

第 5 章



树莓派物联网监测系统



树莓派物联网监测

本章以第 4 章搭建的智能小车作为监测节点,在树莓派内部搭建 LAMP(Linux+Apache+MySQL+PHP)服务器,编写 Web 端监控软件,软件前端发送指令至树莓派服务器后端进行解析,实现远程数据采集、传输以及对监测节点的控制。

5.1 服务器环境搭建

LAMP 架构包括 Linux 操作系统、Apache 网络服务器、MySQL 数据库以及 PHP 编程语言,是目前最为成熟的网站应用模式,能够提供动态 Web 站点服务及其应用开发环境,具有通用、跨平台、高性能的优势。本节使用 LAMP 来构建树莓派 Web 端监控界面,即在 Linux 下开发网站,由 Apache HTTP 服务器提供内容,在 MySQL 数据库中存储内容,用 PHP 来实现程序逻辑。

5.1.1 安装 Apache 服务器

Apache 是一个开源的网页服务器软件,可以通过 HTTP 协议提供 HTML 文件服务。Apache 可以在大多数计算机操作系统中运行,也支持树莓派网页服务。由于其多平台和安全性而被广泛使用,并且可通过简单的 API 扩展,将 Perl/Python 等解释器编译到服务器中。树莓派安装 Apache 服务器的步骤如下:

(1) 在终端输入 `sudo apt-get update` 更新源列表,然后输入 `sudo apt-get install apache2 -y` 安装 Apache。

(2) 安装完成后,需要测试 Apache 服务器是否安装成功。在树莓派浏览器的地址栏输入“127.0.0.1”或“localhost”访问 Apache 服务器,若出现如图 5-1 所示界面,则表示 Apache 服务器安装成功。

在终端输入命令 `sudo service apache2 status` 可以查看 Apache 服务器的状态,分别输入 `sudo service apache2 start/stop/restart` 来控制 Apache 服务器的启动、停止或重启。

5.1.2 安装 MySQL 数据库

数据库是按照数据结构来组织、存储和管理数据的仓库,用户可以对数据库中的数据进行

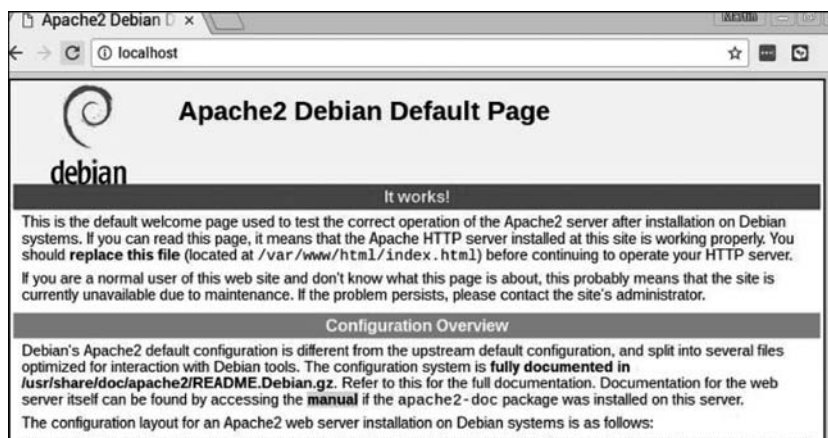


图 5-1 访问 Apache 服务器

行增、删、查、改等操作。MySQL 是最流行的关系型数据库管理系统应用软件之一,因其体积小、速度快、开源的特点,在中小型网站数据库开发中被广泛使用。在树莓派上安装 MySQL 数据库的步骤如下:

(1) 在终端输入命令 `apt-get install mariadb-server`(mariadb 是发展最快的 MySQL 分支版本),待安装完成后,输入 `sudo nano /etc/apache2/apache2.conf` 打开 Apache 服务器的配置文件,将文件中的“AllowOverride None”修改为“AllowOverride All”,如图 5-2 所示。

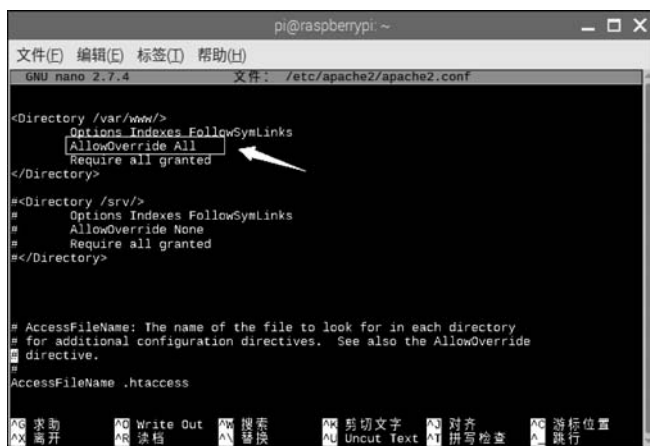


图 5-2 配置 Apache 服务器

(2) 在终端输入 `sudo mysql -u root -p` 命令后直接按 Enter 键即可登录 MySQL 数据库(初始登录时的默认密码为空)。

(3) 在数据库命令行输入 `update mysql. user set plugin='mysql_native_password';` 重置加密方式,再输入 `update mysql. user set password=PASSWORD("12345678") where User='root';` 来更改数据库用户名和密码(本章案例使用的用户名/密码为 root/12345678,读者可

自行设定)。

(4) 最后再输入 **flush privileges**;来刷新权限信息,在键盘上按 Ctrl + C 组合键退出数据库。需要说明的是,对数据库进行操作的命令要以分号结尾。

注意: 如果出现 MySQL 登录 ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password)问题,可以输入 `sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf` 打开配置文件,找到[mysqld]配置项并在该项下添加 `skip-grant-tables`,然后保存文件。输入 `sudo service mysql restart` 重启 MySQL,再次输入 `sudo mysql -u root -p` 命令,当需要输入密码时,直接按 Enter 键便可以不用密码登录到数据库中。

5.1.3 安装 PHP

PHP 是一种基于服务器端来创建动态网站的脚本语言,具有开源免费、简单易懂、跨平台性强的特点。PHP 使用非常广泛,兼容几乎所有的 Web 服务器,支持几乎所有流行的数据库以及操作系统。当服务器通过浏览器收到网页请求时,PHP 解析出需要在页面上显示的内容,然后将该页面发送到浏览器。PHP 是将程序嵌入到 HTML 文档中去执行的,可以比 CGI 或者 Perl 更快速地执行动态网页。树莓派安装 PHP 的步骤如下:

(1) 在终端输入 **sudo apt-get install php php-mysql** 完成 PHP 和 MySQL 连接库的安装。

(2) 为了测试 PHP,输入 **sudo nano /var/www/html/test.php**,在文本编辑器中输入以下内容:

```
<?php
echo "hello!";
echo "欢迎来到树莓派的世界"
?>
```

(3) 在浏览器中访问“`http://192.168.137.3(树莓派 IP 地址)/test.php`”,若出现如图 5-3 所示界面则代表 PHP 测试成功。需要说明的是,后文中网站开发的源程序文件都必须存放到树莓派/`var/www/html`目录下才能正常访问。



图 5-3 PHP 测试

5.1.4 安装 phpMyAdmin

phpMyAdmin 是一个以 PHP 为基础的 MySQL 数据库管理工具,让用户以 Web 接口

管理 MySQL 数据库。通过 phpMyAdmin 可以对数据库进行增、删、查、改等操作。树莓派安装 phpMyAdmin 的步骤如下：

(1) 在终端输入 **sudo apt-get install phpmyadmin** 安装 phpMyAdmin,如图 5-4 所示,选择 **apache2** 后按空格键确定,随后选择默认选项并一直按 **Enter** 键直到界面关闭。

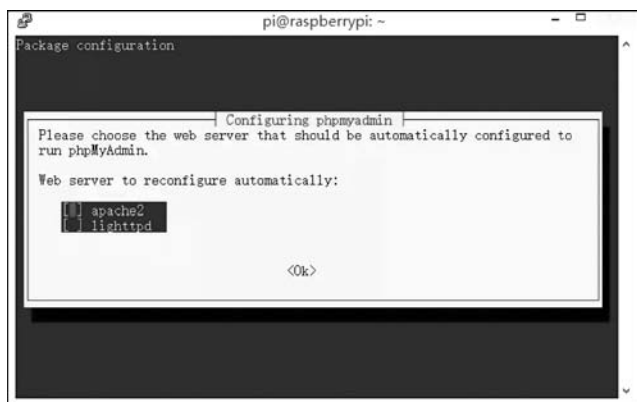


图 5-4 选择 apache2

(2) 安装完成后,需要将 phpmyadmin 链接到 **/var/www/html** 目录下,命令为 **sudo ln -s /usr/share/phpmyadmin /var/www/html**。

(3) 登录 phpmyadmin,在浏览器地址栏输入“树莓派 IP 地址/phpmyadmin”出现如图 5-5 所示的界面,输入之前在安装 MySQL 数据库时设置好的用户名和密码(**root/12345678**)登录即可,接下来即可在数据库管理界面中进行数据的相关操作,如图 5-6 所示。



图 5-5 phpMyAdmin 登录界面



图 5-6 数据库管理界面

经过上述步骤,网站所需的开发环境 LAMP 就已搭建完成。

5.2 树莓派状态读取

为了保证网站服务器稳定可靠地运行,有必要对树莓派的工作状态进行实时监测。Pi Dashboard(Pi 仪表盘)是一款开源的物联网设备监控工具,主要用于树莓派平台以及其他树莓派硬件产品。只需在树莓派上配置好 PHP 环境,就可方便地部署 Pi 仪表盘,并通过 WebUI 来监控树莓派的状态。目前,Pi 仪表盘的监测项包括 CPU 基本信息、状态和使用率,内存、缓存、SWAP 分区、SD 卡使用情况,网络接口的实时数据以及树莓派主机名、操作系统、IP 地址、运行时间等信息。

要实现树莓派状态的读取,首先需要安装 Apache 和 PHP,具体步骤已在 5.1 节中做了介绍,此处不再赘述。然后,在终端输入 `cd /var/www/html` 命令切换到相应的目录,接着输入 `sudo git clone https://github.com/nxez/pi-dashboard.git` 直接通过 GitHub 下载 Pi Dashboard 项目的源码到树莓派的 `/var/www/html` 目录下。

在树莓派中部署好 Pi Dashboard 后,通过 `http://树莓派 IP/pi-dashboard` 即可访问 Pi 仪表盘,如图 5-7 所示。如果页面无法正常显示,则可以尝试在树莓派终端进行如下操作:先切换到 `/var/www/html` 目录,再执行命令 `sudo chown -R www-data pi-dashboard`。该命令会将 `pi-dashboard` 目录及其下面的所有文件与子目录的所有者更改为 `www-data` (apache2 的运行用户是 `www-data`)。



图 5-7 Pi Dashboard 界面

5.3 内网穿透

LAMP 服务器搭建好以后只能在局域网内才可以访问网站,如果需要实现外网访问,则可以使用内网穿透的方法来解决。内网穿透就是进行网络地址转换,将内部网络的私有 IP 地址转换为公网 IP 地址,让外网的终端设备能够访问本地的应用。内网穿透可以通过开放的第三方端口来实现。最常用的方法是安装花生壳内网穿透软件,再添加映射并配置映射端口的信息,外网地址是映射之后访问的域名,这样就可以像平常浏览网页一样通过域名来访问在树莓派上搭建的网站。

花生壳内网穿透的操作步骤如下:

(1) 打开花生壳下载页面(<https://hsk.oray.com/download/>),选择并下载树莓派 5.0 版本(应用平台选择 Raspberry Pi 32 位),如图 5-8 所示。将下载的安装包通过 FileZilla 或 Samba 传输到树莓派中。

(2) 输入命令 `sudo -s` 切换到管理员权限,使用 `cd` 命令切换到安装包所在的目录,输入安装命令 `dpkg -i phtunnel_5_0_rapi_armhf.deb`。安装成功后,会显示花生壳的 SN 码、默认密码(admin)以及远程管理地址 `http://b.oray.com`,如图 5-9 所示。

(3) 输入命令 `phddns` 可以看到扩展功能, `phddns start(启动) | status(状态) | stop(停`



图 5-8 树莓派版花生壳

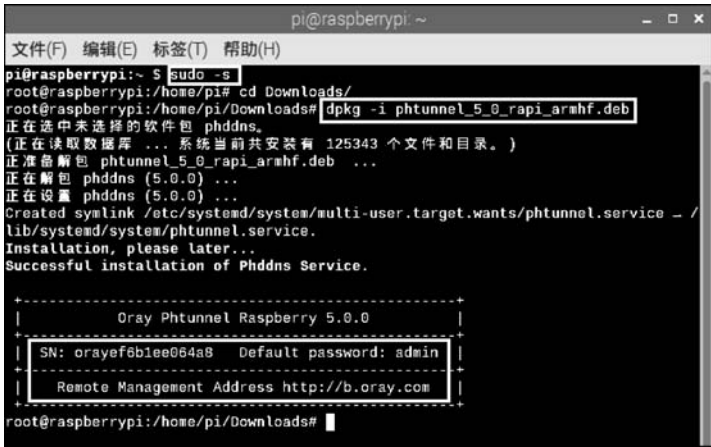


图 5-9 安装树莓派版花生壳

止)| restart(重启)| reset(重置)| enable(开机自启动)| disable(关闭开机自启动)| version (版本),如图 5-10 所示,输入相应命令即可实现对应的功能。

(4) 添加内网穿透映射,具体过程如下:

- ① 花生壳安装完成后,复制生成的 SN 码,在浏览器中访问远程管理地址 <http://b.oray.com>,在页面中输入 SN 码与默认密码(admin)登录,如图 5-11 所示。
- ② 如果是首次登录,则需要通过扫码方式或者密码方式进行激活。前者是指使用花生壳管理 APP 或微信进行扫码激活;后者是输入已注册的 Oray 账号和密码激活,如图 5-12 所示。
- ③ 激活成功后,进入花生壳管理平台。若绑定 SN 码的账号只需要动态域名解析功能,直接单击页面中的“免费开通”按钮,如图 5-13 所示。还可以根据需要将账号升级为带有内网穿透功能的服务版本。

```

root@raspberrypi:/home/pi# phddns
Phtunnel Serive called with unknown argument
(phddns |start|status|stop|restart|reset|enable|disable|version)
root@raspberrypi:/home/pi# phddns start ←
phtunnel service start success !
root@raspberrypi:/home/pi# phddns status ←
+-----+
| Oray PeanutHull Linux 5.0.0 |
+-----+
| Runstatus: ONLINE |
+-----+
| SN: oray7cef70af28d5 |
+-----+
| Remote Management Address http://b.oray.com |
+-----+
root@raspberrypi:/home/pi# phddns stop ←
Sure you want to stop Phtunnel Service ? (y/n)
y
Phtunnel Service Already stopped !
root@raspberrypi:/home/pi# phddns reset ←
Sure you want to reset it? (y/n)
y
reset success !
root@raspberrypi:/home/pi# phddns restart ←
restart phtunnel service success !

```

图 5-10 花生壳功能操作



图 5-11 登录花生壳

④ 开通内网穿透体验版后的界面如图 5-14 所示,单击页面上的“增加映射”按钮,根据页面提示填写映射所需的信息。

下面以映射树莓派 HTTP 服务为例进行说明,如图 5-15 所示,自定义应用名称,自行选择应用图标,映射类型选择 HTTP,外网域名是指用户申请用作外网访问的域名,外网端口选择默认的 80,内网主机为树莓派的内网 IP 地址,内网端口是映射类型对应的 80 端口,带宽默认为 1Mb/s(可购买额外的映射带宽)。确认映射内容无误后,单击“确定”按钮。



图 5-12 激活方式



图 5-13 开通内网穿透功能



图 5-14 增加映射

⑤ 添加映射完成后,生成格式为“域名+端口号”的外网访问地址,如图 5-16 所示。完成上述步骤后,在浏览器地址栏输入域名就可以实现外网访问树莓派了。



图 5-15 映射配置



图 5-16 完成内网穿透

5.4 Web 软件开发

Web 页面呈现给用户的画面称为前端,开发前端界面需要掌握 HTML、CSS、JavaScript 等基础编程语言,其中 HTML 定义网页的内容,CSS 规定网页的布局与样式,JavaScript 对网页逻辑行为进行编程。在前端界面中单击按钮、新增/修改信息等操作需要后台解析,这些对用户是不可见的,称为后端。后端开发一般涉及数据库设计与 PHP 语言编程。前端、后端以及数据库设计完成后,需要将源代码部署到服务器才能正常运行和访问。

本案例中设计的 Web 软件包括实时监测图表、远程控制、树莓派运行状态 3 个功能模块,页面布局如图 5-17 所示。要实现这些功能需要在树莓派/var/www/html 目录下创建 html、css、js 等必要的文件,文件目录如图 5-18 所示。此外,还需要创建 Python 脚本文件完成 Web 软件与树莓派之间的交互。所有文件的目录结构及其实现功能如图 5-19 所示。

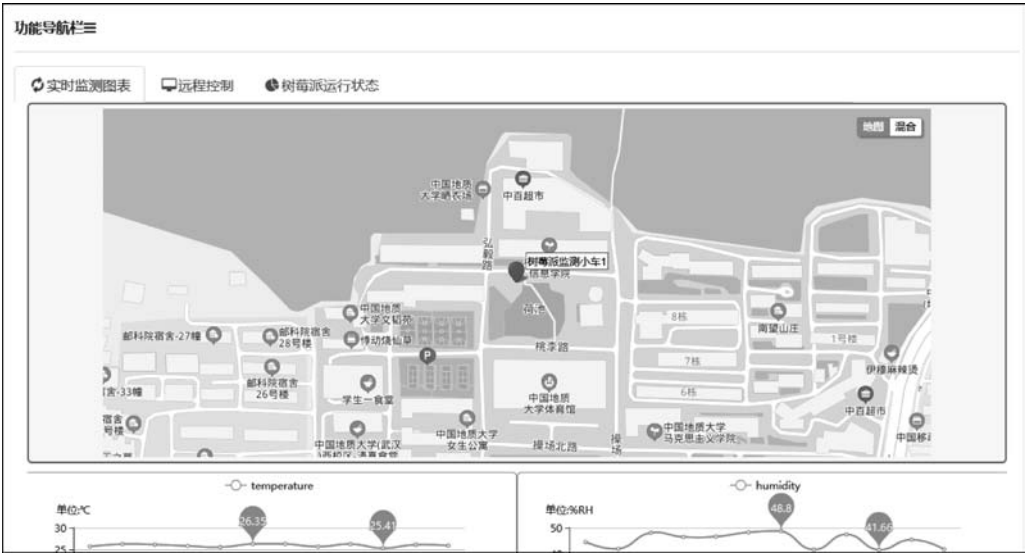


图 5-17 页面布局

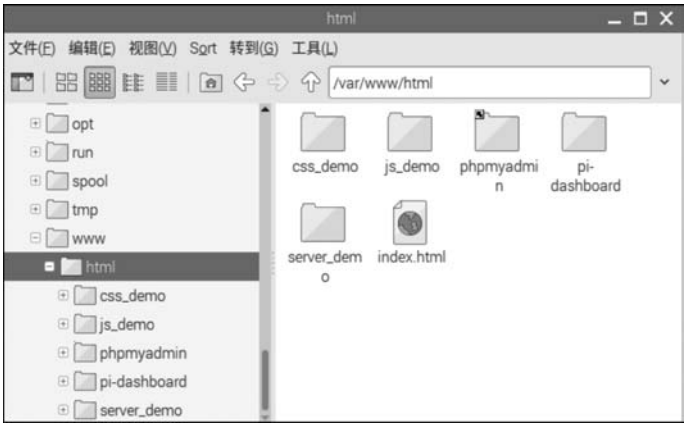
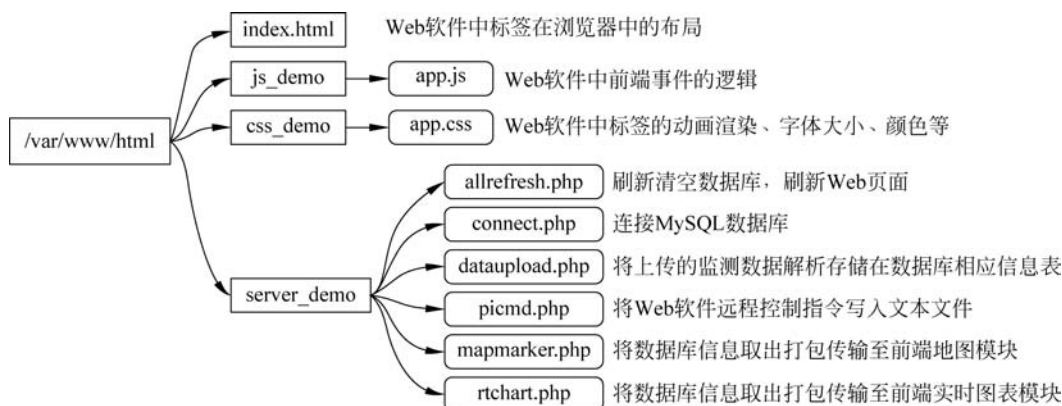
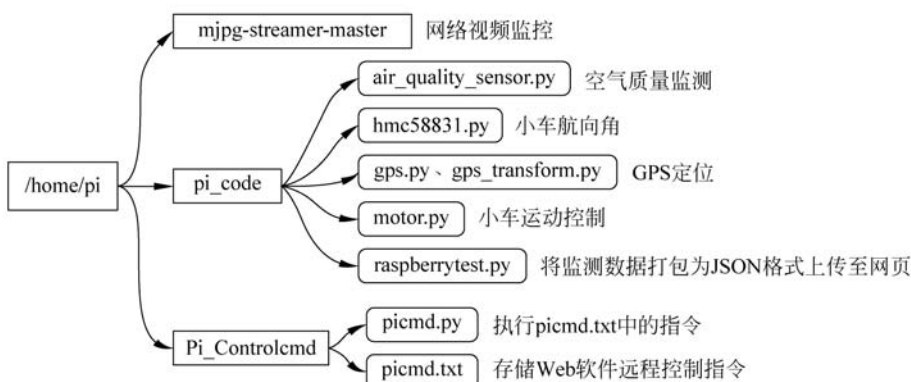


图 5-18 文件目录

图 5-19(a)中的所有文件都在计算机端的 VS Code 环境下进行开发设计,如图 5-20 所示,然后再上传到树莓派/var/www/html 下对应的目录中;图 5-19(b)中 Python 脚本文件直接在 Python3 IDLE 中编写,并保存在相应的目录中。如果读者自行修改了 Python 脚本存放的路径,则需要在后面的代码中做相应的调整。



(a) Web软件



(b) Python脚本

图 5-19 文件结构与功能

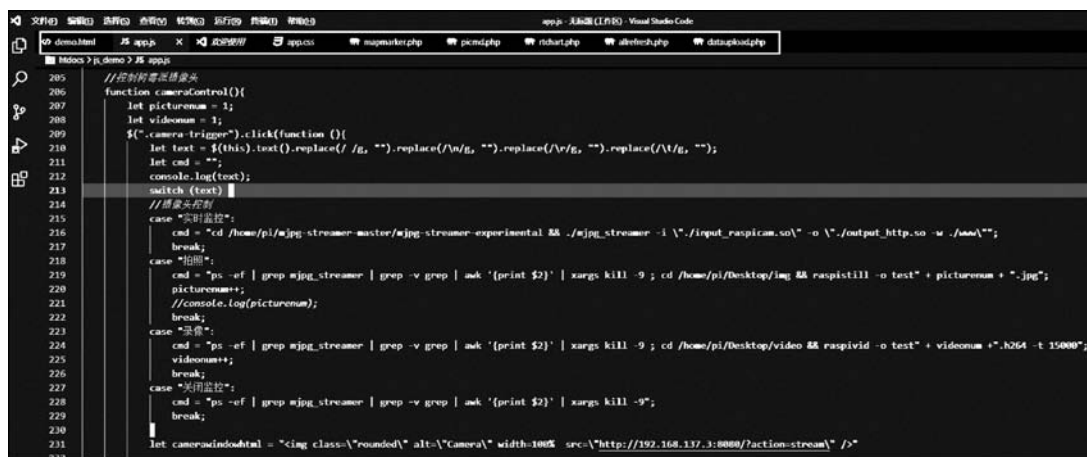


图 5-20 VS Code 开发环境

5.4.1 数据库设计

Web 软件使用 MySQL 数据库存储、查看以及更新监测节点上传的数据,需要在数据库中创建信息表,具体过程如下:

(1) 在终端输入 **sudo mysql -u root -p** 后按 Enter 键确认,会要求输入密码。前面 MySQL 安装过程中已将用户名 root 的密码设置为“12345678”,输入该密码登录成功后将会看到如图 5-21 所示的界面。

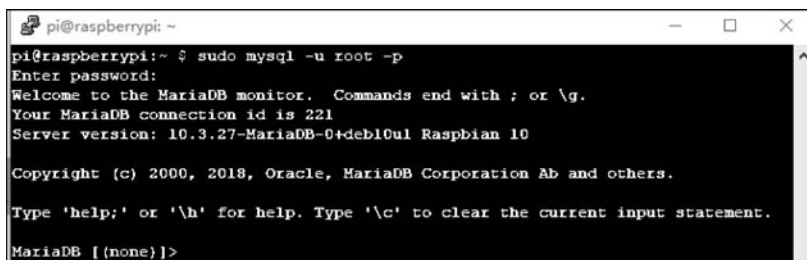


图 5-21 数据库登录成功

(2) 使用 create database 语句创建数据库,命令格式为 **create database 数据库名** (其他选项);,如图 5-22 所示,创建了一个名为 demo 的数据库。创建成功时,数据库终端会输出“Query OK, 1 row affected”。在浏览器地址栏输入“http://树莓派 IP 地址/phpmyadmin”即可看到创建好的数据库,如图 5-23 所示。



图 5-22 创建数据库



图 5-23 phpMyAdmin 显示创建的数据库

(3) 使用 create table 语句创建数据信息表,命令格式为 **create table 表名称** (列声明);,需要说明的是,在创建信息表前要先执行 **use 数据库名**,然后再依次输入以下内容创建名为 rt_node 的信息表。

```

create table rt_node(
    id int(10) unsigned not null auto_increment primary key,
    node_id int(11) not null,
    lng decimal(9,6) not null,
    lat decimal(8,6) not null,
    temperature float(7,2) not null,
    humidity float(7,2) not null,
    co2 float(7,2) not null,
    tvoc float(7,2) not null,
    ch2o float(7,2) not null,
    pm25 float(7,2) not null,
    pm10 float(7,2) not null,
    heading_angle float(7,2) not null,
    time_node timestamp not null
);

```

在数据库终端执行以上命令的结果如图 5-24 所示,创建的信息表用来存储智能小车上传的各种数据信息,说明如表 5-1 所示。

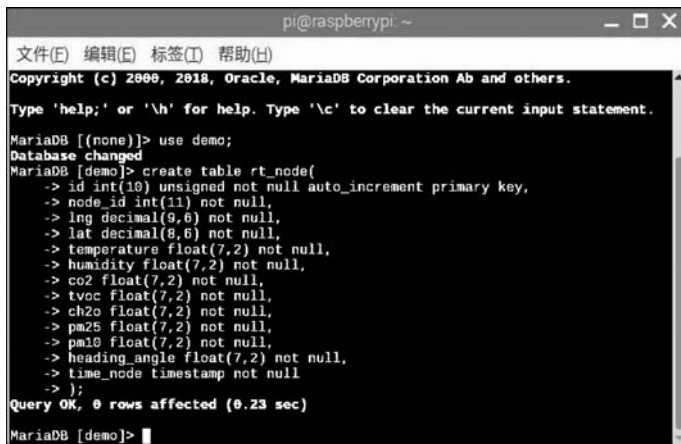


图 5-24 创建信息表

表 5-1 实时数据信息表

字段名称	字段定义	数据类型及长度	特殊要求
id	数据信息编号	int(10)	AUTO_INCREMENT
node_id	监测节点编号	int(10)	NOT NULL
lng	节点经度信息	decimal(9,6)	NOT NULL
lat	节点纬度信息	decimal(8,6)	NOT NULL
temperature	环境温度	float(7,2)	NOT NULL
humidity	环境湿度	float(7,2)	NOT NULL
co2	二氧化碳含量	float(7,2)	NOT NULL

续表

字段名称	字段定义	数据类型及长度	特殊要求
tvoc	挥发性有机物信息	float(7,2)	NOT NULL
ch2o	甲醛含量	float(7,2)	NOT NULL
pm25	空气质量指数 PM2.5	float(7,2)	NOT NULL
pm10	空气质量指数 PM10	float(7,2)	NOT NULL
heading_angle	节点航向角(与正北方的角度)	float(7,2)	NOT NULL
time_node	信息写入时间	timestamp	NOT NULL

5.4.2 地图显示

在 Web 软件中插入百度地图,用气泡在地图中标注出监测节点的位置,当用户单击气泡时会弹出窗口并显示监测节点当前获取的各种数据信息。使用百度地图功能的步骤如下:

(1) 在百度地图开发者平台(<http://lbsyun.baidu.com/index.php>)申请应用,首先登录百度账户(没有百度账户的用户需要先注册再登录),然后单击页面右上角的“控制台”菜单项,进入如图 5-25(a)所示的界面,单击“创建应用”按钮,按照所需功能和要求依次填写相应内容,如图 5-25(b)所示,其中应用名称可以随意填写,应用类型选择浏览器端,默认全选启动服务,Referer 白名单填写网站的 IP 地址(即树莓派 IP 地址),最后单击“提交”按钮。

(2) 应用创建成功后,就可以看到如图 5-26 所示的应用列表,单击“应用配置”栏中的“设置”按钮,可以对启用服务进行选择和修改,也可以修改和增删“Referer 白名单”(只有在白名单上的域名和 IP 地址才能正常使用百度地图),如图 5-27 所示,其中的“应用 AK”是在网站中嵌入百度地图功能最关键的密钥信息。

(3) 为了统一存放与管理树莓派脚本文件,在树莓派/home/pi 目录下建立 pi_code 子目录,在其中创建脚本 raspberrystest.py 并输入以下代码,将智能小车采集的监测数据打包为 JSON 数据格式,通过 HTTP 协议传输至服务器解析。

```
import requests
import json
import time
import random
import gps                                # 从监测节点读取 GPS 信息
import hmc5883l                          # 从监测节点读取方向角
import air_quality_senor                 # 从监测节点读取空气指标参数

url = 'http://192.168.137.3/server_demo/dataupload.php'
headers = {'content-type': 'application/json', 'charset': 'utf-8'}
while True:
    '''读取智能小车传感器采集的数据,具体内容参见本书第 3 章的相关章节,传感器脚本
```



(a) 百度地图应用页面

< 返回 创建应用

应用名称: RaspberryPiWeb ☒ 输入正确

应用类型: 浏览器端

启用服务:

☒ Javascript API	☒ 地点检索	☒ 普通IP定位
☒ 静态图	☒ 全景静态图	☒ 坐标转换
☒ 鹰眼轨迹	☒ 全景URL API	☒ 地理编码
☒ 逆地理编码		

参考文档: Javascript API 地点检索 普通IP定位 正地理编码
逆地理编码 坐标转换

Referer白名单: *192.168.137.3* ☒ 输入正确

只有该白名单中的网站才能成功发起调用

格式: *.mysite.com*,*myapp.com* 多个域名之间请用英文半角逗号隔开

如果不想对任何域名做限制, 设置为英文半角星号*(谨慎使用, 前端代码容易泄露AK, 被其他网站非法调用, 上线前可以用作Debug, 线上正式ak请设置合理的Referer白名单)新申请的Mobile类型的ak与新申请的Browser类型的ak不再支持云存储接口的访问, 如要使用云存储, 请申请server类型ak

提交

(b) 创建应用

图 5-25 申请百度地图功能



图 5-26 应用列表



图 5-27 设置应用

```
文件 gps.py、hmc5883l.py 以及 air_quality_senor.py 也都存放于 pi_code 目录下'''
hmc = hmc5883l.HMC5883()
sto = air_quality_senor.Multisensor()
angle = hmc.read_HMC5883_data()
sto_co2,sto_tvoc,sto_ch20,sto_pm25,sto_pm10,sto_humidity,sto_temp =
    sto.read_sensor_data()
gps_lng,lng_EW,gps_lat,lat_NS = gps.GPS()

body = {"node_id" : "1",
        "lng" : gps_lng,
        "lat" : gps_lat,
        "temperature" : sto_temp,
        "humidity" : sto_humidity,
        "co2" : sto_co2,
        "tvoc" : sto_tvoc,
        "ch2o" : sto_ch20,
        "pm25" : sto_pm25,
        "pm10" : sto_pm10,
        "heading_angle" : angle
    }

response = requests.post(url, data = json.dumps(body), headers = headers)
```

(4) 在树莓派/var/www/html/server_demo目录下创建 dataupload.php 文件,内容如下:

```
<?php
//连接数据库
require ('connect.php');
$post_array = file_get_contents("php://input");

//查询数据库
mysqli_select_db( $ conn, 'demo' );
//解析 JSON, 获取对应的变量值
$obj = json_decode( $ post_array, TRUE);
//得到 Json_list 数组长度
$num = count( $ obj);

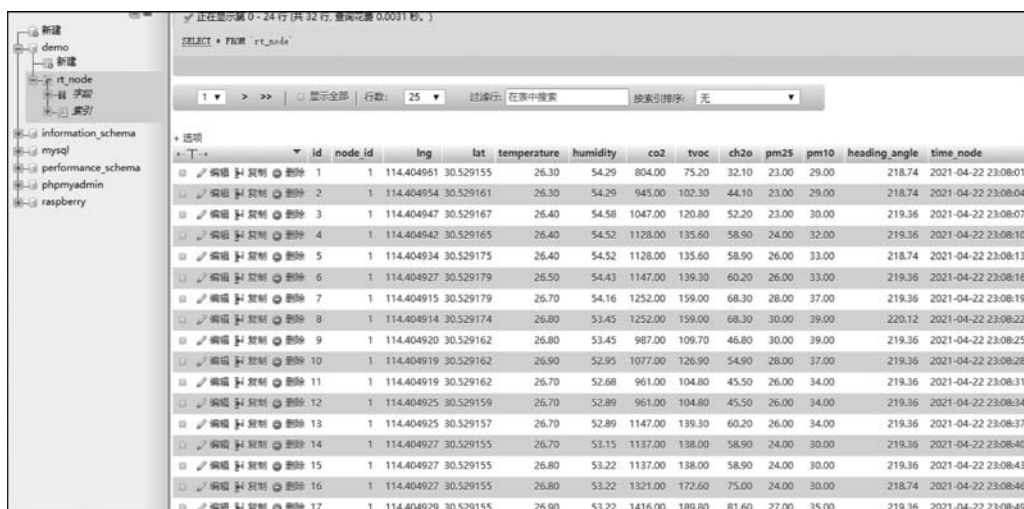
$k = nKey();
$v = nValue();
mysqli_query( $ conn, 'INSERT INTO rt_node ('. $ k. ') VALUES ('. $ v. ')');
//将 JSON 数据的键拼接成字符串
function nKey(){
    global $ obj, $ num;
    $ k = '';
    for( $ i = 0; $ i < $ num; $ i++){
        if( $ i == $ num - 1){
            $ k = $ k. key( $ obj);
            next( $ obj);
        }elseif( $ i < $ num - 1){
            $ k = $ k. key( $ obj);
            $ k = $ k. ',';
            next( $ obj);
        }
    }
    return $ k;
}
//将 JSON 数据的值拼接成字符串
function nValue(){
    global $ obj, $ num;
    $ v = '';
    reset( $ obj);
    for( $ i = 0; $ i < $ num; $ i++){
        if( $ i == $ num - 1){
            $ v = $ v. "'";
            $ v = $ v. current( $ obj);
            $ v = $ v. "'";
            next( $ obj);
        }elseif( $ i < $ num - 1){
            $ v = $ v. "'";
```

```

        $v = $v.current( $obj);
        $v = $v.'", ' ';
        next( $obj);
    }
}
return $v;
}
// 关闭连接
mysqli_close( $ conn);
?>

```

以上代码用于将智能小车上传的监测数据解析存储至数据库信息表中,可通过 phpMyAdmin 查看上传的数据信息,如图 5-28 所示。



id	node_id	lng	lat	temperature	humidity	co2	tvoc	ch2o	pm25	pm10	heading_angle	time_node
1	1	114.404951	30.529155	26.30	54.29	804.00	75.20	32.10	23.00	29.00	218.74	2021-04-22 23:08:01
2	1	114.404954	30.529161	26.30	54.29	945.00	102.30	44.10	23.00	29.00	218.74	2021-04-22 23:08:04
3	1	114.404947	30.529167	26.40	54.58	1047.00	120.80	52.20	23.00	30.00	219.36	2021-04-22 23:08:07
4	1	114.404942	30.529165	26.40	54.52	1128.00	135.60	58.90	24.00	32.00	219.36	2021-04-22 23:08:10
5	1	114.404934	30.529175	26.40	54.52	1128.00	135.60	58.90	26.00	33.00	218.74	2021-04-22 23:08:13
6	1	114.404927	30.529179	26.50	54.43	1147.00	139.30	60.20	26.00	33.00	219.36	2021-04-22 23:08:16
7	1	114.404915	30.529179	26.70	54.16	1252.00	159.00	68.30	28.00	37.00	219.36	2021-04-22 23:08:19
8	1	114.404914	30.529174	26.80	53.45	1252.00	159.00	68.30	30.00	39.00	220.12	2021-04-22 23:08:22
9	1	114.404920	30.529162	26.80	53.45	987.00	109.70	46.80	30.00	39.00	219.36	2021-04-22 23:08:25
10	1	114.404919	30.529162	26.90	52.95	1077.00	126.90	54.90	28.00	37.00	219.36	2021-04-22 23:08:28
11	1	114.404919	30.529162	26.70	52.68	961.00	104.80	45.50	26.00	34.00	219.36	2021-04-22 23:08:31
12	1	114.404925	30.529159	26.70	52.89	961.00	104.80	45.50	26.00	34.00	219.36	2021-04-22 23:08:34
13	1	114.404925	30.529157	26.70	52.89	1147.00	139.30	60.20	26.00	34.00	219.36	2021-04-22 23:08:37
14	1	114.404927	30.529155	26.70	53.15	1137.00	138.00	58.90	24.00	30.00	219.36	2021-04-22 23:08:40
15	1	114.404927	30.529155	26.80	53.22	1137.00	138.00	58.90	24.00	30.00	219.36	2021-04-22 23:08:43
16	1	114.404927	30.529155	26.80	53.22	1321.00	172.60	75.00	24.00	30.00	218.74	2021-04-22 23:08:46
17	1	114.404929	30.529155	26.90	53.22	1416.00	189.80	81.60	27.00	35.00	219.36	2021-04-22 23:08:49

图 5-28 查看监测数据

(5) 在树莓派/var/www/html 目录下的 index. html 文件中(该文件较长,读者可以查看本书配套资源中的完整代码)添加密钥代码<script type="text/javascript" src="//api.map.baidu.com/api?v=2.0&ak=密钥"></script>。

百度地图中数据显示的流程是:先由 Web 前端向后端发起 Ajax 请求,获取数据库中实时数据信息表的监测节点数据;然后,后端将数据以 JSON 字符串的格式打包传输至前端,前端对数据做解析渲染后在百度地图中显示,同时给气泡标注添加单击事件;最后,设置定时器以固定的时间间隔来局部刷新地图界面,实现监测节点信息在地图中的实时显示。

在 js_demo/app.js 文件中对应的前端部分关键代码如下:

```

//百度地图 API 功能
let map = new BMap.Map("allmap");
//创建 Map 实例

```

```

let point = new BMap.Point(114.404252, 30.52761);
map.centerAndZoom(point, 17); //初始化地图,设置中心点坐标和地图级别
let opts = {
width: 100, //信息窗口宽度
height: 300, //信息窗口高度
title: ">>>节点信息如下:", //信息窗口标题
enableMessage:true //设置允许信息窗发送短息
};
map.enableScrollWheelZoom(true); //开启鼠标滚轮缩放
//添加地图类型控件
map.addControl(new BMap.MapTypeControl({
mapTypes:[
BMAP_NORMAL_MAP,
BMAP_HYBRID_MAP
]
})));
map.setCurrentCity("武汉"); // 设置地图显示的城市,此项必须设置

```

对应的后端代码保存在 server_demo/mapmarker.php 文件中,内容如下:

```

<?php
header("content-type:application/json");
//连接数据库
require ('connect.php');
//查询数据库
mysqli_select_db( $ conn, 'demo' );

//当数据库数据超过一定值清空
$ rowsnum = mysqli_fetch_assoc(mysqli_query( $ conn,"select count( * ) as num from rt_node"));
if( $ rowsnum['num']>= 5000){
    mysqli_query( $ conn, 'truncate table rt_node');
}

//获取节点数据
$ sql = 'SELECT node_id,lng,lat,temperature,humidity,co2,tvoc,ch2o,pm25,pm10,heading_angle
FROM rt_node where node_id= 1 ORDER BY id DESC LIMIT 0,1';
$ result = mysqli_query( $ conn, $ sql);
$ results = array();
$ row = mysqli_fetch_assoc( $ result);
$ results[0] = $ row;
//将数组转成 JSON 格式
$ json = json_encode( $ results);
//关闭连接
mysqli_close( $ conn);
echo $ json;
?>

```

地图显示功能最终效果如图 5-29 所示,其中图 5-29(a)标识监测节点在地图中的位置,图 5-29(b)是单击气泡时在弹出窗口中显示节点的监测数据。

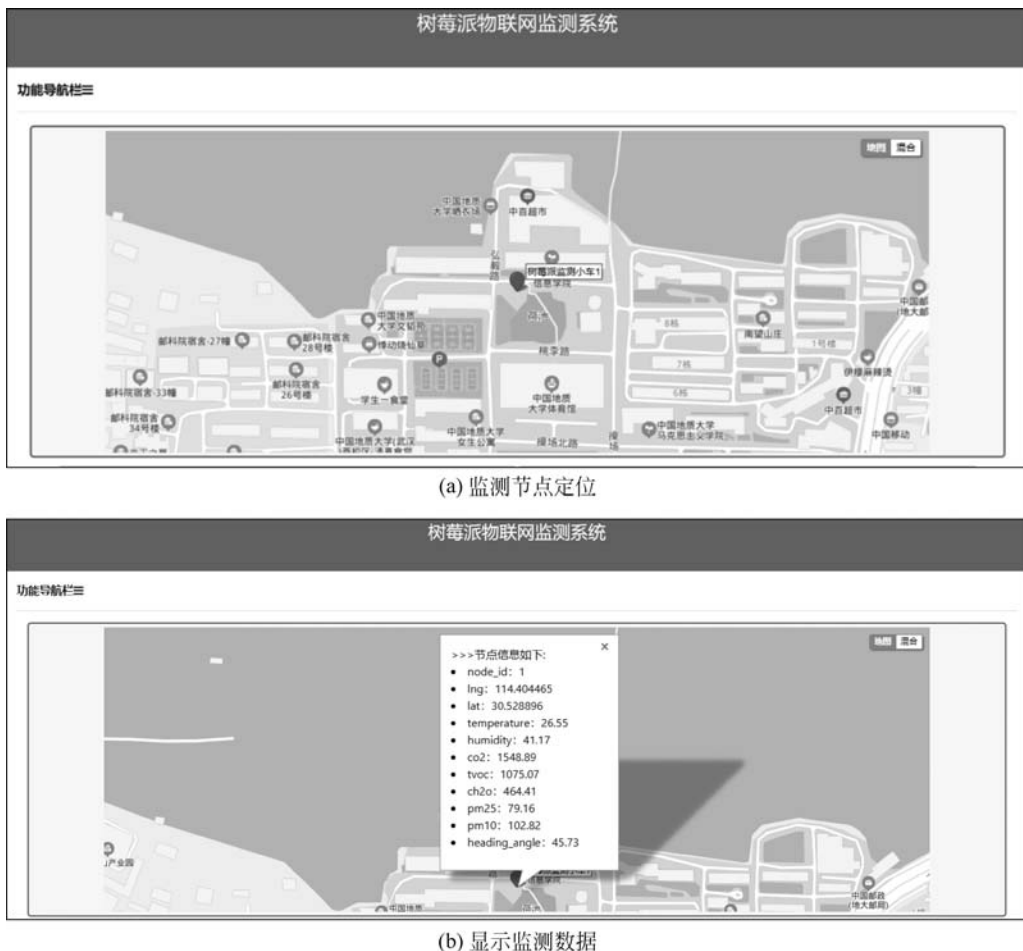


图 5-29 地图显示效果

5.4.3 监测数据图表显示

通过在 Web 软件中加入监测数据实时图表显示的功能,可以直观地查看监测节点采集的数据及其变化趋势。ECharts 是百度旗下的一个基于 JavaScript 的开源数据可视化工具,能够在计算机端和移动设备上流畅运行,可以提供直观、生动、可交互以及个性化定制的数据可视化图表(读者可以访问 <https://echarts.apache.org/zh/index.html> 查看使用教程和示例)。使用 ECharts.js 可以方便地将折线图嵌入到 Web 软件中。

实现监测数据折线图显示,需要在 index.html 文件< head>标签中添加< script src="https://cdn.bootcss.com/echarts/4.2.1-rc1/echarts.min.js"></script>来引入 ECharts.js。

实时数据图表显示的流程如下：前端将节点编号与传感器数据类型通过 Ajax 请求传输至后端服务器；后端接收数据后与数据库中实时数据信息表的最新监测数据进行匹配，并将监测数据取出打包为 JSON 字符串格式回传；前端根据传感器数据种类在界面创建相应数量的 div 图表容器，将回传数据解析后分别写入容器中进行图表渲染。为了实时显示数据图表，设置了 setInterval 定时器，以 500ms 的时间间隔与后端建立长连接，不断获取最新的节点数据。此外，为了方便用户分析数据，在图表中标识出了数据的极值，图表更新过程中的最大值和最小值会以气泡的形式渲染。

图表显示功能在 js_demo/app.js 文件中对应的前端关键代码如下：

```
let clear = setInterval(function(){
    $.ajax({
        //请求方式
        type:"GET",
        //文件位置
        url:"server_demo/rtchart.php?q=" + nodeid + "&v=" + ntype,
        //返回数据格式为 JSON,也可以是其他格式
        dataType: "json",
        //请求成功后要执行的函数
        success:function(result){
            if(result.length != 0){
                $.each(result, function(i,item){
                    rt = item["time_node"].split(" ")[1];
                    if(rt != xdata[11]){
                        xdata.shift();
                        ydata.shift();
                        ydata.push(item[ntype]);
                        xdata.push(rt);
                    }
                });
            }
        });
    myChart.setOption({
        xAxis: {data: xdata},
        series: [{data: ydata}]
    });
},500);
```

对应的后端代码保存在 server_demo/server/rtchart.php 文件中，内容如下：

```
<?php
header("content-type:application/json");
$q = isset( $_GET["q"] ) ? intval( $_GET["q"] ) : '';
$v = isset( $_GET["v"] ) ? strval( $_GET["v"] ) : '';
//连接数据库
```

```

require ('connect.php');
//查询数据库
mysqli_select_db( $ conn, 'demo' );
//当数据库数据超过一定值时清空
$ rownum = mysqli_fetch_assoc(mysqli_query( $ conn,"select count( * ) as num from rt_node"));
if( $ rownum['num'] >= 5000){
    mysqli_query( $ conn, 'truncate table rt_node');
}
//节点数据类型
$ type = ["temperature", "humidity", "co2", "tvoc", "ch2o", "pm25", "pm10", "heading_angle"];
$ t_num = count( $ type);
for( $ i = 0; $ i < $ t_num; $ i++){
    if( $ v == $ type[ $ i]){
        $ result = mysqli_query( $ conn, 'SELECT '. $ type[ $ i].', time_node FROM rt_node WHERE
node_id = "'. $ q. "''. 'ORDER BY id DESC LIMIT 0,1');
        $ results = array();
        while ( $ row = mysqli_fetch_assoc( $ result)) {
            $ results[] = $ row;
        }
        //将数组转成 JSON 格式
        $ data = json_encode( $ results);
        //关闭连接
        mysqli_close( $ conn);
        echo $ data;
    }
}
?>

```

监测数据图表功能最终效果如图 5-30 所示,图中通过折线图实时显示了 7 种环境指标参数以及树莓派小车运动过程中的航向角。

5.4.4 节点远程控制

Web 软件需要具有对监测节点进行远程控制的功能,通过网页上的按钮下达控制指令,例如,开启/关闭摄像头、拍照或录像以及控制智能小车运动。在界面中设置 img 标签来容纳视频监控部分,此外,小车运动控制按钮设置了两种单击事件:按下按钮时触发 mousedown 事件开启节点移动,抬起按钮时触发 mouseup 事件停止节点移动。

监测节点远程控制功能在 js_demo/app.js 文件中对应的前端关键代码如下:

```

//控制树莓派摄像头
function cameraControl(){
    let picturenum = 1;
    let videonum = 1;
    $ (".camera-trigger").click(function (){

```

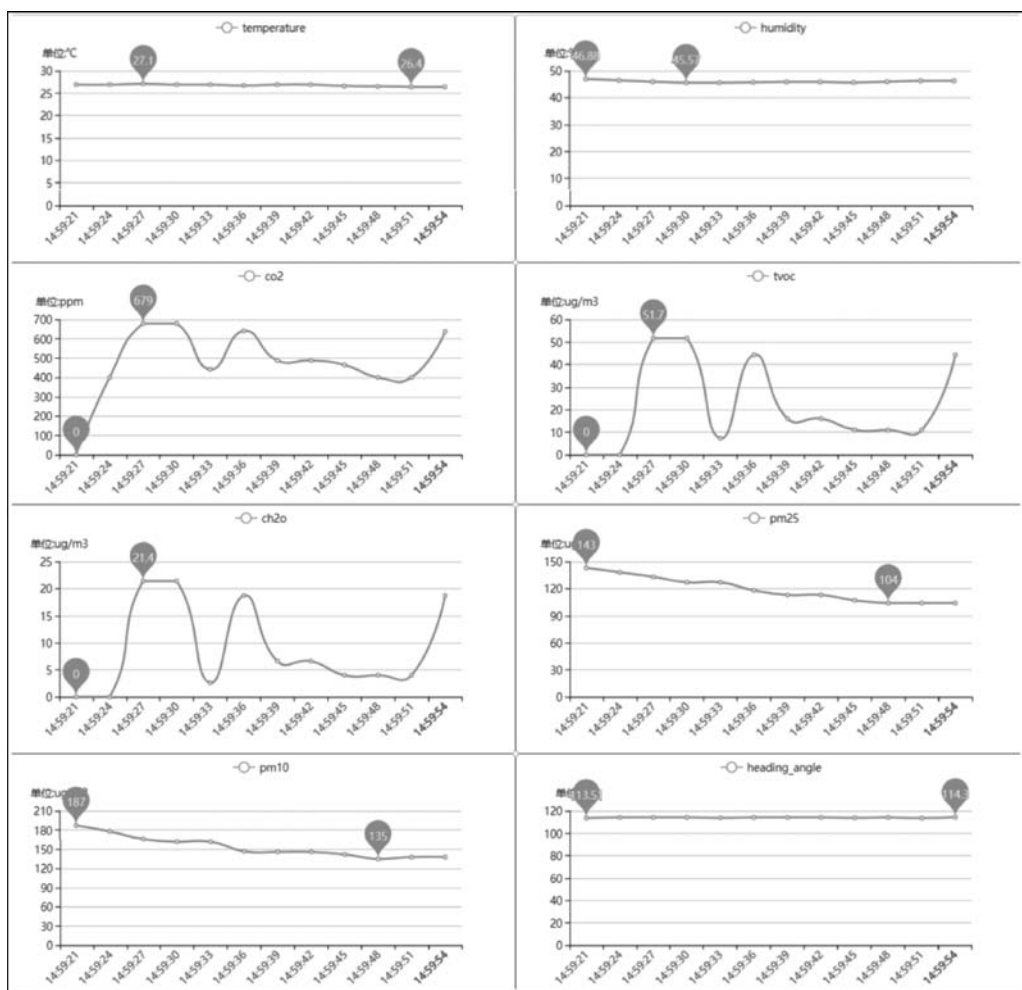


图 5-30 监测数据图表显示

```

let text = $(this).text().replace(/ /g, "").replace(/\n/g, "").replace(/\r/g, "").
replace(/\t/g, "");
let cmd = "";
console.log(text);
switch (text) {
case "实时监控": # 运行 MJPG-streamer 网络视频监控
    cmd = "cd /home/pi/mjpg-streamer-master/mjpg-streamer-experimental && ./mjpg-
streamer -i \"/input_raspicam.so\" -o \"/output_http.so -w ./www\"";
    break;
case "拍照": # 以 test + 序号.jpg 的形式命名图片
    cmd = "ps -ef | grep mjpg_streamer | grep -v grep | awk '{print $2}' | xargs kill -
9 ; cd /home/pi/Desktop/img && raspistill -o test" + picturenum + ".jpg";

```

```

        picturenum++;
        break;
    case "录像": # 以 test + 序号.h264 的形式命名视频
        cmd = "ps -ef | grep mjpg_streamer | grep -v grep | awk '{print $2}' | xargs kill -9";
        cd /home/pi/Desktop/video && raspivid -o test" + videonum + ".h264 -t 15000";
        videonum++;
        break;
    case "关闭监控": # 终止 MJPG-streamer 进程,退出视频监控
        cmd = "ps -ef | grep mjpg_streamer | grep -v grep | awk '{print $2}' | xargs kill -9";
        break;}

    let camerawindowhtml = "<img class = \"rounded\" alt = \"Camera\" width = 100 % src = \"http://192.168.137.3:8080/?action=stream\" />"
    if (confirm("确定要执行该命令吗?")) {
        $.ajax({
            type: "POST",
            url: "server_demo/picmd.php",
            data: {
                action: "set-linux-cmd",
                cmd: cmd
            },
            success: function (result) {
                if(cmd == "cd /home/pi/mjpg-streamer-master/mjpg-streamer-experimental && ./mjpg_streamer -i \"/home/pi/input_raspicam.so\" -o \"/home/pi/output_http.so -w ./www\"") {
                    $ ("."camerawindow").html(camerawindowhtml);
                }
            }
        });
    }

//控制智能小车运动
function directionControl(){
    $ ("."direction-trigger").mousedown(function(){
        let text = $(this).attr("name");
        let cmd = "";
        switch (text) {
            case "前进": # 通过 python3 motor.py 命令行中的参数来指定小车的运行状态
                cmd = "cd /home/pi/pi_code && python3 motor.py 1";
                break;
            case "向左":
                cmd = "cd /home/pi/pi_code && python3 motor.py 4";
                break;
            case "向右":
                cmd = "cd /home/pi/pi_code && python3 motor.py 3";
                break;
            case "后退":
                cmd = "cd /home/pi/pi_code && python3 motor.py 2";
                break;}
    });
}

```

```

console.log("鼠标按下");
$.ajax({
    type: "POST",
    url: "server_demo/picmd.php",
    data: {
        action: "set - linux - cmd",
        cmd: cmd},
    success: function (result) {
    }
});
});
$($(".direction-trigger").mouseup(function(){
    console.log("鼠标松开");
    $.ajax({
        type: "POST",
        url: "server_demo/picmd.php",
        data: {
            action: "set - linux - cmd",
            cmd: "cd /home/pi/pi_code && python3 motor.py 5"},      # 小车停止运动
        success: function (result) {
            console.log(result)
        }
    });
});
}

```

通过以上代码对树莓派摄像头以及智能小车进行控制,其中实时视频监控采用 MJPG-streamer(参见 4.1.3 节)实现,小车运动控制的脚本文件 motor.py 存放在 pi_code 目录下。

每个控制指令通过前端 Ajax 传输至后端写入 picmd.txt 文件,由控制程序读取文件内容来执行指令。实现该功能的后端代码保存在 server_demo/server/picmd.php 文件中,内容如下:

```

<?php
$action = $_POST["action"];
$cmd = $_POST["cmd"];
$myfile = fopen("/home/pi/Pi_Controlcmd/picmd.txt", "w") or die("unable to open file!");
fwrite($myfile, $m_nodeid. $cmd);
$str = file_get_contents("/home/pi/Pi_Controlcmd/picmd.txt");
echo($str);
?>

```

在 Pi_Controlcmd 目录下新建名为 picmd.py 的控制程序负责读取 picmd.txt 文件中的指令并逐条执行,代码如下:

```
import os

while True:
    file1 = open("/home/pi/Pi_Controlcmd/picmd.txt", "r")
    picmd = file1.read()
    file1.close()
    if(picmd):
        file = open("/home/pi/Pi_Controlcmd/picmd.txt", "w")
        file.write("")          # 清空 txt 里的指令
        file.close()
        os.system(picmd + '&')   # 后台运行指令
```

监测节点远程控制功能最终效果如图 5-31 所示。单击“实时监控”将开启网络视频监控功能,在 img 标签内显示实时画面;单击“拍照”将启动摄像头拍照并将采集的图像以 .jpg 格式存储在树莓派桌面 img 目录中;单击“录像”摄像头将录制一段 15s 的视频,以 .h264 格式存储在桌面 video 目录中;单击右侧的方向键将控制小车运动。



图 5-31 智能小车远程操控

5.4.5 树莓派运行状态监控

5.2 节介绍了读取树莓派状态的方法,可以将其集成到 Web 监测软件中来。在 js_demo/app.js 文件中增加 iframe 浮动框,链接树莓派运行状态网址即可,代码如下:

```
//树莓派仪表盘
function pidashboard(){
```

```
let pidashboardhtml = "<br><iframe src = \"http://192.168.137.3/pi - dashboard/\n" width = \"100 %\" height = \"700px\"></iframe>";
$("#pidashboard").html(pidashboardhtml);
}
```

该功能效果如图 5-32 所示,界面左侧显示树莓派 IP 地址和连接时间,中部显示树莓派 CPU 的工作温度、工作频率等,右侧显示树莓派运行时内存的消耗及 SD 卡剩余空间等。树莓派的状态参数会动态变化,方便用户了解树莓派当前的工作状态。

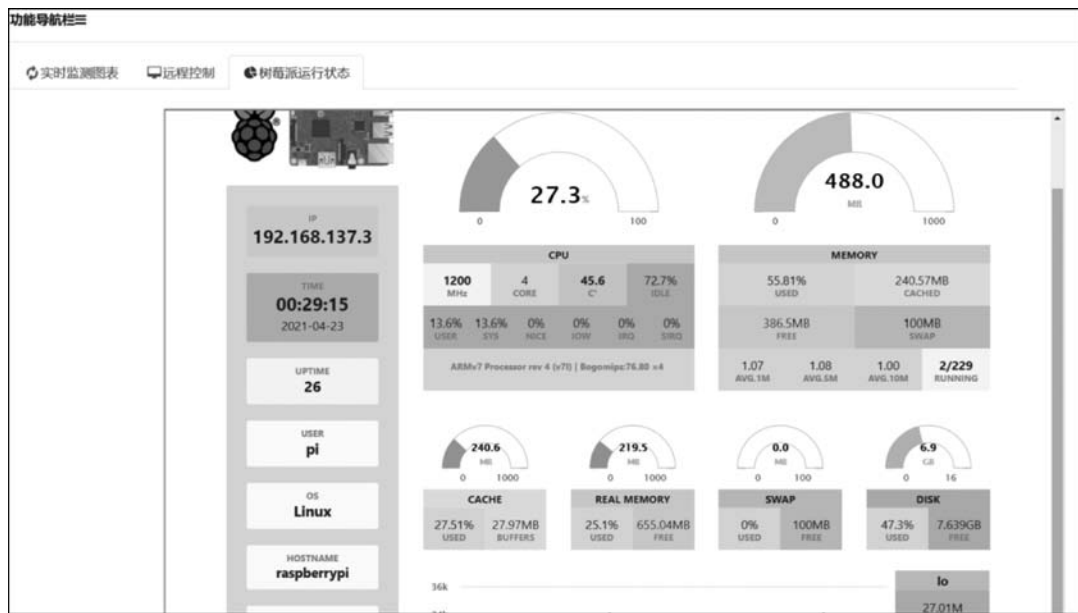


图 5-32 树莓派状态监测

5.4.6 4G 网络远程访问

在前面的介绍中,树莓派是连接笔记本电脑的移动热点,其 IP 地址被固定设置为 192.168.137.3。利用 5.3 节介绍的内网穿透,我们将外网域名与该 IP 地址进行了绑定,输入外网域名即可实现外网访问树莓派监测系统。本节将介绍通过 4G 无线上网模块来实现树莓派的外网远程访问。

接有 4G 无线上网模块的树莓派小车如图 5-33 所示,在这种情况下,树莓派将连接到 4G 上网模块的移动热点。由于该 4G 网卡默认网关是 192.168.0.1,要实现 4G 网络远程访问树莓派,只需修改树莓派的静态 IP 地址,由之前的 192.168.137.3 修改为 192.168.0.1 网关分配的 IP 地址,如 192.168.0.3,并将新的 IP 地址参照 5.3 节的方法重新映射到外网域名。此外,还需要将前面介绍的所有代码中涉及树莓派 IP 地址的内容都做相应的修改,才能正确实现 Web 监测软件的各项功能。

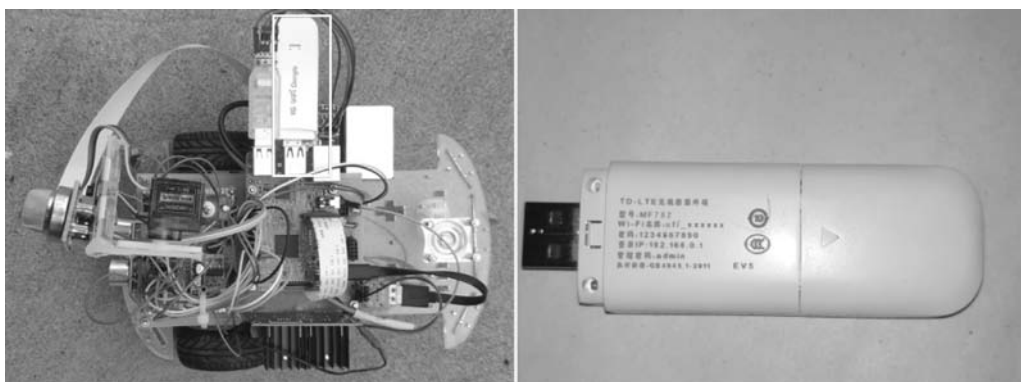


图 5-33 安装 4G 上网模块的树莓派小车