**《嵌入式综合实践》**

**设计报告**

目录

[**一、** **树莓派简介** 3](#_Toc436766043)

[**二、** **配置树莓派** 3](#_Toc436766044)

[1. 树莓派供电 3](#_Toc436766045)

[2. 手动对SD存储卡进行写操作（windows） 3](#_Toc436766046)

[3. 连接笔记本电脑显示器 3](#_Toc436766047)

[**三、** **硬件电路连接** 7](#_Toc436766048)

[**四、** **DHT11简介** 7](#_Toc436766049)

[**五、** **获取DHT11传感器温湿度** 8](#_Toc436766050)

[**六、** **安装本地MYSQL** 10](#_Toc436766051)

[**七、** **连接阿里云RDS数据库** 11](#_Toc436766052)

[**八、** **上传数据到传感云** 13](#_Toc436766053)

[**九、** **Cron 实现定时功能** 14](#_Toc436766054)

1. **树莓派简介**

Raspberry Pi(中文名为“树莓派”,简写为RPi，或者RasPi/RPi)是为学生计算机编程教育而设计，只有信用卡大小的卡片式电脑，其系统基于Linux。随着Windows 10 IoT的发布，我们也将可以用上运行Windows的树莓派。自问世以来，受众多计算机发烧友和创客的追捧，曾经一“派”难求。别看其外表“娇小”，内“心”却很强大，视频、音频等功能通通皆有，可谓是“麻雀虽小，五脏俱全”。

1. **配置树莓派**
2. 树莓派供电

树莓派的供电装置与智能手机的充电器是一样的。基本规格为DC 5V(直流电)，至少达到700mA的输出电流，树莓派2的输出电流应该更大，比如1.5A或2A。

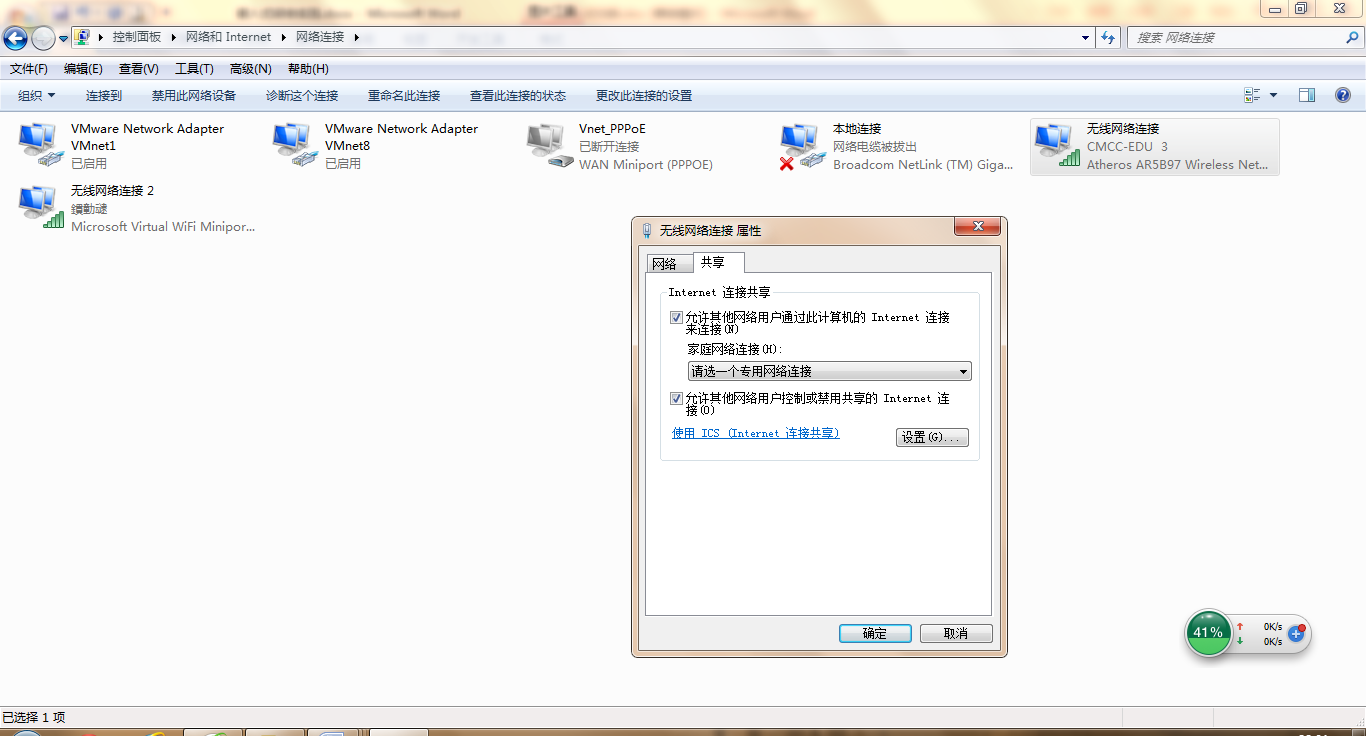
1. 手动对SD存储卡进行写操作（windows）

选择一张4GB以上的SD存储卡，SD卡插入笔记本电脑卡槽(或者需要一个读卡器)。下载官方发行的树莓派操作系统发行包(<http://www.raspberrypi.org/downloads>)，并解压到本地。用管理员权限打开Fedora ARM Installer（<http://bit.ly/ISLPc4>下载），将下载的镜像写入SD存储卡。如下图：

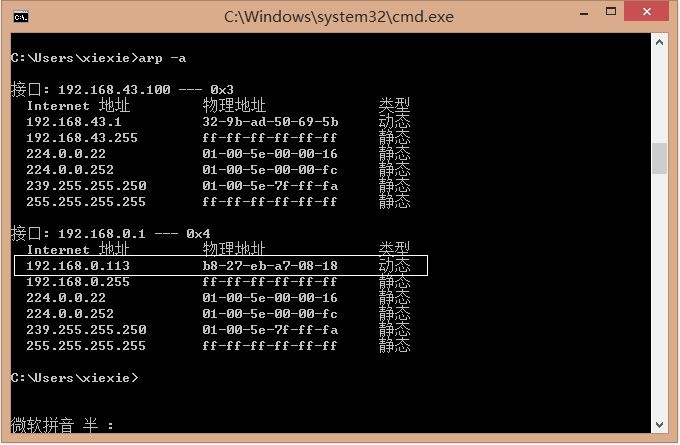


1. 连接笔记本电脑显示器

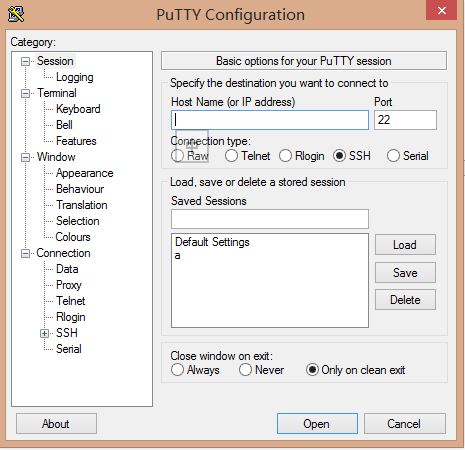
* 网络设置：SD卡插入树莓派的SD卡插槽，把树莓派和路由器用网线连接。打开电脑“网络和共享中心”，点击“WLAN（\*\*\*）”点击“属性”，点击“共享”，在“允许其他网络用户通过此计算机的Internet来连接”前打勾。



* 查找树莓派IP地址：打开“运行”（win+R键），输入cmd”，输入“arp -a”，在下图中的0x4接口下的动态地址就为树莓派的IP地址。



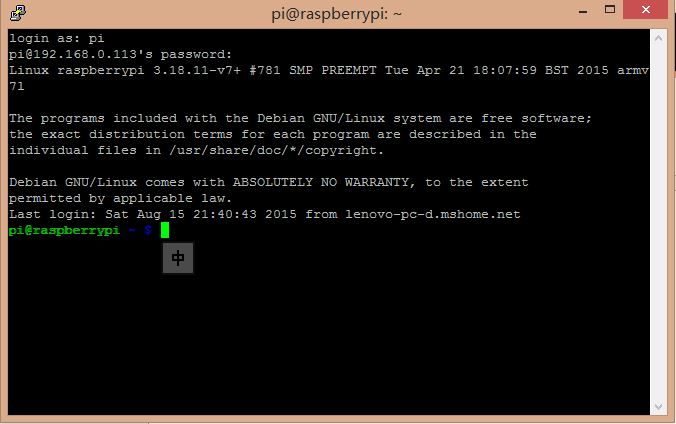
* 下载PuTTy，在Host Name中输入树莓派IP，



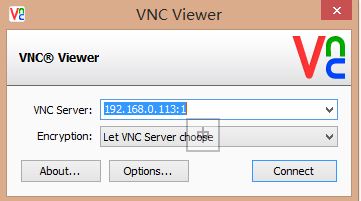
* 登入树莓派：在下图页面中login处输入树莓派账号（pi），在password处输入密码（raspberry）。回车

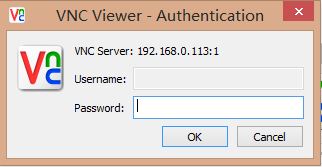


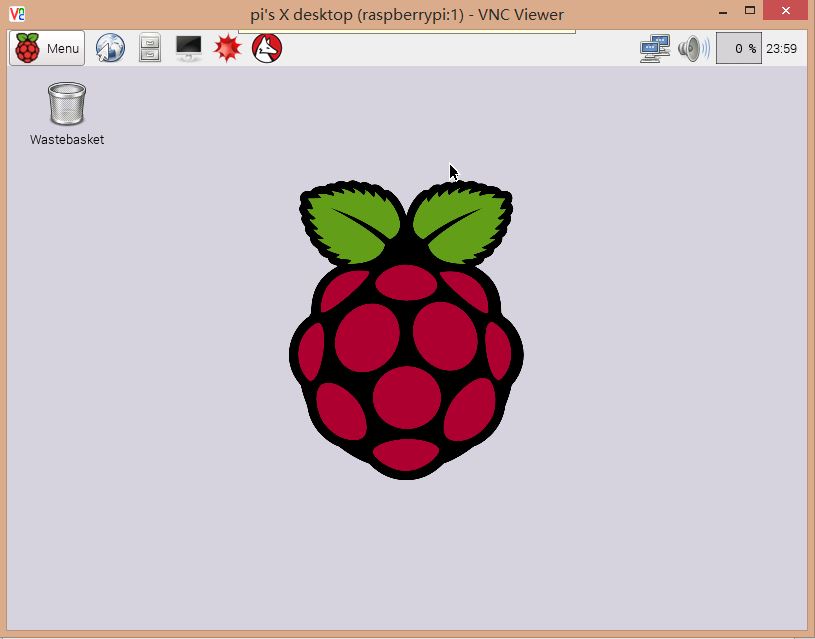
进入下图页面



* 树莓派的桌面环境配置：在上putty中输入sudo apt-get update； sudo apt-get install tightvncserver。安装完成vnc服务后，在上图中继续输入以下命令：vncserver :1 （一定要创建连接端口）。在第一次运行时，会提示你创建密码。在连接时就需要此密码。下载RealVNC（<http://www.realvnc.com>），在下图中输入树莓派IP:1(如192.168.0.113:1。

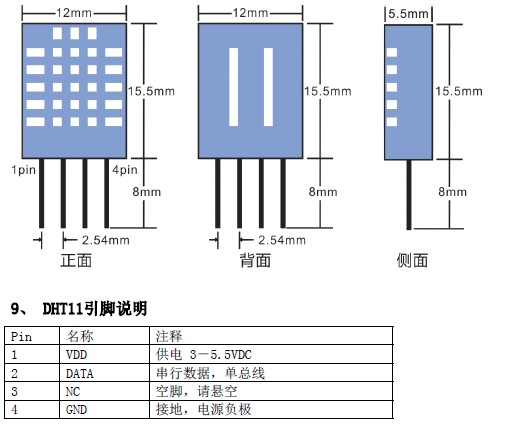
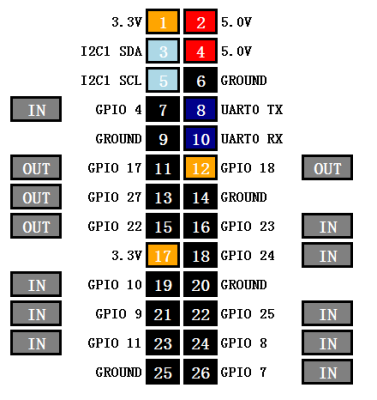


在下图中输入上面创建的密码

进入树莓派桌面

1. **硬件电路连接**

DHT11与树莓派引脚图如下：



具体电路连接如下DHT11的VDD连接树莓派17号管脚，GND连接20号管脚，DATA连接18号管脚。

实物图如下：

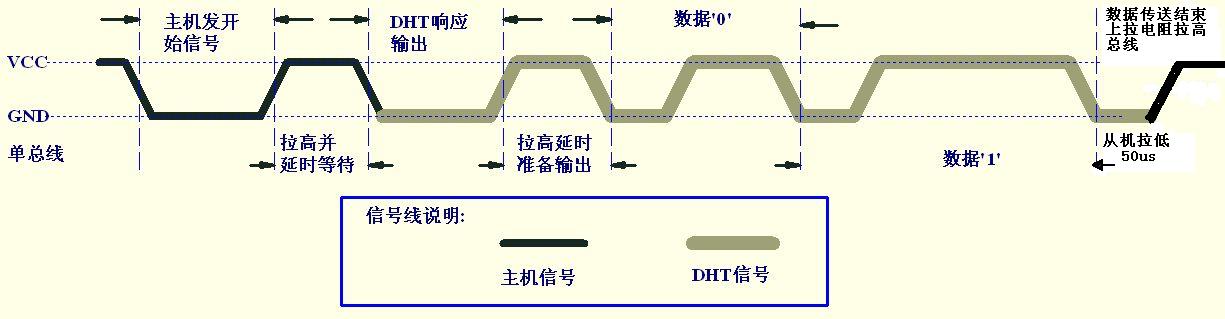
1. **DHT11简介**

数字温湿度传感器是一款含有已校准数字信号输出的温湿度复合传感器。它应用专用的数字模块采集技术和温湿度传感技术，确保产品具有极高的可靠性与卓越的长期稳定性。

DATA 用于微处理器与 DHT11之间的通讯和同步,采用单总线数据格式,一次通讯时间4ms左右,数据分小数部分和整数部分,具体格式在下面说明,当前小数部分用于以后扩展,现读出为零.操作流程如下:

一次完整的数据传输为40bit,高位先出。数据格式:8bit湿度整数数据+8bit湿度小数数据+8bi温度整数数据+8bit温度小数数据+8bit校验和数据传送正确时校验和数据等于“8bit湿度整数数据+8bit湿度小数数据+8bi温度整数数据+8bit温度小数数据”所得结果的末8位。

DHT11发送响应信号,送出40bit的数据,并触发一次信号采集,用户可选择读取部分数据.从模式下,DHT11接收到开始信号触发一次温湿度采集,如果没有接收到主机发送开始信号,DHT11不会主动进行温湿度采集.采集数据后转换到低速模式。通讯过程如图所示



1. **获取DHT11传感器温湿度**

采用Python语言实现温湿度的获取，每执行一次程序读取一次温湿度的值。采用putty中的nano编辑器完成汇编。步骤如下：

* 登入树莓派，获取root权限
* 进入nano编辑器，编辑Python程序，程序如下：#! /usr/bin/env python

import RPi.GPIO as GPIO

import time channel = 12

data = []

j = 0

GPIO.setmode(GPIO.BCM)

time.sleep(1)

GPIO.setup(channel, GPIO.OUT)

GPIO.output(channel, GPIO.LOW)

time.sleep(0.02)

GPIO.output(channel, GPIO.HIGH)

GPIO.setup(channel, GPIO.IN)

while GPIO.input(channel) == GPIO.LOW:

continue

while GPIO.input(channel) == GPIO.HIGH:

continue

while j < 40:

k = 0

while GPIO.input(channel) == GPIO.LOW:

continue

while GPIO.input(channel) == GPIO.HIGH:

k += 1

if k > 100:

break

if k < 8:

data.append(0)

else:

data.append(1)

j += 1

print "sensor is working."

print data

humidity\_bit = data[0:8]

humidity\_point\_bit = data[8:16]

temperature\_bit = data[16:24]

temperature\_point\_bit = data[24:32]

check\_bit = data[32:40]

humidity = 0

humidity\_point = 0

temperature = 0

temperature\_point = 0

check = 0

for i in range(8):

humidity += humidity\_bit[i] \* 2 \*\* (7 - i)

humidity\_point += humidity\_point\_bit[i] \* 2 \*\* (7 - i)

temperature += temperature\_bit[i] \* 2 \*\* (7 - i)

temperature\_point += temperature\_point\_bit[i] \* 2 \*\* (7 - i)

check += check\_bit[i] \* 2 \*\* (7 - i)

tmp = humidity + humidity\_point + temperature + temperature\_point

if check == tmp:

print "temperature : ", temperature, ", humidity : " , humidity

else:

print "wrong"

print "temperature : ", temperature, ", humidity : " , humidity, " check : ", check, " tmp : ", tmp

GPIO.cleanup()

* 保存（Ctrl+O）并退出（Ctrl+X）
* 执行此程序（dht11.py）:python dht11.py

1. **安装本地MYSQL**

* 使用管理员权限运行apt-get获取最新的MySQL及Python编程接口（之后用于数据库编程）：

sudo apt-get install mysql-server python-mysqldb

安装过程中需要输入root管理员的密码，该密码之后用于访问数据库系统。

在本地MYSQL创建库“test”,表“dht”和三个列“dht\_tem”,“dht\_hum”,“dht\_time”。

mysql -u root –p ；登入本地MySQL

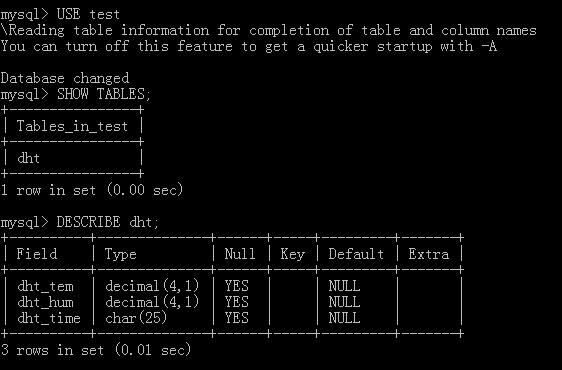
create database test ; 创建数据库”test”

USE test ; 选择数据库”test”

create table dht ;创建表“dht”

CREATE TABLE dht(dht\_tem DECIMAL(4,1), dht\_hum DECIMAL(4,1),dht\_time char(25)) ；创建列“dht\_tem”,“dht\_hum”,“dht\_time”

完成后查看建好的表“dht”,如下图所示：



* 在之前的dht11.py中添加如下几段语句：

import MySQLdb

conn = MySQLdb.connect(host='localhost',

port=3306,

user='root',

passwd='QAZQAZ',

db='test')

cur = conn.cursor()

value = [temperature,humidity,str(time.strftime('%A %X %Z',time.localtime(time.time())))]

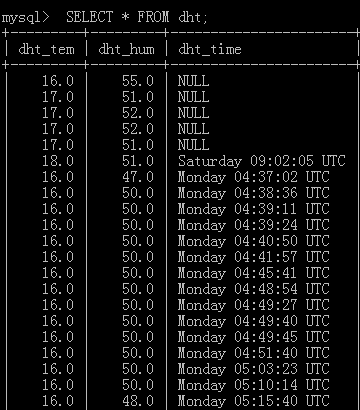
cur.execute("insert into dht (dht\_tem,dht\_hum,dht\_time) values(%s,%s,%s)",value)

conn.commit()

conn.rollback()

* 执行dht11.py：python dht11.py

按照之前数据库的操作方法，可在本地数据库MySQL查看到新插入的温湿度数据。如下图所示（已执行多次程序）：



1. **连接阿里云RDS数据库**

云服务器和云数据库RDS：云服务上安装了下mysql，在云盾里开放3306端口，远程同样能够正常连接数据库，但云服务器里搭建数据库可能不如云数据库来的稳定，云数据库更易维护。利用实名账户在阿里云试用一个月的免费云数据库。具体操作如下：

* 申请阿里云试用一个月的免费云数据库
* 登入阿里云RDS，创建账号”dht\_zzx”和创建数据库”dht”
* 登入自己创建的数据库“dht”,创建表“dht11”和三个列“dht\_tem”,“dht\_hum”,“dht\_time”。结果如下：



* 设置白名单：0.0.0.0/0 即不设置访问限制；

连接信息改为切换外网模式，并将外网地址rds1w17754gww117799w.mysql.rds.aliyuncs.com保存好。

* 在dht11.py中添加如下程序：  
  import requests

import json

db = MySQLdb.connect(host='rds1w17754gww117799w.mysql.rds.aliyuncs.com',

port=3306,

user='dht\_zzx',

passwd='qaz123',

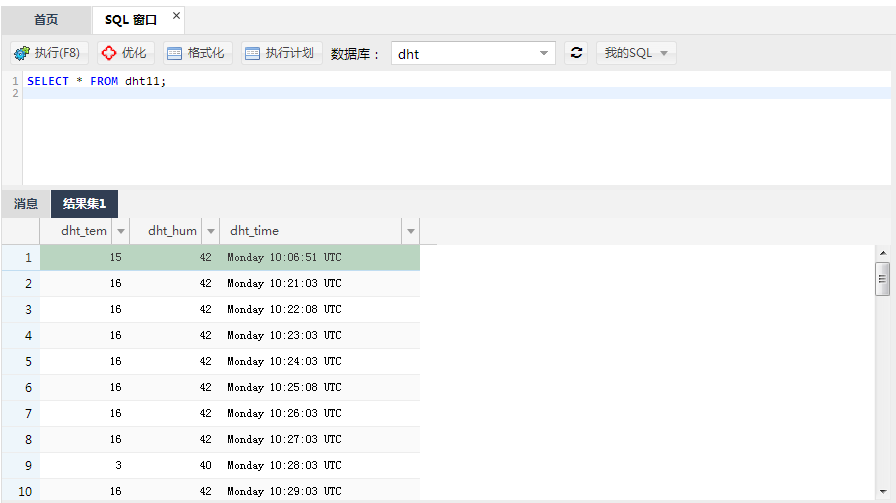
db='dht')

cur\_db = db.cursor()

cur\_db.execute("insert into tb\_dht (dht\_tem,dht\_hum,dht\_time) values (%s,%s,%s)",value)

db.commit()

运行程序，打开阿里云数据库，观察结果，如下：



1. **上传数据到传感云**

* 申请传感云账号
* 添加一个设备“dht11温湿度采集”，添加两传感器“dht11温度”，“dht11湿度”，得到传感器ID分别为：565274cce4b0932584ded623，5652756ee4b0932584ded624；
* 在个人中心查看自己的密钥：c0b419675b8955d2866e224e448b06e1；在API文档的插入数值型传感器数据点查看URL：<http://api.wsncloud.com/data/v1/numerical/insert>

在dht11.py中添加如下程序：  
#wendu

apiurl =' http://api.wsncloud.com/data/v1/numerical/insert'

apiheaders={'ak':'c0b419675b8955d2866e224e448b06e1','id':‘565274cce4b0932584ded623',

’value’: ‘temperature’}

r = requests.post(apiurl, params=apiheaders)

Print (r.text)

# shidu

apiurl =' http://api.wsncloud.com/data/v1/numerical/insert'

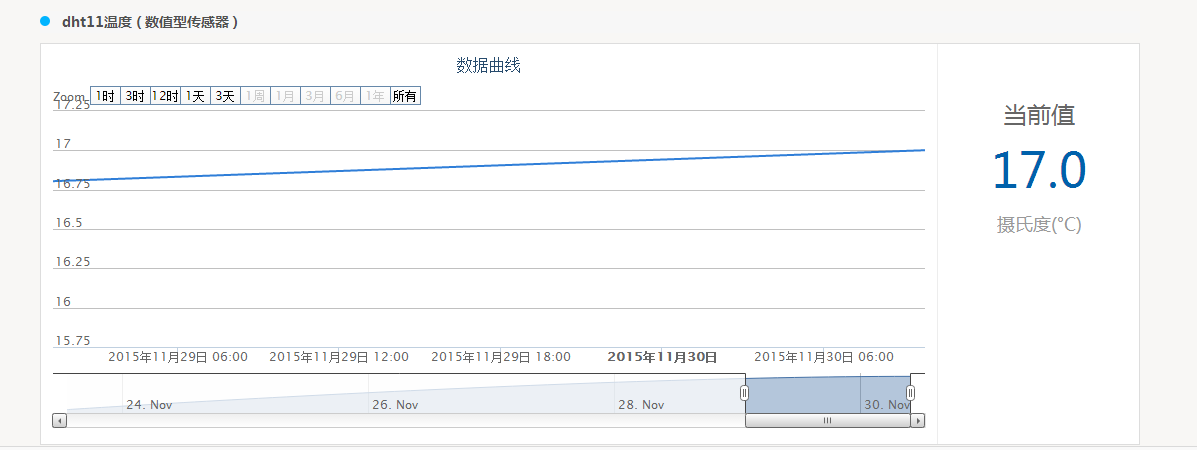
apiheaders={'ak':'c0b419675b8955d2866e224e448b06e1','id':‘5652756ee4b0932584ded624',

’value’: ‘humidity’}

r = requests.post(apiurl, params=apiheaders)

Print (r.text)

运行程序，打开阿里云数据库，观察结果，如下：





1. **Cron 实现定时功能**

Putty下运行如下程序：

Sudo  crontab –e

\*/1 \* \* \* \* /home/pi/zzx/dht.py ；每分钟后台运行dht11.py一次。