

2024年“海河工匠杯”技能大赛暨中华人民共和国第三  
届职业技能大赛

人工智能训练赛项

竞赛  
试题

工位号：\_\_\_\_\_

2024年5月

竞赛时间：480分钟

竞赛准备：

1. 任务书共 16 页，如出现任务书缺页、字迹不清等问题，请及时向裁判申请更换任务书，选手须认真阅读本任务书完成比赛。
2. 每个赛位配有人工智能技术应用实训模组一套，选手在比赛开始前，认真检查工位设备，确认后开始比赛。
3. 比赛共包含四个竞赛模块，总分 100 分，见表 1。

表 1 竞赛任务

序号	名称	配分	时长
1	模块 A：人工智能数据集制作	40	180 分钟
2	模块 B：人工智能算法测试与优化	30	180 分钟
3	模块 C：系统开发测试与运维	25	120 分钟
4	模块 D：安全意识与职业素养	5	贯穿竞赛全程
合计		100	480 分钟

4. 选手全程操作人工智能技术应用实训模组系统选项为 hdkj 密码为：  
hdkj
5. 选手比赛所需要的资料都以电子版形式保存在设备指定位置  
/home/hdkj/desktop/competition/中。
6. 参赛队应在**480分钟**内完成任务书规定内容；选手在竞赛过程中需要存储的电子文件按照任务书指定路径存放，选手必须在桌面上建立储存文件夹用于储存任务书未规定的其他电子文件，并命名文件夹为“场次+工位号”（例如A01），其中A为场次，01代表赛位。未存储到指定位置的运行记录或程序文件均不予给

---

分，选手应及时存储相关程序文件，由于断电或其他原因引起的程序丢失将由选手自行负责，断电时间可根据情况给予补时。

7. 选手提交的报告书、工单等不得出现学校、姓名等与身份有关的信息，只填写工位号，否则成绩无效。

8. 选手比赛过程中，应该严格遵守相关的规章制度和安全守则，若有违反，按照相关规定在竞赛总成绩中扣除相应分值。严格遵守安全操作规范，若发生危及设备或人身安全事故，立即停止比赛，取消参赛资格。

9. 选手须在规定时间内完成竞赛任务，并向裁判举手示意进行评分，除有说明外，不限制各任务评判顺序，且不限任务中各项的先后顺序，选手在实际比赛过程中要根据赛题情况进行操作，每个任务裁判只验收1次，根据任务要求，确认完成后再提请裁判验收。

10. 选手除竞赛结束后备份竞赛结果外，在比赛过程中**禁止使用U盘**等工具，严谨携带通讯设备、存储设备和技术资料，如有发现取消竞赛资格。未经允许，选手禁止离开赛位或与其他选手交流，禁止赛场大声喧哗，严重影响赛场秩序，如有发生取消竞赛资格。

11. 竞赛结束后，选手需将所有文件置于工位上，禁止选手携带任何材料离开竞赛场地。

---

## 竞赛模块 A：人工智能数据集制作

本模块的目标是使用提供的原始数据和现场采集的数据，制作一个规定格式的数据集，该数据集主要用于训练出一个能够识别出智能零售类型和质量的模型权重文件。因此要求参赛队基于提供的原始数据和素材，制定数据采集和标注的方案，完成图像处理、采集、清洗和标注。通过提供的标注软件labeling完成数据集的制作。

### (一) 图像处理

本模块为选手提供了原始数据，按照任务要求完成数据处理工作，提供给选手的原始数据存放在/home/hdkj/desktop/competition/data 中。

#### 任务要求：

(1) 参赛选手编辑/home/hdkj/desktop/competition/Deduplication.py 文件，去掉重复的数据，并将处理后的数据保存至/home/hdkj/desktop/“场次+工位号”（例如 A01）中，并命名文件夹为 Deduplication。

(2) 参赛选手编辑/home/hdkj/desktop/competition/conversion.py 文件，对去重过后的图像进行通道转换处理，转换后颜色应与现场实物颜色一致，并将处理后的数据保存至/home/hdkj/desktop/“场次+工位号”（例如 A01）中，并命名文件夹为 conversion。

(3) 参赛选手编辑/home/hdkj/desktop/competition/concatenation.py 文件，对原始数据与通道转换后的数据进行 2\*3 左右对称拼接（一张即可），并将最后拼接的图像保存至/home/hdkj/desktop/“场次+工位号”（例如 A01）中，并命名文件夹为 concatenation，数据拼接示意图如下：



### 测试要求：

- (1) 要求选手在裁判评分时，展示 **Deduplication** 文件夹。
- (2) 要求选手在裁判评分时，展示 **conversion** 文件夹。
- (3) 要求选手在裁判评分时，展示 **concatenation** 文件夹。

### (二) 图像采集

本模块为选手提供了现场素材、相关设备和软件，按照任务要求完成数据采集工作。

(1) 参赛选手编写 `/home/hdkj/desktop/competition/gather.py` 调用外置摄像头采集图像，并自行清洗数据，要求现场提供的素材采集共100张，分别为可乐 (cola) 25张、雪碧 (sprite) 25张、芬达 (fanta) 25张、口香糖 (gum) 25张，其中各个采集任务文件夹名称分别为括号对应的标签名，采集完成后需将各类数据保存至总文件夹 **collect** 中，将总文件夹保存至 `/home/hdkj/desktop/“场次+工位号”`（例如A01）中。

(2) 选手将处理过的数据 **conversion** 与采集的 **collect** 文件夹中的数据保

---

存至/home/hdkj/desktop/“场次+工位号”（例如A01）中，并将最终数据集命名为**images**。

**测试要求：**

（1）要求选手在裁判评分时，展示图像采集界面功能。

（2）要求选手在裁判评分时，展示 **collect** 文件夹中各个采集任务文件夹数据张数。

**(三) 图像标注**

选手使用标注软件 **labelimg** 完成数据标注工作，制作数据集。

**任务要求：**

（1）创建数据标注任务，将整合好的**images**图像数据使用数据标注工具进行标注。

（2）创建数据标注任务，并依次设置标签名称为**可乐(cola)**、**雪碧(sprite)**、**芬达(fanta)**、**口香糖(gum)**。

（3）完成数据标注工作，生成.xml格式文件，并将标注好的结果保存至/home/hdkj/desktop/“场次+工位号”（例如A01）中，并命名为**Annotations**。

（4）选手自行查看标注后的数据，完成数据审核工作。

（5）将图像数据与标注数据保存至一个文件夹内制作一个标准的VOC格式数据集并保存至/home/hdkj/desktop/“场次+工位号”（例如A01）中，并将文件夹命名为**VOC**。

**测试要求：**

（1）要求选手在裁判评分时，展示创建的**标签名称**。

（2）要求选手在裁判评分时，展示创建的 **Annotations** 文件夹。

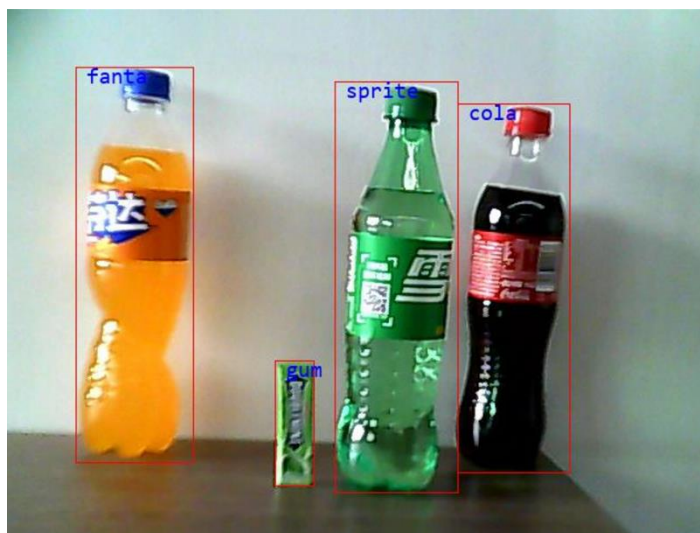
（3）要求选手在裁判评分时，展示创建的 **VOC** 文件夹内容。

#### (四) 标注结果解析

参赛选手对标注结果进行解析，绘制坐标至图像上。

##### 任务要求：

(1) 参赛选手编辑 `/home/hdkj/Desktop/competition/xmlanalysis.py` 文件,对 **Annotations** 中的.xml 数据进行解析,对 **images** 中的原始图像绘制,保存至 `/home/hdkj/desktop/“场次+工位号”` (例如 A01) 中,并命名文件夹为 **analysisImage**。解析示意图如下：



##### 测试要求：

(1) 要求选手在裁判评分时，展示创建的 **analysisImage** 文件夹内容。

#### (五) 标注结果裁剪

参赛选手对标注结果进行裁剪，裁剪标注结果保存至文件夹内。

##### 任务要求：

(1) 参赛选手编辑 `/home/hdkj/desktop/competition/xmlcrop.py` 文件,对 **Annotations** 中的.xml 文件进行裁剪,对 **images** 中的原始图像裁剪,将裁剪的各个标签的数据以文件夹名称(分别为括号对应的标签名),裁剪完成后需将各类数据保存至总文件夹 **crop**,将总文件夹保存至 `/home/hdkj/desktop/“场次+工位号”` (例如 A01) 中。裁剪示意图如下：



### 测试要求：

(1) 要求选手在裁判评分时，展示创建的 **crop** 文件夹中各个文件夹内容。

(六) 填写《人工智能数据集制作工单》

### 竞赛模块 B：人工智能算法测试与优化

本模块的目标是要求参赛队基于制作的图像数据集训练出一个能以较高的准确率对未知图像进行检测和分类的模型。本模块的考核指标为目标检测的准确率。

#### (一) 模型调参及模型训练

本赛题为参赛队提供了基于基线算法和模型。参赛队可以在基线模型基础上，通过参数调优、网络重构等手段，训练出分类准确率更高的模型。

### 任务要求：

(1) 选手使用提供的开发环境和部分脚本训练模型，yolo路径为 `/home/hdkj/darknet`。部分脚本文件可参照 `/home/hdkj/darknet/hdkj` 中脚本文件使用。比赛过程中将提供**预训练模型**。

(2) 选手使用竞赛模块A创建的数据集文件，完成数据切分，制作训练集和验证集。

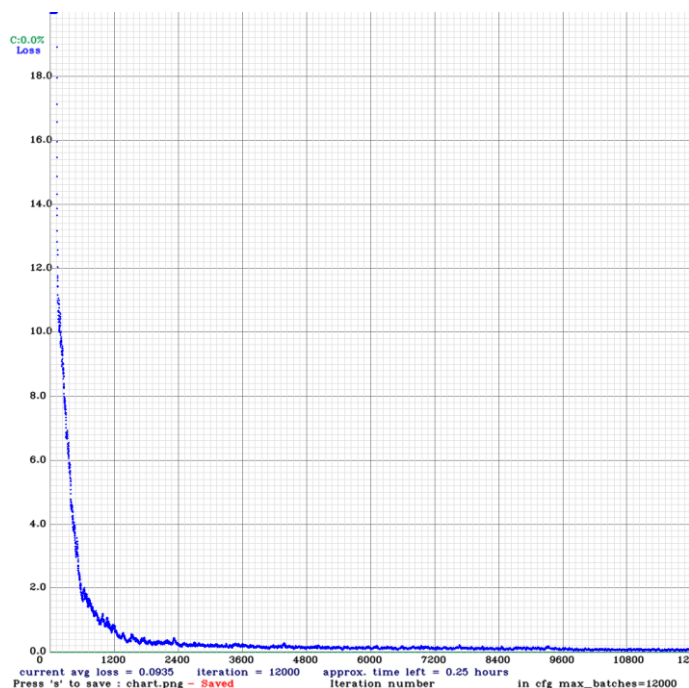
(3) 选手使用标注完成的.xml格式文件，将其转化为.txt格式文件。



(4) 配置训练环境，修改文件路径、数据类别、特征层等。使用预训练模型完成模型训练。

(5) 保存训练日志，生成.log 文件。

(6) 生成合乎规律的 loss 曲线图，最大训练次数应不少于 500，loss 曲线图示意图如下：



测试要求：

(1) 要求选手在裁判评分时，展示切分的训练集和验证集。

(2) 要求选手在裁判评分时，展示转化的.txt 格式文件。

(3) 要求选手在裁判评分时，展示.data 文件。

(4) 要求选手在裁判评分时，展示训练时产生的权重文件。

(5) 要求选手在裁判评分时，展示 loss 曲线图。

(6) 要求选手在裁判评分时，展示.log 日志。

## (二) 模型验证

为了提高模型识别的准确率，对采集图像进行模型识别验证；也可以重新

---

回到训练平台上重新调整网络模型参数以更好地提高模型识别的准确率。

**任务要求：**

- (1) 基于训练好的模型和提供的素材，使用提供的  
`/home/hdkj/desktop/competition/test.jpg`进行测试，并保存测试结果。
- (2) 如果模型识别效果不理想可以重新对模型参数进行调优。
- (3) 将选手自己的`.cfg`、`.data`、`.weights`、`.log`日志，loss曲线图文件及测试图片结果图保存至`/home/hdkj/desktop/“场次+工位号”`（例如A01）中，并命名文件夹为`model`。

**测试要求：**

- (1) 要求选手在裁判评分时，现场实时展示测试图片的识别效果。
- (2) 要求选手在裁判评分时，展示 `model` 文件夹。
- (三) 完成《人工智能算法测试与优化任务工单》填写

**竞赛模块C：系统开发测试与运维**

本模块的目标是要求参赛队基于训练好的模型应用到人工智能技术应用实训模组上，完成任务的识别要求。本模块的考核指标为模型的分类准确率和模型识别结果在嵌入式边缘计算平台上的控制准确率。

**任务背景：**

你来到了一家名为“智能购物”的自助贩卖机店，这是一种全新的零售体验方式。无需人工服务，通过智能化技术，你可以自助选购商品并完成支付，你来到了自助贩卖机区域，贩卖机上都贴着商品列表和价格，贩卖机配备了智能商品识别系统，当你站在贩卖机前，它会自动检测你的面部特征或者扫描你手机上的二维码，识别你的身份和账户信息，选好商品后，你可以选择支付方式，支付完成后，贩卖机会自动将商品送到取货口，你可以取走自己购买的商品，

买到自己需要的东西。

## (一) 应用场景开发

### 任务概述：

将我们的商品置于自助贩卖机中，来实现我们智能零售系统场景。

### 任务要求：

(1) 选手需要编辑/home/hdkj/desktop/competition/control.py, 能够对实训模组的正常通信，能够正常调用设备的舵机、蜂鸣器、电磁锁、微型风扇、警示灯、步进电机等控制执行器件。

(2) 选手需要编辑/home/hdkj/desktop/competition/system.py, 完成智能零售系统场景开发。需要给我们的自助贩卖机放入我们的货物，需放入可乐 (cola)、雪碧 (sprite)、芬达 (fanta)、口香糖 (gum) 各 99 个，参考如图所示：



### 测试要求：

(1) 要求选手在裁判评分时，运行 system.py 文件，展示我们开发的智

能零售系统场景。

## (二) 应用场景验证

### 任务概述：

模拟选手作为购买者来购买商品，通过人工智能技术应用实训模组上面的摄像头，对提供的素材进行数据采集，调用模型识别，根据识别结果模拟智能零售场景下的目标识别与功能开发，并根据识别结果，控制模组实现相应的功能，验证智能零售系统场景。

### 任务要求：

选手需要编辑/home/hdkj/desktop/competition/validate.py, 完成智能零售场景验证。

(1) 选手调用模块 B 训练出来的模型，对可乐 (cola)、雪碧 (sprite)、芬达 (fanta)、口香糖 (gum) 进行实时识别，需静止不动识别 5 秒，5 秒过后确定识别商品，根据识别结果在对应位置绘制图像框与名称，并显示置信度，并将识别商品图片保存至/home/hdkj/desktop/“场次+工位号”（例如 A01）中，并命名图片为 result.jpg。口香糖 (gum) 为红色图像框与标签，其它饮料为绿色图像框与标签进行区分。



(2) 根据串口屏界面对应的物品价格表，计算物品金额并写入串口屏中。

(3) 点击串口屏上的‘确认支付’，打开摄像头扫描提供的二维码付款。

(4) 付款之后，串口界面关闭，同时电磁锁打开，3 秒之后拿完商品且电磁锁关闭。

(5) 我们买完我们想要的商品后，系统的库存会自动减少。



(6) 代码相关指令与硬件控制协议见附件一、串口屏界面模板及相关协议见附件二。

**测试要求:**

(1) 要求选手在裁判评分时，运行 **validate.py** 文件，验证智能零售场景实现。识别饮料框为红色，口香糖框为绿色，需要显示识别置信度。

(2) 智能零售系统界面中商品数量显示正确，商品价格合计显示正确。

(3) 点击‘确认支付’调出识别二维码界面，扫描完成后电磁锁打开，3 秒后关闭。

(4) 展示商品购买完成后的智能零售界面。

(三) 完成《系统开发测试与运维任务工单》填写

---

## 附件一：代码相关命令

### 关键命令：

1、`cv2.VideoCapture()` 参数是 0，表示打开内置摄像头，参数是视频文件路径则打开视频，如 `vc = cv2.VideoCapture("../test1.mp4")`

2、`ret, frame = vc.read()`

`vc.read()` 按帧读取视频，`ret, frame` 是获 `cap.read()` 方法的两个返回值。其中 `ret` 是布尔值，如果读取帧是正确的则返回 `True`，如果文件读取到结尾，它的返回值就为 `False`。`frame` 就是每一帧的图像，是个三维矩阵。

3、`cv2.waitKey(1)`，`waitKey()` 方法本身表示等待键盘输入，

参数是 1，表示延时 1ms 切换到下一帧图像，对于视频而言；

参数为 0，如 `cv2.waitKey(0)` 只显示当前帧图像，相当于视频暂停；

参数过大如 `cv2.waitKey(1000)`，会因为延时过久而卡顿感觉到卡顿。

4、按键判断得到的是键盘输入的 ASCII 码，`esc` 键对应的 ASCII 码是 27，即当按 `esc` 键是 `if` 条件句成立

5、调用 `release()` 释放摄像头，调用 `destroyAllWindows()` 关闭所有图像窗口。

`cv2.imshow()` 函数可以在窗口中显示图像。该窗口和图像的原始大小自适应（自动调整到原始尺寸）。

6、`cv2.rectangle` 这个函数的作用是在图像上绘制一个简单的矩形。

`cv2.rectangle(image, start_point, end_point, color, thickness)`。

7、`cv2.putText()` 的功能是在一个图片上写文字，其用法为 `cv2.putText(img, str(i), (123, 456)), font, 2, (0, 255, 0), 3)`。

8、`cv2.resize()` 对图片进行缩放的作用，`cv2.resize(src, dsize, dst=None, fx=None, fy=None, interpolation=None)`。`src`（必需）输入原图像，`dsize`

---

(必需) 输出图像的大小, fx (可选) width 方向的缩放比例, fy (可选) height 方向的缩放比例, interpolation (可选) 这个是指定插值的方式。

9、cv2.imwrite 写入保存图片。imwrite(img\_path\_name,img,name),  
img\_path\_name: 保存的文件名, img: 文件对象。

10、cv2.cvtColor() 函数来改变图像的颜色空间, 例如该函数形式为:  
cv2.cvtColor(frame,cv2.COLOR\_BGR2RGB)。

11、serial.Serial 串口设置, 例如 serial.Serial("/dev/ttyUSB0",9600,  
timeout=0.5) 分别设置端口, 波特率, 连接时间设置。

12、ser.isOpen() 看看这个串口是否已经被打开。

13、ser.open() 打开端口

14、ser.read(10) 从端口读 10 个字节

15、ser.write("hello") 向端口写数据

16、ser.readline() 是读一行, 以/n 结束, 要是没有/n 就一直读, 阻塞。  
ser.readlines() 和 ser.xreadlines() 都需要设置超时时间。

17、flushOutput(): 终止当前写操作, 并丢弃发送缓存中的数据。

18、ser.flushInput() 清除缓存。

19、ser.close() 关闭端口。

---

## 附件二：硬件通信协议

### 1. 指示灯控制：B-111!

B: 指示灯功能 /111: 三个灯全灭（顺序：红绿黄） /!: 帧尾

例：B-101! 含义：点亮指示灯模块绿灯

### 2. 舵机旋转：A-000!

A: 舵机控制功能 /-: 帧头 /000: 舵机旋转角度，范围 0-180 /!: 帧尾

例：A-90! 含义：舵机旋转到 90° 位置

### 3. 蜂鸣器控制：E-!

E: 蜂鸣器单次报警 /-: 帧头 /!: 帧尾

例：E-! 含义：蜂鸣器单次报警

### 4. 风扇控制：F-1000!

F: 风扇控制功能 /-: 帧头 /0000: 风扇速度，范围 0-1999 /!: 帧尾

例：F-170! 含义：风扇以 170 的速度旋转（数字越小越快）

F-2000! 含义：风扇停止旋转

### 5. 步进电机控制：H-1!

H: 步进电机控制功能 /-: 帧头 /0: 步进电机停止旋转 /1: 步进电机顺时针旋转 /2: 步进电机逆时针旋转 /!: 帧尾

例：H-0! 含义：步进电机停止旋转

H-1! 含义：步进电机顺时针旋转

H-2! 含义：步进电机逆时针旋转

I: 步进电机速度控制功能 /-: 帧头 /001: 步进电机速度参数，1-999 区间，单位 ms /!: 帧尾

例：I-003! 含义：步进电机以 3ms 的间隔转动（默认为 3）

### 6. 电磁锁控制：C-!

C: 电磁锁打开 3 秒后关闭

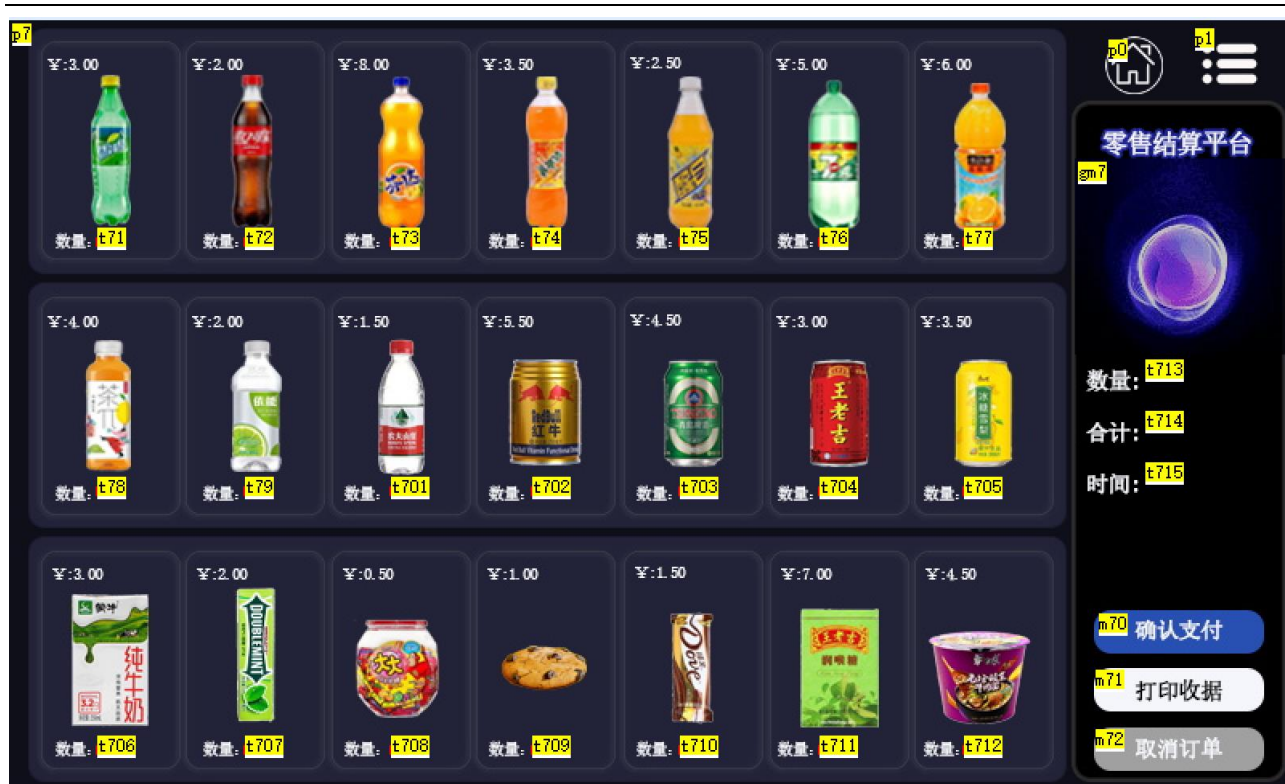
### 7. 串口屏协议：M!

返回：AT:0!代表当前处于非透传状态，此时可控制模组硬件部分

返回：AT:1!代表当前处于透传状态，此时可控制模组串口屏部分

界面切换：page 7





## 数据协议

标题文本: t111.txt= “\*\*\*”, 支持 30 字符

标题颜色: t111.pco= “\*\*\*”

横向文本 1: t112.txt= “\*\*\*” 支持 6 字符

颜色: t112.pco= “\*\*\*”