# MES接口说明

MES服务通过MinMesSock.dll（vs2019 c++版本）提供接口，设备软件分别调用该DLL提供的接口函数实现入站和出站。

入站和出站接口函数传入的ip地址为MES服务器所在的机器ip，根据传入的SN、工位名称、槽位号组装json包，执行创建socket、connect服务器ip、send包、recv服务器返回、解析服务返回的json包、将解析后的调用结果（success或失败具体信息）返回调用端。

## 入站检测

设备软件**每轮测试扫描条码后**，通过调用该接口，执行MES的入站管控功能，根据函数返回bool值判断成功/失败，outputMsg返回具体字符串信息（“success”或“具体失败信息”），**返回成功继续测试，返回失败停止测试**。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口名称 | bool ValidateBoardWorkState(const char \*IpAddr, const char \*barcode, const char\* StationName, const char \*ProductName, char\* outputMsg, int nLengthMsg) | | | |
| 功能描述 | 入站检测，该条码是否应在当前工位测试 | | | |
| 输入参数 | 必选/可选 | 类型 | 长度 | 描述 |
| IpAddr | 必选 | char\* | NA | HonorMinMES服务器IP地址 |
| barcode | 必选 | string | NA | sn条码 |
| StationName | 必选 | string | NA | 工位名称 |
| *ProductName* | *可选* | *Char\** | *NA* | *产品名称* |
| nLengthMsg | 必选 | Int | NA | 输入的outputMSG的长度 |
| 输入/输出参数 | 必选/可选 | 类型 | 长度 | 描述 |
| outputMsg | 必选 | Char\* | NA | 调用成功返回“success”  调用失败返回“不是当前工位/测试已打最大次数” |

## 出站上报

设备软件每轮**所有测试结束**后，通过调用该接口，执行MES的出站功能，根据函数返回bool值判断成功/失败，outputMsg返回具体字符串信息（“success”或“具体失败信息”）。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口名称 | bool SubmitBoardWorkState(const char \*IpAddr, const char \*barcode, const char\* StationName, const char\* ProductName, int Slot\_ID, bool result, char\* outputMsg, int nLengthMsg) | | | |
| 功能描述 | 出站，上报该条码在当前工位/槽位本轮测试的结果 | | | |
| 输入参数 | 必选/可选 | 类型 | 长度 | 描述 |
| IpAddr | 必选 | char\* | NA | HonorMinMES服务器IP地址 |
| barcode | 必选 | string | NA | sn条码 |
| StationName | 必选 | string | NA | 工位名称 |
| *ProductName* | *可选* | *Char\** | *NA* | *产品名称* |
| *Slot\_ID* | *可选* | *int* | *1* | *槽位号* |
| result | 必选 | bool | 1 | true 本轮成功，false 本轮失败 |
| nLengthMsg | 必选 | Int | NA | 输入的outputMSG的长度 |
| 输入/输出参数 | 必选/可选 | 类型 | 长度 | 描述 |
| outputMsg | 必选 | Char\* | NA | 调用成功返回“success”  调用失败返回“服务器具体失败信息” |

# 设备软件调用接口说明

1. 设备软件参数设置，**增加一项是否使用MES**，参数名为**UsingMES**，该参数为true，设备软件开始测试和结束测试分别调用MinMesSock.dll的入站和出站接口函数，该参数为false，不调用MinMesSock.dll的接口函数
2. 设备软件参数设置，**增加一项MES的IP地址**，参数名为**MESIPAddress**，调用接口函数传入该IP地址参数值
3. 设备软件参数设置，**增加一项本设备的工位名称**，参数名为**StationName**，调用接口函数传入该StationName参数值，各个工位对应的名称如下

|  |  |
| --- | --- |
| 定位柱测试 | PINT |
| 螺丝机 | LSJ01 |
| 扭矩测试 | TORQUET |
| 转轴异响测试 | R&BT |
| 角度一致性测试 | ANGLET |
| 螺丝机 | LSJ02 |

1. 将接口文件MinMesSock.dll/MinMesSock.pdb和设备软件DLL放在同一目录
2. 加载MinMesSock.dll
3. 开始测试前调用入站接口函数ValidateBoardWorkState，测试结束调用出站接口函数SubmitBoardWorkState

# 设备软件调用接口示例

## 设备软件调用示例（VS C#示例）

1. 加载DLL

namespace TestCSharp

{

public static class Honor

{

// 入站检测

[DllImport("MinMesSock.dll", CallingConvention = CallingConvention.Cdecl)]

public static extern bool ValidateBoardWorkState(string IpAddr, string barcode, string StationName, string ProductName, [MarshalAs(UnmanagedType.LPStr)]StringBuilder ouputMsg, Int32 nLengthMsg);

// 出站上报

[DllImport("MinMesSock.dll", CallingConvention = CallingConvention.Cdecl)]

public static extern bool SubmitBoardWorkState(string IpAddr, string barcode, string StationName, string ProductName, Int32 Slot\_ID, Int32 result, [MarshalAs(UnmanagedType.LPStr)]StringBuilder outputMsg, Int32 nLengthMsg);

}

}

2、入站调用代码示例

private void button1\_Click(object sender, EventArgs e)

{

string strIP = "10.1.8.201"; // MES 服务器所在ip地址

string strSN = "1234567890abcdef02"; // 设备每轮测试的SN

string strStaionName = "PINT"; // 设备软件参数设置中本设备的工位名称StationName值

string strProductName = "product\_name"; // 当前设备测试的产品名称（可选）

StringBuilder chOutMsg = new StringBuilder(1024); // 接收服务端接口执行返回的具体信息

bool bResult = HonorMinMesSock.ValidateBoardWorkState(strIP, strSN, strStaionName, strProductName, chOutMsg, 1024);

if(bResult == false)

{

MessageBox.Show("入站失败");

}

else

{

MessageBox.Show("入站成功");

}

}

3、出站调用代码示例

private void button2\_Click(object sender, EventArgs e)

{

string strIP = "10.1.8.201"; // MES服务器所在ip地址

string strSN = "1234567890abcdef02";

string strStaionName = "PINT";

string strProductName = "product\_name";

Int32 slotID = 1; // 当前设备测试的槽位号（可选）

bool testResult = true; // 本轮测试结果; true 成功，false 失败

StringBuilder chOutMsg = new StringBuilder(1024);

bool bResult = HonorMinMesSock.SubmitBoardWorkState(strIP, strSN, strStaionName, strProductName, slotID, testResult, chOutMsg, 1024);

if(bResult == false)

{

MessageBox.Show("出站失败");

}

else

{

MessageBox.Show("出站成功");

}

}

## 设备软件调用示例（VS C++示例）

1. 加载DLL

CString strExePath = String2Cstring(getExePath()) + \_T("\\MinMesSock - x64.dll ");

m\_hSockDll = ::LoadLibrary(strExePath);

hValidBoardWorkStation = (ValidBoardWorkState)GetProcAddress(m\_hSockDll, "ValidateBoardWorkState");

hSubBoardWorkState = (SubBoardWorkState)GetProcAddress(m\_hSockDll, "SubmitBoardWorkState");

1. 入站调用代码示例

string strIP = 设备软件参数设置中的MES的IP地址值;

string strSN = 设备每轮测试的SN;

string strStaionName = 设备软件参数设置中本设备的工位名称StationName值;

string strProductName = 当前设备测试的产品名称（可选）;

char strMSGInfo[MAX\_PATH] = { 0 }; // 服务端接口执行返回的具体信息

bool result; // 服务端接口执行的结果：true 成功，false 失败

result = hValidateBoardWorkState(strIP.c\_str(), strSN.c\_str(), strStaionName.c\_str(), strProductName.c\_str(),strMSGInfo, sizeof(strMSGInfo));

strMSGInfo保存的服务端返回的具体信息示例如下：

工位不存在

OK

PINT工位不是当前测试工位

PINT工位已达到最大测试次数

多跳了工位，返回失败ANGLET工位不是当前测试工位

1. 出站调用代码示例

string strIP = 设备软件参数设置中的MES的IP地址值;

string strSN = 设备每轮测试的SN;

string strStaionName = 设备软件参数设置中本设备的工位名称StationName值;

string strProductName = 当前设备测试的产品名称（可选）;

int nSlotID = 当前设备测试的槽位号（可选）;

bool bTestResult = 本轮测试结果; // true 成功，false 失败

char strMSGInfo[MAX\_PATH] = { 0 }; // 服务端接口执行返回的具体信息

bool result; // 服务端接口执行的结果：true 成功，false 失败

result = hSubmitBoardWorkState(strIP.c\_str(), strSN.c\_str(), strStaionName.c\_str(), strProductName.c\_str(), nSlotID, bTestResult,

strMSGInfo, sizeof(strMSGInfo));

strMSGInfo保存的服务端返回的具体信息示例如下：

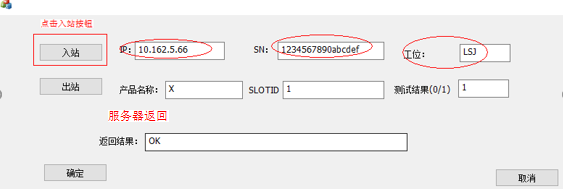
OK

“服务端异常信息”

## 设备软件模拟器示例

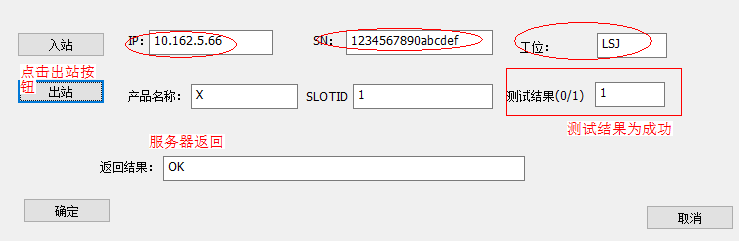
运行HonorMachineClient.exe（模拟设备软件）

1、入站模拟测试：



2、出站模拟测试：

（1）测试结果为成功的出站模拟测试



（2）测试结果为失败的出站模拟测试

