

# GQY 机器人接口文档

(试用版)

新世纪机器人有限公司

## 郑重声明

本手册内容若有变动，恕不另行通知。未得到新世纪机器人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。本文档可能涉及新世纪机器人有限公司的专利（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© 新世纪机器人有限公司

目录

GQY 机器人接口文档 ..... 1

版本信息 ..... 4

概述 ..... 4

    目的 ..... 4

    范围 ..... 4

GQY 机器人开放资源和开发接口 ..... 4

    GQY 机器人开放资源 ..... 4

    GQY 机器人开发接口 ..... 5

导航 APP 功能介绍 ..... 16

定制知识库 ..... 17

定制广告 ..... 18

定制 VIP 识别 ..... 19

附录 ..... 19

    附录 1 ..... 19

# 版本信息

| 版本   | 日期       | 说明  |
|------|----------|-----|
| V1.0 | 2017-6-8 | 试用版 |

## 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           |
| 导航控制及其结果的接 | 客户程序可以请求导航                | 无限制，导航 APP 的介 |

|                |  |   |
|----------------|--|---|
| 受              | 功能和获悉导航结果；结合导航 APP 实现导航到目的地                                  | 绍参考附录 1   |
| 表情控制           | 客户程序可以请求眼睛和嘴部表情  | 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 <a href="#">表情表格</a> |
| 动作控制           | 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                              | 2 个头部动作，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/后退、左转/右转   |
| 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   |

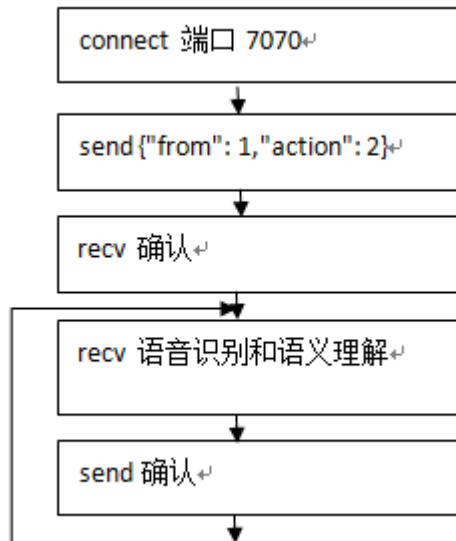
资源表格

## 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。

### 1. 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：



客户程序循环接受语音识别和语音理解结果的 json 格式：

`{"question": "xxx", "behavior": <行为>, "parameter" : "", "content" : "xxx", "context" : "", "url" : ""}`，其中 `question` 的键值是语音识别的结果，`content` 的键值是语音理解的结果，`url` 的键值是音乐资源链接。<行为>取值为 38 时，json 包中只有 `question`，没有 `content`。

c 语言例程：

```
const char *req="{\"from\":1,\"action\": 2}";
char buff[1024];
```

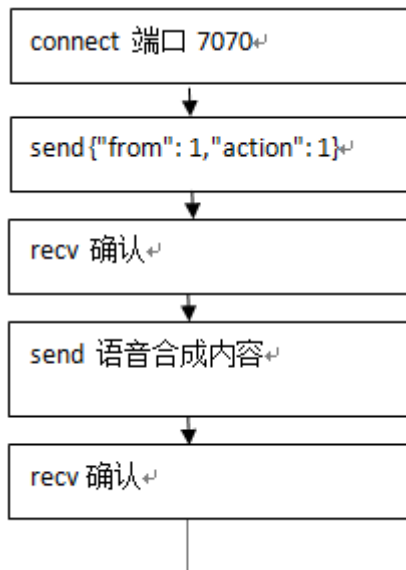
```
SOCKET client_socket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
struct sockaddr_in serAddrsound;
serAddrsound.sin_family = AF_INET;
serAddrsound.sin_port = htons(7070);
serAddrsound.sin_addr.S_un.S_addr = inet_addr("127.0.0.1");//填写机器人平板 IP
client_socket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
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
while(1)
{
    recv(client_socket, buff, 1024, 0);//获得语音识别和语义理解结果
    send(client_socket, "ok", 2, 0);//发送确认
}
```

```
/*处理 buff 里面的语音识别结果和语音理解结果 2*/
}
```

## 2. 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：



客户程序发送语音合成内容的编码为 utf8，格式为：{"content":xxx}

c 语言例程：

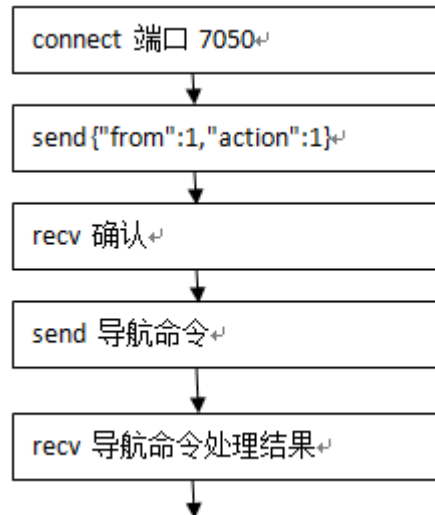
```
const char *req="{\"from\":1,\"action\": 1}";
char buff[1024];
```

```
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\": xxx}" //xxx 是发声的 utf8
send(client_socket, ttsB, strlen(ttsB), 0);
recv(client_socket, buff, 1024, 0);//接受确认。
```

### 3. 导航控制接口

客户程序通过 socket 访问端口 7050，具体流程：



导航命令: {"from": 1,"cmd":73,"subcmd":<导航目标点>}; <导航目标点>为导航点 ID，取值为 0-20, 使用导航 APP 设置导航点。

导航命令处理结果格式：

{"from": 3,"cmd":73,"subcmd":<导航目标点>,"resp":<接受否>}, <接受否>取值 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
```

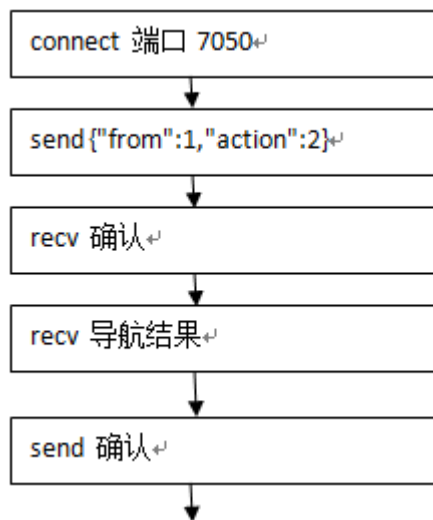
```
const char *n1="{\"from\": 1,\"cmd\":73,\"subcmd\":1}";//导航到一号地点
```

```
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
```

#### 4. 导航结果接受接口

客户程序通过 socket 访问端口 7050， 具体流程：



导航结果格式：

{"from": 3,"cmd":73,"subcmd":<导航目标点>,"resp":<导航结果>}, <导航结果>

<导航结果>取值 1 表示正在进行,2 表示导航失败,3 表示导航成功。

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");//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));
```

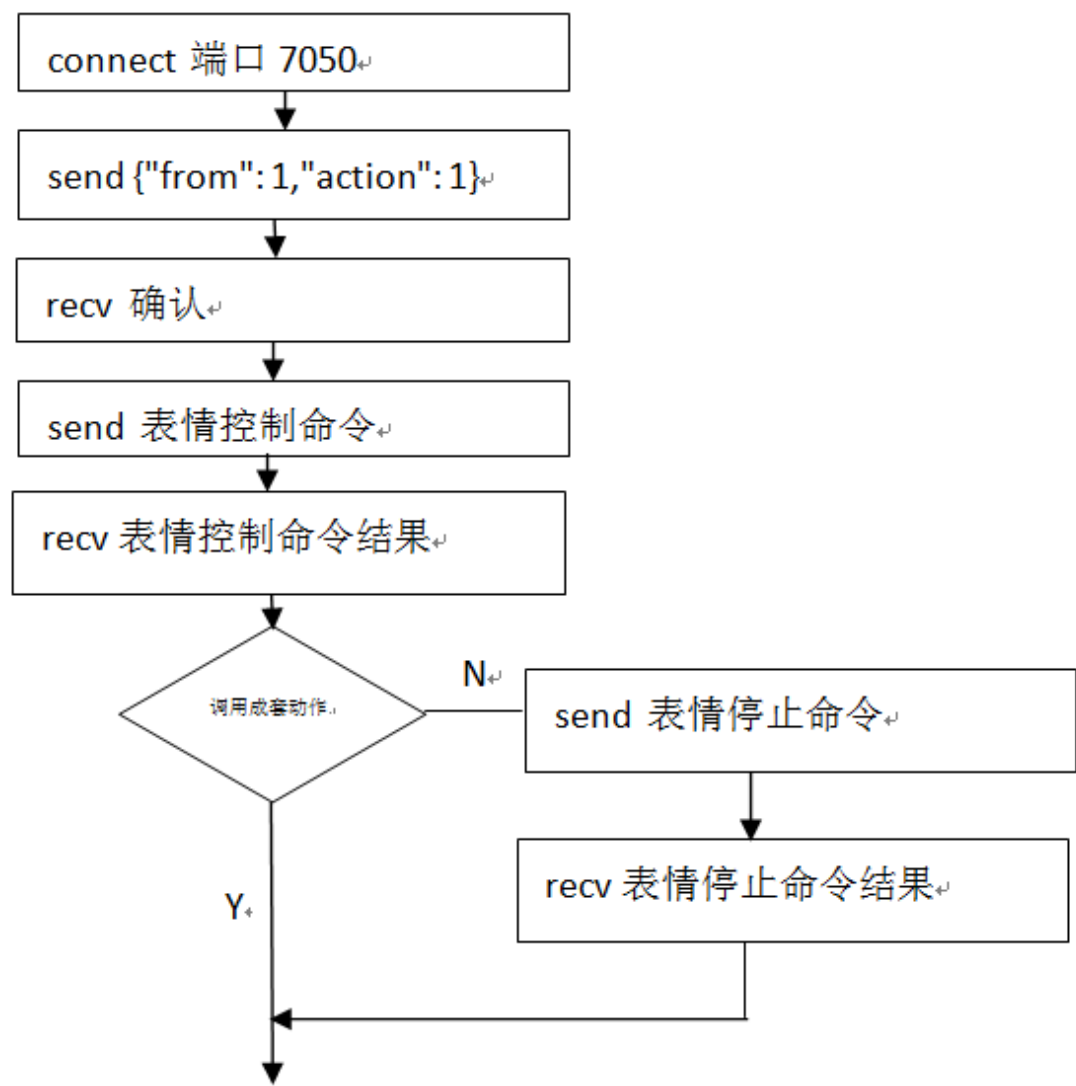
```
send(client_socket, req,strlen(req) , 0);
```

```
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);
```

5. 表情控制接口

客户程序通过 socket 访问端口 7050，具体流程：



表情控制命令格式：{"from": 1,"cmd":102,"subcmd":<表情或者动作 ID>}

<表情或者动作 ID>的取值格式为三个字节组成的整数：







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









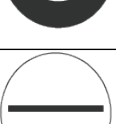
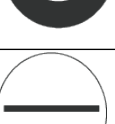
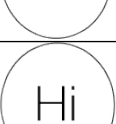
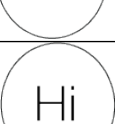
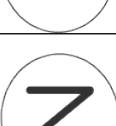

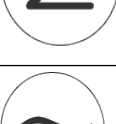

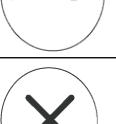



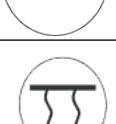

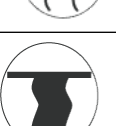

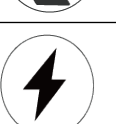

每个字节的取值范围:

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。

| 低字节值 | 动作意义   |
|------|--------|
| 1    | 指示平板   |
| 2    | 解答问题   |
| 3    | 卖萌 1   |
| 4    | 卖萌 2   |
| 5    | 迎宾     |
| 6    | 大屏介绍   |
| 7    | 请      |
| 8    | 舞蹈 1   |
| 9    | 握手     |
| 10   | 敬礼     |
| 11   | 摆 pose |
| 12   | 舞蹈 2   |

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  |
|------|---------|---|
| 1    | welcome |   |
| 2    | 桃心      |   |
| 3    | 三角      |   |






|    |      |   |   |
|----|------|---|---|
| 4  | 微笑   |    |    |
| 5  | 失落   |    |    |
| 6  | 眼睫毛  |    |    |
| 7  | 圆圈   |    |    |
| 8  | 横线   |    |    |
| 9  | Hi   |    |    |
| 10 | Z    |   |   |
| 11 | 波浪   |  |  |
| 12 | 叉    |  |  |
| 13 | 电池没电 |  |  |
| 14 | 哭泣   |  |  |
| 15 | 嚎啕大哭 |  |  |
| 16 | 闪电   |  |  |

|    |           |   |  |
|----|-----------|---|--|
| 17 | 问号        |    |    |
| 18 | 向右看       |    |    |
| 19 | 向左看       |    |    |
| 20 | 眩晕        |    |    |
| 21 | 音乐 1      |    |    |
| 22 | 音乐 2      |    |    |
| 23 | 感叹号       |   |   |
| 24 | 充电        |  |  |
| 25 | 心电图       |  |  |
| 26 | 警示符       |  |  |
| 27 | byebye    |  |  |
| 28 | Hi 动画     |   |  |
| 29 | 音乐条动画     |   |  |
| 30 | 心电图动画     |   |  |
| 31 | 波浪线动画     |   |  |
| 32 | 哭泣动画      |   |  |
| 33 | 向左看动画     |   |  |
| 34 | 向右看动画     |   |  |
| 35 | 桃心动画      |   |  |
| 36 | byebye 动画 |   |  |
| 37 | 睡觉动画      |   |  |

|    |            |  |
|----|------------|--|
| 38 | 思考动画       |  |
| 39 | 程序更新动画     |  |
| 40 | 眨眼睛一次动画    |  |
| 41 | 眨眼睛两次动画    |  |
| 42 | 眨眼睛两次后熄灭动画 |  |
| 43 | 清屏         |  |
| 44 | 故障         |  |
| 45 | 电量不足动画     |  |
| 46 | 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  |
|------|--------|---|
| 1    | 微笑     |   |
| 2    | 失落     |  |
| 3    | 口型     |  |
| 4    | 横线     |  |
| 5    | 清屏     |  |
| 6    | 嘴部说话 1 |   |
| 7    | 嘴部说话 2 |   |
| 8    | 嘴部说话 3 |   |
| 9    | 嘴部说话 4 |   |
| 10   | 嘴部说话 5 |   |
| 11   | 嘴部说话 6 |   |
| 12   | 嘴部说话 7 |   |
| 13   | 嘴部说话 8 |   |

表情控制命令结果格式：{"from": 1,"cmd":102,"subcmd":<表情或者动作

ID>,"resp":<接受否>}， <接受否>取值 1 表示接受， 2 表示拒绝。

表情停止命令格式： {"from": 1,"cmd":65,"subcmd":}

表情停止命令结果格式： {"from": 1,"cmd":65,"subcmd":,"resp":<接受否>},

<接受否>取值

1 表示接受， 2 表示拒绝。

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 serAddrssound;
serAddrssound.sin_family = AF_INET;
serAddrssound.sin_port = htons(7070);
serAddrssound.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 *)&serAddrssound, sizeof(serAddrssound));
```

```
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
```

## 6. 人脸识别接口

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：



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(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));

```

```

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
}

```

人脸识别结果格式: {"id": xxxxxx, "gender":<性别>, "emotion":<表情>  
,"age":<年龄>,"attr":<颜值>

xxxxxx 表示 VIP 号, 整数型

<性别>取值: 0 表示女, 1 表示男, 2 表示不确定

<表情>取值:

| 表情值 | 表情意义 |
|-----|------|
| 0   | 愤怒   |
| 1   | 平静   |
| 2   | 困惑   |
| 3   | 厌恶   |
| 4   | 高兴   |
| 5   | 悲伤   |
| 6   | 惊恐   |



|   |    |
|---|----|
| 7 | 诧异 |
| 8 | 斜视 |
| 9 | 尖叫 |

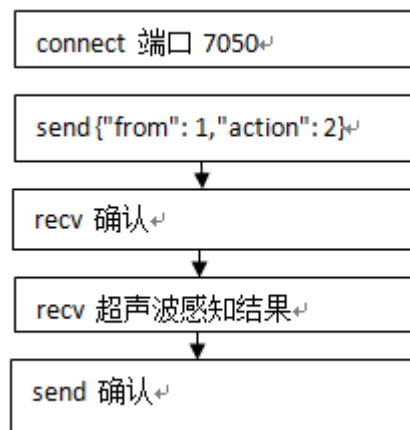
<年龄>取值: 0-100 岁

<颜值>取值:0-100,0 最低分，100 最高分

## 7. 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：



超声波感知结果格式：{"from":3, "cmd":78, "subcmd":<感知状态>, "resp":1}

<感知状态>取值：1 表示有人进入，0 表示人离开

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(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, "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
//处理超声波感知结果
```

## 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。
- 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；

完整示例如下表：

| 问题          | 答案               |
|-------------|------------------|
| <b>Name</b> | <b>content</b>   |
| 这里有洗手间吗     | 有，前面往左转          |
| 起诉需要什么流程    | 如果你带了身份证，请在这里填单子 |
| 办卡 我要办卡     | 请带上身份证到柜台填写单子    |
|             |                  |

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：

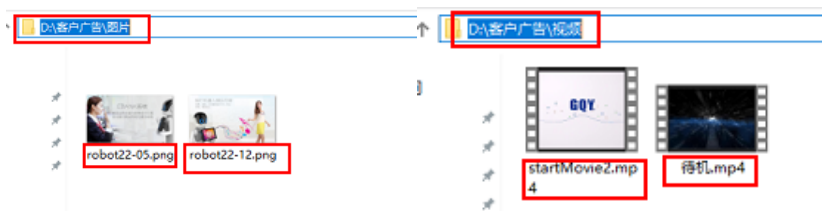


7. 重起 Surface 平板，完成客户知识库的定制

## 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
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。文件名为该 VIP 的称呼
3. 文件组织如下：

A screenshot of a Windows file explorer window. The address bar shows the path 'D:\VIP客户图片'. The window displays a list of files with columns for '名称' (Name), '修改日期' (Modified Date), '类型' (Type), and '大小' (Size).

| 名称 | 修改日期           | 类型     | 大小    |
|----|----------------|--------|-------|
| 李总 | 2017-5-5 10:22 | JPG 文件 | 17 KB |
| 王总 | 2017-5-5 10:22 | JPG 文件 | 17 KB |
| 王总 | 2017-5-5 10:22 | JPG 文件 | 17 KB |
| 周总 | 2017-5-5 10:22 | JPG 文件 | 17 KB |
| 朱总 | 2017-5-5 10:22 | JPG 文件 | 17 KB |

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 `excl` 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>