GQY机器人接口文档

(试用版)

宁波GQY视讯股份有限公司

**郑重声明**

本手册内容若有变动，恕不另行通知。未得到宁波GQY视讯股份有限公司

明确的书面许可，不得为任何目的、以任何形式或手段（电子的或机械的）复制或传播手册的任何部分。本文档可能涉及宁波GQY视讯股份有限公司

的专利（或正在申请的专利）、商标、版权或其他知识产权，除非得到新世纪机器人有限公司的明确书面许可协议，本文档不授予使用这些专利（或正在申请的专利）、商标、版权或其他知识产权的任何许可协议。本手册提及的其它产品和公司名称均可能是各自所有者的商标。

版权所有©宁波GQY视讯股份有限公司

目录

[GQY机器人接口文档 1](#_Toc492309024)

[版本信息 4](#_Toc492309025)

[概述 4](#_Toc492309026)

[目的 4](#_Toc492309027)

[范围 4](#_Toc492309028)

[GQY机器人开放资源和开发接口 5](#_Toc492309029)

[GQY机器人开放资源 5](#_Toc492309030)

[GQY机器人开发接口 6](#_Toc492309031)

[获得语音识别和语音理解结果接口 6](#_Toc492309032)

[访问语音合成接口 7](#_Toc492309033)

[导航控制接口(导航到某处，暂停和继续导航) 8](#_Toc492309034)

[导航结果接受接口 10](#_Toc492309035)

[表情与运动控制接口 12](#_Toc492309036)

[运动控制接口 18](#_Toc492309037)

[人脸识别接口 20](#_Toc492309038)

[超声波感知接口 21](#_Toc492309039)

[导航APP功能介绍 23](#_Toc492309040)

[定制知识库 24](#_Toc492309041)

[定制广告 24](#_Toc492309042)

[定制VIP识别 25](#_Toc492309043)

[附录 26](#_Toc492309044)

[附录1 26](#_Toc492309045)

# 版本信息

|  |  |  |
| --- | --- | --- |
| **版本** | **日期** | **说明** |
| V1.0 | 2017-7-4 | 试用版 |
| V1.1 | 2017-8-18 | 校正 |
| V1.2 | 2017-9-4 | 校正 |
| V1.3 | 2017-9-19 | 增加机器人版本差别 |
| V1.4 | 2018-3-28 | 增加关机指令 |

# 概述

GQY机器人接口文档是介绍GQY机器人对客户开放的资源和接口，指导客户如何使用这些接口以及和二次开发配套的软件和文档说明。

## 目的

本手册目录目的是帮助客户了解和开发GQY机器人，方便地使用GQY机器人的开放的资源和开发接口。

## 范围

本手册描述GQY机器人开发资源和接口，不描述机器人结构和与开放资源及接口无关的机器人功能。

# GQY机器人开放资源和开发接口

## GQY机器人开放资源

|  |  |  |
| --- | --- | --- |
| **开放资源** | **说明** | **开放额度** |
| 语音识别 | 客户程序可以请求语音识别的结果 | 无限制 |
| 语义理解 | 客户程序可以请求语义理解的结果；客户可以定制知识库 | 无限制 |
| 语音合成 | 客户程序可以请求发声 | 无限制 |
| 机器人状态控制 | 通过命令让机器人处于某种状态或者行为 | 无限制 |
| 导航控制及其结果的接受 | 客户程序可以请求导航功能和获悉导航结果；  结合导航APP实现导航到目的地 | 无限制，导航APP的介绍参考附录1 |
| 表情与动作控制 | * 客户程序可以请求眼睛和嘴部表情 * 客户程序可以请求单个头部动作和单个手臂动作；也可以请求整套动作 | * 眼睛表情（27种静态+11动态），嘴部表情（4种静态+6动态），表情图片具体参考[表情表格](#表情表格) * 2个头部动作和4个手臂动作 * 13套整体动作(卖萌、敬礼、飞吻等) |
| 运动控制 | 客户程序可以请求运动控制 | 前进,左转,右转，转一圈 |
| 超声波感知 | 机器人具有感知是否有人靠近和离开的能力，客户程序可以请求该服务 | 无限制 |
| 人脸识别 | 客户程序可以请求人脸识别的结果 | VIP号，年龄,性别,表情和颜值属性 |
| Windows 程序运行环境 | 在机器人核心服务不受影响的情况下，允许客户的Windows程序运行在GQY机器人的Windows10系统中。 | 有限 |
| （选配）外设访问 | 客户程序可以请求外设访问如打印机，身份证和银行卡读卡器 | 外设驱动驱动动态库参考附录1 |

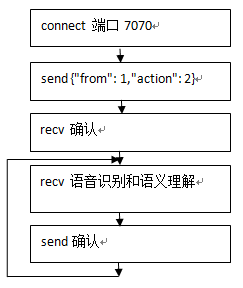
**资源表格**

## GQY机器人开发接口

**说明：使用该接口需要关闭CBC界面程序；中文编码为utf8。**

### 获得语音识别和语音理解结果接口

客户程序通过socket访问端口7070，具体流程图：



握手请求json:

{

"from":1,

"action": 2

}

语音识别和语音理解结果示例:

{

"question":"你叫什么名字",

"behavior":0,

"parameter" :"",

"content" : "我是GQY银行大堂经理小民",

"context" : "",

"url" : ""

}

返回参数说明

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| question | string | 语音识别的结果(问题) |
| behavior | int | 在此应用中可以忽略 |
| parameter | string | 在此应用中可以忽略 |
| content | string | 语义理解的结果(答案) |
| context | string | 在此应用中可以忽略 |
| url | string | 在此应用中可以忽略 |

c语言例程：

const char \*req=”{\"from\":1,\"action\": 2}”;

char buff[1024];

SOCKET client\_socket = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

struct sockaddr\_in serAddrsound;

serAddrsound.sin\_family = AF\_INET;

serAddrsound.sin\_port = htons(7070);

serAddrsound.sin\_addr.S\_un.S\_addr = inet\_addr("127.0.0.1")//surface IP

client\_socket = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

connect(client\_socket, (struct sockaddr \*)&serAddrsound, sizeof(serAddrsound));

send(client\_socket, req,strlen(req) , 0);

recv(client\_socket ,”ok”,2,0)

while(1)

{

recv(client\_socket, buff, 1024, 0);

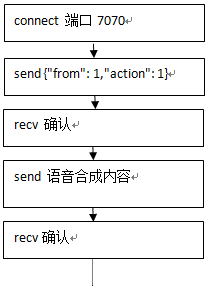
send(client\_socket, “ok”,2 , 0);

// process the result

}

### 访问语音合成接口

客户程序通过socket访问端口7070，具体流程图：



握手请求json:

{

"from":1,

"action": 1

}

语音合成内容示例:

{

"content":"你好，欢迎来到GQY公司"

}

语音合成请求参数说明

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| content | string | 发声内容 |

c语言例程：

const char \*req=”{\"from\":1,\"action\": 1}”;

char buff[1024];

SOCKET client\_socket = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

struct sockaddr\_in serAddrsound;

serAddrsound.sin\_family = AF\_INET;

serAddrsound.sin\_port = htons(7070);

serAddrsound.sin\_addr.S\_un.S\_addr = inet\_addr("127.0.0.1")//填写机器人平板IP

client\_socket = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

connect(client\_socket, (struct sockaddr \*)&serAddrsound, sizeof(serAddrsound));

send(client\_socket, req,strlen(req) , 0);

recv(client\_socket ,buff ,1024,0)//接受确认

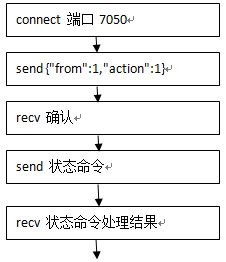
const char \*ttsB=”{\”content\”: "你好，欢迎来到GQY公司"}” //xxx是发声的utf8

send(client\_socket, ttsB,strlen(ttsB) , 0);

recv(client\_socket, buff, 1024, 0);//接受确认。

### 机器人状态控制(关机和跟随)

客户程序通过socket访问端口7050，具体流程：



握手请求json:

{

"from":1,

"action": 1

}

状态命令示例：

{

"from": 1,

"cmd":xxx,

"subcmd":yyy

}

状态命令参数说明

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| from | int | 固定为1 |
| cmd | int | 状态命令xxx：关机为81，跟随为68 |
| subcmd | int | yyy：10，无意义，保留。 |

关机命令处理结果示例：

{

"from": 3,

"cmd":81,

"subcmd":10，

"resp":1

}

关机命令处理结果参数说明

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| from | int | 固定为3 |
| cmd | int | 关机命令81 |
| subcmd | int | 10，无意义，保留 |
| resp | int | 1表示接受状态命令，  2表示拒绝状态 |

C语言例程：

const char \*req=”{\"from\":1,\"action\": 1}”;

char buff[1024];

SOCKET client\_socket = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

struct sockaddr\_in serAddrsound;

serAddrsound.sin\_family = AF\_INET;

serAddrsound.sin\_port = htons(7050);

serAddrsound.sin\_addr.S\_un.S\_addr = inet\_addr("127.0.0.1")//填写机器人平板IP

client\_socket = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

connect(client\_socket, (struct sockaddr \*)&serAddrsound, sizeof(serAddrsound));

send(client\_socket, req,strlen(req) , 0);

recv(client\_socket ,buff,1024,0);//接受确认

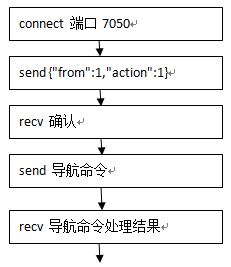
const char \*n1=”{\"from\": 1,\"cmd\":81,\"subcmd\":10};//关闭机器人

send(client\_socket, n1,strlen(n1) , 0);

recv(client\_socket, buff, 1024, 0);//接受状态命令处理结果，查看resp键值是接受还是拒绝。

### 导航控制接口(导航到某处，暂停和继续导航)

客户程序通过socket访问端口7050，具体流程：



握手请求json:

{

"from":1,

"action": 1

}

导航命令示例：

{

"from": 1,

"cmd":73,

"subcmd":10

}

导航命令参数说明

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| from | int | 固定为1 |
| cmd | int | 导航命令，固定为73 |
| subcmd | int | * 0-20 机器人导航到达位置号，使用导航APP设置； * 0xff，暂停导航(进入了导航模式有效，即发过0-20导航点命令，且没有到达预定位置) * 0xfe，继续导航(进入了导航模式有效) |

导航命令处理结果示例：

{

"from": 3,

"cmd":73,

"subcmd":10，

"resp":1

}

导航命令处理结果参数说明

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| from | int | 固定为3 |
| cmd | int | 导航命令，固定为73 |
| subcmd | int | 导航命力下发的导航点或者0xff/0xfe |
| resp | int | 1表示接受导航，  2表示拒绝导航 |

C语言例程：

const char \*req=”{\"from\":1,\"action\": 1}”;

char buff[1024];

SOCKET client\_socket = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

struct sockaddr\_in serAddrsound;

serAddrsound.sin\_family = AF\_INET;

serAddrsound.sin\_port = htons(7050);

serAddrsound.sin\_addr.S\_un.S\_addr = inet\_addr("127.0.0.1")//填写机器人平板IP

client\_socket = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

connect(client\_socket, (struct sockaddr \*)&serAddrsound, sizeof(serAddrsound));

send(client\_socket, req,strlen(req) , 0);

recv(client\_socket ,buff,1024,0);//接受确认

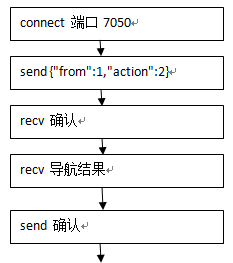
const char \*n1=”{\"from\": 1,\"cmd\":73,\"subcmd\":1};//导航到一号地点

send(client\_socket, n1,strlen(n1) , 0);

recv(client\_socket, buff, 1024, 0);//接受导航命令处理结果，查看resp键值是接受还是拒绝

### 导航结果接受接口

客户程序通过socket访问端口7050，具体流程：



握手请求json:

{

"from":1,

"action": 2

}

导航结果示例:

{

"from": 3,

"cmd":73,

"subcmd":10,

"resp":3

}

导航结果参数说明:

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| from | int | 固定为3 |
| cmd | int | 导航命令，固定为73 |
| subcmd | int | 导航命力下发的导航点(0-20)或者0xff/0xfe |
| resp | int | 1表示导航正在进行  2表示导航失败  3表示导航成功  4表示导航临时遇到障碍 |

c语言例程：

const char \*req=”{\"from\":1,\"action\": 2}”;

const char \*n1="ok";

char buff[1024];

SOCKET client\_socket = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

struct sockaddr\_in serAddrsound;

serAddrsound.sin\_family = AF\_INET;

serAddrsound.sin\_port = htons(7050);

serAddrsound.sin\_addr.S\_un.S\_addr = inet\_addr("127.0.0.1")//填写机器人平板IP

client\_socket = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

connect(client\_socket, (struct sockaddr \*)&serAddrsound, sizeof(serAddrsound));

send(client\_socket, req,strlen(req) , 0);

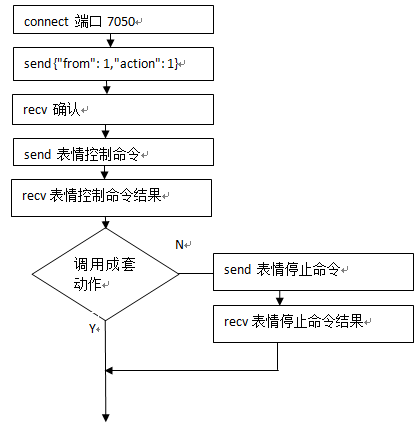
recv(client\_socket ,buff,1024,0);//接受确认

recv(client\_socket, buff, 1024, 0);//接受导航结果，查看

send(client\_socket, n1,strlen(n1) , 0);

### 表情与运动控制接口

客户程序通过socket访问端口7050，具体流程：



握手请求json:

{

"from":1,

"action": 1

}

表情控制命令示例：

{

"from":1,

"cmd":102,

"subcmd":1792

}

表情控制命令参数说明：

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| from | int | 固定为1 |
| cmd | int | 动作命令，固定为102 |
| subcmd | int | 表情或者动作号 |

表情控制命令结果示例：

{

"from": 1,

"cmd":102,

"subcmd":1792,

"resp":1

}

表情控制命令结果参数说明：

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| from | int | 固定为1 |
| cmd | int | 动作命令，固定为102 |
| subcmd | int | 表情或者动作号 |
| resp | int | 1 表示接受动作指令  2 表示拒绝动作指令 |

表情停止命令示例：

{

"from": 1,

"cmd":65,

"subcmd":0

}

表情控制命令参数说明：

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| from | int | 固定为1 |
| cmd | int | 动作命令，固定为65 |
| subcmd | int | 值为0，忽略它 |

表情停止命令结果示例：

{

"from": 1,

"cmd":65,

"subcmd":0,

"resp":1

}

表情停止命令结果参数说明：

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| from | int | 固定为1 |
| cmd | int | 动作命令，固定为65 |
| subcmd | int | 值为0，忽略它 |
| resp | int | 1 表示接受动作指令  2 表示拒绝动作指令 |

表情或者动作号的取值格式为三个字节组成的整数如下图：

|  |  |  |
| --- | --- | --- |
| 高 | 中 | 低 |

每个字节的取值范围:

**低字节**：1-12，可以调用GQY定制好的整套表情和动作；和其他字节是互斥，如果该字节为非0，其他字节必须为0。



**中字节**：1-46，可以调用眼睛表情；该字节可以和高字节做或运算，实现眼睛和嘴巴动作组合。

|  |  |  |
| --- | --- | --- |
| **中字节值** | **表达意思** | **眼睛形状** |
| 0 | welcome | 1_40 |
| 1 | 桃心 | 1_08 |
| 2 | 三角 | 1_63 |
| 3 | 微笑 | 1_79 |
| 4 | 失落 | C:\Users\Administrator\Documents\WeChat Files\jxshln\Files\图像\单只眼睛轮廓_35.png |
| 5 | 眼睫毛 | 1_35 |
| 6 | 圆圈 | 1_16 |
| 7 | 横线 | 1_65 |
| 8 | Hi | 1_13 |
| 9 | Z | 1_68 |
| 10 | 波浪 | 1_88 |
| 11 | 叉 | 1_03 |
| 12 | 电池没电 | 1_57 |
| 13 | 哭泣 | c:\Temp\WeChat Files\362823253350320111.png |
| 14 | 嚎啕大哭 | C:\Users\Administrator\Documents\WeChat Files\jxshln\Files\图像\3_83.png |
| 15 | 闪电 | 1_84 |
| 16 | 问号 | 1_33 |
| 17 | 向右看 | 1_43 |
| 18 | 向左看 | 1_11 |
| 19 | 眩晕 | 1_59 |
| 20 | 音乐1 | C:\Users\Administrator\Documents\WeChat Files\jxshln\Files\图像(1)\单只眼睛轮廓_56.png |
| 21 | 音乐2 | C:\Users\Administrator\Documents\WeChat Files\jxshln\Files\图像(1)\4_17.png |
| 22 | 感叹号 | 1_82 |
| 23 | 充电 | 1_37 |
| 24 | 心电图 | 1_70 |
| 25 | 警示符 | C:\Users\Administrator\Documents\WeChat Files\jxshln\Files\9_84\9_84.png |
| 26 | byebye | 1_78 |
| 27 | Hi动画 |  |
| 28 | 音乐条动画 |  |
| 29 | 心电图动画 |  |
| 30 | 波浪线动画 |  |
| 31 | 哭泣动画 |  |
| 32 | 向左看动画 |  |
| 33 | 向右看动画 |  |
| 34 | 桃心动画 |  |
| 35 | byebye动画 |  |
| 36 | 睡觉动画 |  |
| 37 | 思考动画 |  |
| 38 | 程序更新动画 |  |
| 39 | 眨眼睛一次动画 |  |
| 40 | 眨眼睛两次动画 |  |
| 41 | 眨眼睛两次后熄灭动画 |  |
| 42 | 清屏 |  |
| 43 | 故障 | 1_06 |
| 44 | 电量不足动画 |  |
| 45 | 充电动画 |  |

**表情表格**

**高字节**：1-14，可以调用嘴巴表情；

|  |  |  |
| --- | --- | --- |
| **高字节值** | **表达意思** | **嘴巴形状** |
| 0 | 微笑 | 2_59 |
| 1 | 失落 | 2_25 |
| 2 | 口型 | 2_86 |
| 3 | 横线 | 2_22 |
| 4 | 清屏 | 2_57 |
| 5 | 嘴部说话1 |  |
| 6 | 嘴部说话2 |  |
| 7 | 嘴部说话3 |  |
| 8 | 嘴部说话4 |  |
| 9 | 嘴部说话5 |  |
| 10 | 嘴部说话6 |  |
| 11 | 嘴部说话7 |  |
| 12 | 嘴部说话8 |  |

C语言例程：

让眼睛显示圈圈：

const char \*req=”{\"from\":1,\"action\": 1}”;

char buff[1024];

SOCKET client\_socket = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

struct sockaddr\_in serAddrsound;

serAddrsound.sin\_family = AF\_INET;

serAddrsound.sin\_port = htons(7050);

serAddrsound.sin\_addr.S\_un.S\_addr = inet\_addr("127.0.0.1")//填写机器人平板IP

client\_socket = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

connect(client\_socket, (struct sockaddr \*)&serAddrsound, sizeof(serAddrsound));

send(client\_socket, req,strlen(req) , 0);

recv(client\_socket ,”ok”,2,0)//接受确认

send(client\_socket, req,strlen(req) , 0);

recv(client\_socket ,buff,1024,0);//接受确认

const char \*n1=”{\"from\": 1,\"cmd\":102,\"subcmd\":1792}; //1792的16进制为0x700

send(client\_socket, n1,strlen(n1) , 0); //动作控制命令

recv(client\_socket, buff, 1024, 0);//接受表情控制命令结果，查看resp键值是接受还是拒绝

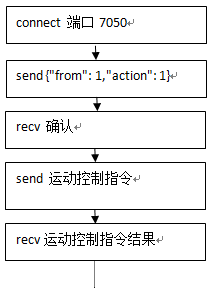
const char \*n2=”{\"from\": 1,\"cmd\":65,\"subcmd\":}; //停止动作命令

send(client\_socket, n2,strlen(n2) , 0);

recv(client\_socket, buff, 1024, 0); //停止动作命令结果

### 运动控制接口

客户程序通过socket访问端口7050，可以控制机器人前进，左转，右转以及转一圈。具体流程：



握手请求json:

{

"from":1,

"action": 1

}

运动控制指令示例：

{

"from":1,

"cmd":69,

"subcmd":1

}

表情控制命令参数说明：

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| from | int | 固定为1 |
| cmd | int | 运动控制命令，固定为69 |
| subcmd | int | 1表示左转  2 表示右转  3 表示转圈  4 表示前进 |

运动控制指令结果示例：

{

"from":1,

"cmd":69,

"subcmd":1,

"resp":1

}

运动控制指令结果参数说明：

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| from | int | 固定为1 |
| cmd | int | 运动控制命令，固定为69 |
| subcmd | int | 1表示左转  2 表示右转  3 表示转圈  4 表示前进 |
| resp | int | 1 表示接受  2 表示拒绝 |

C语言例程：

const char \*req=”{\"from\":1,\"action\": 1}”;

char buff[1024];

SOCKET client\_socket = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

struct sockaddr\_in serAddrsound;

serAddrsound.sin\_family = AF\_INET;

serAddrsound.sin\_port = htons(7050);

serAddrsound.sin\_addr.S\_un.S\_addr = inet\_addr("127.0.0.1")//填写机器人平板IP

client\_socket = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

connect(client\_socket, (struct sockaddr \*)&serAddrsound, sizeof(serAddrsound));

send(client\_socket, req,strlen(req) , 0);

recv(client\_socket ,buff,1024,0);//接受确认

const char \*n1=”{\"from\": 1,\"cmd\":69,\"subcmd\":1};//左转

send(client\_socket, n1,strlen(n1) , 0);

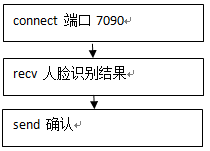
recv(client\_socket, buff, 1024, 0);//接受左转命令结果，查看resp键值是接受还是拒绝

### 人脸识别接口

**人脸识别接口在A2和A3接口是不同的。请认清机器人版本，选择对应的接口。**

**A2机器人**：

客户程序通过socket访问端口7090，可以获得人脸识别的结果。具体流程：



人脸识别结果示例：

{

"id": 300011111,

"gender":1,

"emotion":1,

"age":28,

"attr":96

}

人脸识别结果参数说明：

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| id | int | VIP 号码 |
| gender | int | 性别，0表示女，1表示男，2表示不确定 |
| emtion | int | 0表示愤怒  1表示平静  2表示困惑  3表示厌恶  4表示高兴  5表示悲伤  6表示惊恐  7表示诧异  8表示斜视  9表示尖叫 |
| age | int | 年龄，0-100 |
| attr | int | 颜值，0-100 |

c语言例程：

char buff[1024];

SOCKET client\_socket = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

struct sockaddr\_in serAddrsound;

serAddrsound.sin\_family = AF\_INET;

serAddrsound.sin\_port = htons(7090);

serAddrsound.sin\_addr.S\_un.S\_addr = inet\_addr("127.0.0.1")//填写机器人平板IP

client\_socket = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

connect(client\_socket, (struct sockaddr \*)&serAddrsound, sizeof(serAddrsound));

while(1)

{

recv(client\_socket ,buff,1024,0);//接受人脸识别结果

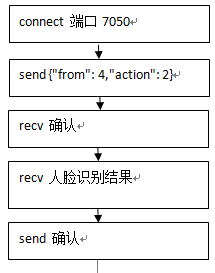
send(client\_socket, "ok",2, 0);//发送确认

//处理人脸识别结果

}

**A3机器人**：

客户程序通过socket访问端口7050，可以获得人脸识别的结果。具体流程：



人脸识别结果示例：

{

"id": 300011111,

"gender":99,

"age":26,

"emotion\_t":4,

"mouth\_open":

}

人脸识别结果参数说明：

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| id | int | VIP 号码 |
| gender | int | 性别为男性的百分比置信度(0-100) |
| emtion\_t | int | 0表示愤怒  1表示平静  2表示困惑  3表示厌恶  4表示高兴  5表示悲伤  6表示惊恐  7表示诧异  8表示斜视  9表示尖叫 |
| age | int | 年龄，0-100 |
| attr | int | 颜值，0-100 |

char buff[1024];

const char \*req=”{\"from\":4,\"action\": 2}”;

SOCKET client\_socket = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

struct sockaddr\_in serAddrsound;

serAddrsound.sin\_family = AF\_INET;

serAddrsound.sin\_port = htons(7050);

serAddrsound.sin\_addr.S\_un.S\_addr = inet\_addr("127.0.0.1")//填写机器人平板IP

client\_socket = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

connect(client\_socket, (struct sockaddr \*)&serAddrsound, sizeof(serAddrsound));

send(client\_socket, req,strlen(req) , 0);

recv(client\_socket ,buff,1024,0);//接受确认

while(1)

{

recv(client\_socket ,buff,1024,0);//接受人脸识别结果

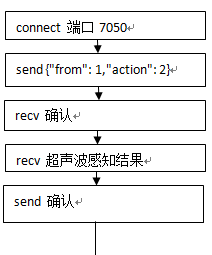
send(client\_socket, "ok",2, 0);//发送确认

//处理人脸识别结果

}

### 超声波感知接口

客户程序通过socket访问端口7050，感知机器人前面有人进入或者离开。具体流程：



请求json:

{

"from":1,

"action",2

}

超声波感知结果示例：

{

"from":3,

"cmd":80,

"subcmd":1,

"resp":1

}

超声波感知结果参数

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **说明** |
| from | int | 固定值3 |
| cmd | int | 固定值80 |
| subcmd | int | 该值得掩码是0x400，与运算后的结果为0表示人离开，为1表示人进来 |
| resp | int | 固定值1，忽略它 |

c语言例程：

char buff[1024];

const char \*req=”{\"from\":1,\"action\": 2}”;

SOCKET client\_socket = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

struct sockaddr\_in serAddrsound;

serAddrsound.sin\_family = AF\_INET;

serAddrsound.sin\_port = htons(7050);

serAddrsound.sin\_addr.S\_un.S\_addr = inet\_addr("127.0.0.1")//填写机器人平板IP

client\_socket = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

connect(client\_socket, (struct sockaddr \*)&serAddrsound, sizeof(serAddrsound));

send(client\_socket, req,strlen(req) , 0);

recv(client\_socket ,”ok”,2,0)//接受确认

recv(client\_socket ,buff,1024,0);//接受超声波感知结果

send(client\_socket, "ok",2, 0);//发送确认

//接受结果和0x400掩码作与运算，获得超声波感知结果

# 导航APP功能介绍

GQY提供了手机app实现建图和导航点设置。结合导航接口，客户可以全方位控制导航。本手册只介绍和导航点设置相关功能，其它建图功能参考附录1

进入建图模式，建图后标注导航点

1输入初始导航点的名字，机器人上电点和充电点



2 反复输入其它导航点



3 确认后保存地图和导航点



# 定制知识库

GQY机器人提供了便捷的定制语义理解知识库方法，具体步骤如下:

1. 在U盘中，建立目录：新知识库

2. 在客户知识库目录下创建exec表格文件，文件名如：1.xls，文件夹中可以多个exec表格文件;文件名不能是中文。

3. 每个exec文件有两列：问题和答案，表必须保留前两行表头内容，从第三行开始填入自己定义的问题和答案。

4. 在name列填写自定义问题，在对应的content列填写自定义答案，多个问题或答案可用“|”分隔，例如name填“今天天气|天气怎么样|天气”，对应content填“天气晴|晴空万里|好天气”，每一句句首句尾均不加标点；

在github中可以下载完整execl表格文件示例，参考附录1。

5. 参考附录1中的客户知识库目录。

6. 插入U盘到机器人充电座后盖的USB接口，如下图红圈标示处：



7. 重起Surface平板，完成客户知识库的定制

# 定制广告

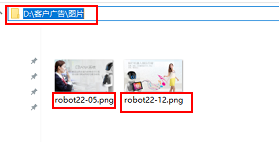
GQY机器人提供了便捷的定制客户广告方法，具体步骤如下:

1. 在U盘中，建立目录：客户广告\图片，客户广告\视频

2. 客户广告\图片目录下保存客户的广告图片，格式为png，分辨率推荐为1600\*1000

3. 客户广告\视频目录下保存客户的广告视频，格式为mp4

4. 如下图：



5. 插入U盘到机器人充电座后盖的USB接口，如下图红圈标示处：



7. 重起Surface平板，完成客户广告的定制。

# 定制VIP识别

GQY机器人提供了便捷的定制客户VIP识别，具体步骤如下:

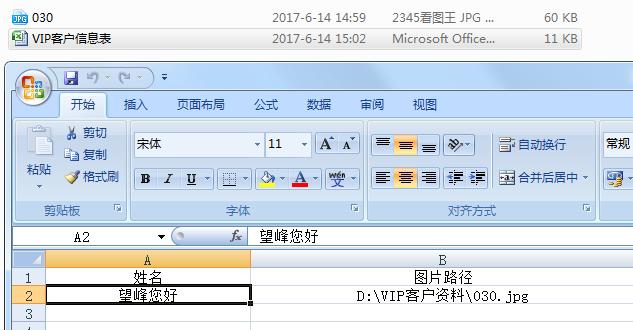
更换主题和增加VIP数据的方法。具体的使用方法如下：

1. 在U盘中，建立目录：VIP客户资料。

2. 在目录中保存VIP图片,格式为jpg，分辨率为1280\*960左右，大小为500K左右，单个人脸的正面照。

3. 创建VIP客户信息表.xlsx 文件，建立照片和称呼的对应。

4. 示例: VIP客户资料目录下的030.jpg 照片和对应的称呼文档。



4.插入U盘到机器人充电座后盖的USB接口，如下图红圈标示处：



7.打开VIP录入程序，完成VIP识别定制。

# 附录

## 附录1

该附录包括目录和文件：

1. 外设：包括了外设的帮助文档和头文件

2. 客户知识库：包括了execl文件，展示如何编写知识库

3. 导航APP说明:介绍导航APP功能和使用方法

下载地址：https://github.com/43970117/GQY