

# 上位机通讯协议

## 连接信息

- Broker: mqtt.gycis.me
- Port: 1883
- Username: smartcar
- Password: smartcar
- topic:
  - 路况信息: /smartcar/{mac}/traffic
  - 起终点信息: /smartcar/{mac}/task
  - 命令信息: /smartcar/{mac}/command
  - 位置信息: /smartcar/{mac}/position
- {mac} 说明：为树莓派 wlan0 接口 MAC 地址的后三段，如 01a2ee，获取代码如下：

```
import re
with open('/sys/class/net/wlan0/address') as f:
    mac = re.sub(r'.+((:\w\w){3})\n',r'\1',f.read()).replace(':', '')

topic = "/smartcar/{mac}/xxx".format(mac=mac)
```

## 数据格式

### 基本格式

在 MQTT 协议上传输的 payload 全部为十六进制数组，即 Python 中 bytes 类型数据，每个 topic 中传输的数据具体格式不同，见下文。

### 路况信息

路况信息的 payload 由多个连续的 record 组成，每个 record 包含3个字节，其中第0个字节表示一段路的起点编号，第1个字节表示该段路的终点编号，第2个字节表示该段路的路况。路况字节 x 与小车行驶在该路段时电机将获得的最高电压 y 的关系为  $y=12-x$  (V)，其中 x 取值范围为 0~9，0 表示完全不拥堵，9 表示最拥堵。路况信息会保证遍历且刚好遍历到地图上所有路段。

示例如下：

字节	0	1	2	3	4	5	6	7	8	.....
数据	0x01	0x02	0x03	0x01	0x03	0x00	0x05	0x07	0x08	.....
含义	(1、2 节点间拥堵为 3)			(1、3 节点间拥堵为 0)			(5、7 节点间拥堵为 8)			.....

## 起终点信息

起终点信息的 `payload` 为两个字节，第0字节表示起点编号，第1字节表示终点编号。起终点信息只会在每轮比赛开始时发送一次，有效期为当前一轮比赛。

## 命令信息

命令信息为控制本轮比赛开始和结束的命令，`payload` 为只含有一个字节的十六进制数组。该字节为 `0x00` 时表示本轮比赛开始；为 `0x01` 时表示到达终点，本轮比赛正常结束；为 `0x02` 时表示小车冲出赛道等异常情况。

当比赛开始时，上位机会发送开始命令，同时控制电机电源打开。比赛正常或异常结束时，上位机会发送结束命令，同时强制关闭电机电源。

## 位置信息

位置信息为上位机实时获取的小车位置信息，其中同时含有小车在路段上的位置和在整个地图中的绝对位置，选手程序中可选择两种信息中的任何一种进行处理。

位置信息共包含8个字节，第0字节表示小车当前所处路段的起点编号，第1字节表示终点编号，第2、3字节表示小车距离起点的距离，第4、5字节表示在地图中 X 方向坐标，第6、7字节表示地图中 Y 方向坐标，低字节在前，长度单位均为 mm。

参考处理代码如下：

```
start_num = msg.payload[0]          # 路段起点编号
end_num = msg.payload[1]            # 路段终点编号
dist = msg.payload[3] * 0xFF + msg.payload[2]  # 小车距离路段起点距离

x = msg.payload[5] * 0xFF + msg.payload[4]     # 小车 X 坐标
y = msg.payload[7] * 0xFF + msg.payload[6]     # 小车 Y 坐标
```

## 发送时间节点

以上各信息发送的时间节点如下：一轮比赛开始前，选手将车模放置在赛道起点处，打开车模上的电机开关，上位机会发送本轮比赛的起终点。比赛开始时，上位机会发送开始命令，这时比赛开始。比赛的过程中，上位机会循环发送位置信息和路况信息，位置信息会以最快的速度实时发送；路况约每5-10秒刷新一次，路况信息会在路况刷新时发送。