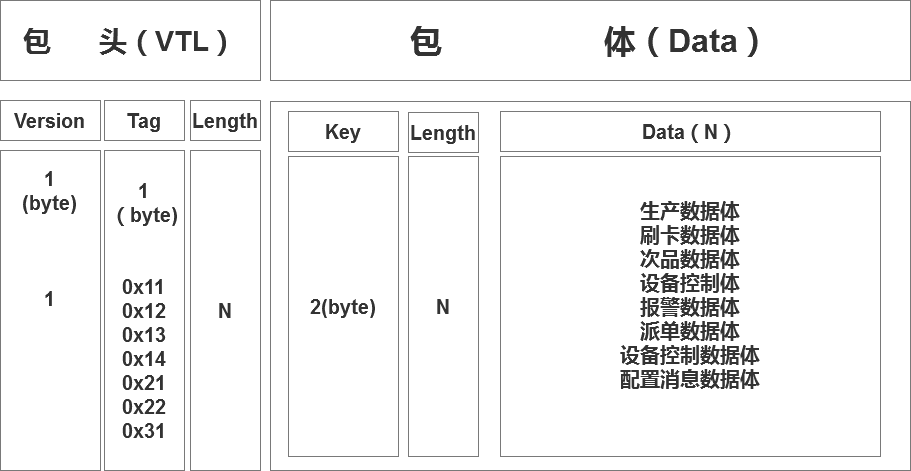
**通信协议**

1. 概述&目的：

后端数据传输协议定义了下位机与上位机之间数据通信格式，通过该协议明确上下位机之间的数据包组成以及方便扩充和修改。

1. 协议组成：

后端数据通信协议主要有包头+包体组成（如下图）：



VTL: 1B 1B 2B

KLD: 2B 2B 2B

Data部分全部为Json。

包体数据说明：

JSON格式字段内容只有两种Integer与String。

对于数组或列表方面的格式的内容使用JSONArray

以下文档内容中：

ASC为字符串类型（最大长度不超过预设长度）

HEX为Integer.(如果有4字节为长整型LongInt须要注意说明)

数据响应时的Data(JSON):

{ “result”: 0 //int 0 成功；1出错

“msg”:”” } //string 消息信息

Json示例如下，其它包体参照类型和数据组合。

**生产数据示例:**

**1包头（VTL）组成如下**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段 | 名字 | 长度（单位Byte） | 备注 |
| 1 | [Versions](file:///C:/Users/huang/AppData/Local/youdao/dict/Application/7.0.1.0214/resultui/dict/../dict/result.html%3Fkeyword=versions) | 版本号 | 1 |  |
| 2 | Tag | 包类型 | 1 | 具体定义见TAG表 |
| 3 | Length | 包体长度 | 2 | 包体的总字节长度（Value)  网络字节序  用memcpy复制两边可对应 |
| 4 | Value | 包体 | n | 具体定义见Value表 |

**1.1 TAG****表（包类型）**

| TAG | 名 称 | 数据类型 | | 描 述 |
| --- | --- | --- | --- | --- |
| 下位机上传数据: (主题) d2s/设备序列号/协议版本/data | | | | |
| 0x11 | 生产数据 | 上发数据  (不需响应) | | 实时的生产数据（包括温度压力） |
| 0x12 | 刷卡数据 | 上发数据  (需要响应) | | 刷卡产生的上传数据 |
| 0x13 | 次品数据 | 上发数据  (需要响应) | | 生产次品数 |
| 0x14 | 报警数据 | 上发数据 | | 下位机相关报警 |
| 服务器下发数据：(主题) s2d/设备序列号/协议版本/data  配置消息(主题) s2d/设备序列号/协议版本/config | | | | |
| 0x21 | 机器派单 | 下发数据 | 服务端派工单  须要回复 | |
| 0x22 | 设备控制 | 下发数据 | 服务端下发设备控制指令  须要回复 | |
| 0x31 | 配置消息 | 下发数据  （参数配置） | 系统（本地化）参数（计数信号和正负逻辑等。  须要回复 | |
| 设备离线遗嘱主题：will/设备序列号 {“sn”:””,”onLineId”:11231} | | | | |
| 设备上线时主题：online/设备序列号 {“sn”:””,”onLineId”:11231,其他信息待定 }  onLineId: 设备在线时的ID号，只要与遗嘱一样就离线。 | | | | |

服务器下发主题 ：

s2d/设备序列号/协议版本/data

s2d/设备序列号/协议版本/config

采集器上传(包含回复)主题：

d2s/设备序列号/协议版本/data

采集器回复：VTL(包头): KLD(包体)

包体：Key为下发到采集器的Key

Data为JSON数据：

{ “result”: 0, //integer类型,

///结果状态(错误码):0成功;1失败须重发;2失败不重发,

///0xFFFF其它

“msg”:”....” //失败原因说明

}

**1.2 VALUE表（包体）**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段 | 名称 | 长度(Byte) | 备注 |
| 1 | Key | 数据ID | 2 | 上传与下发设置为同一个key |
| 2 | Length | Value的长度 | 2 | Data长度 |
| 4 | Data | 数据体 | n | 根据TAG的类型定义长度（见表1.1.1 至1.1.7） |

**下位机上传数据**

**1.2.1 生产数据** **（TAG值 0x11)**

| 序号 | 名 称 | 字节数 | 数据类型 | 描 述 |
| --- | --- | --- | --- | --- |
| 1 | PD\_DispatchNo | 20 | ASC | 派工单号 |
| 12 | PD\_StartCycle | 6 | HEX | 生产数据开始时间（成型周期开始时间） |
| 6 | PD\_EndCycle | 6 | HEX | 生产数据产生时间（成型周期结束时间） |
|  | PD\_TemperNum | 1 | HEX | 温度路数(N) |
| 7 | PD\_TemperValue | 2\*N | Array | 4路温度值，每路2字节依序为  温度1 （2字节）  温度2 （2字节）  温度3 （2字节）  温度4 （2字节）  ....温度N （逗号隔开） |
| 8 | PD\_MachineCycle | 4 | HEX | 机器周期，毫秒 |
| 9 | PD\_FillTime | 4 | HEX | 填充时间，毫秒 |
| 10 | PD\_CycleTime | 4 | HEX | 成型周期，毫秒 |
| 11 | PD\_TotalNum | 4 | HEX | 模次（成模次数，啤数） |
|  | PD\_KeepPressNum | 1 | HEX | 压力点个数（N） |
| 12 | PD\_KeepPress | 2\*N | HEX | 100个压力点（逗号隔开） |

**1.2.2 刷卡数据（TAG值 0x12)**

| 序号 | 名 称 | 字节数 | 数据类型 | 描 述 |
| --- | --- | --- | --- | --- |
| 1 | CC\_DispatchNo | 20 | ASC | 派工单号 |
| 5 | CC\_CardID | 10 | HEX | 卡号 |
| 7 | CC\_CardType | 1 | HEX | 刷卡原因编号[功能ID] |
| 8 | CC\_CardDate | 6 | HEX | 刷卡数据产生时间 |
| 9 | CC\_IsBeginEnd | 1 | HEX | 刷卡开始或结束标记，  0表示开始  1表示结束 |

|  |
| --- |
| 刷卡原因编号 |
| "上班",1, false  "下班",2, false  "交班",3, true  "换模",4, true  "换料",5, true  "换单",6, true  "待人",7, true  "待料",8, true  "换班",9, true  "机器故障",10, true  "模具故障", 11, true  "辅设故障", 12, true  "无订单", 13, true  "我的派工单列表", 255, true  “BindDataArray”:[ //设备上的派工单与件数组  {MO\_DispatchNo,PCS\_ItemNo,CreatedTime}  ] |

**1.2.3 次品数据（TAG值 0x13)**

| 序号 | 名 称 | 字节数 | 数据类型 | 描 述 |
| --- | --- | --- | --- | --- |
| 1 | FDAT\_DispatchNo | 20 | ASC | 派工单号 |
| 2 | FDAT\_CardID | 10 | HEX | 卡号 |
| 3 | FDAT\_CardDate | 6 | HEX | 次品数据产生时间 |
| 4 | FDAT\_MultiNum | 1 | HEX | 总件数N（0< N <= 100） |
| 5 | FDAT\_BadReasonNum | 1 | HEX | 次品原因总数（M） |
| 6 | FDAT\_Data | N\*Bad\_Data\_SIZE | HEX | N件产品的次品**（见1.1.3.1****）** |

**1.2.3.1 每件产品次品数据（FDAT\_Data）**

| 序号 | 名 称 | 字节数 | 数据类型 | 描 述 |
| --- | --- | --- | --- | --- |
| 1 | Bad\_ItemNo | 20 | ASC | 产品编号 |
| 2 | Bad\_BadQty | 4 | DWORD | 次品总数 |
| 3 | Bad\_BadData | M \*4 | HEX | 每件产品的次品数据（对应次品原因的数据）（见1.1.3.1.1） |

**1.2.3.1.1**

| 序号 | 名 称 | 字节数 | 数据类型 | 描 述 |
| --- | --- | --- | --- | --- |
| 1 | MO\_BadItem1 | 4 | DWORD | 次品原因1数量 |
| 2 | MO\_BadItem2 | 4 | DWORD | 次品原因2数量 |
| …… | | | | |
| M | MO\_BadItemM | 4 | DWORD | 次品原因M数量 |

**1.2.4 报警（TAG值 0x14)**

| 序号 | 名 称 | 字节数 | 数据类型 | 描 述 |
| --- | --- | --- | --- | --- |
| 1 | WR\_TYPE | 1 | HEX | 警报类型   1. 设备报警（监控机器） 2. 本地报警（HMI报警） 3. 其它报警 |
| 2 | WR\_NAME | 20 | ASC | 警报名称（自定义） |
| 3 | WR\_TIMER | 6 | HEX | 警报产生产生时间 |
|  | FDAT\_DispatchNo | 20 | ASC | 派工单号 |

**上位机下发数据**

**1.2.5 上位机派单 （TAG 指令****0x21)**

| 名 称 | 字节数 | 数据类型 | 描 述 |
| --- | --- | --- | --- |
| FDAT\_DataType | . | integer | 派单类型  0x00：标准主单  0x01：标准次单 |
| FDAT\_Data | W | HEX,  JSON不须要 | 根据FDAT决定本数据段的长度和具体有效数据**(见1.2.5.1****）** |

**1.2.5.1 派工单主体（FA\_Data)**

| 序号 | 名 称 | 字节数 | 数据类型 | 描 述 |
| --- | --- | --- | --- | --- |
| 1 | MO\_DispatchNo | 20 | string | 派工单号 |
| 3 | MO\_ProcCode | 20 | string | 工序代码 |
| 4 | MO\_ProcName | 50 | string | 工序名称 |
| 5 | MO\_StaCode | 10 | string | 站别代码 |
| 6 | MO\_StaName | 20 | string | 站别名称（工作中心） |
| 7 | MO\_StandCycle | 4 | DWORD | 标准周期，毫秒 |
| 8 | MO\_TotalNum | 4 | DWORD | 模次,生产次数（原啤数） |
| 9 | MO\_MultiNum | 1 | integer | 总件数N（0< N <= 100） |
| 11 | MO\_BadReasonNum | 1 | integer | 次品原因类型  总数M（0< M <= 100） |
| 12 | MO\_Data | N\*PCS\_Data\_SIZE | Array  可决定N与M | N件产品的主单数据（见1.2.5.2） |

派工单下面有多个产品，这些产品编号是不同的。

**1.2.5.2一模多件每件主体（MO\_DAT)**

| 序号 | 名 称 | 字节数 | 数据类型 | 描 述 |
| --- | --- | --- | --- | --- |
| 1 | PCS\_MO | 20 | string | 工单号（件） |
| 2 | PCS\_ItemNO | 20 | string | 产品编号 |
| 3 | PCS\_ItemName | 50 | string | 产品描述 |
| 4 | PCS\_MouldNo | 20 | string | 模具编号 |
| 5 | PCS\_DispatchQty | 4 | DWORD | 派工数量 |
| 6 | PCS\_SocketNum1 | 1 | integer | 该件模穴数 |
| 7 | PCS\_BadQty | 4 | DWORD | 次品总数 |
| 8 | PCS\_BadData | M \*4 | Array | 每件产品的次品数据（对应次品原因的数据）（见1.2.5.3） |

**1.2.5.3 PCS\_BadData,Array**

| 序号 | 名 称 | 字节数 | 数据类型 | 描 述 |
| --- | --- | --- | --- | --- |
| 1 | MO\_BadItem1 不须要 | 4 | DWORD | 次品原因1数量 |
| 2 | MO\_BadItem2不须要 | 4 | DWORD | 次品原因2数量 |
| …… | | | | |
| M | MO\_BadItemM不须要 | 4 | DWORD | 次品原因M数量 |

**1.2.6 上位机下发设备控制指令 （TAG 指令0x22)**

| 名 称 | 字节数 | 数据类型 | 描 述 |
| --- | --- | --- | --- |
| FDAT\_CommandType | 1 | HEX | 控制类型  0x01：HMI关机  0x02：HMI重启  0x03: HMI 升级  0x04自定义 |

**1.2.7 配置消息下发 （TAG 指令类型0x31)**

| 序号 | 名 称 | 字节数 | 数据类型 | 描 述 |
| --- | --- | --- | --- | --- |
| 1 | FDAT\_ConfigType | 1 | HEX | 控制类型  0x01：设置采集参数  0x02：设置HMI时间  0x03: 自定义 |
| 2 | FDAT\_ConfigData | N | 根据类型 | 见1.2.7.1 和1.2.7.2 |

**1.2.7.1 设置采集参数（FDAT\_ConfigData 0x01)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | Default | String |  | 默认扫描配置  ScanConfig-GPIO  ScanConfig-Kang |
| 2 | ScanConfig-GPIO | Object |  |  |

|  | LineA | Object | “StartPort”: {“GPIO”:,”value”:},  “EndPort”: {} | 线A扫描信息  开始点与结束点  GPIO 采集器那个口  value 高电平底电平 |
| --- | --- | --- | --- | --- |
|  | LineB | Object |  |  |
|  | LineC | Object |  |  |

|  |
| --- |
| 列子： |
| {  "FDAT\_ConfigType": 1,  "FDAT\_ConfigData": {  "Default": "ScanConfig-GPIO",  "ScanConfig-Kang": {  "LineA": {  "StartPort": {  "GPIO": 1,  "value": 1  },  "EndPort": {  "Port": 1,  "Value": 0  }  },  "LineB": {  "StartPort": {  "GPIO": 1,  "value": 1  },  "EndPort": {  "GPIO": 1,  "Value": 0  }  },  "LineC": {  "StartPort": {  "GPIO": 1,  "value": 1  },  "EndPort": {  "GPIO": 1,  "Value": 0  }  }  }  }  } |

**1.2.7.1 设置采集Time（FDAT\_ConfigData 0x02)**

| 序号 | 名 称 |  | 数据类型 | 描 述 |
| --- | --- | --- | --- | --- |
|  | 年月日时分秒 | string | Time | YYYY-MMM-DDD HH:MM:SS  2017-01-02 24:12:32 |