

物联网大作业说明

组员信息：

181250120戎鑫

181250213朱盛盛

181250146王宇博

181250035高金比尔

1. 相关设备

1. Arduino

采用Arduino开发板，板上连有DHT11温湿度传感器和蜂鸣器

2. 边缘设备

采用笔记本电脑作为边缘设备。在边缘设备上，用Python写了一个服务器，用来读取温湿度数据并发送给EdgeX边缘计算平台，同时接收规则引擎判断成功后使蜂鸣器响的请求

3. EdgeX

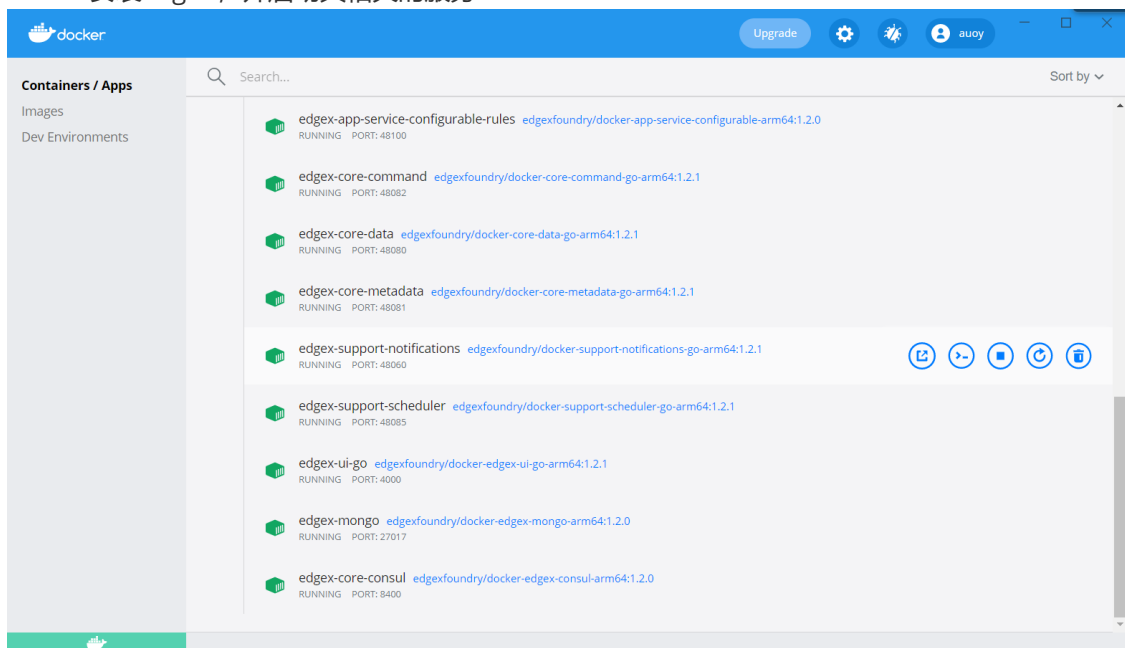
我们使用Docker搭建了EdgeX的运行环境，通过HTTP REST请求的方法去调用EdgeX的命令和接口，通过配置文件以及命令调用去存取数据，配置设备和服务

4. Huawei Atlas 200DK

我们使用了200DK，有人脸识别功能，当识别到人脸时候，发出发现人脸的数据。

2. 实现步骤

1. docker安装EdgeX，并启动其相关的服务



2. 编写EdgeX平台上创建Device所需的配置文件

```
name: "edgex_arduino"
manufacturer: "Dell"
model: "Cam123456"
labels:
  - "arduino"
description: "arduino profile"
coreCommands:
  -
    name: temperature
    put:
      path: "/api/v1/devices/{deviceId}/temperature"
      responses:
        -
          code: "204"
          description: "arduino temperature"
          expectedValues: ["temperature"]
        -
          code: "503"
          description: "service unavailable"
          expectedValues: ["carerror"]
```

3. 按照EdgeX官方文档的教程在EdgeX平台上创建一个DeviceService

The screenshot shows the EdgeX Foundry Console interface. On the left sidebar, the 'DeviceService' tab is selected. The main panel displays a table of DeviceServices. A red arrow points to the entry 'arduino device service'.

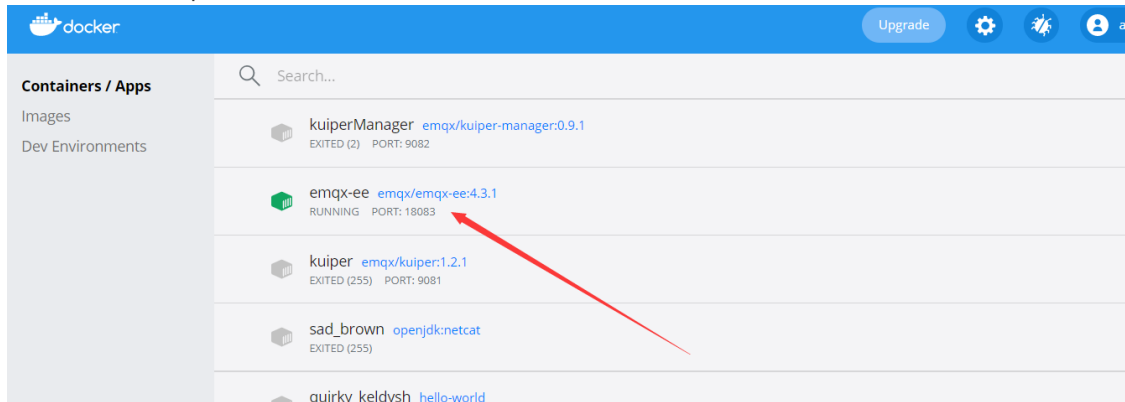
#	ID	Name	Description	Labels	Addressable	OperatingState	AdminState	Devic
1	8948b49e-0fa2-4576-be52-49d65c1fc83	edgex-device-rest			Q	ENABLED	UNLOCKED	
2	9abbcba-9343-4d11-9d1f-4078479c549c	device-virtual			Q	ENABLED	UNLOCKED	
3	29fd75e4-3bb2-4223-a4e5-d2e929a6177f	device-random			Q	ENABLED	UNLOCKED	
4	eca3c07a-7f78-4a57-b0d3-5771f0a97255	edgex-device-mqtt			Q	ENABLED	UNLOCKED	
5	1d5aa45b-0a77-48e9-9bde-abe496e47369	camera control device service	Manage human and dog counting cameras	camera,counter	Q	ENABLED	UNLOCKED	
6	5d54a6a1-570f-43d0-8925-90a6a9c797a8	arduino control device service	environment temperature	arduino,temperature	Q	ENABLED	UNLOCKED	
7	9ca8da8a-d39a-4aa5-b68d-c8ec0c3e2cc8	arduino control device service1	environment temperature	arduino,temperature	Q	ENABLED	UNLOCKED	
8	8e739115-c1b6-48a0-bf00-870c85f3955f	arduino device service	dht11 temperature	arduino,temperature	Q	ENABLED	UNLOCKED	

4. 再根据步骤1中的配置文件创建Device

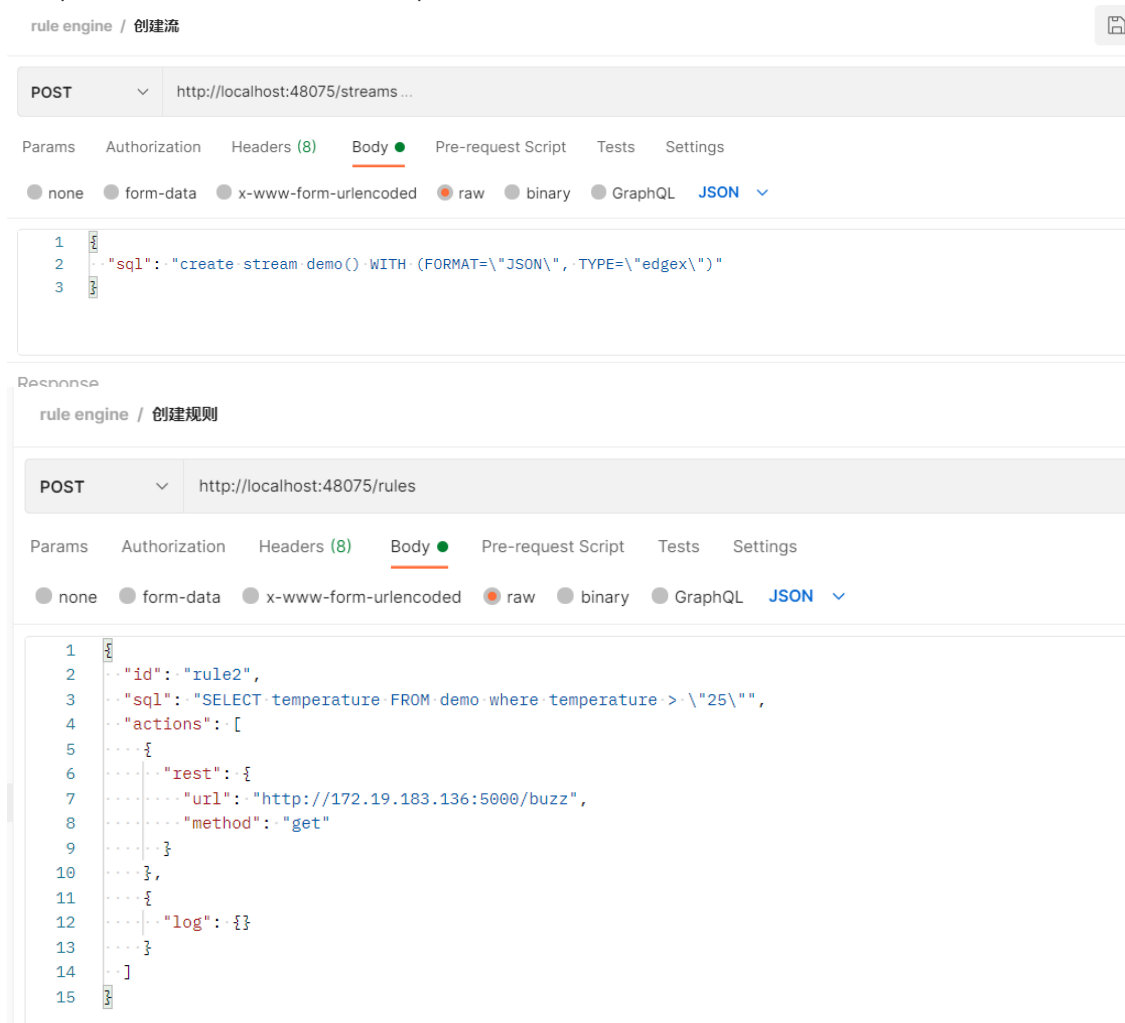
The screenshot shows the EdgeX Foundry Console interface. On the left sidebar, the 'DeviceService' tab is selected. The main panel displays a table of Devices. A red arrow points to the entry 'arduino'.

#	ID	Name	Description	Labels	Commands	Profile	OperatingState	AdminState	Created Time	Modif
1	7470a891-360f-41df-b3af-5187772a7657	arduino	arduino	arduino,temperature	>...	edgex_arduino	ENABLED	UNLOCKED	2021-07-01 20:05:55	2021-07-

5. docker安装emqx-ee, 并启动



6. 使用postman发送创建流和规则的api如下, 进行流和规则的创建



7. 使用python编写server与EdgeX平台通信，其中buzz函数用来通过串口向Arduino发送信号，使蜂鸣器报警；read函数用来从串口读取DHT11获取到的温度数据，并将其发送到EdgeX平台上。根据上一步的规则引擎，当温度大于25℃时，会自动调用本机5000端口server中的buzz函数

```
@app.route("/buzz")
def buzz():
    ser.write('a'.encode())
    print("蜂鸣器响")
    return "succeed"

@app.route("/read")
def read():
    count = ser.inWaiting() # 获取串口缓冲区数据
    if count != 0:
        recv = ser.read(ser.in_waiting).decode("gbk") # 读出串口数据，数据采用gbk编码
        index = recv.find('Temperature', 0, len(recv))
        if index != -1:
            temstr = recv[index + 18: index + 20]
            print(temstr)
            data = '{"device": "arduino", "readings": [{"name": "temperature", "value": "' + str(temstr) + '"}]}'
            headers = {'Content-Type': 'application/json'}
            rep = requests.post(url='http://127.0.0.1:48080/api/v1/event', data=data, headers=headers)
            print(rep)
        return temstr

if __name__ == '__main__':
    app.run(host="172.19.183.136", port=5000)
```

8. 另外用一个python程序每三秒通过url调用上一步中的read函数读串口数据

```
def getTemperature():
    while True:
        requests.get("http://172.19.183.136:5000/read")
        time.sleep(3) # 间隔

if __name__ == '__main__':
    getTemperature()
```

9. 修改Arduino的代码，当从串口读到字符'a'时，报警

```
t=(float)DHT11.temperature;
Serial.print("Temperature (oC): ");
Serial.println(t, 2);

//读取数据
if(Serial.available()>0)//如果串口有数据进入的话
{
    c=Serial.read();//每次读一个字符，是ASCII码的
}

if(c=='a'){
    digitalWrite(buzzPin,LOW);        //输入低电平，响
    delay(1000);
    digitalWrite(buzzPin,HIGH);
    c = 'b';
}
else if(c=='b'){
    digitalWrite(buzzPin,HIGH);        //输入高电平
}
delay(2000);
```

10. 人脸识别相关代码实现

```
void FaceDetect::PostProcess(vector<DetectionResult>& detectResults,
                             uint32_t imageWidth, uint32_t imageHeight,
                             vector<InferenceOutput>& modelOutput) {
    uint32_t dataSize = 0;
    float* detectData = (float *)modelOutput[kBBoxDataBufId].data.get();
    uint32_t* boxNum = (uint32_t *)modelOutput[kBoxNumDataBufId].data.get();

    uint32_t totalBox = boxNum[0];

    for (uint32_t i = 0; i < totalBox; i++) {
        DetectionResult oneResult;
        Point point_lt, point_rb;
        uint32_t score = uint32_t(detectData[SCORE + i * kItemSize] * 100);
        if (score < 70) {
            break;
        }
        ATLAS_LOG_INFO("Find a people\n");

        point_lt.x = detectData[TOPLEFTX + i * kItemSize] * imageWidth;
        point_lt.y = detectData[TOPLEFTY + i * kItemSize] * imageHeight;
        point_rb.x = detectData[BOTTOMRIGHTX + i * kItemSize] * imageWidth;
        point_rb.y = detectData[BOTTOMRIGHTY + i * kItemSize] * imageHeight;
```

3. 最终效果

server控制台输出

```

25
192.168.144.1 - - [03/Jul/2021 11:52:20] "GET /read HTTP/1.1" 200 -
<Response [200]>
26
192.168.144.1 - - [03/Jul/2021 11:52:23] "GET /read HTTP/1.1" 200 -
<Response [200]>
蜂鸣器响
192.168.144.1 - - [03/Jul/2021 11:52:23] "GET /buzz HTTP/1.1" 200 -
26
<Response [200]>
蜂鸣器响
192.168.144.1 - - [03/Jul/2021 11:52:26] "GET /read HTTP/1.1" 200 -
192.168.144.1 - - [03/Jul/2021 11:52:26] "GET /buzz HTTP/1.1" 200 -
26
<Response [200]>
蜂鸣器响
192.168.144.1 - - [03/Jul/2021 11:52:29] "GET /read HTTP/1.1" 200 -
192.168.144.1 - - [03/Jul/2021 11:52:29] "GET /buzz HTTP/1.1" 200 -
25
192.168.144.1 - - [03/Jul/2021 11:52:32] "GET /read HTTP/1.1" 200 -
<Response [200]>
25

```

EdgeX平台接收到的数据

edgex device / get temperature

GET http://localhost:48080/api/v1/reading/name/temperature/3 ...

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Body Cookies Headers (3) Test Results Status: 200 OK Time: 6

Pretty Raw Preview Visualize JSON

```

1  [
2    {
3      "id": "ce3139f2-0729-44cf-af7d-d067181fd82d",
4      "created": 1625284354783,
5      "modified": 1625284354783,
6      "device": "arduino",
7      "name": "temperature",
8      "value": "25"
9    },
10   {
11     "id": "ee40d942-03f4-4a0b-b395-72e2ac983578",
12     "created": 1625284351733,
13     "modified": 1625284351733,
14     "device": "arduino",
15     "name": "temperature",
16     "value": "25"
17   },
18   {
19     "id": "1217eba9-44be-4bd9-9cdf-5c291e72a3ba",
20     "created": 1625284348626,
21     "modified": 1625284348626,
22     "device": "arduino",
23     "name": "temperature",
24     "value": "26"
25   }
26 ]

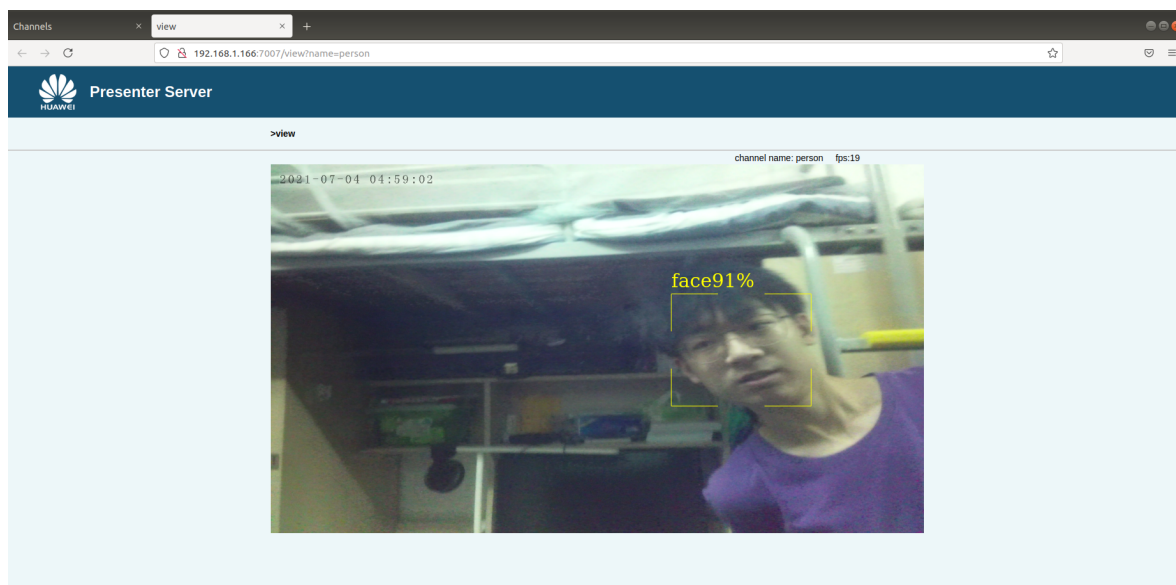
```

分别运行虚拟机和200dk开发板端程序

```
HwHIAIUser@davinci-mini:~/face_detection_camera/out$
File Edit View Search Terminal Help
[INFO] Finalize acl ok
HwHIAIUser@davinci-mini:~/face_detection_camera/out$ ./main
[INFO] Acl init ok
[INFO] Open device 0 ok
[INFO] Use default context currently
content_type config string:1
[ERROR] Open channel failed, error 2
[ERROR] Init resource failed, error 1
[INFO] Model(../model/face_detection.on) had not been loaded or unload already
[INFO] Reset device 0 ok
[INFO] Finalize acl ok
HwHIAIUser@davinci-mini:~/face_detection_camera/out$ ./main
[INFO] Acl init ok
[INFO] Open device 0 ok
[INFO] Use default context currently
content_type config string:1
[INFO] Dvpp init resource ok
[INFO] Load model ../model/face_detection.on success
[INFO] Create model description success
[INFO] Create model(../model/face_detection.on) output success
[INFO] Init model ../model/face_detection.on success
[INFO] No specified camera id, use camera1
[INFO] Open camera 1 success

HwHIAIUser@ubuntu:~/samples/common$
File Edit View Search Terminal Help
HwHIAIUser@ubuntu:~/samples/common$ cd $HOME/samples/common/
HwHIAIUser@ubuntu:~/samples/common$ bash run_presenter_server.sh ../cplussplus/le
vel2_stmple_inference/2_object_detection/face_detection_camera/scripts/face_de
tction.conf
Check python3 libs .....
Looking in indexes: http://mirrors.aliyun.com/pypi/simple/
Requirement already satisfied: protobuf==3.11.3 in /home/HwHIAIUser/.local/lib/p
ython3.0/site-packages (3.11.3)
Requirement already satisfied: setuptools in /usr/lib/python3/dist-packages (fro
m protobuf==3.11.3) (39.0.1)
Requirement already satisfied: six>=1.9 in /usr/lib/python3/dist-packages (from
protobuf==3.11.3) (1.11.0)
python3 libs have been prepared.
check ip 192.168.1.166
check ip 192.168.1.166
7006
Use 192.168.1.166:7006 to connect to Atlas...
Use 192.168.1.166 to show information in browser...
Finish to prepare display presenter server ip configuration.
HwHIAIUser@ubuntu:~/samples/common$ Presenter socket server listen on 192.168.1.
166:7006
Please visit http://192.168.1.166:7007 for display
```

人脸识别情况

[illegible]