文档编号：HL2023-1015001

# 环力车间检测设备
# 存储检测明细数据开发说明书

**东莞市环力智能科技有限公司**

2023 年 10月

# 前　言

## 1、文档控制

1、编号：HL2023-1015001

背景说明：基于项目需求和品质追溯需要，车间所有检测设备的明细数据需要调用接口集中存储。

2、建立日期：　2023-10-15

3、有效日期：一个年内有效

4、文档去向：设备供应商

5、技术对接：资讯部 侯叶军 18925274990 （微信同号）Email:hlit@dghuanli.com

## 2、文档说明

1. **保密控制**：因车间数据为公司生产、加工的内部核心机密，此文档也受保密协议控制；
2. **阅读范围**：设备供应商;

# 开发需求

## 一、现有界面增加“明细数据上传”开关

设备软件在保持现有逻辑控制的基础上，
A、增加一个选项开关：“明细数据上传”，默认为开启状态；
B、增加权限，只有管理员才可以有权打开或关闭“明细数据上传”选项；

## 二、接口调用技术、参数相关说明

# 1. 接口概要

## 1.1. 接口技术规范

接口方式

东莞市环力智能科技有限公司车间检测设备数据存储的接口通过WebAPI方式实现，采用基于HTTP的GET/POST调用方式，返回XML或JSON数据串给调用端。

## 1.2. 数据处理和控制

WebAPI方式

- WebAPI基于RESTful架构，使用HTTP协议进行通信；
- WebAPI支持跨平台，可以在任何平台上使用；

车间检测设备程序调用服务器接口提交检测数据，集成服务接口将接收到的数据存储在数据库中，通过返回True或False标志进行状态交互

## 2.2. 接口结构定义

## 2.1.接口类型

| 接口编号 | 类型 | 接口方向 | 备注 |
| --- | --- | --- | --- |
| HLQC | 数据存储 | 设备端→数据存储中心库 | |
| HLSEARCH | 数据查询 | 任何客户端→数据存储中心库 | |

### 2.2.1. 数据存储接口

- 接口说明

  数据存储接口执行对车间设备的检测数据进行保存；

- 接口方向
  设备端→数据存储中心库

  - 触发事件

  车间设备软件在检测完成后，提交检测结果（NG或OK）给MES系统的同时，提交数据给“集成服务”接口，收到返回的结果时需要有相应提示给检测软件操作人员；

接口字段
（“检测明细数据”按各设备检测的数据字段多少顺序从001-400增加。）

| 描述 | 数据库字段 | 类型（长度） | 示例 | 备注 |
| --- | --- | --- | --- | --- |
| 数据标记 | DATAFLAG | 文本字符型（2） | 默认为 01 | NOT NULL |
| 产品SN码 | SN | 文本字符型（27） | 0235AFXU**H231017AA10G00931 | 与MES参数一致，不为空 |
| 项目编码 | PRID | 文本字符型（5） | HL005 /HL063 | 三楼为005 二楼为063 不为空 |
| 产品编码 | PNUMBER | 文本字符型（50） | HL001 | 与MES参数一致，不为空 |
| 工序编码 | GXID | 文本字符型（4） | B002 | 与MES参数一致，不为空 |
| 工序名称 | GXNAME | 文本字符型（4） | 螺母浮高检测 | 与MES参数一致，不为空 |
| 检测结果 | RESULT | 文本字符型（10） | TUER | TUER即OK，FALSE即NG，不为空 |
| 开始检测时间 | STARTTIME | 长日期时间型 | 2023-10-31 06:07:59:666 | 设备本次开始检测的时间，不为空 |
| 结束检测时间 | ENDTIME | 长日期时间型 | 2023-10-31 06:09:59:666 | 设备本次结束检测的时间，不为空 |

| | | | | |
|---|---|---|---|---|
| 提交数据时间 | SUBMITTIME | 长日期时间型 | 2023-10-31 06:09:59:866 | 软件本次提交数据的时间，不为空 |
| 检测明细数据001 | QC001 | 文本字符型（50） | 6.25 | 按现有检测数据提交。 |
| 检测明细数据002 | QC002 | 文本字符型（50） | 2.30 | |
| 检测明细数据003 | QC003 | 文本字符型（50） | 4.20 | |
| ……400 | ……QC400 | 文本字符型（50） | 按现有日志填充。 | 支持最大400列数据存储 |

### 2.2.2. 数制查询接口

- 接口说明

  集成接口接收查询参数，如产品SN、工位信息等返回相关产品的检测明细数据

- 接口方向

  任何客户端→数据存储中心库

- 触发事件

查询时触发。

接口字段

| 描述 | 数据库字段 | 类型（长度） | 示例 | 备注 |
|---|---|---|---|---|
| 数据标记 | DATAFLAG | 文本字符型（2） | 默认为02 | NOT NULL |
| 产品SN码 | SN | 文本字符型（27） | 0235AFXU**H231017 AA10G00931 | 与MES参数一致，不为空 |
| 项目编码 | PRID | 文本字符型（4） | B002 | 与MES参数一致，不为空 |
| | | | | |

### 2.2.3. 接口调用（开发测试阶段）

**URL: http://58.253.84.177:36118/execute?action=HLQC**

**URL: http://58.253.84.177:36118/execute?action=**HLSEARCH

**C#代码：**

```
public class HttpWebResponseUtility
        {
            private static readonly string DefaultUserAgent = "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100 Safari/537.36";


            private static bool CheckValidationResult(object sender, X509Certificate certificate, X509Chain chain, SslPolicyErrors errors)
            {
                return true; //总是接受
            }

            /// <summary>
            /// 创建GET方式的HTTP请求
            /// </summary>
            /// <param name="url">请求的URL</param>
            /// <param name="timeout">请求的超时时间</param>
            /// <param name="userAgent">请求的客户端浏览器信息，可以为空</param>
            /// <param name="cookies">随同HTTP请求发送的Cookie信息，如果不需要身份验证可以为空</param>
            /// <returns></returns>
            public static HttpWebResponse CreateGetHttpResponse(string url, int? timeout, string userAgent, CookieCollection cookies)
            {
                if (string.IsNullOrEmpty(url))
                {
                    throw new ArgumentNullException("url");
                }
                HttpWebRequest request = WebRequest.Create(url) as HttpWebRequest;
                request.Method = "GET";
                request.UserAgent = DefaultUserAgent;
                if (!string.IsNullOrEmpty(userAgent))
                {
                    request.UserAgent = userAgent;
                }
                if (timeout.HasValue)
                {
                    request.Timeout = timeout.Value;
                }
                if (cookies != null)
                {
                    request.CookieContainer = new CookieContainer();
                    request.CookieContainer.Add(cookies);
                }
                return request.GetResponse() as HttpWebResponse;
            }


            /// <summary>
            /// 创建POST方式的HTTP请求
            /// </summary>
            /// <param name="url">请求的URL</param>
            /// <param name="parameters">随同请求POST的参数名称及参数值字典</param>
            /// <param name="timeout">请求的超时时间</param>
            /// <param name="userAgent">请求的客户端浏览器信息，可以为空</param>
            /// <param name="requestEncoding">发送HTTP请求时所用的编码</param>
            /// <param name="cookies">随同HTTP请求发送的Cookie信息，如果不需要身份验证可以为空</param>
            /// <returns></returns>
            public static string CreatePostHttpResponse(string url, IDictionary<string, string> parameters, int?
```

```
timeout, string userAgent, Encoding requestEncoding, CookieCollection cookies)
            {
                if (string.IsNullOrEmpty(url))
                {
                    throw new ArgumentException("url");
                }
                if (requestEncoding == null)
                {
                    throw new ArgumentException("requestEncoding");
                }
                HttpWebRequest request = null;
                //如果是发送HTTPS请求
                if (url.StartsWith("https", StringComparison.OrdinalIgnoreCase))
                {
                    ServicePointManager.ServerCertificateValidationCallback = new
                        RemoteCertificateValidationCallback(CheckValidationResult);
                    request = WebRequest.Create(url) as HttpWebRequest;
                    request.ProtocolVersion = HttpVersion.Version10;
                }
                else
                {
                    request = WebRequest.Create(url) as HttpWebRequest;
                }
                request.Method = "POST";
                request.ContentType = "application/x-www-form-urlencoded";

                if (!string.IsNullOrEmpty(userAgent))
                {
                    request.UserAgent = userAgent;
                }
                else
                {
                    request.UserAgent = DefaultUserAgent;
                }
                if (timeout.HasValue)
                {
                    request.Timeout = timeout.Value;
                }
                if (cookies != null)
                {
                    request.CookieContainer = new CookieContainer();
                    request.CookieContainer.Add(cookies);
                }

                //如果需要POST数据
                if (!(parameters == null || parameters.Count == 0))
                {
                                    var postdata = string.Join("&", parameters.Select(
                        p => string.Format("{0}={1}", p.Key, System.Web.HttpUtility.UrlEncode(p.Value,
requestEncoding))).ToArray());
                    //byte[] data = requestEncoding.GetBytes(buffer.ToString());
                    byte[] data = requestEncoding.GetBytes(postdata);
                    using (Stream stream = request.GetRequestStream())
                    {
                        stream.Write(data, 0, data.Length);
                    }

                }
                // 获得响应流
                HttpWebResponse response = (HttpWebResponse)request.GetResponse();
                string r = null;
                using (StreamReader myStreamReader = new StreamReader(response.GetResponseStream(),
```

```
Encoding.UTF8))
                {
                    string retString = myStreamReader.ReadToEnd();
                    r = Regex.Unescape(retString);//将返回的unicode转化为中文
                }


                //myStreamReader.Close();
                //myResponseStream.Close();

                if (response != null)
                {
                    response.Close();
                }
                if (request != null)
                {
                    request.Abort();
                }

                return r;

            }


            public static string Get(string Url)
            {

                HttpWebRequest request = (HttpWebRequest)WebRequest.Create(Url);//创建请求
                request.Proxy = null;//IP代理池
                request.KeepAlive = false;//是否一直保持连接
                request.Method = "GET";//请求方法
                request.ContentType = "application/json; charset=UTF-8";//请求内容返回格式和编码
                request.AutomaticDecompression = DecompressionMethods.GZip;//允许压缩内容，可以提高请求效率

                HttpWebResponse response = (HttpWebResponse)request.GetResponse();//获得响应
                Stream myResponseStream = response.GetResponseStream();//将响应内容生成流
                StreamReader myStreamReader = new StreamReader(myResponseStream, Encoding.UTF8);
                string retString = myStreamReader.ReadToEnd();//读取所有的响应内容并将结果生成字符串
                string r = Regex.Unescape(retString);//将返回的unicode转化为中文

                myStreamReader.Close();
                myResponseStream.Close();

                if (response != null)
                {
                    response.Close();
                }
                if (request != null)
                {
                    request.Abort();
                }

                return r;
            }

            public static string Post(string Url, string Data, string Referer)
            {
                HttpWebRequest request = (HttpWebRequest)WebRequest.Create(Url);
                request.Method = "POST";
                request.Referer = Referer;
                byte[] bytes = Encoding.UTF8.GetBytes(Data);
                string sb = Encoding.UTF8.GetString(bytes);
```

```
                //将URL编码后的字符串转化为字节
                //byte[] bytes = Encoding.UTF8.GetBytes(HttpUtility.UrlEncode(Data,Encoding.UTF8));
                request.Accept = "application/json, text/javascript, */*"; //"text/html, application/xhtml+xml,
*/*";
                request.ContentType = "application/json/x-www-form-urlencoded; charset=UTF-8";
                request.ContentLength = bytes.Length;
                string r = null;
                //获得请求流
                using (Stream myResponseStream = request.GetRequestStream())
                {
                    //将请求参数写入流
                    myResponseStream.Write(bytes, 0, bytes.Length);
                }
                // 获得响应流
                HttpWebResponse response = (HttpWebResponse)request.GetResponse();
                using (StreamReader myStreamReader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))
                {
                    string retString = myStreamReader.ReadToEnd();
                    r = Regex.Unescape(retString);//将返回的unicode转化为中文
                }
                //myStreamReader.Close();
                //myResponseStream.Close();
                if (response != null)
                {
                    response.Close();
                }
                if (request != null)
                {
                    request.Abort();
                }

                return r;
            }


        }
```

按键执行：

```
    private void button14_Click(object sender, EventArgs e)
        {
               string URL = "http://192.168.10.82:8585/execute?action=HLQC";
            string Data =
"{\"DATAFLAG\":\"01\",\"SN\":\"99093527**H231015AA10A01061\",\"PRID\":\"H005\",\"PNUMBERr\":\"HL06301\",\"GXID\
":\"B001\",\"GXNAME\":\"螺母浮高\",\"RESULT\":\"True\",\"CheckTime\":\"2023-10-04
00:00:00\",\"SubmitTime\":\"2023-10-04 00:00:00\",\"CreateTime\":\"2023-10-04 17:41:10\"}";
            string jsonStrlist = HttpWebResponseUtility.Post(URL, Data, "");
            Rootobject rtlist = Newtonsoft.Json.JsonConvert.DeserializeObject<Rootobject>(jsonStrlist);
            MessageBox.Show(rtlist.Message.ToString());

          }
```

### 2.2.4 测试方法

Postman或其它在线API测试平台工具，360浏览器等；

**URL:**

http://58.253.84.177:36118/execute?action=HLQC&SN=0235AFYA88H231013AA10G013&GXID=B002&PNUMBER=HL06301&PRID=H063&RESULT=TRUE&QC001=2.25&QC002=5.5369&QC003=12.69&GXNAME=螺母浮高&STARTTIME=2023-10-31 06:09:59:666&ENDTIME=2023-10-31 06:09:59:666&SUBMITTIME=2023-10-31 06:09:59:666

**JSON：**

{"SN":"0235AFYA88H231013AA10G013","GXID":"B002","PNUMBER":"HL06301","PRID":"H063","RESULT":"TRUE","QC001":"2.25","QC002":"5.5369","QC003":"12.69","GXNAME":"螺母浮高","STARTTIME":"2023-10-31 06:09:59:666","ENDTIME":"2023-10-31 06:09:59:666","SUBMITTIME":"2023-10-31 06:09:59:666"}

### 2.2.5 返回数据

1、成功返回

{"ResultTime":"'2023-10-17 13:38:55'","State":"true","Message":"成功导入数据中心库。","Fnote":"99093527**H231015AA10A01061"}

2、提交数据不规范返回：

{ "ResultTime":"'2023-10-17 14:01:33'","State":"false", "Message":"产品SN码不合法。", "Fnote":"0235AFYA88H231013AA10G013"}

----------全文完----------