

TiBox-NB100 NB-IoT可编程数传控制器开发指南

TiBox-NB100是钛云物联基于钛极OS(TiJOS)物联网操作系统开发的NB-IoT可编程数传控制器，用户可通过Java语言开发控制器内部的应用和控制逻辑以及与云平台交互过程。

钛云物联同时提供了钛极OS(TiJOS)物联网开发套件-**钛极小龟WIFI版和NB-IoT版**，并提供了从基础入门、开发进阶到综合例程等丰富的配套教程，让用户能够快速进入钛极OS(TiJOS)的开发世界。

在使用TiBox-NB100 可编程数据控制器开发之前，建议用户先通过钛极OS(TiJOS)物联网开发套件熟悉相关的开发过程，相关套件可通过在线商城进行购买，在本产品的SDK中包含了部分教程方便用户快速了解开发过程。

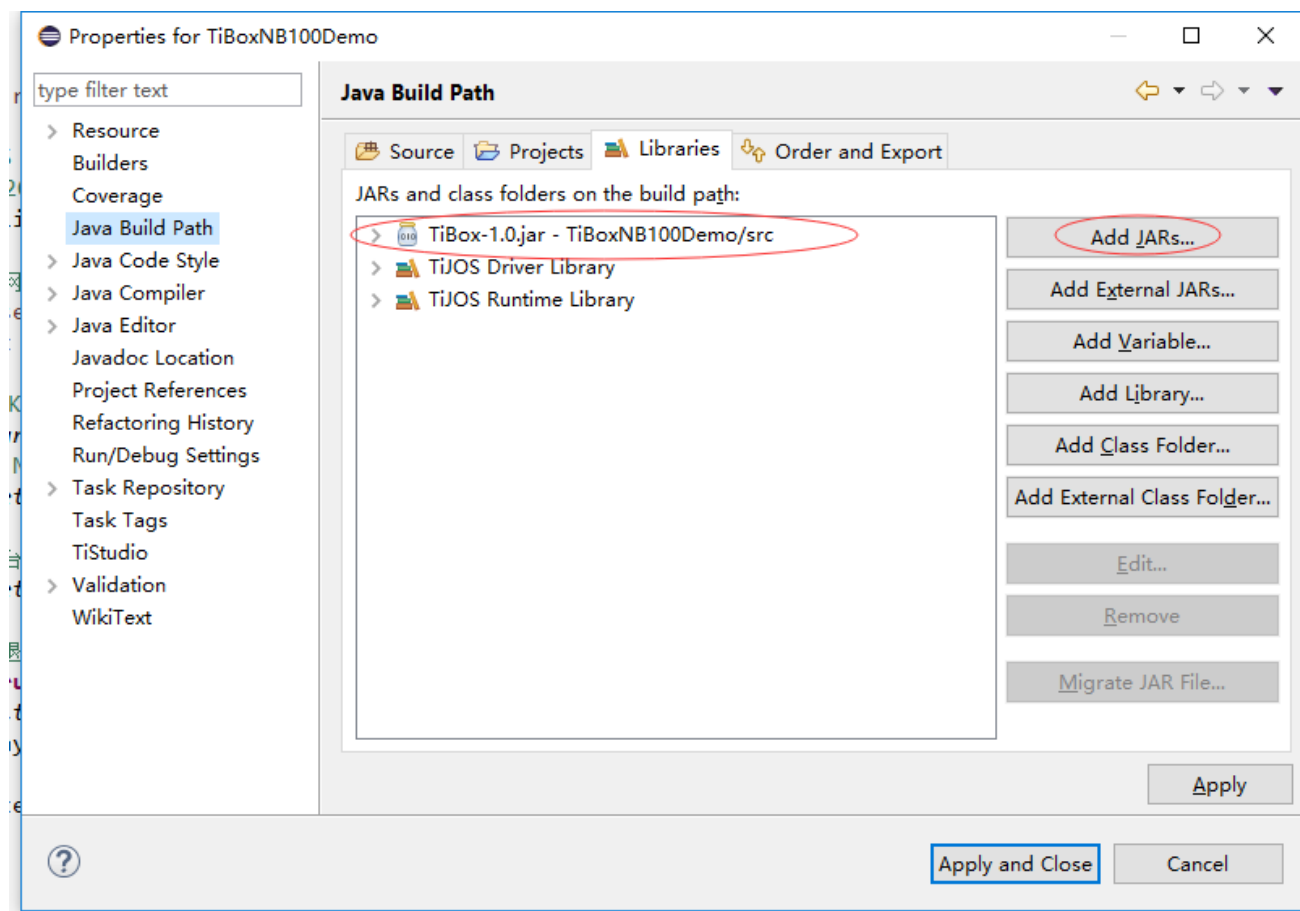
准备开发环境

安装TiStudio

在进行开发之前，请先安装Eclipse开发环境及TiStudio开发插件，具体安装过程请参考<钛极OS(TiJOS)开发环境搭建>文档。

创建TiJOS Application工程

TiBox-NB100提供了相关例程，用户可直接使用Eclipse打开例程进行修改或者新建一个TiJOS Application工程，加入TiJOS Driver Library，具体过程请参考<新建工程Hello TiJOS>文档，在新建工程后，将TiBox-1.0.jar加入到工程中，将在工程属性中将该Jar包加入到Java Build Path中，如下图所示：

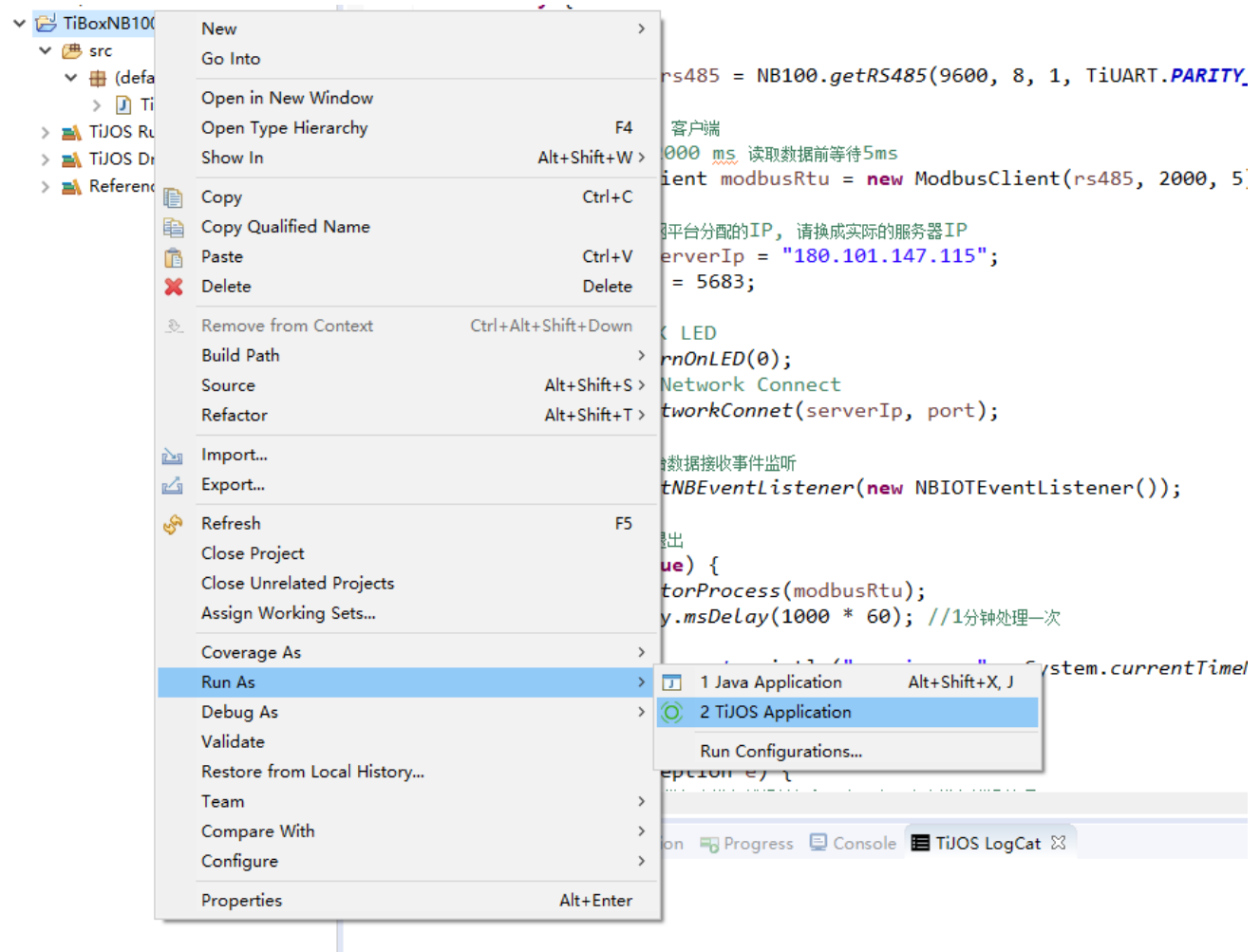


编码

此时，即可在Eclipse中进行相应的代码编写。

下载、运行

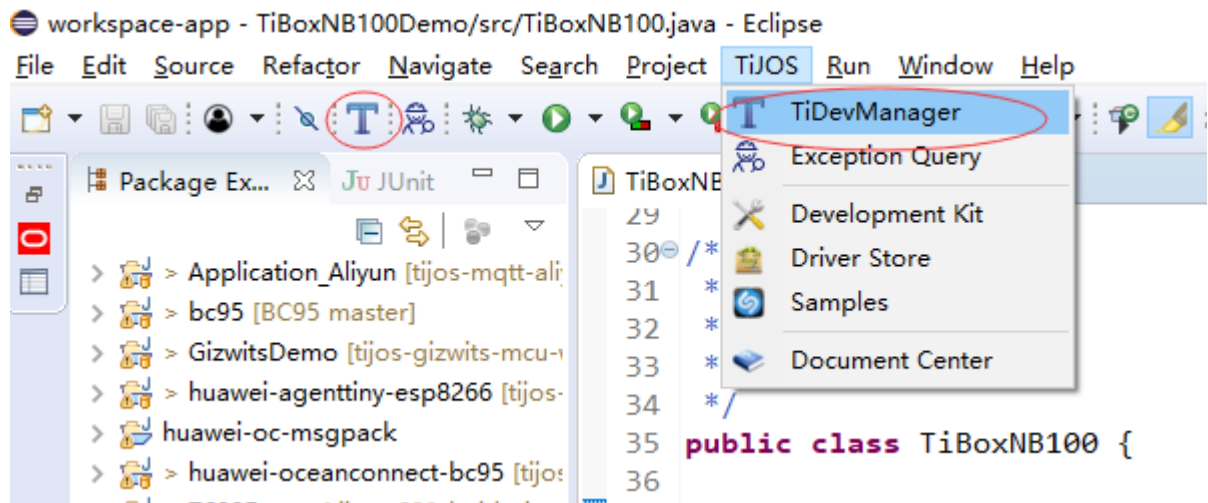
代码无误后，可通过Run As菜单选择"TiJOS Application"运行，在运行之前请确保已正确连接在TiBox的USB编程口，可从TiJOS LogCat中查看日志或打开TiDevManager查看日志。



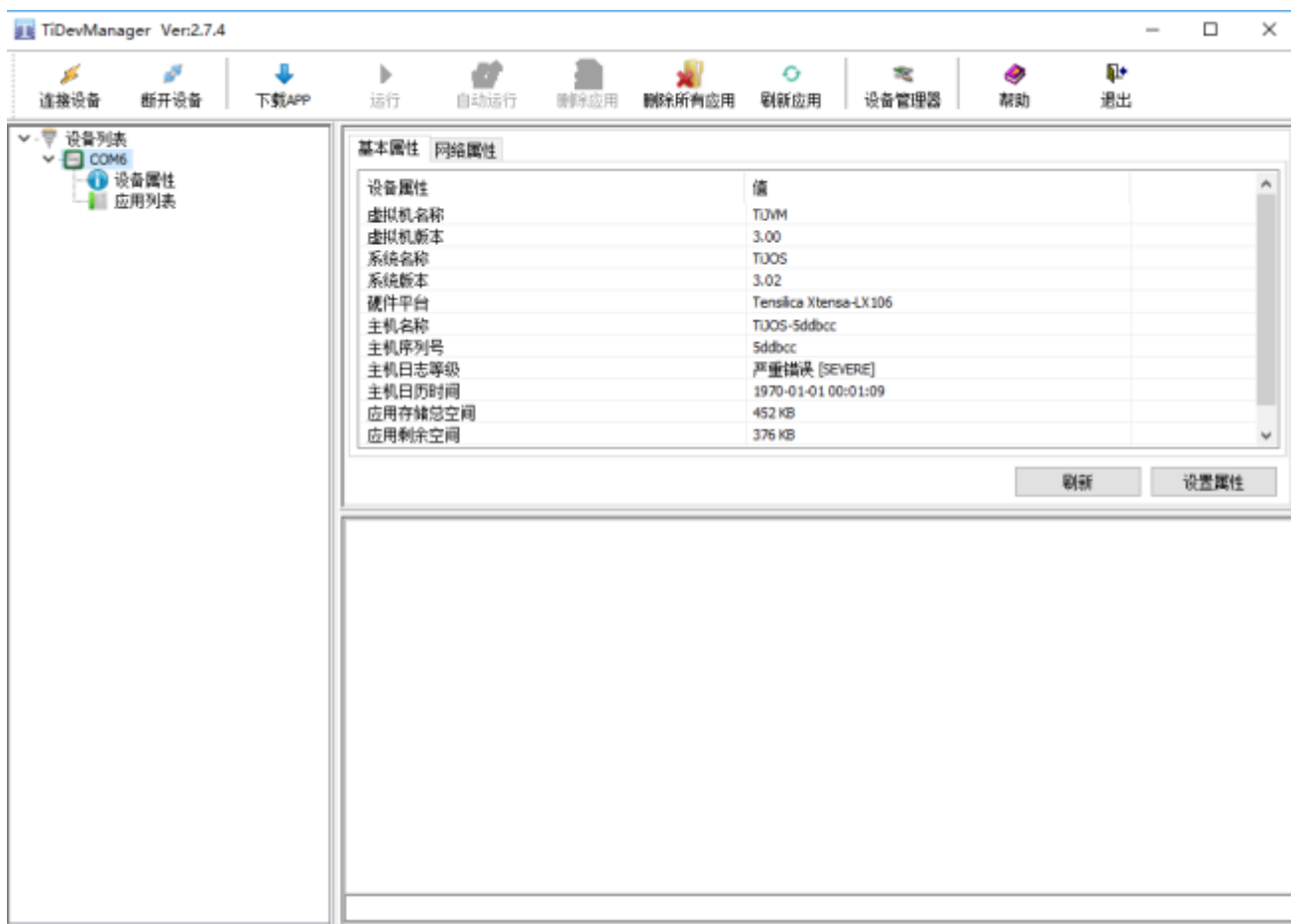
TiDevManager设备管理器

TiDevManager设备管理器是钛极OS(TiJOS)开发套件TiStudio的组成部分，用于查看设备信息及应用管理的工具，也可单独运行，详细使用方法请参考文档<TiDevManager设备管理器应用>。

TiDevManager可通过Eclipse的菜单启动。



启动后，可连接设备查看设备及应用信息



电信OceanConnect云平台接入指南

TiBox-NB100 提供了电信OceanConnect平台接入例程，在进行该例程的测试之前，请先通过中国电信申请相关的平台账号并在平台中进行配置，具体请参考<中国电信OceanConnect云平台接入向导>文档。

TiBox-NB100 编程开发说明

TiBox-NB100 内置钛极OS(TiJOS) 操作系统，支持通过Java语言进行应用开发，可通过钛极OS(TiJOS) 开发工具链IDE进行应用开发，钛极OS(TiJOS)在线文档可参考 doc.tijos.net

TiBox-NB100 Java类使用 说明

TiBox.NB100类提供了TiBox-N100所支持的硬件资源访问，包括RS485, RS232, NBIOT, GPIO等等，用户可通过在TiStudio中进行简单的开发即可支持各种应用，同时基于钛极OS(TiJOS)支持的MODBUS协议类，可以很方便地与支持MODBUS RTU协议的设备进行数据交互。

NB100 主要方法说明

方法	说明
TiSerialPort getRS485(int baudRate, int dataBitNum, int stopBitNum, int parity)	获取RS485接口，参数：波特率，数据位，停止位，校验位
TiSerialPort getRS232(int baudRate, int dataBitNum, int stopBitNum, int parity)	获取RS232接口，参数：波特率，数据位，停止位，校验位
void networkConnet(String serverIp, int port)	连接NB-IOT云平台，建议使用电信云。 serverIp/port: 电信云平台IP 及端口
void networkCoAPSend(byte[] dataBuffer)	发送数据到云平台, dataBuffer 待发送数据
void turnOnLED(int id)	打开指定LED灯, id = 0 : LED 灯, id=1 : NET灯
void turnOffLED(int id)	关闭指定LED灯, id = 0 : LED 灯, id=1 : NET灯
void startFlashLED()	闪烁指定LED灯, id = 0 : LED 灯, id=1 : NET灯
void stopFlashLED()	停止指定LED灯, id = 0 : LED 灯, id=1 : NET灯
void setNBEventListener	设置NB-IOT平台数据监听对象

IDeviceEventListener 数据监听

方法	说明
void onCoapDataArrived(byte []message);	当收到NBIOT云平台COAP协议数据时该接口被调用，meessage为云平台下发数据
void onUDPDataArrived(byte [] packet);	当收到NBIOT云平台UDP数据时该接口被调用，一般不使用该接口

TiSerialPort 串口类主要方法使用说明

通过getRS485/getRS232获取串口后，即可对串口进行读写操作

方法	说明
<code>void write(byte [] buffer ,int start ,int length)</code>	写入数据到串口 buffer: 待写入数据 start 缓存区开始位置 length 写入长度
<code>boolean readToBuffer(byte[] buffer, int start, int length, int timeOut)</code>	从串口读取指定长度数据 buffer: 读入数据缓存区, start 缓存区开始位置, length 读取长度, timeOut超时, 单位毫秒

一般调用过程 - MODBUS RTU为例

场景：

设备通过RS485连接到TiBox-NB100，通讯MODBUS RTU协议进行数据交互

设备通讯参数

参数	值
设备 ID	1
波特率	9600
数据位	8
停止位	1
停止位	无

寄存器：INPUT REGISTER (03)

寄存器地址	内容	操作权限	数值范围
0x0000	空气湿度	只读	0x00(0)--0x03E7(999) 对应 0%--99.9% 数值放大了
0x0001	空气温度	只读	0x8190(-400)--0x0320(800) 对应 -40℃--80℃ 负数

代码调用过程

1. 打开RS485并获取TiSerialPort对象

```
//通讯参数
TiSerialPort rs485 = NB100.getRS485(9600, 8, 1, TiUART.PARITY_NONE);
```

2. 创建MODBUS协议对象并挂接RS485

```
//MODBUS 客户端
//通讯超时2000 ms 读取数据前等待5ms
ModbusClient modbusRtu = new ModbusClient(rs485, 2000, 5);
```

3. 连接NB-IOT网络

```
//电信物联网平台分配的IP, 请换成实际的服务器IP
String serverIp = "180.101.147.115";
int port = 5683;

//NBIOT Network Connect
NB100.networkConnet(serverIp, port);

//设置NBIOT 电信云平台数据接收事件监听
NB100.setNBEventListener(new NBIOTEventListener());
```

4. 通过MODBUS协议读取寄存器数据

```
// MODBUS Server 设备地址
int serverId = 1;
// Input Register 开始地址
int startAddr = 0;
// Read 2 registers from start address 读取个数
int count = 2;

//读取Holding Register
modbusRtu.InitReadHoldingsRequest(serverId, startAddr, count);
int result = modbusRtu.execRequest();

//读取成功进行数据解析
if (result == ModbusClient.RESULT_OK) {
    //获取第1个寄存器值 - 温度
    int temperature = modbusRtu.getResponseRegister(modbusRtu.getResponseAddress(), false);
    //获取第2个寄存器值 - 湿度
    int humidity = modbusRtu.getResponseRegister(modbusRtu.getResponseAddress() + 1, false);
}
```

5. 将数据上报至云平台

```
//在电信云平台中需进行相应的PROFILE和插件配置, 具体请参考电信云平台相关文档
byte[] dataBuffer = new byte[5];

dataBuffer[0] = 0; // message id
dataBuffer[1] = (byte) (humidity >> 8);
dataBuffer[2] = (byte) (humidity & 0xFF);
dataBuffer[3] = (byte) (temperature >> 8);
dataBuffer[4] = (byte) (temperature & 0xFF);
```

```
try {  
    NB100.networkCoAPSend(dataBuffer);  
} catch (IOException e) {  
    e.printStackTrace();  
}
```

6. 当收到云平台数据时，在事件监听中进行相应命令解析和执行

```
/**  
 * NB-IOT 收到数据事件回调，电信云平台 通过onCoapDataArrived事件来进行发送数据到设备，  
 onUDPDataArrived 可忽略  
 */  
class NBIOTEventListener implements IDeviceEventListener  
{  
    @Override  
    public void onCoapDataArrived(byte []message) {  
        System.out.println("onCoapDataArrived");  
    }  
  
    @Override  
    public void onUDPDataArrived(byte [] packet) {  
        System.out.println("onUDPDataArrived");  
    }  
}
```

附：MODBUS 协议类使用说明

条目	说明
驱动名称	MODBUS RTU Client
适用	该驱动适用于符合MODBUS RTU 协议的设备
通讯方式	RS485/RS232/UART
Java Class	ModbusClient.java
图片	

主要接口

函数	说明
ModbusClient(TiSerialPort serialPort, int timeout, int pause)	实初始化, timeout: 通讯超时, pause: 发送命令后等待时间后开始读取数据
InitReadCoilsRequest(int serverId, int startAddress, int count)	初始化Read Coils 请求
InitWriteCoilRequest(int serverId, int coilAddress, boolean value)	初始化WRITE COIL register 请求- 单寄存器操作
InitWriteCoilsRequest(int serverId, int startAddress, boolean[] values)	初始化WRITE MULTIPLE COILS registers 请求- 多寄存器操作
InitReadHoldingsRequest(int serverId, int startAddress, int count)	初始化READ HOLDING REGISTERs 请求
InitReadDInputsRequest(int serverId, int startAddress, int count)	初始化READ DISCRETE INPUT REGISTERs 请求
InitReadAInputsRequest(int serverId, int startAddress, int count)	初始化READ INPUT REGISTERs 请求
InitWriteRegisterRequest(int serverId, int regAddress, int value)	初始化WRITE SINGLE REGISTER 请求 - 单寄存器操作
InitWriteRegistersRequest(int serverId, int startAddress, int[] values)	初始化WRITE MULTIPLE 请求 - 多寄存器操作
int execRequest()	执行MODBUS 请求并获得响应
int getExceptionCode()	获得返回的MODBUS异常码
int getResponseAddress()	获取返回数据的开始地址
int getResponseCount()	获取返回数据寄存器个数
boolean getResponseBit(int address)	获取指定地址COIL寄存器值
int getResponseRegister(int address, boolean unsigned)	获取指定地址InputRegister/HoldingRegister的值, unsigned: 返回值 为无符号或有符号

使用方法

第一步：RS485 初始化

获取TiBox-NB100的RS485对象


```
// 485端口
TiSerialPort rs485 = NB100.getRS485(9600, 8, 1, TiUART.PARITY_NONE);
```

第二步: MODBUS 客户端设置

创建ModbusClient对象, 设置RS485及通讯参数

```
// Modbus 客户端
// 通讯超时2000 ms 读取数据前等待5ms
ModbusClient mc = new ModbusClient(rs485, 2000, 5);
```

第三步: 操作寄存器

进行寄存器操作, 步骤:

1. 通过InitXXXRequest初始化参数,
2. execRequest执行请求, 并获取响应
3. getResponseRegister

```
//初始读取Holding Register参数, 设备地址, 寄存器开始地址, 个数
mc.InitReadHoldingRequest(serverId, startAddr, count);
//执行请求
int result = mc.execRequest();
//执行成功
if (result == ModbusClient.RESULT_OK) {
    //解析寄存器地址及值(无符号或有符号)
    int humidity = mc.getResponseRegister(mc.getResponseAddress(), false);
    int temperature = mc.getResponseRegister(mc.getResponseAddress() + 1, false);
}
```

技术支持

如果您有任何技术问题, 可通过电话, QQ群等方式与我们联系, 同时钛云物联可提供产品定制, 通讯协议开发, 云端接入, 技术培训等多种服务。

更多资源

TiBox-NB100是钛云物联的钛极OS(TiJOS)物联网操作系统的一个典型应用, 关于钛极OS(TiJOS)物联网操作系统可参考如下资源:

资源	url
钛极OS官网	www.tijos.net
钛极OS开发者社区	bbs.tijos.net
钛极OS(TiJOS) 文档中心	http://doc.tijos.net
钛极OS(TiJOS) 驱动仓库	http://store.tijos.net/
钛极OS(TiJOS) JDK API文档	http://dev.tijos.net/javadoc
微信公众号 - 钛极OS	
钛极OS物联网开发交流群QQ - 737547181	

联系方式

北京钛云物联科技有限公司

商务合作: 13911058165

品牌热线: 010-86462928

公司网址: www.tijos.net

电子邮件: tijos@tijos.net

在线购买: <https://shop423269048.taobao.com/>