

MOUSIKA 小车快速 操作指南 v3.0

MOUSIKA MINI CAR Quick Guide v3.0

*Internal Only
Do not distribute*

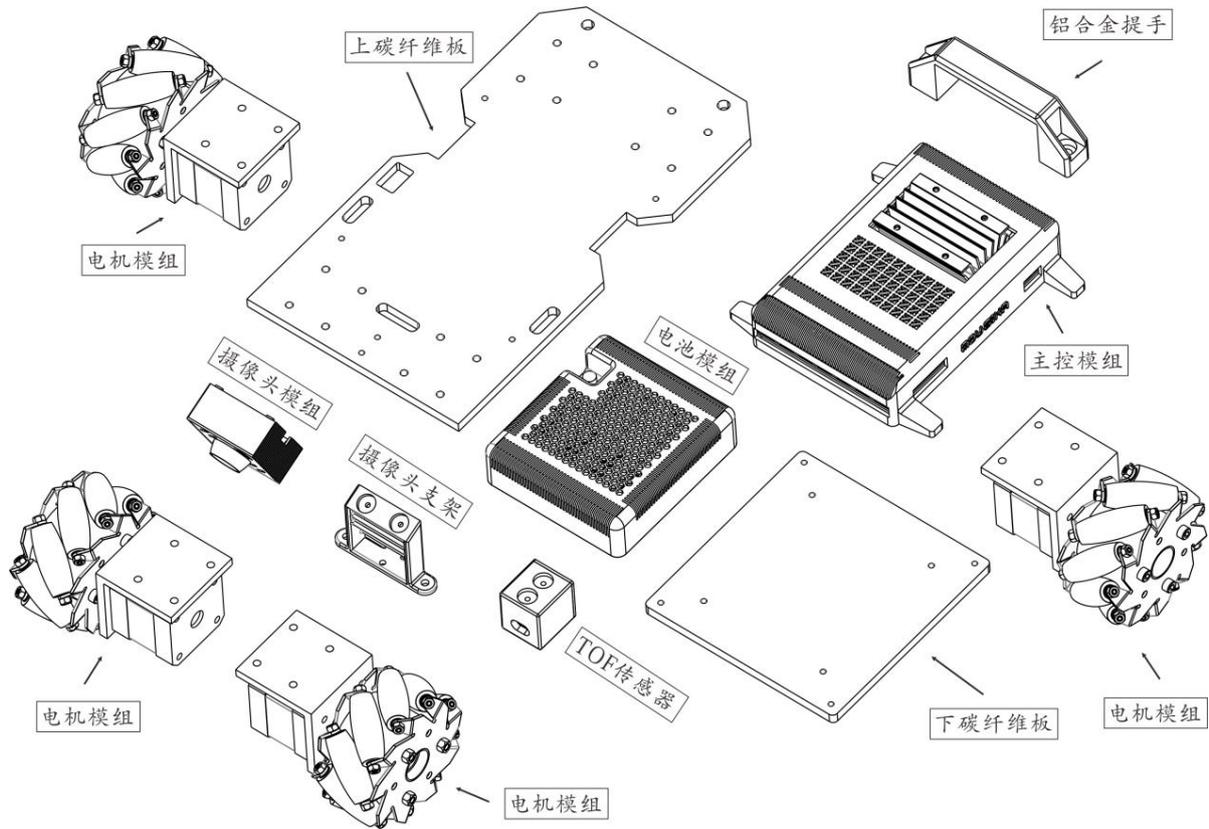
修订

<i>Date</i>	<i>Description</i>
<i>Nov 2020</i>	<i>Initial release</i>
<i>Dec 2020</i>	<i>Document upgrade</i>
<i>Mar 2021</i>	<i>Content supplement</i>

目 录

修订.....	2
1 零部件总览.....	4
2 零部件目录.....	6
2.1.电机模组.....	6
2.2.主控模组.....	6
2.3.3S 电池模组.....	7
2.4.TOF(Time of Flight) 激光传感器模组.....	7
2.5.摄像头模组.....	7
3 小车轨道搭建图示.....	8
4 MOUSIKA 小车组装教程.....	9
5 小车软件概览.....	17
6 小车开发所用的技术及框架.....	18
7 主要功能介绍.....	18
7.1.WIFI 连接.....	18
7.2.蓝牙连接.....	20
7.3.小车控制.....	23
7.3.1.手动驾驶.....	23
7.3.2.自动控制.....	25
7.4.小车初始化.....	26
7.5.小车运行状态.....	26
8 VDI 登录.....	27
9 GPU 训练.....	28
10 VGPU 训练.....	29
11 运行流程图示.....	31
12 关键模块运行错误的判定方法.....	32

1 零部件总览



Mousika Mini Car 采用可快速拆卸式模组设计，核心主控器与动力驱动部分分离，采用总线数字信号驱动动力模组，并且拥有 TOF 激光距离传感器，相机模组等扩展模块。

- 清单：

序号	名称	数量	备注
1	上碳纤维板	1	
2	下碳纤维板	1	
3	主控模组	1	
4	电机模组	4	
5	车轮	4	
6	电机安装座	4	

7	摄像头模组	1	
8	摄像头支架	1	
9	铝合金提手	1	
10	电池模组	1	
11	TOF 传感器	1	可能未配置

• 工具盒螺丝型号说明:



• 工具盒清单:

物 料	数 量
铜柱	6
磁铁	14
m2*8 平头内六角螺丝	5
m2*12 平头内六角螺丝	6
m3*7 杯头内六角螺丝	24
m4*20 杯头内六角螺丝	20
m5*15 杯头内六角螺丝	4

物 料	数 量
m3 螺母	10
m4 螺母	20
m5 螺母	4
m2 内六角扳手	1
m2.5 内六角扳手	1
m3 内六角扳手	1
m1.5 内六角扳手	1

2 零部件目录

2.1 电机模组

本机所采用的动力模组使用 42 型步进电机作为核心动力驱动来源, 电机本体具有 1.8 度细分动力角控制, 最大 1.7A 相电流, 0.55N.m 静力矩, 3.0 欧姆电阻等特性。

步进电机驱动模组使用最高 128 细分的驱动芯片 THB6128 驱动, 支持最高 3A 电流持续输出, 并且拥有驱动模组片上单片机进行数字细分, 数字电流控制等特性。使用数字总线协议进行通讯, 最高支持 256 级驱动级联。

模组使用 TYPE-C 型接口进行连接, 同时支持单线供电+总线数据传输的模式与 Mousika 核心模组进行通讯和接受控制。

步进电机动力模组接口 SPECS:

- . 步进电机动力模组 Type-C 接口 X1
- . 步进电机动力模组级联 Type-C 接口 X1
- . 传感器扩展 Type-C 接口 X1
- . 数字地址配置可拨档位设置开关 X1

2.2 主控模组

本机采用 Nvidia Jetson 系列 AI 计算芯片作为核心主控器, 全面支持 CUDA 加速边缘计算功能, 拥有多核心并发计算能力。

核心模组支持最高 4S/14.8V 动力电池供电输入能力, 设计为常规 3S/11.1V 动力电池输入, 请勿使用更高电压的动力电池作为动力来源, 以免损坏核心模组, 以及其他连接在模组上的设备。

核心模组在设计上具有 DC-DC 变压输出 5V/5A 以及 3V3/2A 的供电能力, 支持给 Jetson 计算卡最高 20W 的供电能力, 支持 6W PCIE M.2 插槽供电能力, 以及最高 60W 动力模组供电输出能力。

核心模组拥有 4 路动力模组供电 TYPE-C 型接口, 3 路摄像头 USB TYPE-C 型接口, 4 路传感器扩展 TYPE-C 型接口, 2 路 2.0 HUB USB-A 型接口, 1 路 1000M 全双工 RJ45 型以太网接口, 一路 TF 读卡器扩展接口, 一路电源输入接口。

核心模组接口 SPECS:

- . 步进电机动力模组 Type-C 接口 X4

- . 摄像头扩展 USB Type-C 接口 X3
- . 传感器扩展 Type-C 接口 X4
- . USB2.0 HUB 扩展 Type-A 接口 X2
- . 1000M Base RJ45 以太网接口 接口 X1
- . TF Card Reader 存储卡扩展接口 X1
- . Jetson Debug USB Host Type-C 接口 X1
- . Jetson Debug UART Type-C 接口 X1

2.3 3S 电池模组

本机采用 3S 动力电池模组作为动力来源，使用 18650 电芯来保证电池组的安全性能以及放电性能，推荐放电电流最大为 12V 5A，推荐充电电流最大为 1A，模组拥有输出开关的功能以及 3S 平衡冲接口，便于电池的平衡充电以及高级维护功能。

电池模组接口 SPECS:

- . XT60 供电输出接口 X1
- . 3S XH2.54 4P 平衡冲接口 X1
- . 电池模组输出开关 X1

2.4 TOF (Time of Flight) 激光传感器模组

本机采用高精度 TOF 激光飞行时间距离传感器来进行前方障碍物距离的辅助判定，最大测距距离为 2M，工作光强范围为室内光强，请在室外开启本传感器以免对主控运动控制进行错误判断。

TOF 激光传感器接口 SPECS:

- . 摄像头扩展 USB Type-C 接口 X1

2.5 摄像头模组

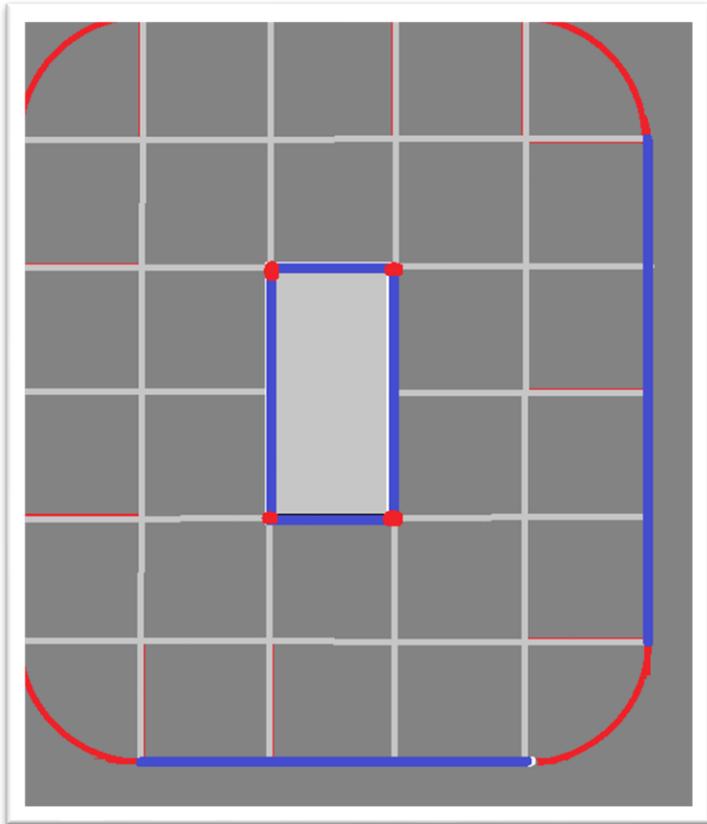
本机采用 120 度广角 FOV 相机镜头，USB2.0 协议，最高支持 1920*1080 分辨率，30FPS 采样率，可看清前方较为详细的视觉图像，25 度倾角对地面，适合 AI 实际模型训练中会遇到的路况。

摄像头模组接口 SPECS:

- . 摄像头扩展 USB Type-C 接口 X1

3 小车轨道搭建图示

搭建示意图：



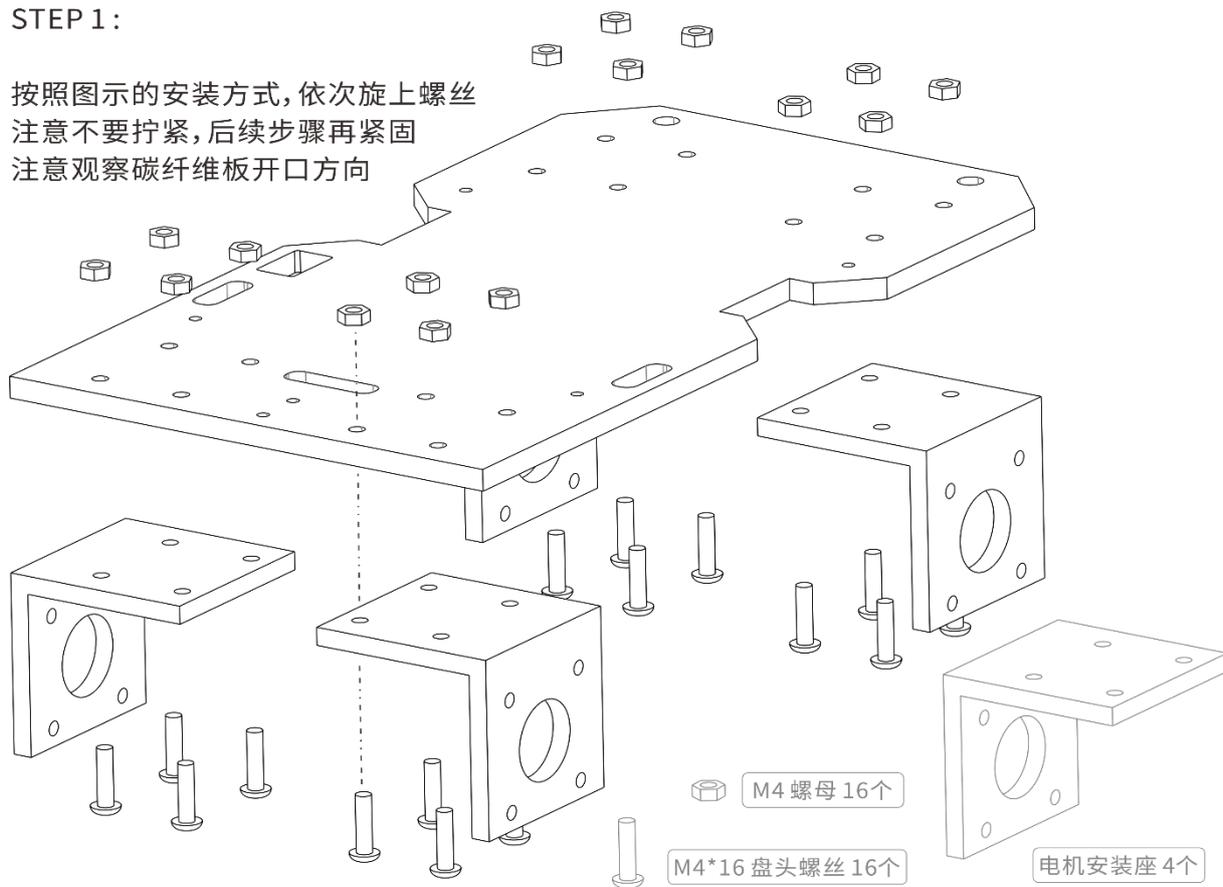
搭建完成图示：



4 MOUSIKA 小车组装教程

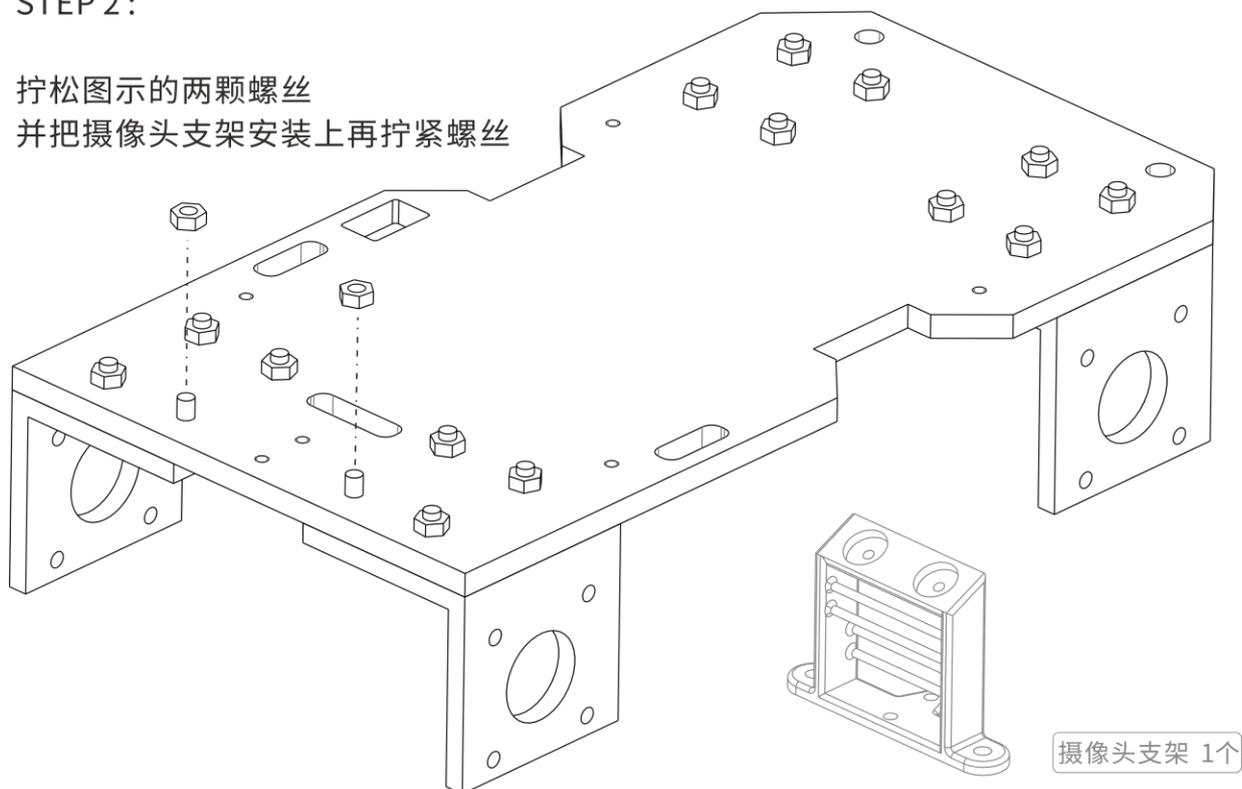
STEP 1:

按照图示的安装方式,依次旋上螺丝
注意不要拧紧,后续步骤再紧固
注意观察碳纤维板开口方向

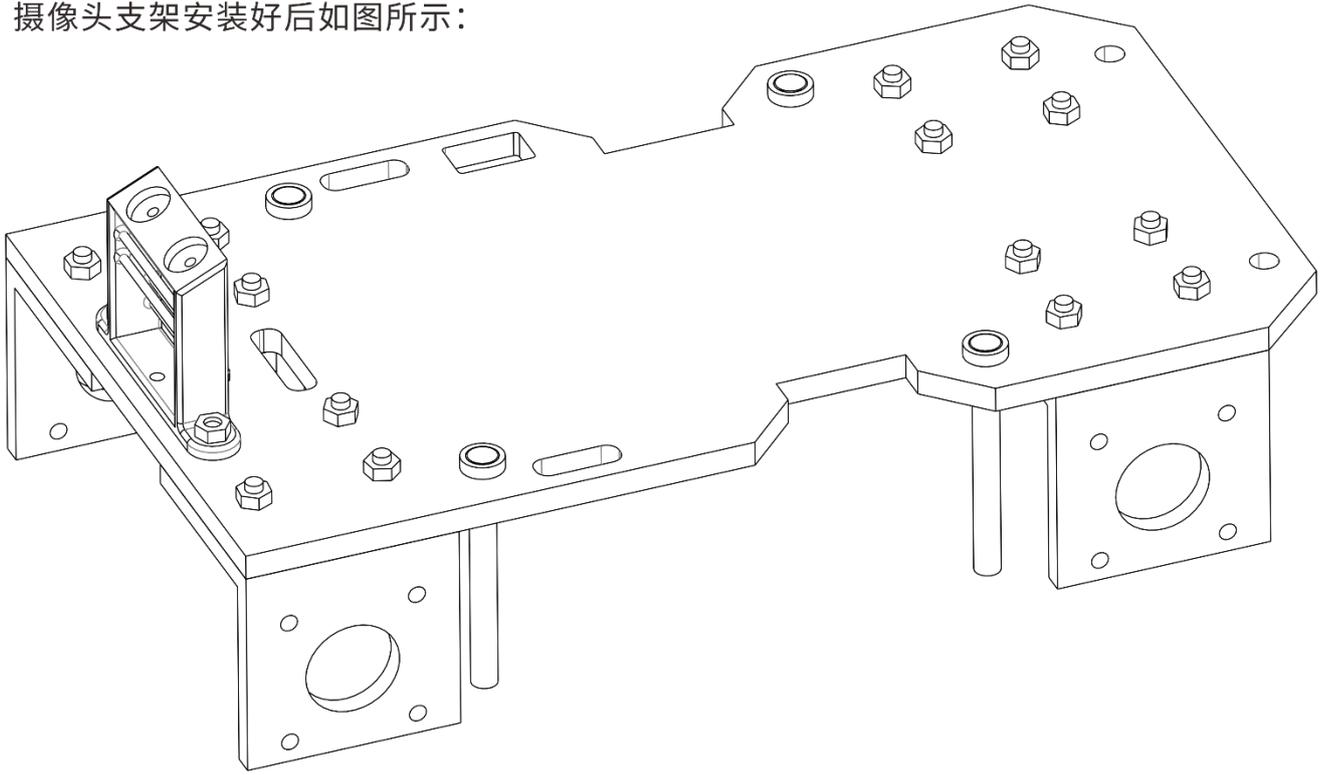


STEP 2:

拧松图示的两颗螺丝
并把摄像头支架安装上再拧紧螺丝

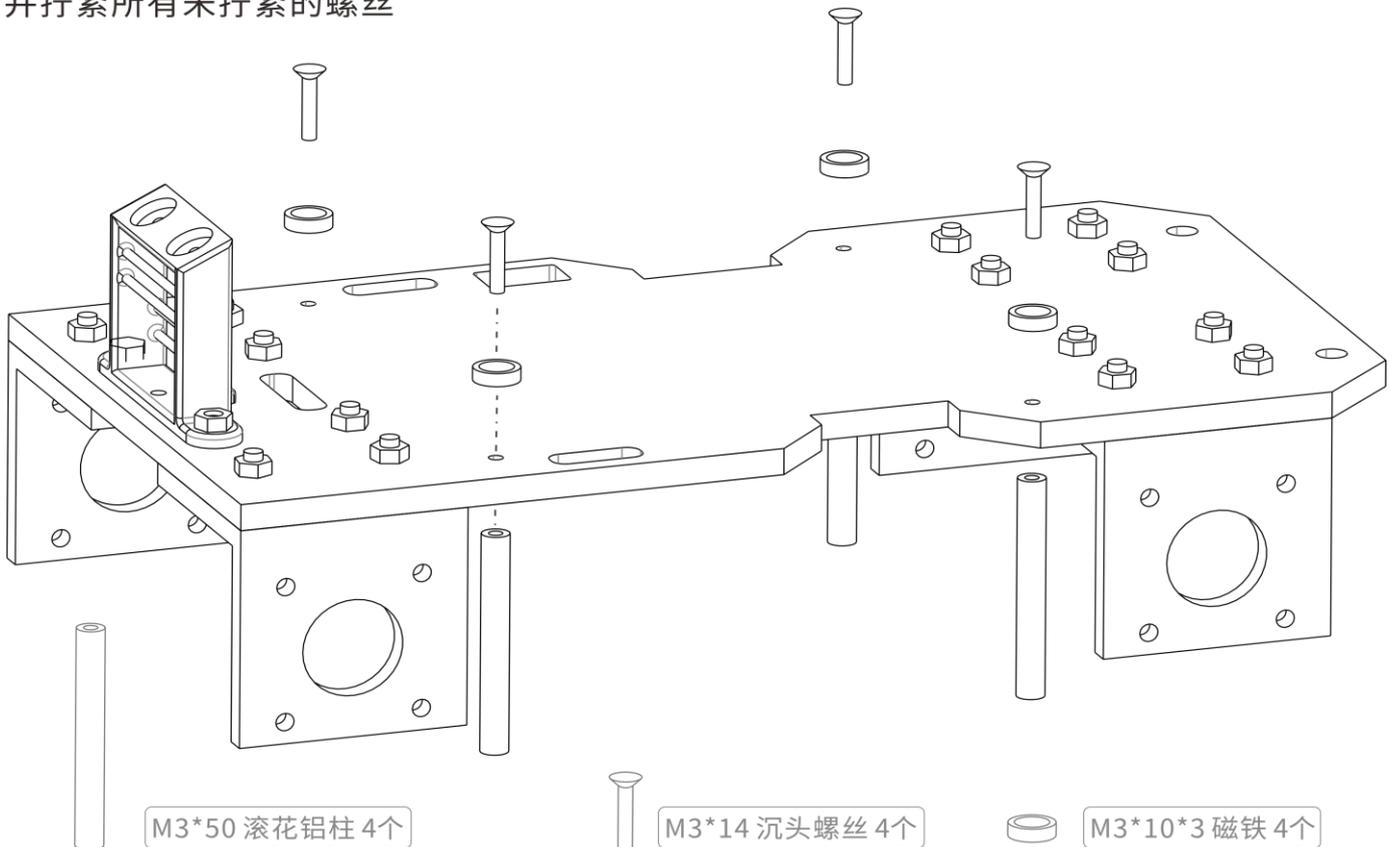


摄像头支架安装好后如图所示：

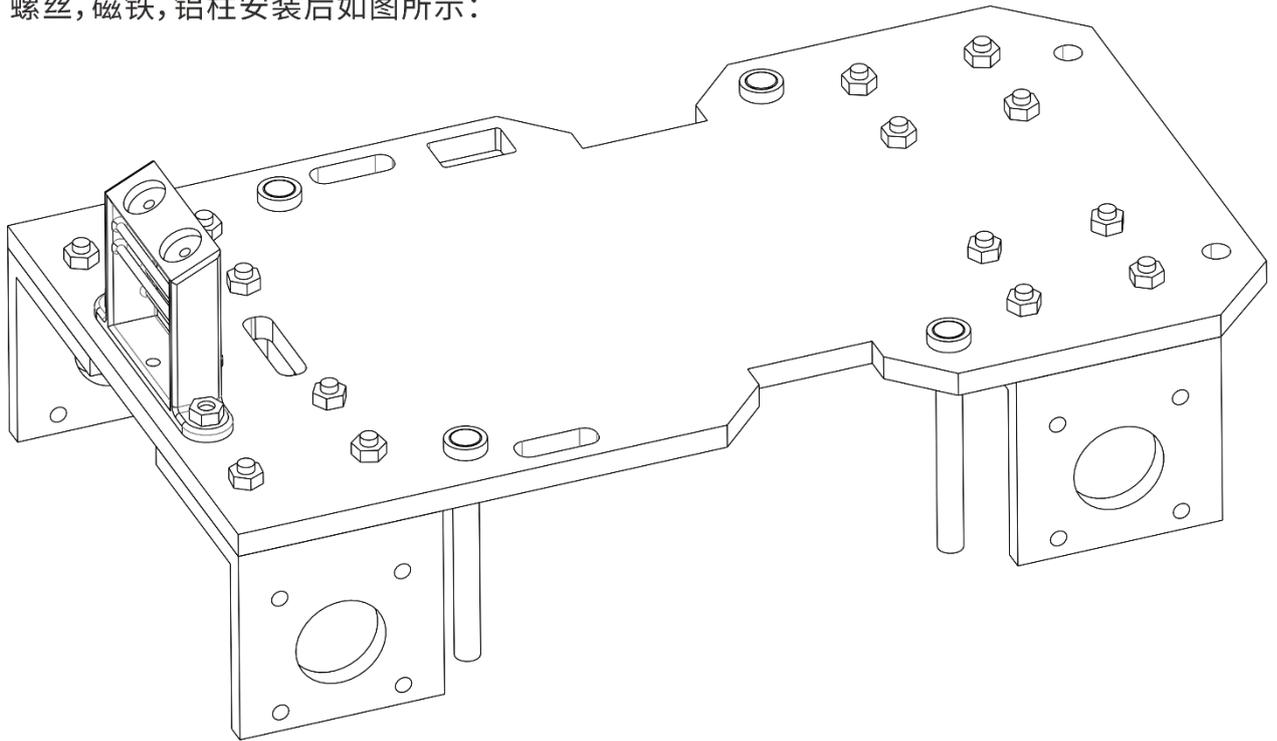


STEP 3:

按照图示的安装方式
依次安装螺丝, 磁铁, 铝柱
并拧紧所有未拧紧的螺丝

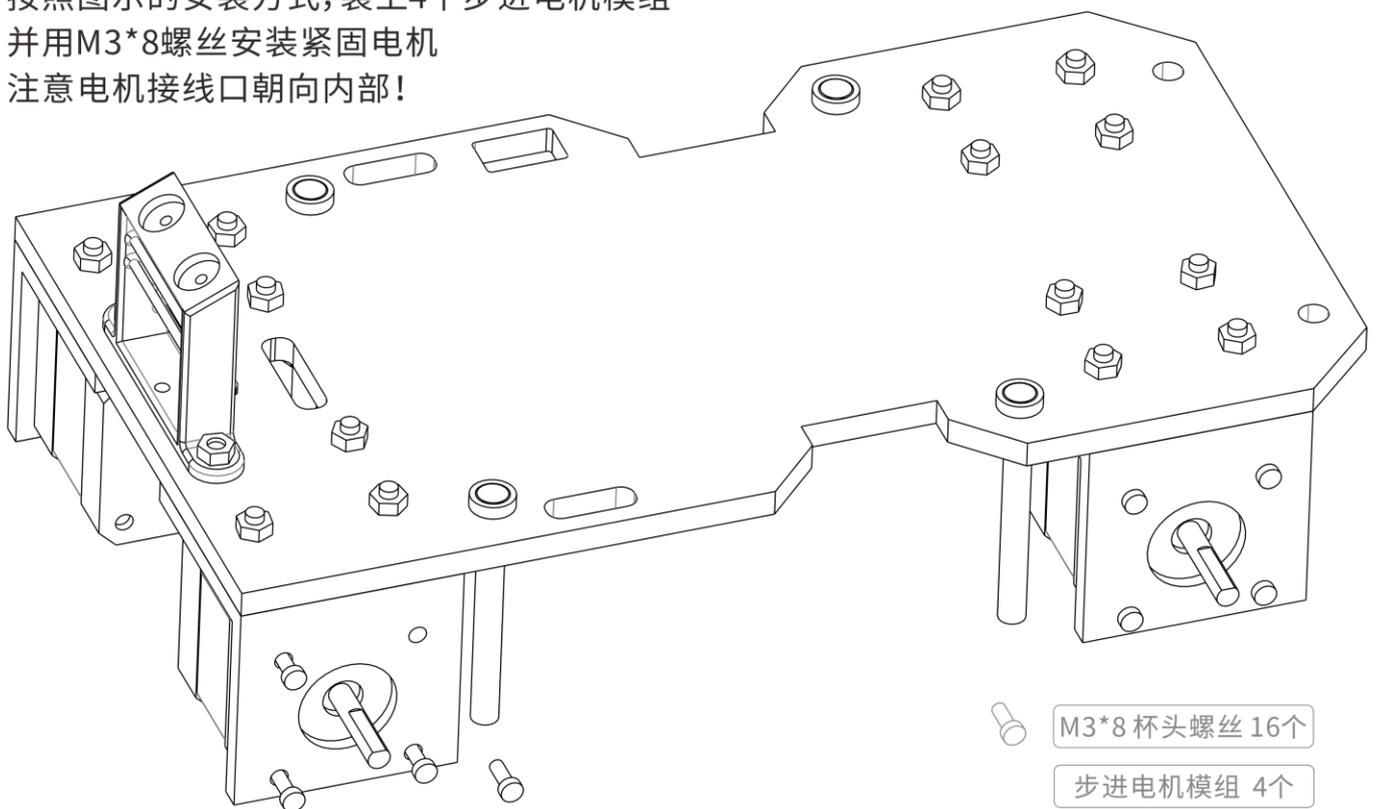


螺丝, 磁铁, 铝柱安装后如图所示:



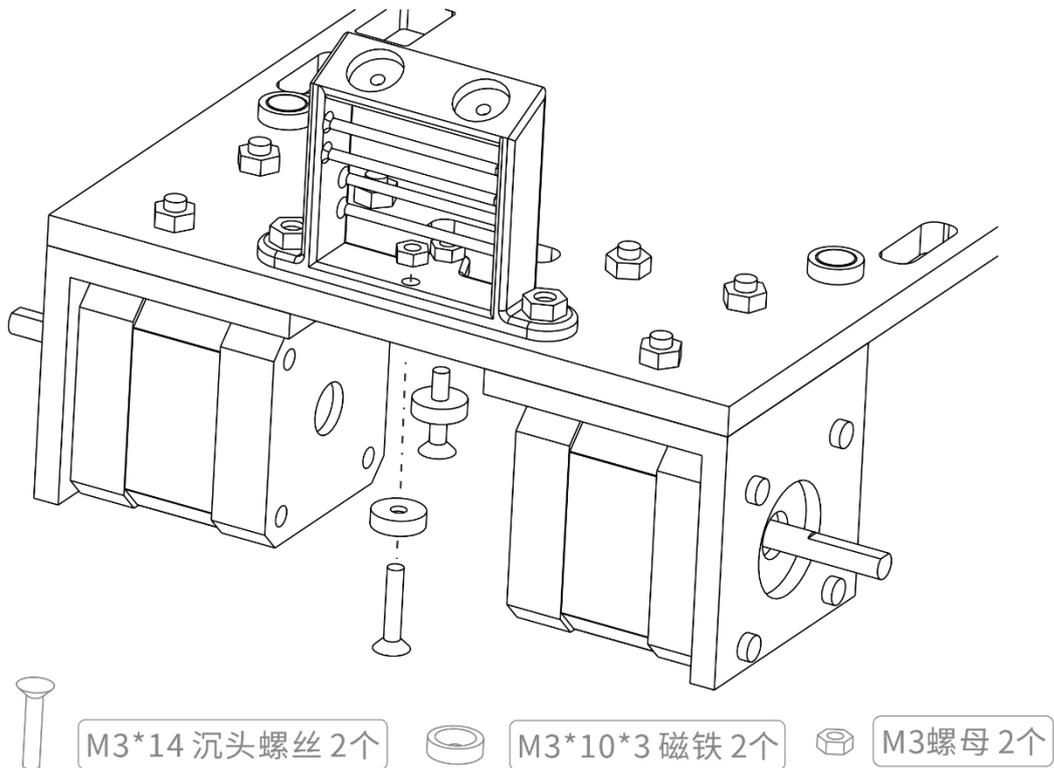
STEP 4:

按照图示的安装方式, 装上4个步进电机模组
并用M3*8螺丝安装紧固电机
注意电机接线口朝向内部!



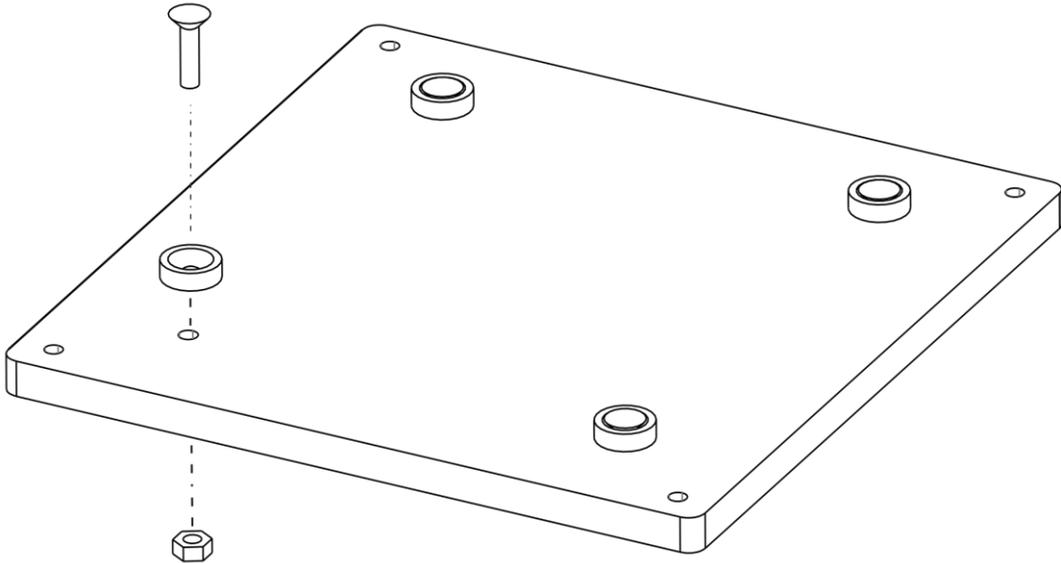


STEP 5:
按照图示的安装方式
依次安装螺丝, 磁铁, 螺母



STEP 6:

按照图示的安装方式
依次安装螺丝, 磁铁, 螺母



M3*10 沉头螺丝 4个

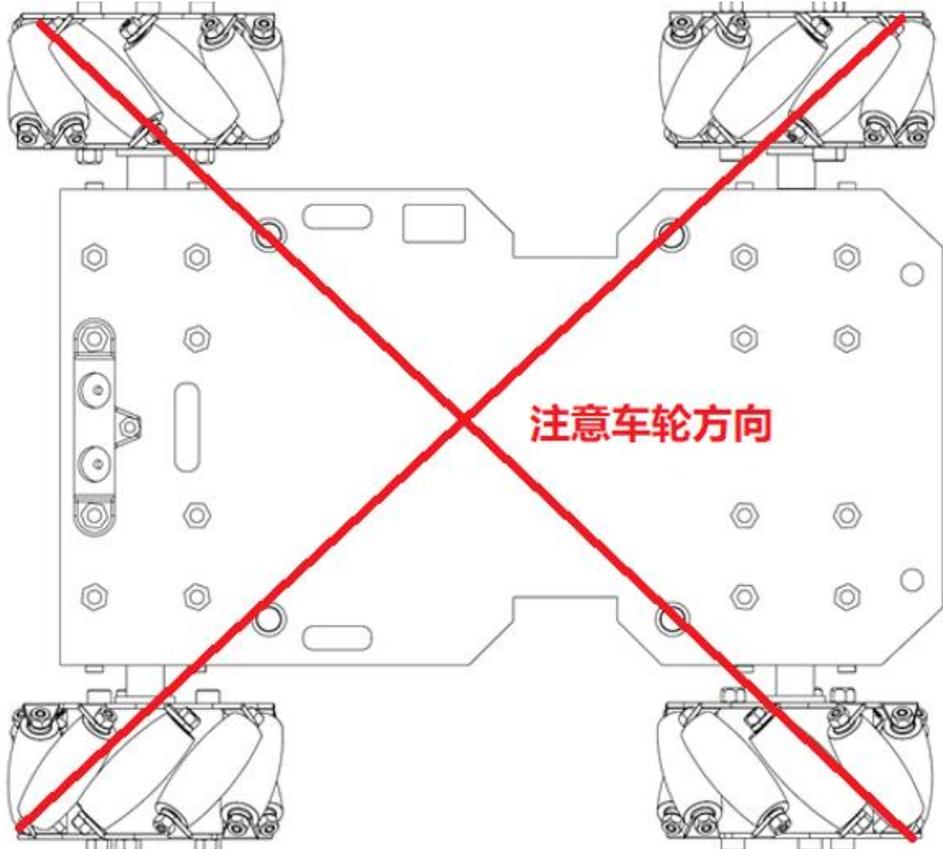


M3*10*3 磁铁 4个



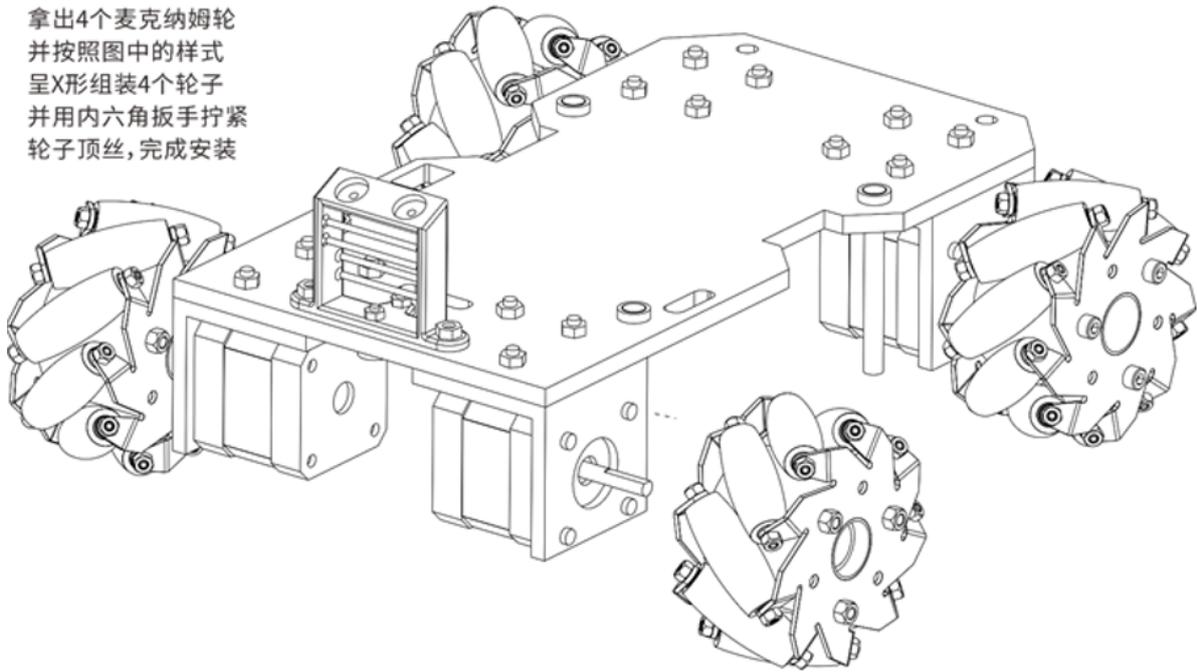
M3螺母 4个

麦克纳姆轮 4个
按照图示螺旋
方向安装



STEP 7:

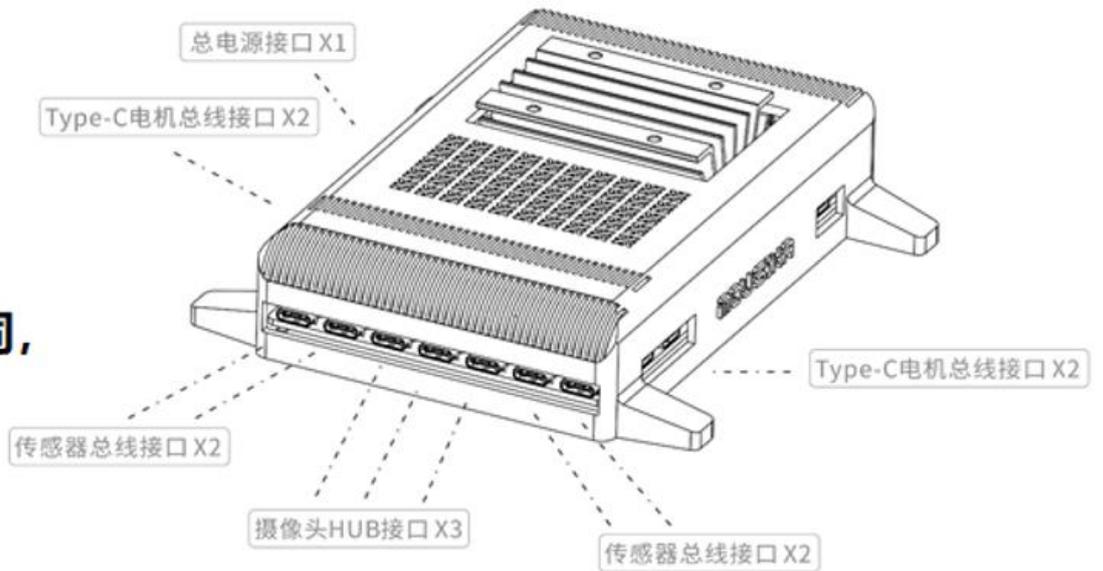
拿出4个麦克纳姆轮
并按照图中的样式
呈X形组装4个轮子
并用内六角扳手拧紧
轮子顶丝,完成安装



TOF传感器模组
用磁吸连接车体

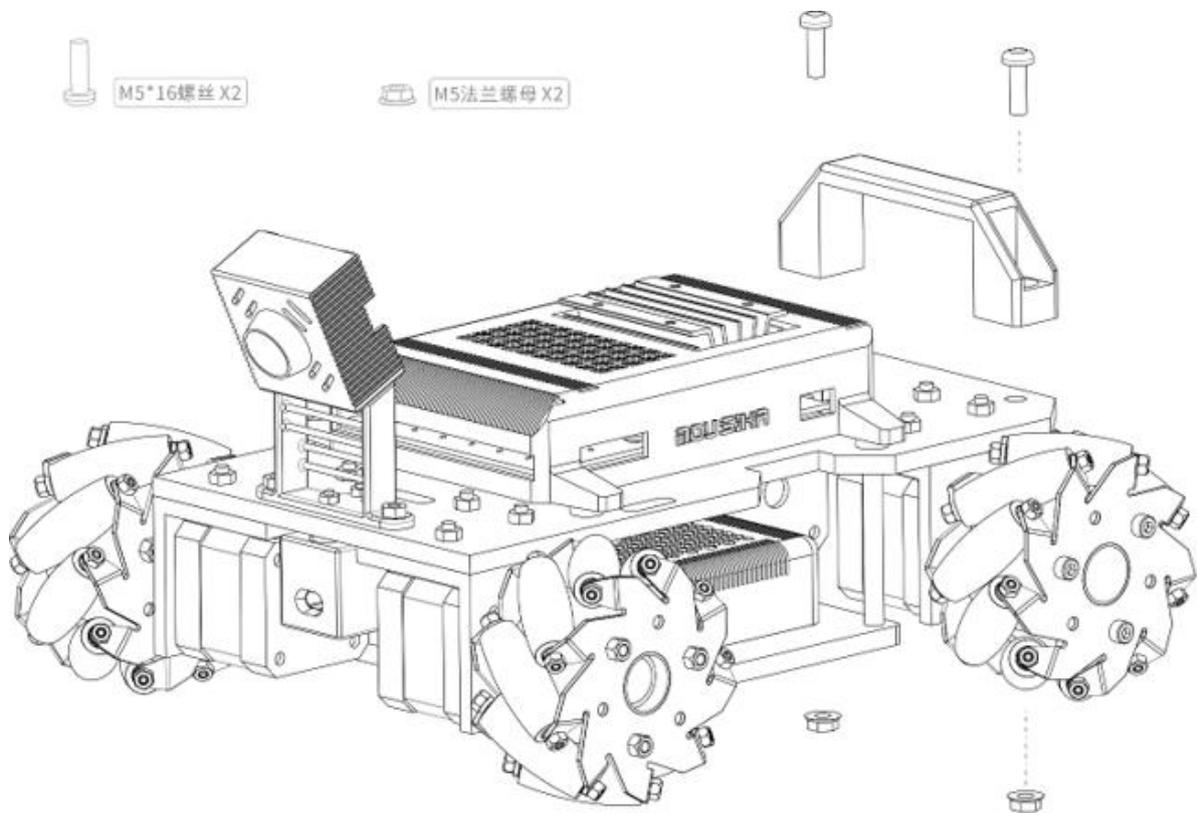
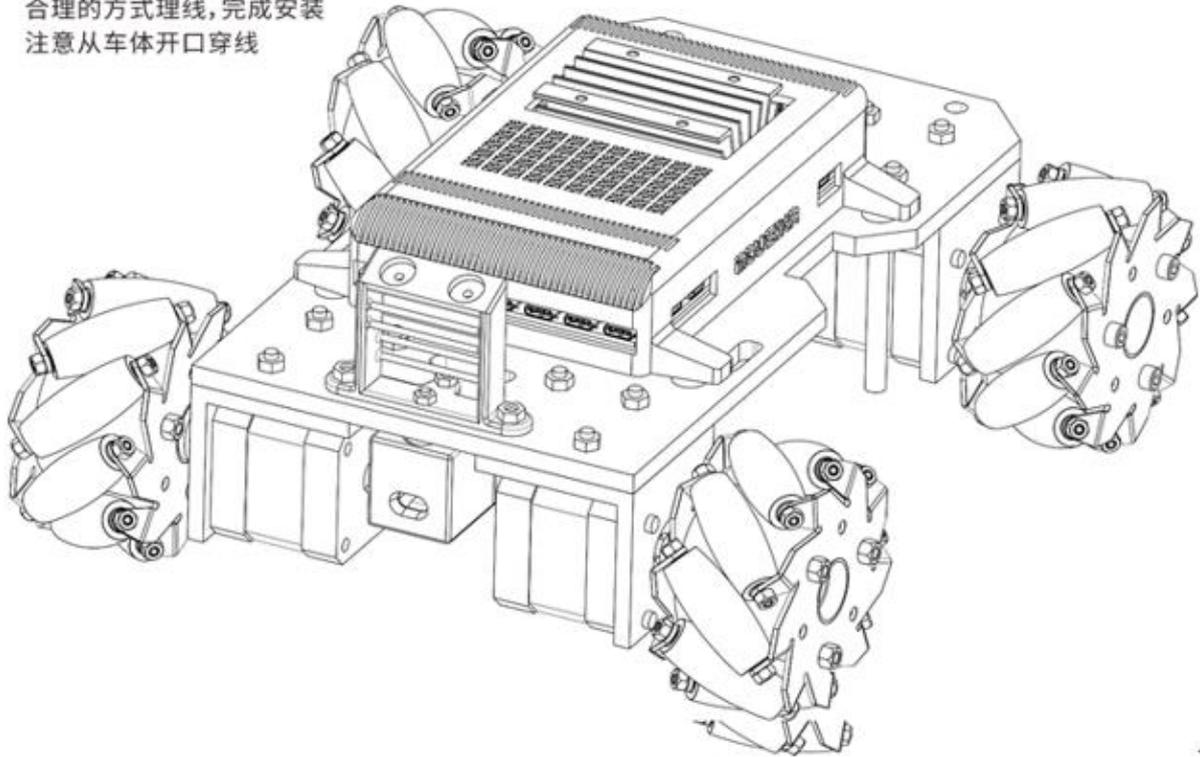


**根据型号不同,
可能不配备**



STEP 8:

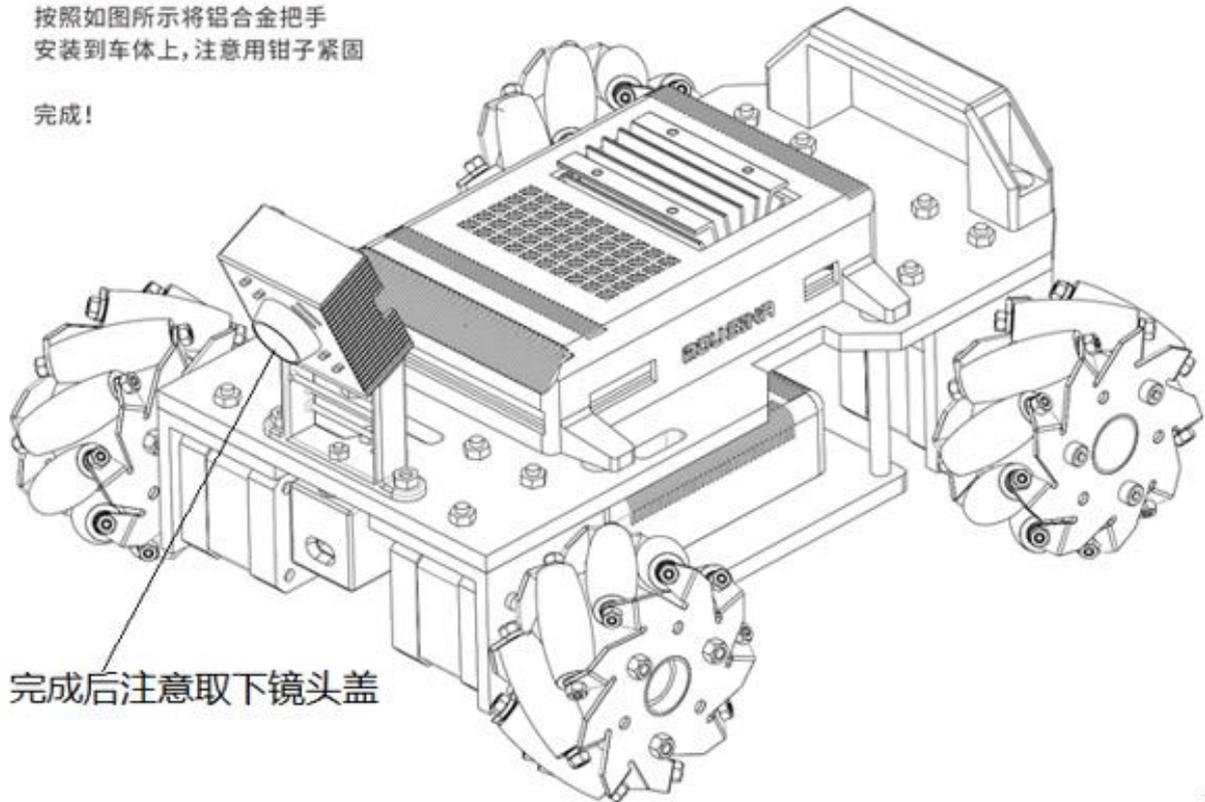
将TOF传感器模组和主控模组
按照图里的方式和车体磁吸
拿出5根Type-C连接线
合理的方式理线,完成安装
注意从车体开口穿线



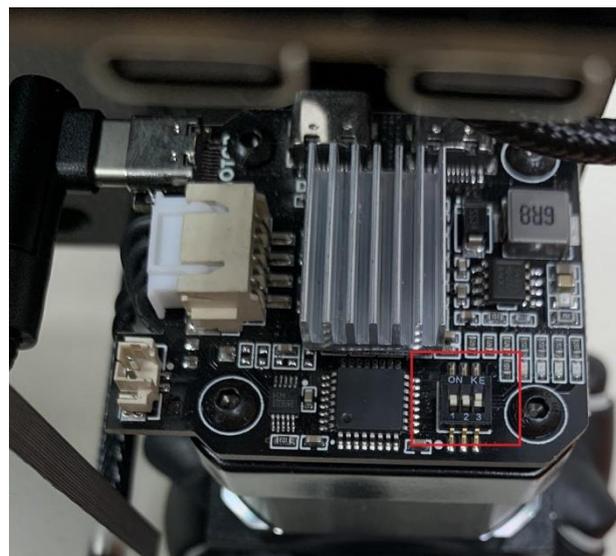
STEP 9:

将摄像头模组磁吸到支架上
按照如图所示将铝合金把手
安装到车体上,注意用钳子紧固

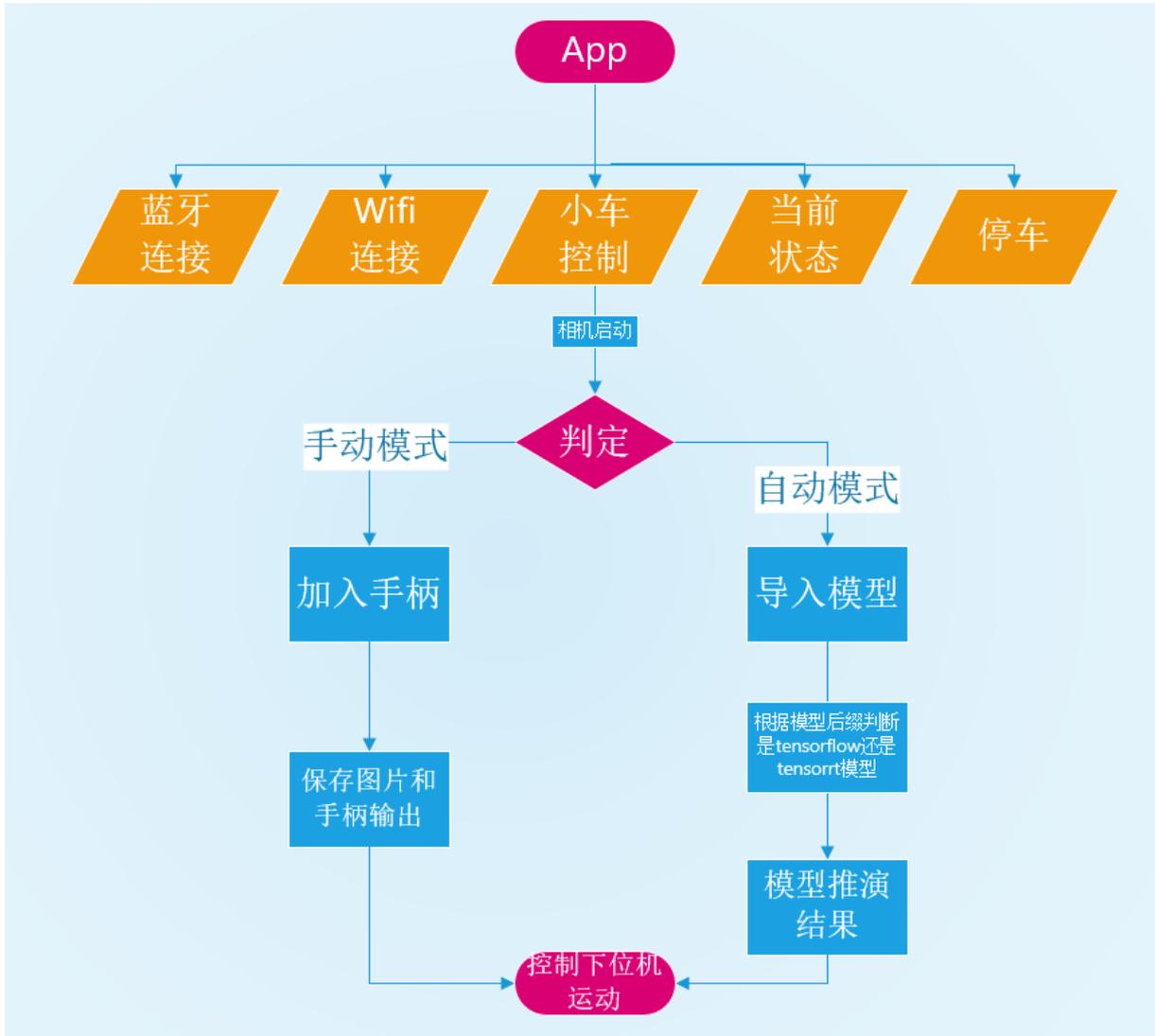
完成!



安装完成后需要定义电机编号, 编号标识如图, 0为无需跳线。



5 小车软件概览



6 小车开发使用的技术及框架

小车使用 Flask 后端框架，部署使用 gunicorn 启动 Flask 服务，入口函数在 `/home/mousika/mycar/app.py`

自动驾驶部分使用 tensorflow 和 tensorRT 框架实现

通信用到了蓝牙和串口(serial)协议

摄像头的调用使用 opencv 实现

注：小车的出厂用户名：mousika

小车的出厂密码：mousika

7 主要功能介绍

小车的主要功能包括：WIFI 连接、蓝牙连接、手动驾驶、自动驾驶、小车初始化和小车运行状态。

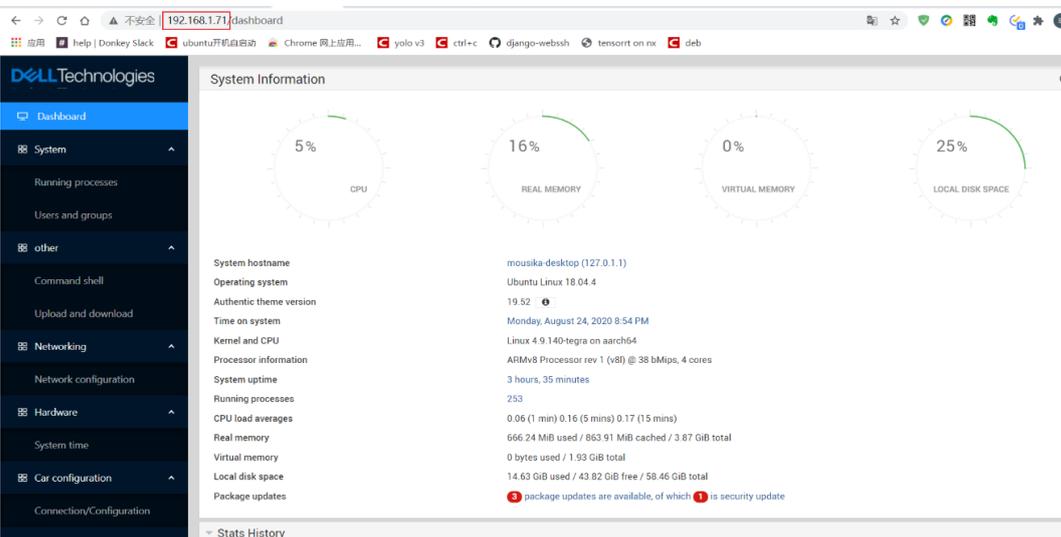
7.1 WIFI 连接

第一次配置一辆新的小车需要先通过网线配置小车的 WIFI。

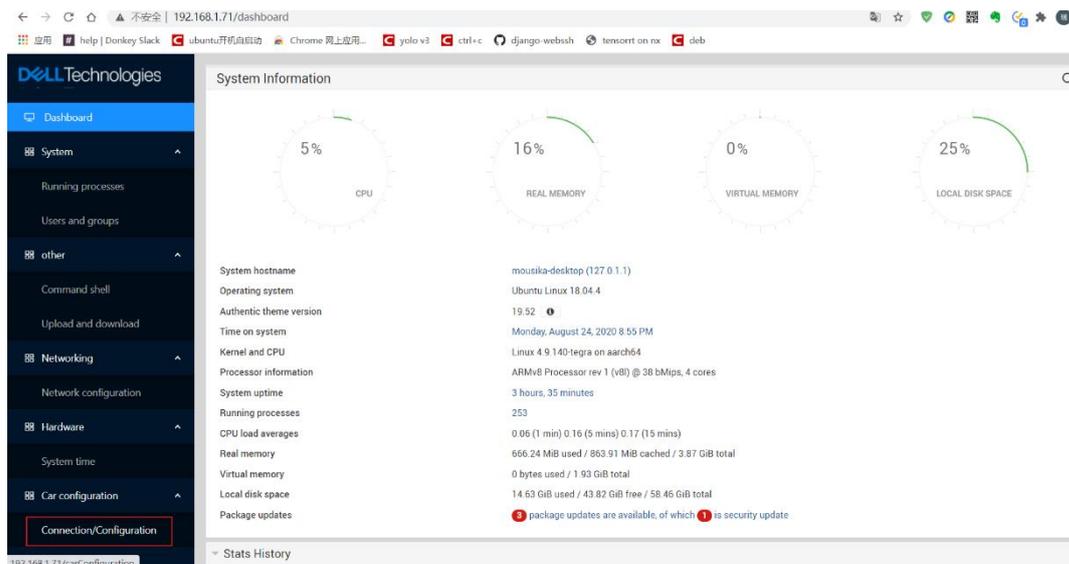
- 1) 小车通过网线连接到路由器。
- 2) 登录到路由器查看小车的 IP。

所有名单				
互联网	图标	客户端名称	客户端 IP 地址	客户端 MAC 地址
		mousika-desktop	192.168.1.71	00:28:F8:C2:ED:3B

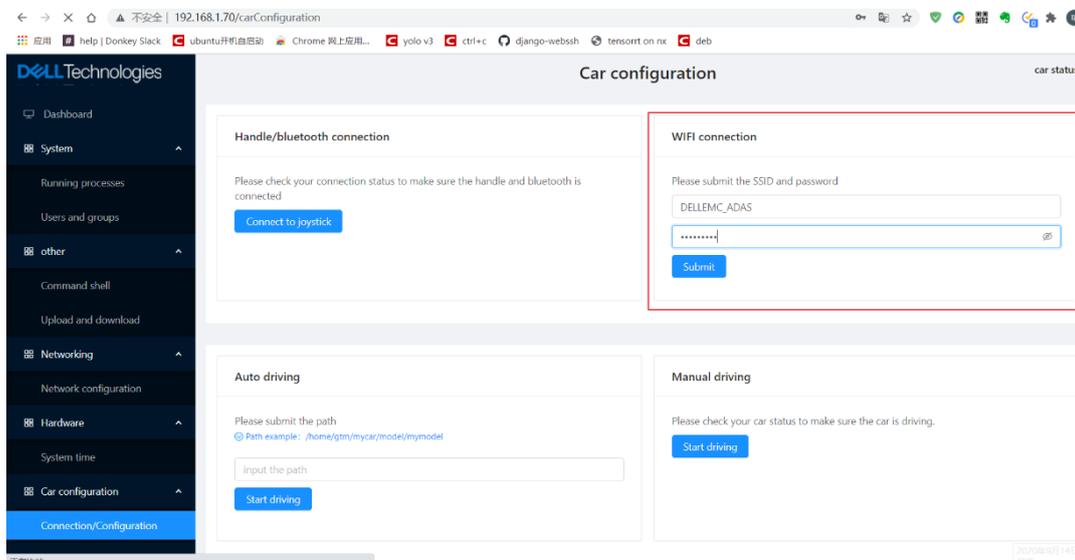
- 3) 用一台同一网段的电脑，在浏览器输入小车的 IP。



4) 点击左下角的 Connection/Configuration.



5) 在右上角的输入框内输入 SSID 和 Passwd.

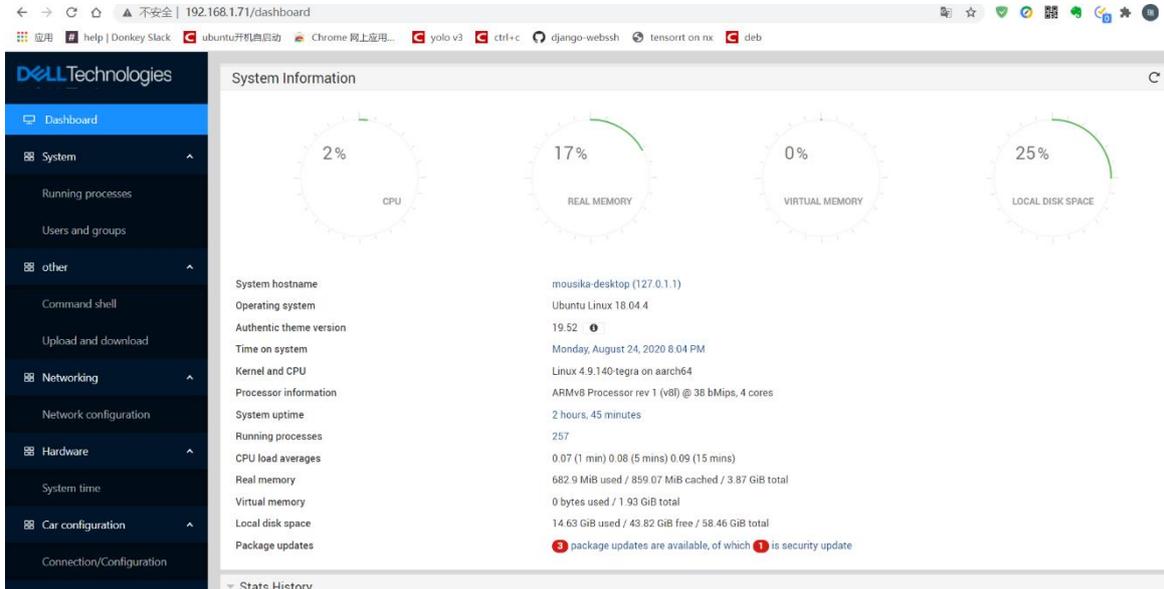


6) 根据网页上返回的 IP 地址重新输入 IP 地址, 此时可以拔掉网线, 之后每次系统启动, 小车会自动连接到这个 SSID.

7.2 蓝牙连接

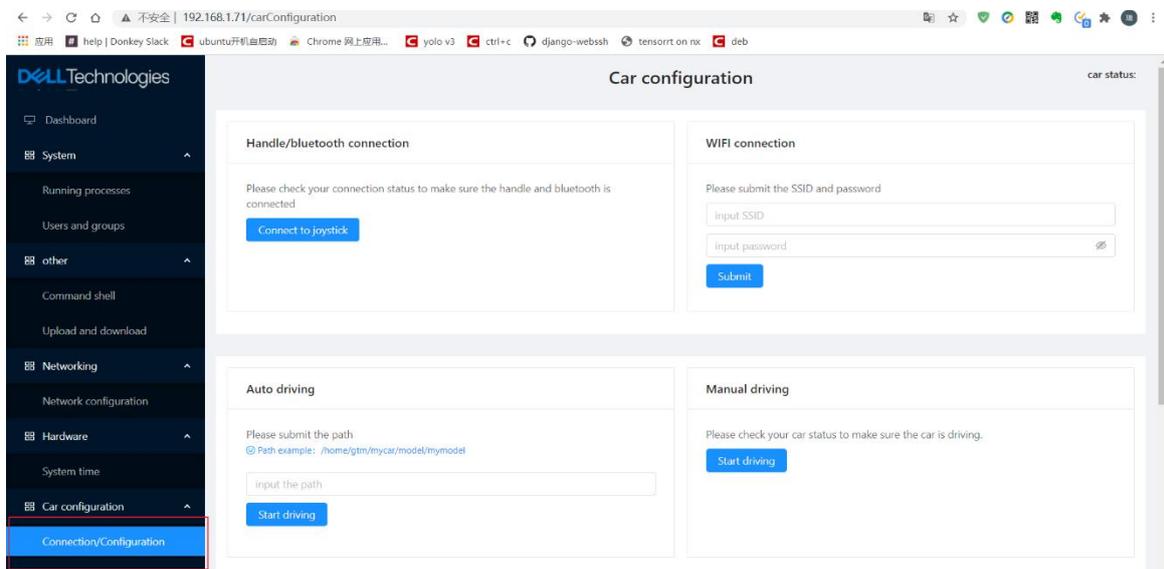
当需要手动操控小车或者收集训练数据时，先连接蓝牙。

1) 打开浏览器，输入小车 IP。



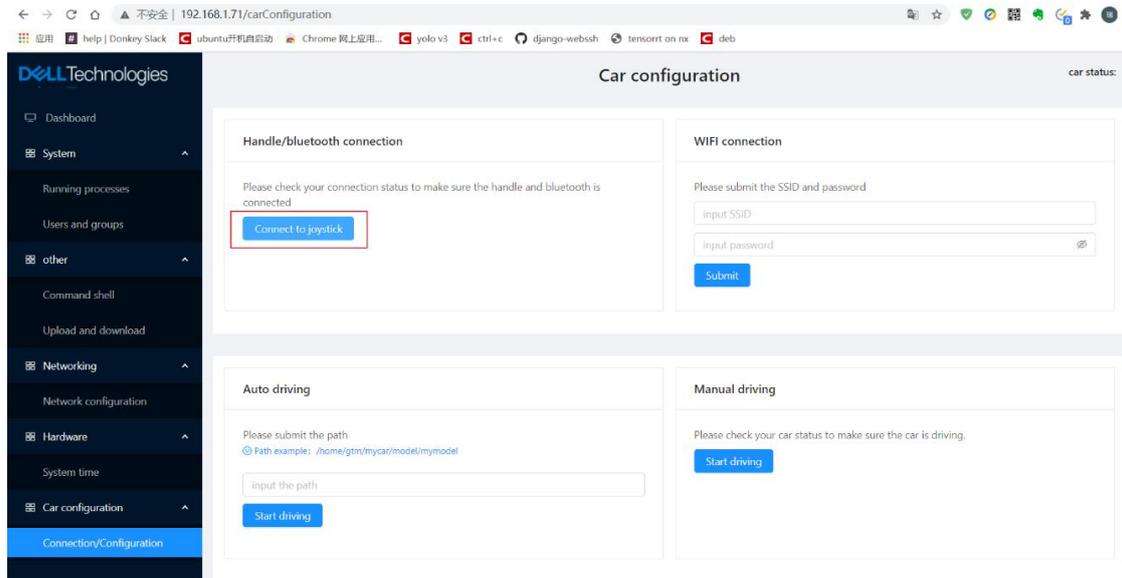
The screenshot shows a web browser at the address 192.168.1.71/dashboard. The page displays system information for a device named 'mousika-desktop' running Ubuntu Linux 18.04.4. Key metrics are shown in circular gauges: CPU at 2%, REAL MEMORY at 17%, VIRTUAL MEMORY at 0%, and LOCAL DISK SPACE at 25%. Below these, system details include hostname, operating system, authentic theme version (19.52), time on system (Monday, August 24, 2020 8:04 PM), kernel and CPU (Linux 4.9.140-tegra on aarch64), processor information (ARMv8 Processor rev 1 (v8l) @ 38 bMips, 4 cores), system uptime (2 hours, 45 minutes), running processes (257), CPU load averages (0.07 (1 min) 0.08 (5 mins) 0.09 (15 mins)), real memory (682.9 MiB used / 859.07 MiB cached / 3.87 GiB total), virtual memory (0 bytes used / 1.93 GiB total), local disk space (14.63 GiB used / 43.82 GiB free / 58.46 GiB total), and package updates (3 available, 1 security update).

2) 选择左下角的 Connection Configuration.



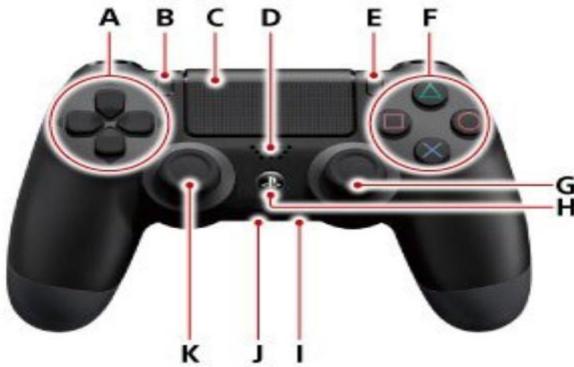
The screenshot shows the 'Car configuration' page at 192.168.1.71/carConfiguration. The page is divided into four sections: 'Handle/bluetooth connection' with a 'Connect to joystick' button; 'WIFI connection' with input fields for SSID and password and a 'Submit' button; 'Auto driving' with a path input field and a 'Start driving' button; and 'Manual driving' with a 'Start driving' button. The 'Connection/Configuration' menu item in the left sidebar is highlighted with a red box.

3) 点击 Connect to Joystick。

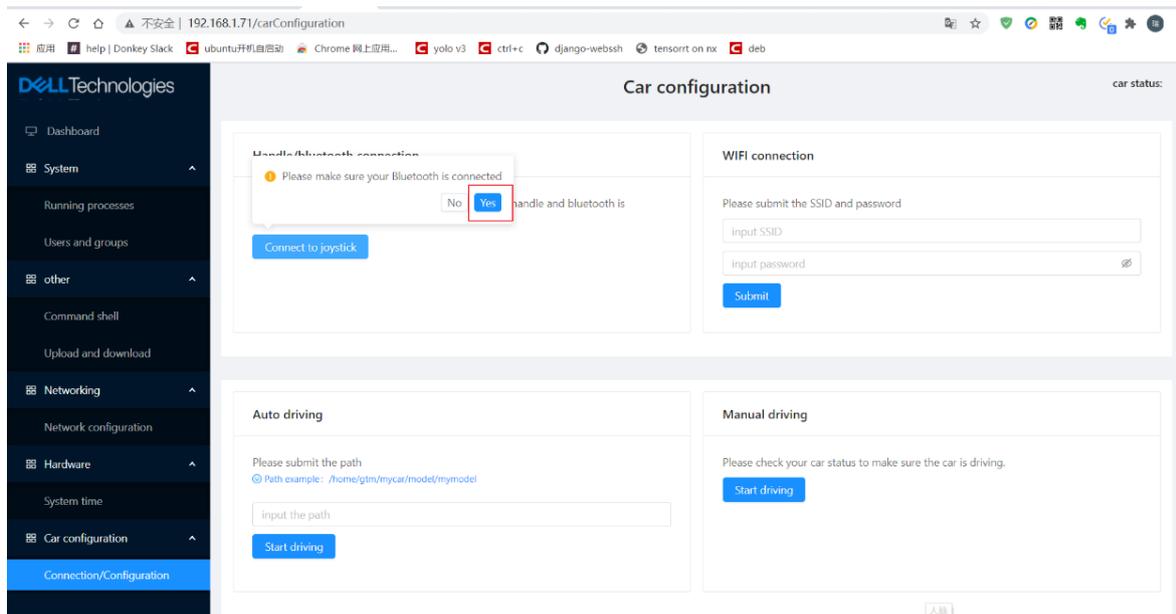


4) 同时按住手柄的 Share 键 (B) 和图标键 (H)，当手柄的指示灯快速闪烁时松开 Share 键和图标键。

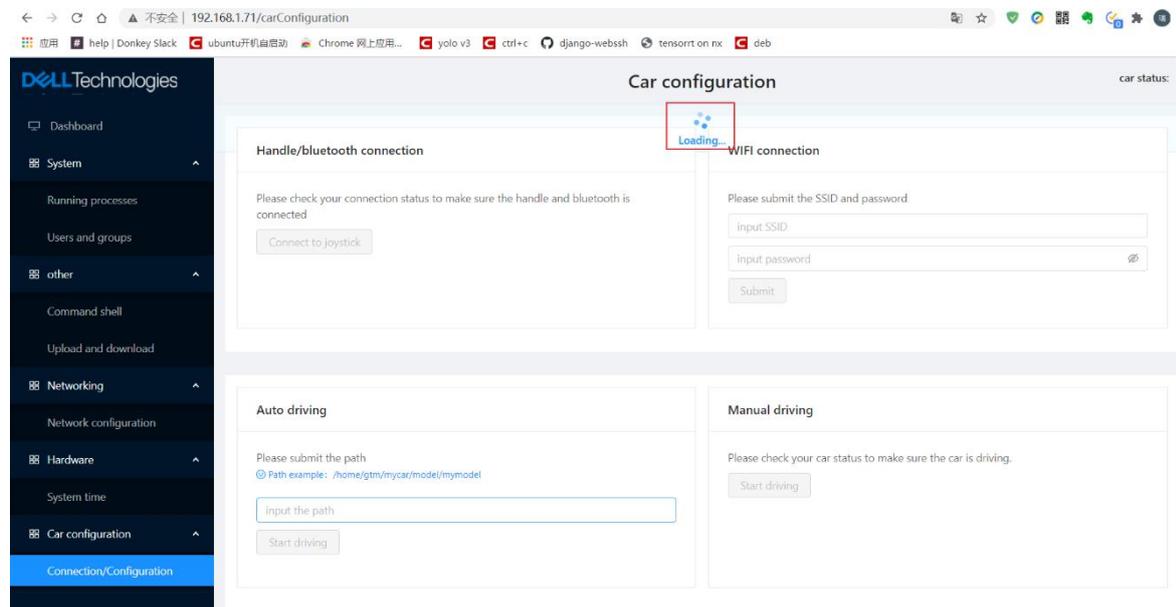
正面



5) 点击 Yes 按钮。



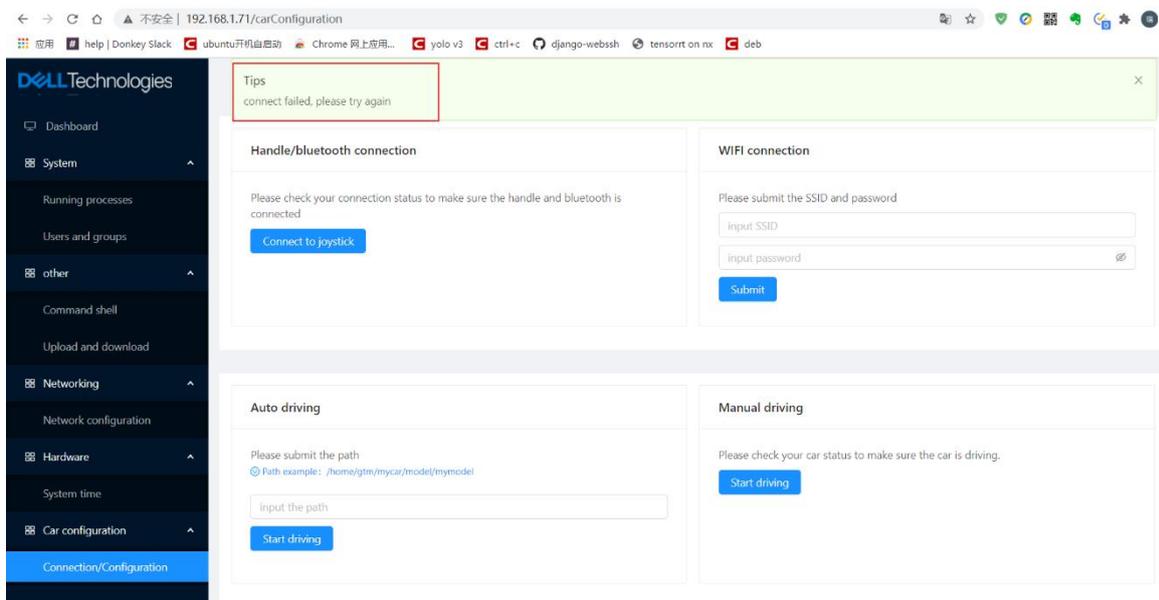
6) 等待手柄连接。



- 7) 当 Web 页面返回 “Connect failed, please try again” 时，重复以上步骤。（注意手柄指示灯，如果不亮了需要立即重新按住 share 键和图标键，直至指示灯闪烁）。
- 8) 当手柄指示灯常亮时代表连接成功。
- 9) 下一次再用同一辆小车连接同一个手柄时，只需在小车开机 30 秒后按下图标键就会自动建立连接。

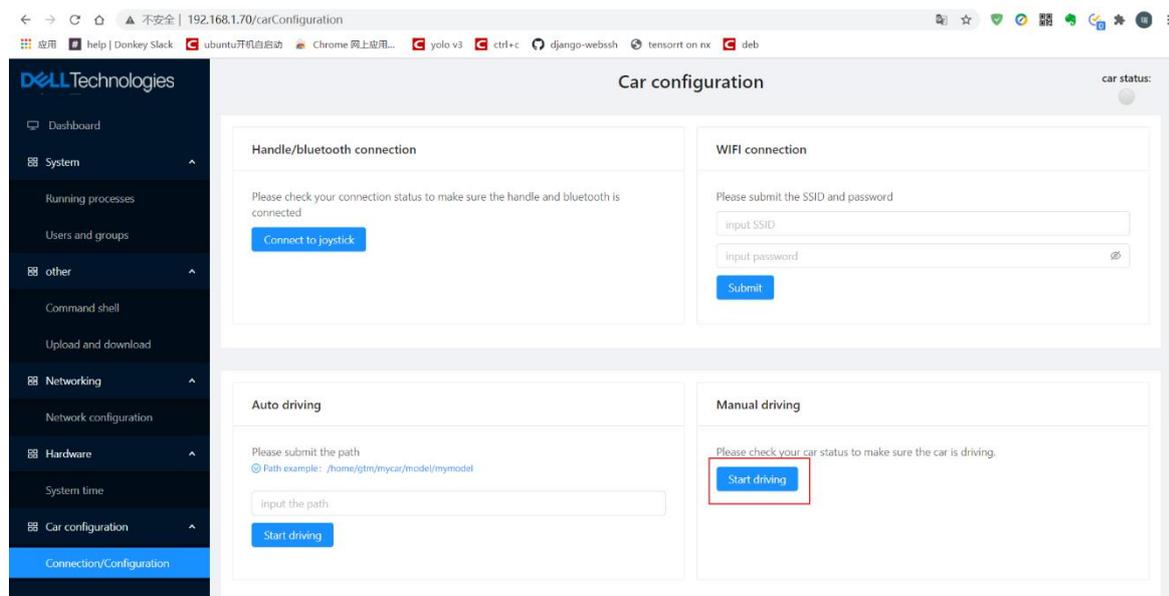
7.3 小车控制

小车控制分为手动驾驶和自动驾驶两个模式。手动驾驶时需要连接手柄，且当速度不为 0 时会以每秒 20 帧的速度记录摄像头数据和手柄的 records 数据。如启用自动驾驶模式则无需连接手柄，也不会记录数据。

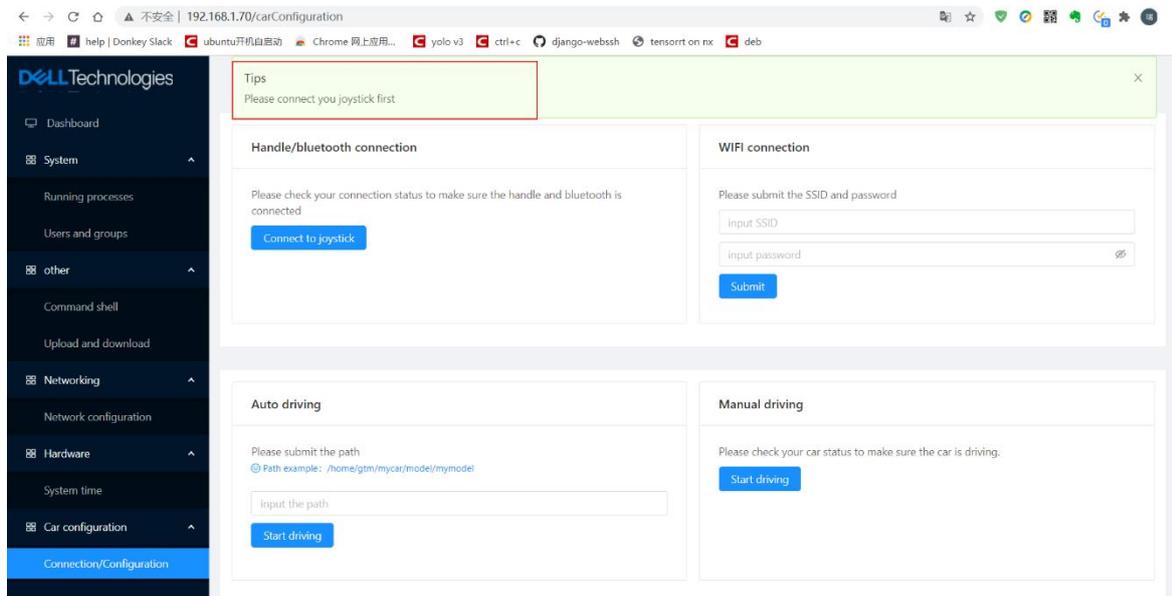


7.3.1 手动驾驶

1) 在 Web 页面的 Manual Driving 模块点击 Start Driving 按键。



2) 如果没有连接手柄就点击 Start Driving 按键会有提示，且不会成功启动。

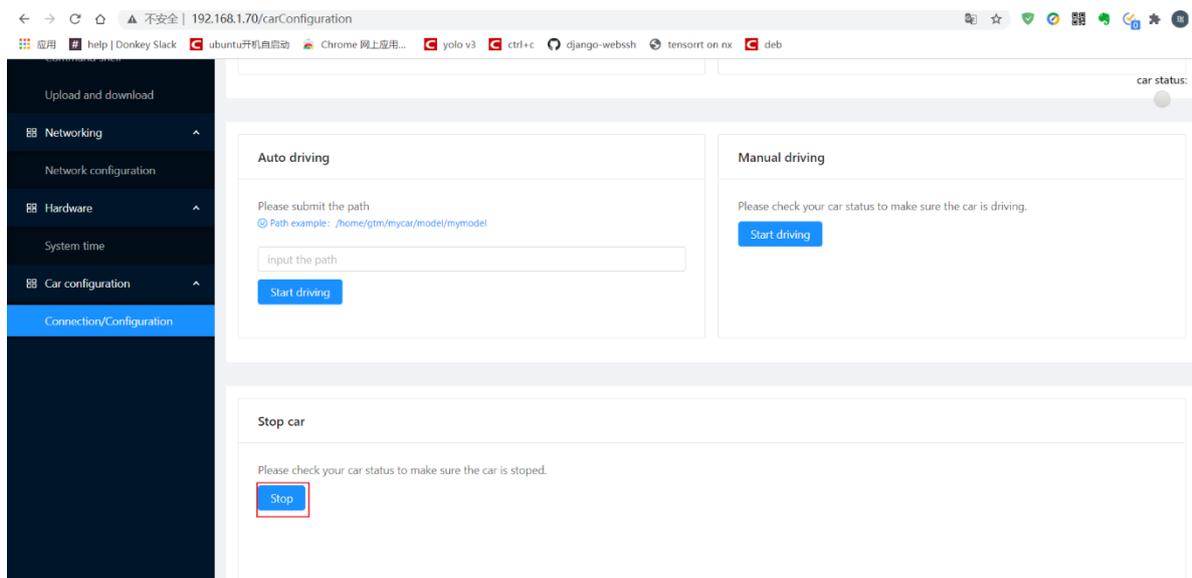


3) 直至 Tips 出现，启动后可以正常遥控小车收集数据。

左边的摇杆左右方向摇动表示转向，右边的摇杆上下方向摇动表示前进后退（只有在前进或者后退的状态小车才可以转向）



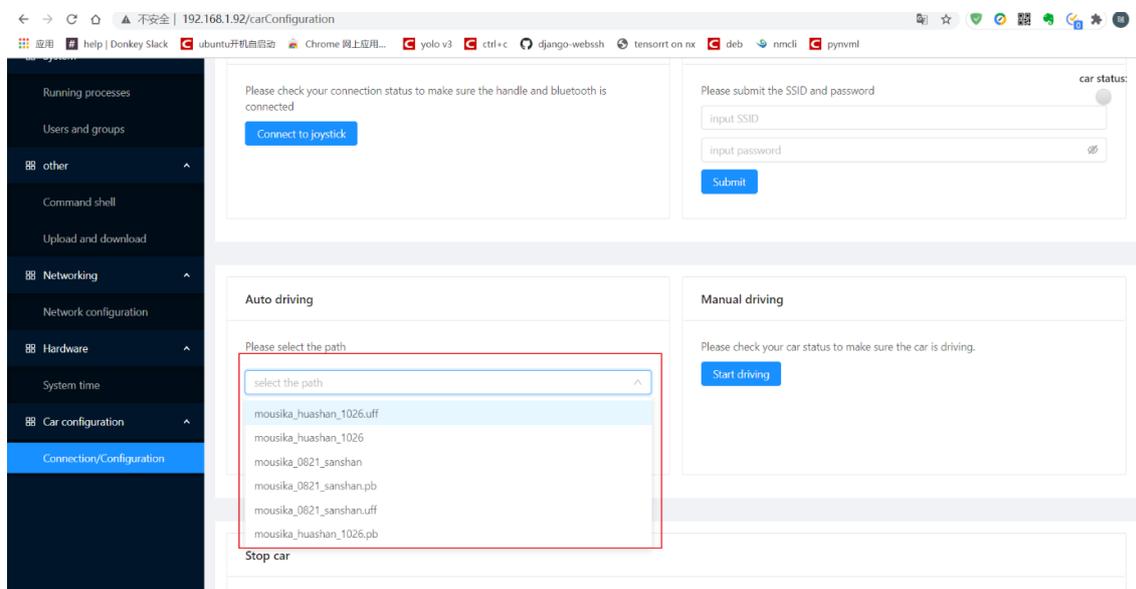
4) 当需要停止手动驾驶模式时，需要点击 Web 页面 Stop Car 模块的 Stop 按键。



7.3.2 自动驾驶

自动驾驶目前支持 2 种类型的模型，tensorflow 模型和 tensorRT 模型。

1) 点击 Select the Path 的下拉框选择想要运行的模型。



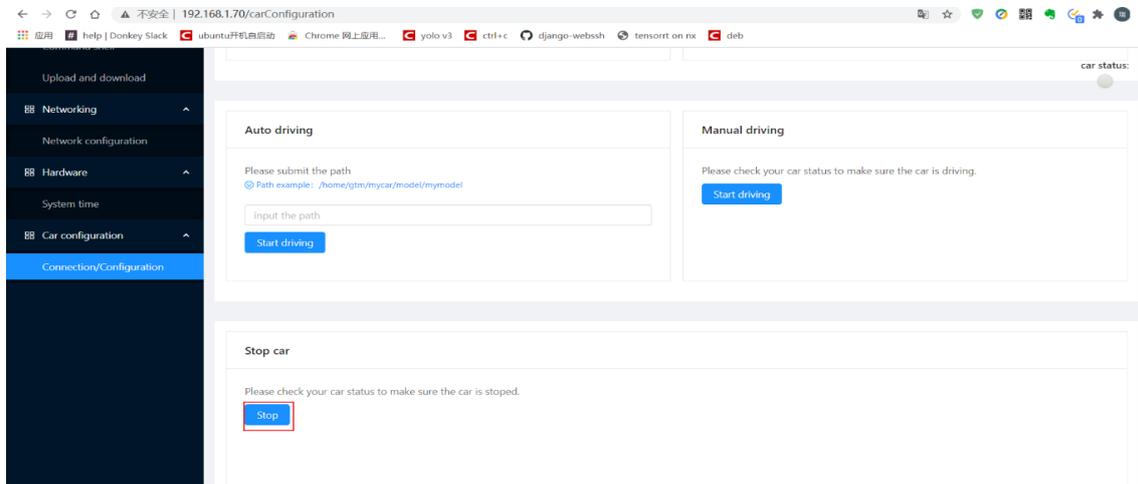
2) 点击 Start Driving 按钮，小车会根据模型的后缀自动判断模型种类。

3) 程序成功运行提示 Success。

4) 当需要停止时点击 Stop Car 模块的 Stop 按键。

7.4 小车初始化

当需要停止手动或者自动驾驶模式的程序时，需要通过 Stop Car 的方式来实现，以下是几种常用的使用 Stop Car 的场景。



- 1) 当需要停止手动驾驶模式时；
- 2) 当需要停止自动驾驶模式时；
- 3) 当需要从手动驾驶切换到自动驾驶模式时，需要先 Stop Car 再切换；
- 4) 当需要从自动驾驶切换到手动驾驶模式时，需要先 Stop Car 再切换；
- 5) 当遇到一些突发的异常导致无法正常启动自动驾驶或者手动驾驶时，可以尝试 Stop Car 刷新小车状态。
- 6) 在启动手动驾驶或者自动驾驶模式之前，如果 Car Status 颜色不是灰色时，可以尝试 Stop Car 刷新小车状态，直至 Car Status 显示灰色。

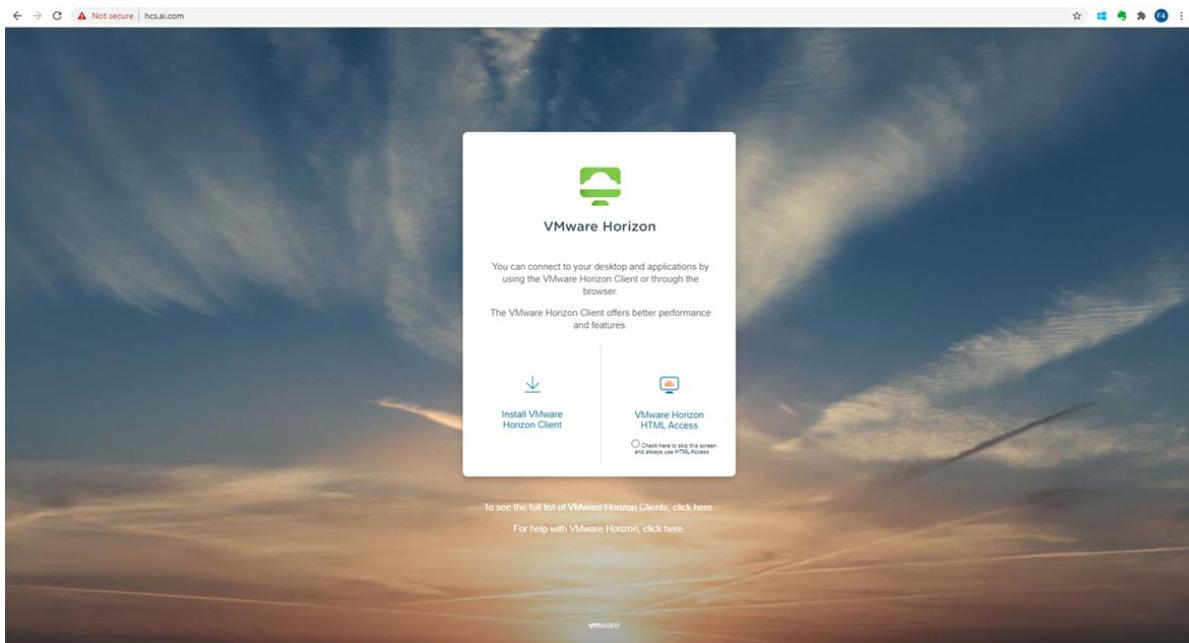
7.5 小车运行状态

在 Web 页面右上角有一个检测小车当前状态的图标，一共有三种颜色，即灰色，绿色和红色。

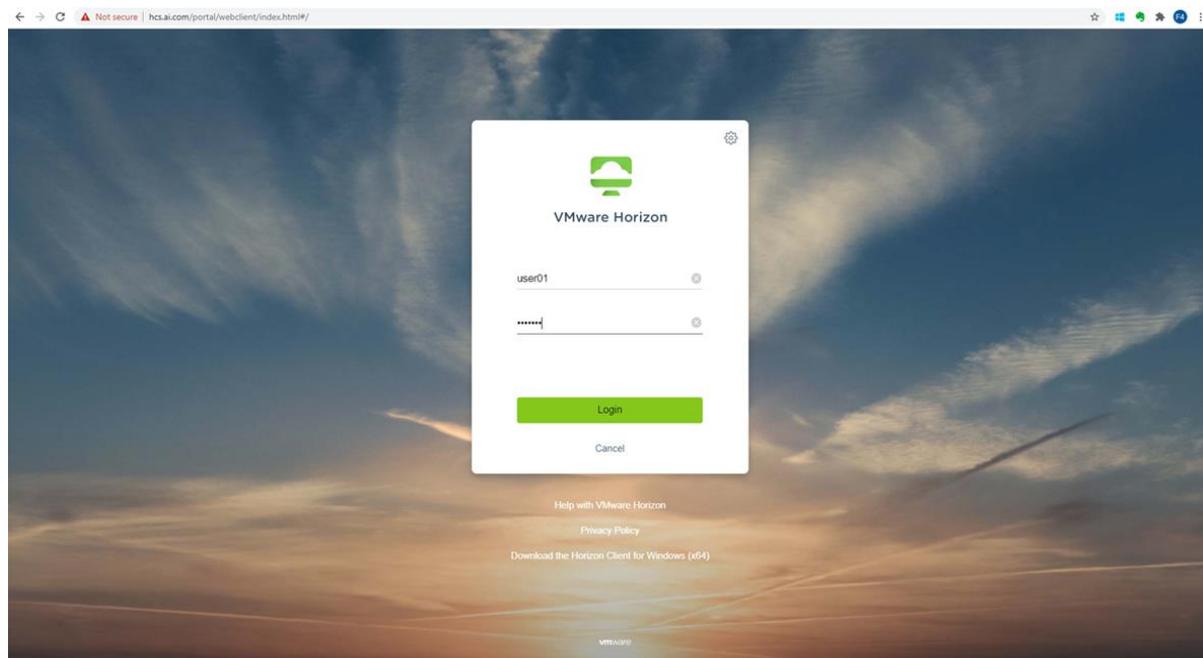
- 1) 小车不处于 Drive 状态时，Car Status 是灰色。
- 2) 小车处于正常启动状态时，Car Status 是绿色。
- 3) 小车程序遇到问题不能正常工作时，Car Status 是红色。

8 VDI 登录 (如果为 vGPU 模式, 此步可忽略, 请直接调到 9)

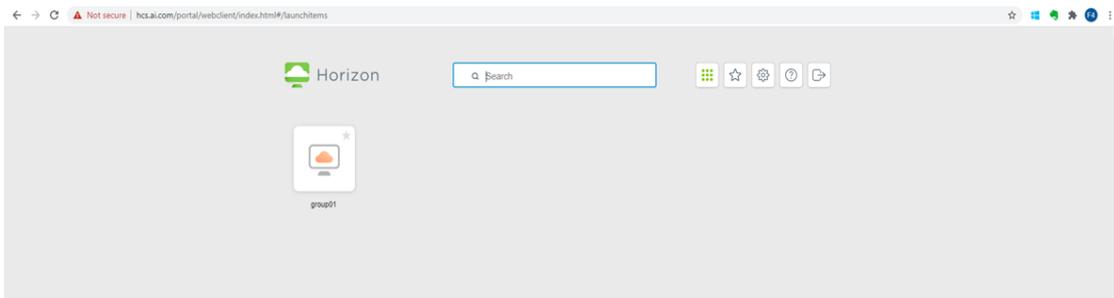
- 1) 请编辑 C:\Windows\System32\drivers\etc\目录下的 hosts 文件, 在 hosts 文件最下边添加 “10.84.126.55hcs.ai.com” 这一行。
- 2) 使用 Chrome 浏览器打开 https://hcs.ai.com, 选择 VMware Horizon HTML Access 访问。



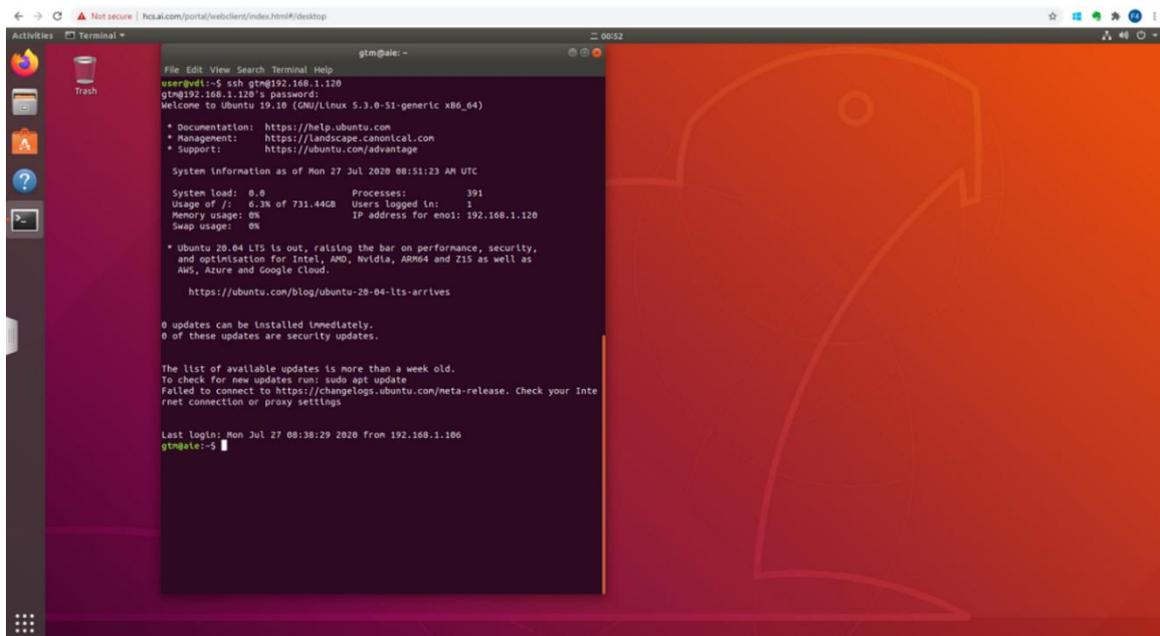
- 3) 使用用户名 “user01”, 密码 “abc123!” 登录虚拟化桌面



4) 点击 group01, 进入虚拟化桌面



5) 虚拟化桌面的账号密码为 “user/abc123!” 打开一个终端, ssh 访问 GPU 服务器, GPU 服务器的账号密码为 “gtm/abc123!”



9 GPU 训练 (如果为 vGPU 模式, 此步可忽略, 请直接调到 9)

用户名&IP: gtuadmin@192.168.1.20 密码: abc123!

- 1) 登录 VDI 远程桌面, 将小车数据 (mycar/tub 目录下) 传入此桌面。

```
rsync -aP mousika@192.168.1.71:~/mycar/tub ./
```

- 2) 查看 mycar/tub 目录中小车的数据, 手动剔除错误数据。

- 3) 将清洗后的数据上传至 GPU 服务器 (推荐放到 mycar/tub 目录下):

```
rsync -aP tub gtuadmin@192.168.1.20:mycar/data/
```

输入密码: abc123! 等待数据上传完成。

4) 在 VDI 中, 通过 ssh 登录到 GPU 服务器:

ssh gpuadmin@192.168.1.20 密码: abc123!

5) 切换到 mycar 目录: cd mycar

6) 训练模型:

`python3 manage.py train --tub <要训练的数据路径(data/tub)> --model <路径+模型名 (models/mymodel)> --base_model <路径+base model 名 (models/base_model)>`

出现如下界面代表程序已经正常运行:

```
0.0 12198.0
leftMemory: 0 TotalMemory: 12198
2020-08-05 11:07:01.591191: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled t
o use: AVX2 AVX512F FMA
2020-08-05 11:07:03.093966: I tensorflow/compiler/xla/service/service.cc:150] XLA service 0x1d95fe0 executing computations on platform CUDA. Devices:
2020-08-05 11:07:03.094081: I tensorflow/compiler/xla/service/service.cc:158] StreamExecutor device (0): Tesla P100-PCIE-12GB, Compute Capability 6.0
2020-08-05 11:07:03.120180: I tensorflow/core/platform/profile_utils/cpu_utils.cc:94] CPU Frequency: 1800000000 Hz
2020-08-05 11:07:03.124631: I tensorflow/compiler/xla/service/service.cc:150] XLA service 0x1d8c290 executing computations on platform Host. Devices:
2020-08-05 11:07:03.124692: I tensorflow/compiler/xla/service/service.cc:158] StreamExecutor device (0): <undefined>, <undefined>
2020-08-05 11:07:03.125065: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1433] Found device 0 with properties:
name: Tesla P100-PCIE-12GB major: 6 minor: 0 memoryClockRate(GHz): 1.3285
pciBusID: 0000:3b:00:0
totalMemory: 11.916GB freeMemory: 11.656GB
2020-08-05 11:07:03.125122: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1512] Adding visible gpu devices: 0
2020-08-05 11:07:03.127412: I tensorflow/core/common_runtime/gpu/gpu_device.cc:984] Device interconnect StreamExecutor with strength 1 edge matrix:
2020-08-05 11:07:03.127449: I tensorflow/core/common_runtime/gpu/gpu_device.cc:990] 0
2020-08-05 11:07:03.127469: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1003] 0: N
2020-08-05 11:07:03.127627: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1115] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 w
ith 5489 MB memory) -> physical GPU (device: 0, name: Tesla P100-PCIE-12GB, pci bus id: 0000:3b:00:0, compute capability: 6.0)
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python/ops/resource_variable_ops.py:435: colocate_with (from tensorflow.python.frame
work.ops) is deprecated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python/keras/layers/core.py:143: calling dropout (from tensorflow.python.ops.nn_ops)
with keep_prob is deprecated and will be removed in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep_prob`.
2020-08-05 11:07:03.702938: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1512] Adding visible gpu devices: 0
2020-08-05 11:07:03.702996: I tensorflow/core/common_runtime/gpu/gpu_device.cc:984] Device interconnect StreamExecutor with strength 1 edge matrix:
2020-08-05 11:07:03.703006: I tensorflow/core/common_runtime/gpu/gpu_device.cc:990] 0
2020-08-05 11:07:03.703014: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1003] 0: N
2020-08-05 11:07:03.703100: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1115] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 w
ith 5489 MB memory) -> physical GPU (device: 0, name: Tesla P100-PCIE-12GB, pci bus id: 0000:3b:00:0, compute capability: 6.0)
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python/ops/math_ops.py:3066: to_int32 (from tensorflow.python.ops.math_ops) is depre
cated and will be removed in a future version.
Instructions for updating:
Use tf.cast instead.
tub_names None
train: 82953, validation: 20739
steps_per_epoch 648
Epoch 1/100
2020-08-05 11:07:12.495080: I tensorflow/stream_executor/dso_loader.cc:152] successfully opened CUDA library libcublas.so.10.0 locally
6/648 [.....] - ETA: 8:36 - loss: 1.2122 - angle_out_loss: 1.3460 - throttle_out_loss: 0.0078
```

7) 等待小车模型训练结束之后, 将模型回传到小车。在小车上输入命令:

`rsync gpuadmin@192.168.1.20:mycar/models/<模型名> ~/mycar/models/`

8) 在小车上验证模型:

`cd mycar`

`sudo python3 manage.py drive --model <模型名>`

10 VGPU 训练

用户名&IP: gtm@10.25.1.199

密码: Emsee123!

- 1) ssh 登录小车终端，将小车数据（mycar/tub 目录下）上传至 GPU 服务器（推荐放到 mousika/tub 目录下）：

```
rsync -aP tub gtm@10.25.1.199:~/mousika/data/
```

输入密码：Emsee123! 等待数据上传完成。

- 2) 通过 ssh 登录到 GPU 服务器：

```
ssh gtm@10.25.1.199          密码：Emsee123!
```

- 3) 切换到 mousika 目录：cd mousika

- 4) 训练模型：

```
python3 manage.py train --tub <要训练的数据路径(data/tub)> --model <路径+模型名  
(models/mymodel)>
```

出现如下界面代表程序已经正常运行：

```
0.0 12198.0  
leftMemory: 0 TotalMemory: 12198  
2020-08-05 11:07:01.591191: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled t  
o use: AVX2 AVX512F FMA  
2020-08-05 11:07:03.093966: I tensorflow/compiler/xla/service/service.cc:150] XLA service 0x1d95fe0 executing computations on platform CUDA. Devices:  
2020-08-05 11:07:03.094081: I tensorflow/compiler/xla/service/service.cc:158] StreamExecutor device (0): Tesla P100-PCIE-12GB, Compute Capability 6.0  
2020-08-05 11:07:03.120180: I tensorflow/core/platform/profile_utils/cpu_utils.cc:94] CPU Frequency: 1800000000 Hz  
2020-08-05 11:07:03.124631: I tensorflow/compiler/xla/service/service.cc:150] XLA service 0x1d8c290 executing computations on platform Host. Devices:  
2020-08-05 11:07:03.124692: I tensorflow/compiler/xla/service/service.cc:158] StreamExecutor device (0): <undefined>, <undefined>  
2020-08-05 11:07:03.125065: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1433] Found device 0 with properties:  
name: Tesla P100-PCIE-12GB major: 6 minor: 0 memoryClockRate(GHz): 1.3285  
pciBusID: 0000:3b:00.0  
totalMemory: 11.9161B freeMemory: 11.6561B  
2020-08-05 11:07:03.125122: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1512] Adding visible gpu devices: 0  
2020-08-05 11:07:03.127412: I tensorflow/core/common_runtime/gpu/gpu_device.cc:984] Device interconnect StreamExecutor with strength 1 edge matrix:  
2020-08-05 11:07:03.127449: I tensorflow/core/common_runtime/gpu/gpu_device.cc:998] 0  
2020-08-05 11:07:03.127469: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1683] 0: N  
2020-08-05 11:07:03.127627: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1115] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 w  
ith 5489 MB memory) -> physical GPU (device: 0, name: Tesla P100-PCIE-12GB, pci bus id: 0000:3b:00.0, compute capability: 6.0)  
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python/ops/resource_variable_ops.py:435: colocate_with (from tensorflow.python.frame  
work_ops) is deprecated and will be removed in a future version.  
Instructions for updating:  
Colocations handled automatically by placer.  
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python/keras/layers/core.py:143: calling dropout (from tensorflow.python.ops.nn_ops)  
with keep_prob is deprecated and will be removed in a future version.  
Instructions for updating:  
Please use 'rate' instead of 'keep_prob'. Rate should be set to 'rate = 1 - keep_prob'.  
2020-08-05 11:07:03.702938: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1512] Adding visible gpu devices: 0  
2020-08-05 11:07:03.702996: I tensorflow/core/common_runtime/gpu/gpu_device.cc:984] Device interconnect StreamExecutor with strength 1 edge matrix:  
2020-08-05 11:07:03.703068: I tensorflow/core/common_runtime/gpu/gpu_device.cc:998] 0  
2020-08-05 11:07:03.703614: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1603] 0: N  
2020-08-05 11:07:03.703169: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1115] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 w  
ith 5489 MB memory) -> physical GPU (device: 0, name: Tesla P100-PCIE-12GB, pci bus id: 0000:3b:00.0, compute capability: 6.0)  
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python/ops/math_ops.py:3066: to_int32 (from tensorflow.python.ops.math_ops) is depre  
cated and will be removed in a future version.  
Instructions for updating:  
Use tf.cast instead.  
tub_names None  
train: 82953, validation: 20739  
steps per epoch 648  
Epoch 1/100  
2020-08-05 11:07:12.495080: I tensorflow/stream_executor/dso_loader.cc:152] successfully opened CUDA library libcublas.so.10.0 locally  
6/648 [.....] - ETA: 8:36 - loss: 1.2122 - angle_out_loss: 1.3460 - throttle_out_loss: 0.0078
```

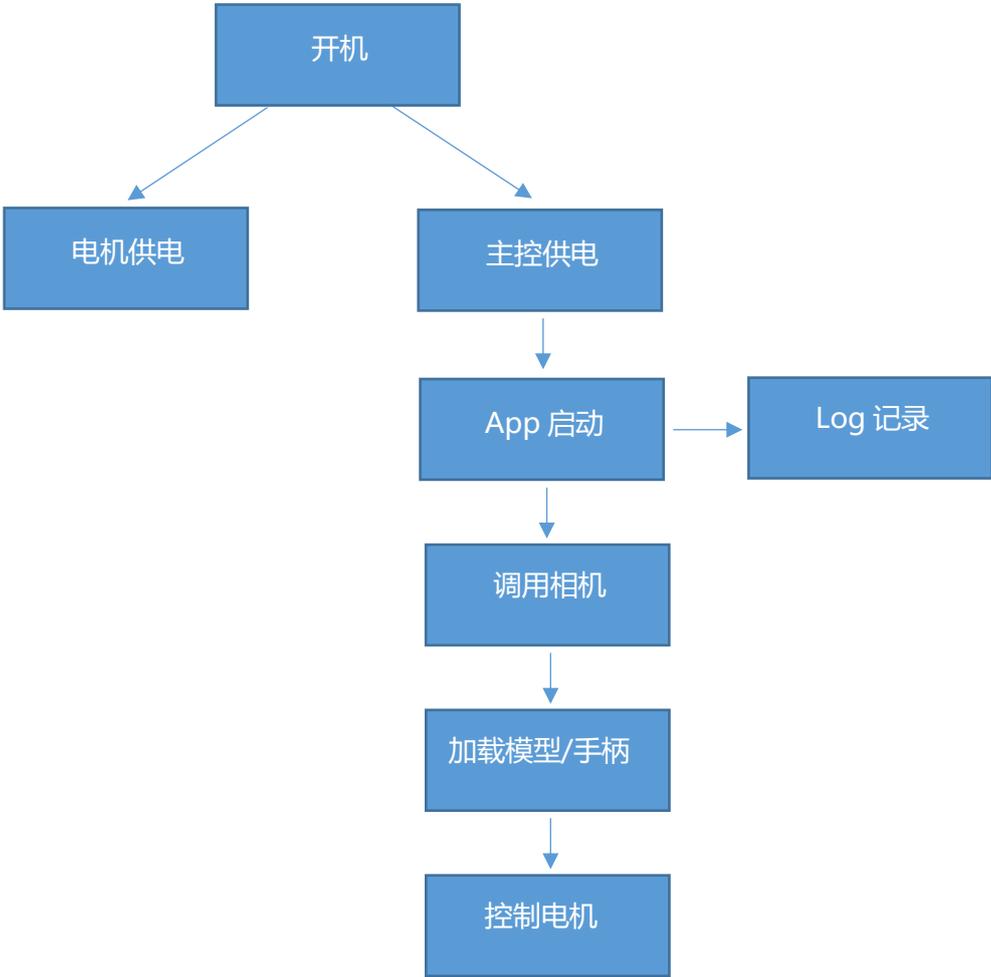
- 5) 等待小车模型训练结束之后，将模型回传到小车。在小车上输入命令：

```
rsync gtm@10.25.1.199:~/mousika/models/<模型名> ~/mycar/models/
```

- 6) 在小车上验证模型：

浏览器打开小车地址，在自动驾驶菜单选择训练完成的模型。

11 运行流程图示



12 关键模块运行错误的判定方法

1) 电池

接通电源后，电机提示灯和主板提示灯开启；如果所有灯都没亮说明电池坏了，需要更换电池。

2) 主控

电源接通 10 秒左右主控模组上粉色的灯常亮，如果粉色的灯未被点亮说明主控故障，需更换。

3) 电机

使用 ssh 登录小车，在确定没有小车运行的情况下（可以先在 Web 界面点击 Car Stop 按键），运行 `sudo python3 /home/mousika/mycar/unitest/control_test.py` 此时小车会前进、转向等行为运行 1 分钟，如果存在某个电机不转的情况，说明电机故障，需更换。

4) 相机

使用 ssh 登录小车，在确定没有小车运行的情况下（可以先在 Web 界面点击 Car Stop 按键），运行 `sudo python3 /home/mousika/mycar/unitest/camera_test.py` 此时小车会将相机数据以矩阵的格式打印出来，如果报错说明相机故障，需更换。

5) 其他报错

当程序不能正常运行，且以上的 troubleshooting 步骤都通过的情况下，可以到 `/home/mousika/mycar/logs/gun.conf_error.log` 查看报错日志，根据日志提示 troubleshooting。