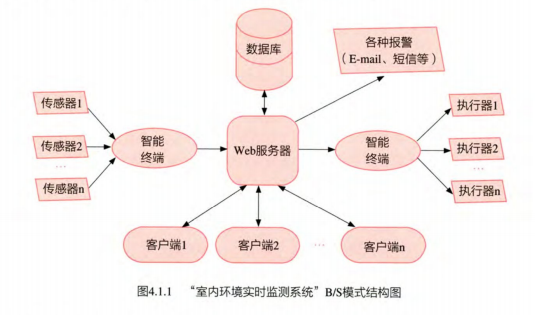
**项目实践：室内环境实时监控系统**

一、室内环境实时监控系统开发模式：



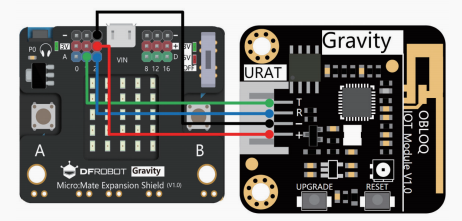
二、硬件选择：Micro:bit主板、扩展板、IoT（物联网）模块、温度传感器、蜂鸣器模块和服务器

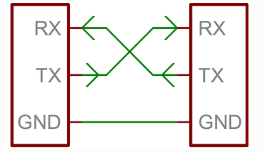
三、系统搭建：

**实践1、智能终端联网**

实践目标：把obloq模块，接入网络，在点阵屏上显示IP地址。

1、硬件连接：将Micro:bit主板插入扩展板，连接好IoT模块。





**发送端**

**接收端**

IoT模块接口：

TX口：连接扩展板的Pin1口

RX口：连接扩展板的Pin2口

“+”口：连接扩展板的正极

“-”口：连接扩展板的负极

2、在BXY中编写代码，并将代码烧录到Micro:bit运行，确认IoT模块连入无线网络。

from microbit import \*

import Obloq #导入OBLOQ IoT模块

SSID="jf3" #Wi-Fi名称,英文字符

PASSWORD="a12345678" #Wi-Fi密码，英文字符

uart.init(baudrate=115200, bits=8, parity=None, stop=1, **tx=pin2, rx=pin1**)

#串口设置，波特率，传输字节，奇偶校验，停止位数值，引脚设置

while Obloq.connectWifi(SSID,PASSWORD,10000) != True:

display.show(".") #连接不成功，显示“.”

display.scroll(Obloq.ifconfig()) #连接成功，显示IP地址

**实践2：搭建数据库**

数据库采用SQLite，数据库中有两个基本表，存储各种传感器数据。

|  |  |  |  |
| --- | --- | --- | --- |
| 表名称 | 字段 | 数据类型 | 作用 |
| sensorlist | sensorid | integer,自动编号 | 传感器唯一编号，主键 |
| sensorname | text | 传感器名称 |
| maxvalue | integer | 传感器报警最大值 |
| minvalue | integer | 传感器报警最低值 |
| sensorlog | logid | integer,自动编号 | 记录编号，主键 |
| sensorid | integer | 传感器编号 |
| sensorvalue | float | 传感器值 |
| updatetime | time | 上传时间 |

**代码如下：**

import sqlite3

import datetime

DATABASE = 'data/data.db'

def setup\_db():

db = sqlite3.connect(DATABASE)

cur = db.cursor()

cur.execute("CREATE TABLE IF NOT EXISTS sensorlist(sensorid INTEGER PRIMARY KEY autoincrement, sensorname text,maxvalue int,minvalue int)")

cur.execute("CREATE TABLE IF NOT EXISTS sensorlog(logid INTEGER PRIMARY KEY autoincrement , sensorid int,sensorvalue float,updatetime time)")

db.commit()

cur.execute("SELECT COUNT(\*) FROM sensorlist")

if cur.fetchall()[0][0] == 0:

cur.execute('INSERT INTO sensorlist(sensorid,sensorname,maxvalue,minvalue) VALUES(1,"温度传感器",39,5)')

cur.execute('INSERT INTO sensorlist(sensorid,sensorname,maxvalue,minvalue) VALUES(2,"湿度传感器",80,20)')

db.commit()

cur.execute("SELECT COUNT(\*) FROM sensorlog")

now = datetime.datetime.now()

now = now.strftime('%Y-%m-%d %H:%M:%S')

if cur.fetchall()[0][0] == 0:

cur.execute("INSERT INTO sensorlog(logid,sensorid,sensorvalue,updatetime) VALUES(1,1,39,'%s')"%now)

db.commit()

if \_\_name\_\_ == "\_\_main\_\_":

setup\_db()

**实践3：服务器搭建**

实践目标：

①终端接入服务器：采集温度信息，发送到服务器(POST)；

②搭建服务器：页面显示温度信息。

1、硬件连接：将温度传感器连接到扩展板的Pin0口，电脑连入无法网络，查看获取到的IP地址（即Web服务器的IP地址）。

2、在BXY中编写代码，利用micro:bit读取温度传感器的数据，并将数据传输给服务器。

from microbit import \*import ObloqIP="192.168.1.102" #Web服务器的地址PORT="8088" #Web服务器的端口SSID="jf3" #Wi-Fi的SSIDPASSWORD="a12345678" #Wi-Fi的密码uart.init(baudrate=115200, bits=8, parity=None, stop=1, tx=pin2, rx=pin1)while Obloq.connectWifi(SSID,PASSWORD,10000) != True: display.show(".")display.scroll(Obloq.ifconfig())Obloq.httpConfig(IP,PORT) while True: temp=round((pin0.read\_analog()/1024)\*3000/10.24,1) #读取温度传感器的数据 errno,resp=Obloq.get("input?id=1&val="+str(temp),10000) #向服务器发送温度数据 if errno == 200: display.show(str(resp)) else: display.show(str(errno)) sleep(1000\*5)

3、搭建服务器：获取传感器数据，存入数据库，并在页面中实时显示温度信息

import sqlite3

import datetime

from flask import Flask,render\_template, request

DATABASE = 'data/data.db'

app = Flask(\_\_name\_\_)

@app.route("/")

def hello():

db = sqlite3.connect(DATABASE)

cur = db.cursor()

cur.execute("SELECT \* FROM sensorlog WHERE sensorid =1")

data = cur.fetchall()

cur.close()

db.close()

temp1 = data[len(data) - 1]

temp=temp1[2]

return render\_template('vews.html', data=data,temp=temp)

#Adding data

@app.route("/input",methods=['POST','GET'])

def add\_data():

if request.method == 'POST':

sensorid = int(request.form.get('id'))

sensorvalue = float(request.form.get('val'))

else:

sensorid = int(request.args.get('id'))

sensorvalue = float(request.args.get('val'))

nowtime = datetime.datetime.now()

nowtime = nowtime.strftime('%Y-%m-%d %H:%M:%S')

db = sqlite3.connect(DATABASE)

cur = db.cursor()

cur.execute("INSERT INTO sensorlog(sensorid,sensorvalue,updatetime) VALUES(%d,%f,'%s')" %(sensorid,sensorvalue,nowtime) )

db.commit()

cur.execute("SELECT \* FROM sensorlist where sensorid = %d"% sensorid)

rv = cur.fetchall()

cur.close()

db.close()

maxrv = rv[0][2]

minrv = rv[0][3]

if sensorvalue > maxrv or sensorvalue < minrv:

return '1'

else:

return '0'

if \_\_name\_\_ == "\_\_main\_\_":

app.run(host="192.168.1.102",port=8088)

程序运行后，使用浏览器打开http://192.168.1.102:8088 即可实时查看温度数据。

**实践4：服务器搭建**

实践目标：以JSON格式返回特定传感器的所有数据

在搭建web服务器程序中增加以下路由视图函数代码，使用浏览器访问http:// 192.168.1.102:8088/get?id=1， 即可以以JSON格式返回温度传感器的所有数据。

#Get data

@app.route("/get",methods=['GET'])

def get\_data():

sensorid=int(request.args.get('id'))

db = sqlite3.connect(DATABASE)

cur = db.cursor()

cur.execute("SELECT \* FROM sensorlog WHERE sensorid = %s"% sensorid)

data = cur.fetchall()

dbsum=len(data)

dset={'sensorid':sensorid}

temp=[]

for i in range(dbsum):

value={}

value['sensorvalue']=data[i][2]

value['updatetime']=data[i][3]

temp.append(value)

dset['value']=temp

djson=json.dumps(dset)

return djson