1. **初步了解Arduino**

**1.何为Arduino**

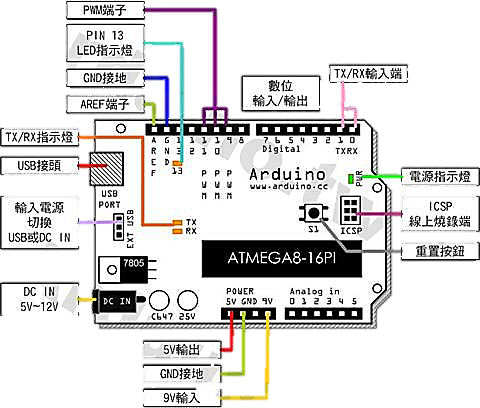
Arduino是一款便捷灵活、方便上手的开源电子原型平台。包含硬件（各种型号的Arduino板）和软件（Arduino IDE)。

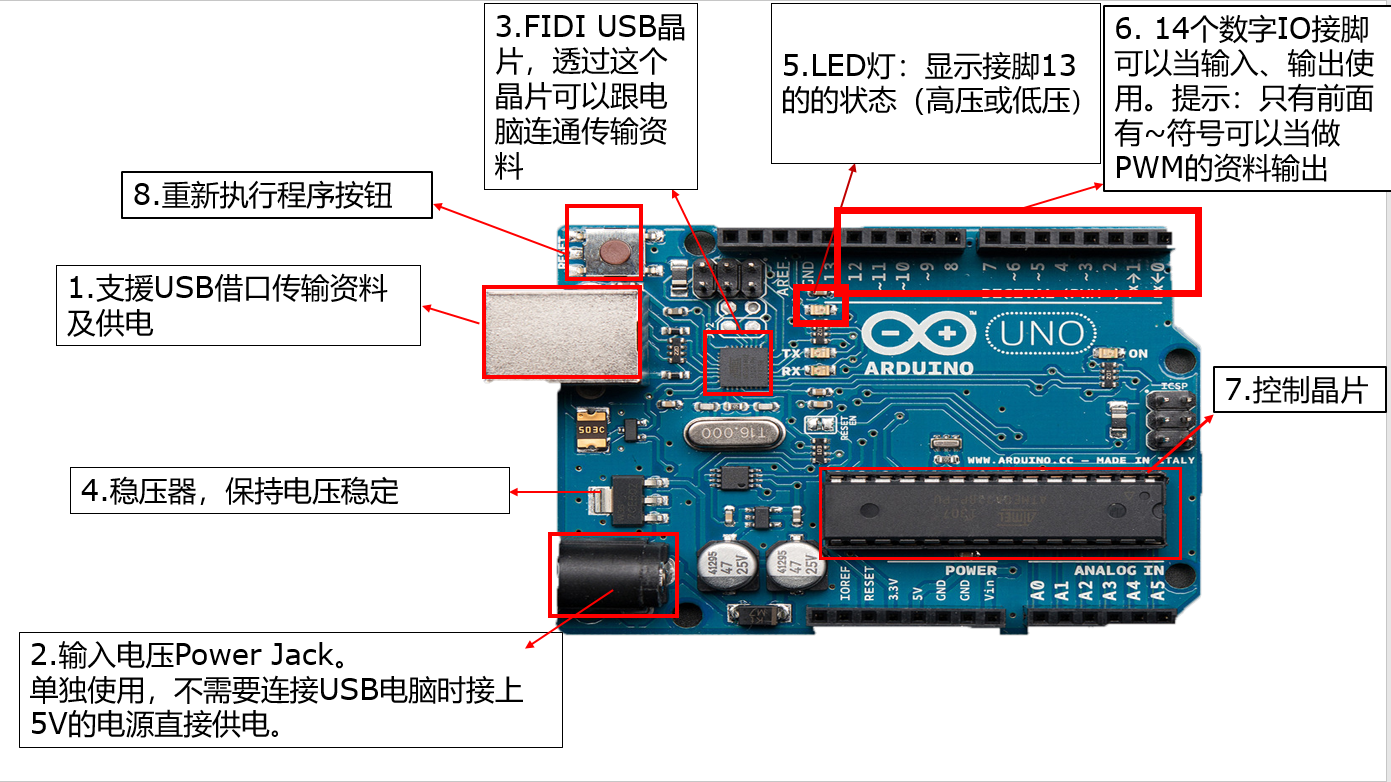
**2.Arduino的优势**

便捷灵活、容易上手的开源硬件产品，具有丰富的接口，有数字I/O口，模拟I/O口，同时支持SPI,IIC,UART串口通信。

**3.Arduino的基本硬件结构**

**a．常用的Arduino开发板**





**b.面包板原理图**

面包板的两边拥有4组8条横向相连的插孔，这4组我们称之为电源孔，一般是作为电源引入的通路，中间是上下两部分，纵向每5个孔为一个通道（这5个孔是联通的），这个就是主工作区，用来插接原件和跳线。

**4.开发板的供电方式**

**a．外部直流电源通过电源插座供电**

**b．电池连接电源连接器的GND和VIN引脚**

**c．USB接口直流供电**

1. **Arduino基本语法**
2. **基本语法架构**

**a．setup（）函数和loop（）函数**



Void loop():这是Arduino的主函数。这套程序会一直重复执行，直到电源被断开。

Void setup()：当Arduino板起动时setup()函数会被调用。用它来初始化变量，引脚模式，开始使用某个库，等等。该函数在Arduino板的每次上电和复位时只运行一次

**b．常用的语法**

Serial.begin(9600); 这是设置波特率，用于串口通信，9600是指传输速度为9600bit/s。常见的波特率有4800，9600,14400，38400等。

delay(1000); 延迟一秒钟，单位为毫秒。

Serial.println()，在串口监视器中打印出值

Serial.aviliable():返回串口缓冲区中当前剩余的字符个数，一般用于判断缓冲区有无数据

Serial.Read():用来读取数据

其他语法(如循环结构和判断结构等)和C语言相似。

**c．常用的函数**

**pinMode(pin,mode)：将指定的引脚配置成输出或输入模式。**

**其中pin：要设置模式的引脚;mode:引脚的模式（有INPUT和OUTPUT），由引脚使用情况决定输入输出；**

**digitalWrite(pin,value):给一个数字引脚写入高(HIGH)低(LOW)电平。**

**pin:引脚编号，如（1,5，13，A0，A3）；value:HIGH或者是LOW；**

**digitalRead(pin):读取引脚值，其返回值为HIGH或者是LOW。**

**pin:想要读取的引脚号；**

**analogRead(pin):从指定模拟引脚读取数值，其返回值为0~1023的整数值。**

**analogWrite(pin,value):从一个引脚输出模拟值。**

**pin:引脚号，value:占空比（0~255；**

**tone(pin,frequency,dration):在一个引脚上产生一个特定频率的方波。**

**pin:引脚号,frequency：产生声音的频率(Hz),dratioin:声音持续的时间(ms)；**

**noTone(pin):停止由tone(pin)产生的方波。**

**d.软串口通信**

**SoftwareSerial mySerial=SoftwareSerial(rxPin,txPin)**

**或者**

**SoftwareSerial mySerial(rxPin,txPin)**

**Myserial:自定义软串口对象，**

**rxPin：软串口接收引脚，**

**txPin：软串口发送引脚**

**如：**SoftwareSerial mySerial(10, 11);

即是新建一个名为mySerial的软串口，并将2号引脚作为RX端，3号引脚作为TX端。

1. **Arduino常见器件的应用**
2. **点亮LED灯**

**a.代码：**

**void setup() {**

**// initialize digital pin LED\_BUILTIN as an output.**

**pinMode(LED\_BUILTIN, OUTPUT);//设置13号引脚为输出模式**

**}**

**// the loop function runs over and over again forever**

**void loop() {**

**digitalWrite(LED\_BUILTIN, HIGH); // 设置13号引脚为高电平，此时LED灯亮**

**delay(1000); // 等一秒钟**

**digitalWrite(LED\_BUILTIN, LOW); // 设置13号引脚为低电平，此时LED灯灭**

**delay(1000); // 等一秒钟**

**}**

**b.硬件：**

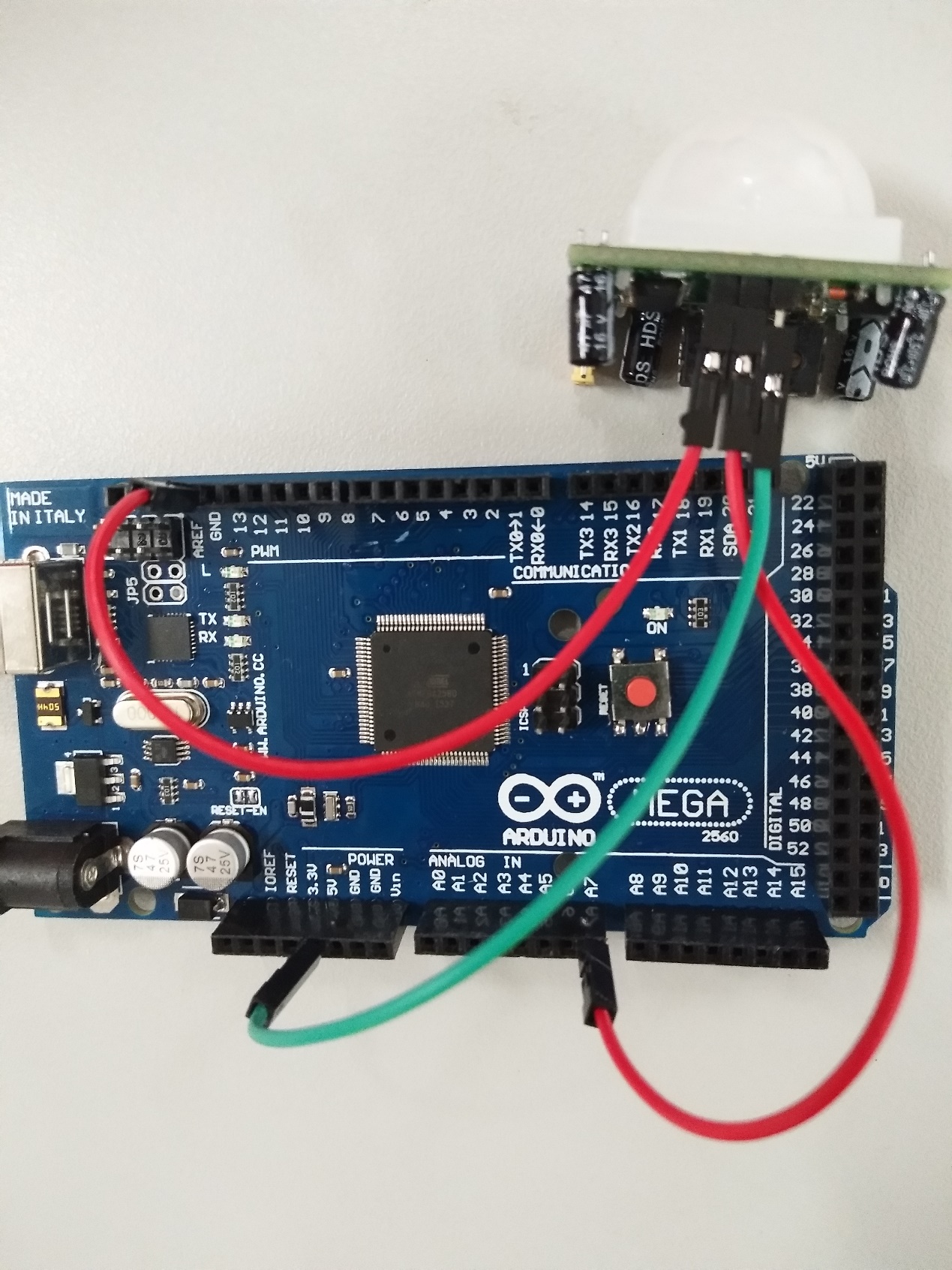
**LED灯长引脚为正极，短引脚为负极，负极接开发板GND，正极接开发板13号引脚（正极引脚可自定义，例子里定义为13号引脚），开发板和电脑相连。**

1. **人体红外传感器**
2. **原理：**

利用人体恒定在37℃的体温，会发出特定波长10um左右的红外线，被动式探头就是靠探测人体发射的10um左右的红外线而进行工作的。

1. **硬件：**

三个引脚分别为VCC、OUT、GND，VCC和GND用于接开发板对应的VCC和GND，OUT接模拟接口。后面的两个十字螺旋分别为灵敏度调节（顺时针灵敏度越高）和延时调节（顺时针延迟加长）。

****

1. **代码：**

**int PIR\_sensor = A5; //指定PIR模拟端口 A5**

**int val = 0;**

**void setup()**

**{**

**pinMode(PIR\_sensor, INPUT); //设置PIR模拟端口为输入模式**

**Serial.begin(9600); //设置串口波特率为9600**

**}**

**void loop()**

**{**

**val=analogRead(PIR\_sensor);//读取A0口的电压值并赋值到val**

**Serial.println(val); // 向串口发送val值**

**delay(800);**

**}**

1. **门磁开关：**
2. **硬件：**

**连接此部分需要用到面包板，一个电阻，门磁和Arduino板。门磁的两根导线分别和GND和电阻的一端，其中门磁和电阻相连的那一列需要和Arduino板中的模拟端口，电阻的另一端和VCC相连（若VCC和GND接的顺序不一样，会导致模拟端口的输出值不同）。**

1. **代码：**

**int BUTTON=7; //声明变量，按钮接在第7针脚；**

**int val=0; //声明变量，用来存储按钮返回值HIGH或者LOW；**

**void setup() //初始化；**

**{**

**pinMode(BUTTON,INPUT);//定义BUTTON为输出模式；**

**Serial.begin(9600);**

**}**

**void loop()**

**{**

**val=digitalRead(BUTTON);//读取数字针脚的值；**

**Serial.println(val);**

**if(val==0)//比较前一次和现在的按钮返回的值，**

**//只有现在按下按钮这个状态下，才改变state的值**

**{**

**Serial.println("Open door");**

**}**

**else**

**{**

**Serial.println("Close door");**

**}**

**}**

1. **蜂鸣器**
2. **硬件：**

**蜂鸣器有两种，一种是有源蜂鸣器，另一种是无源蜂鸣器，其中源指的是震荡源。短脚为负极，长脚为正极。分别接GND和数字引脚。**

1. **代码：**

**int speakerPin = 3;**

**void setup()**

**{**

**pinMode(speakerPin , OUTPUT); //设定3号引脚为输出模式**

**}**

**void loop()**

**{**

**tone(speakerPin,262, 100);**

**delay(1000);**

**}**

1. **蓝牙（以HC-05为例）**
2. **硬件介绍：**

**蓝牙模块有四个接口，分别为VCC、GND、TXD、RXD。其中VCC和GND用于供电，RXD和TXD用于接收和传输数据。**

1. **连接方式：**

**蓝牙模块的TXD要和开发板的RXD相连，蓝牙模块的RXD和开发板的TXD相连。**

1. **进入AT模式：接入电源之前长按蓝牙上面的黑色按钮，直到接入电源后看到指示灯两秒闪一次时才松手。同时波特率要设置为38400，在串口监视器里输入“AT+ ”进入AT模式可以查看蓝牙模块的地址、名称、密码等。**
2. **进入工作状态：断开电源，重新接入电源，看到指示灯闪烁频率为一秒一次时即可。**
3. **代码：**

#include <SoftwareSerial.h>

SoftwareSerial BT(10, 11);

char val;

void setup() {

Serial.begin(9600);

Serial.println("BT is ready!");

// HC-05默认，38400

BT.begin(9600);

pinMode(13,OUTPUT);

}

void loop() {

//从串口发送到手机

if (Serial.available()) {

val = Serial.read();

BT.print(val);

}

// 从手机发送到串口

if (BT.available()) {

val = BT.read();

Serial.print(val);

}

}

1. **通过蓝牙控制LED灯**
2. **硬件介绍：**

**蓝牙TX、RX注意要和开发板上的TX、RX反接。LED正极接13号引脚，负极接GND**

1. **代码：**

#include <SoftwareSerial.h>

int txPin = 7;//接蓝牙TX口

int rxPin = 6;//接蓝牙RX口

SoftwareSerial BTSerial(txPin,rxPin);

char val;

int ledPin=13;

void setup()

{

pinMode(ledPin,OUTPUT);

Serial.begin(9600);//

BTSerial.begin(9600);//这里设置蓝牙通信波特率，刚买来的一般是38400

}

void loop()

{

if (Serial.available()) {

val = Serial.read();

BTSerial.print(val);

}

if ( BTSerial.available()) {

val = BTSerial.read();

Serial.print(val);

}

if (val =='1')

{

digitalWrite(ledPin, HIGH); //把13号引脚写入高电平

BTSerial.println("LED ON");

}

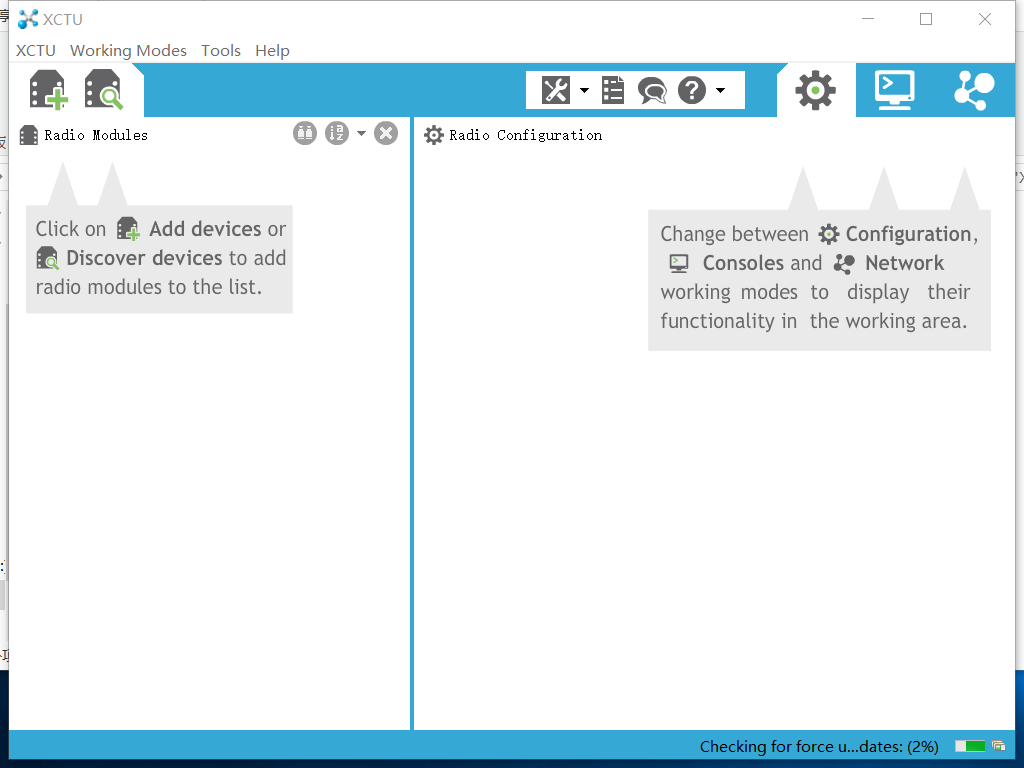
else if (val == '0')

{

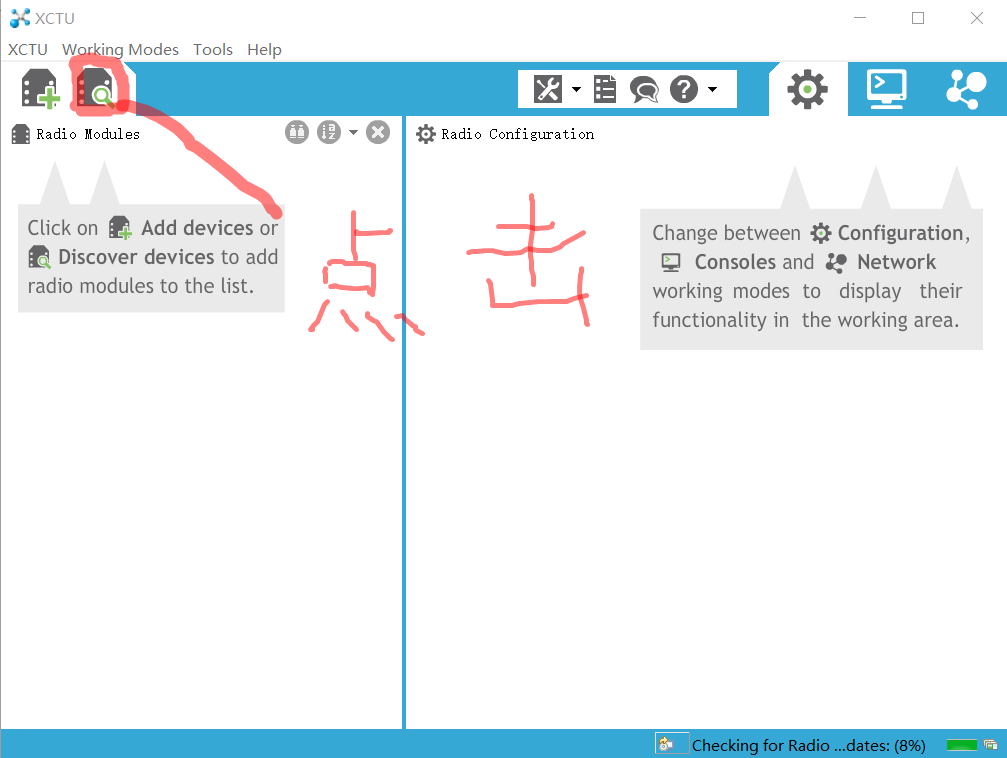
digitalWrite(ledPin, LOW); BTSerial.println("LED OFF");

}

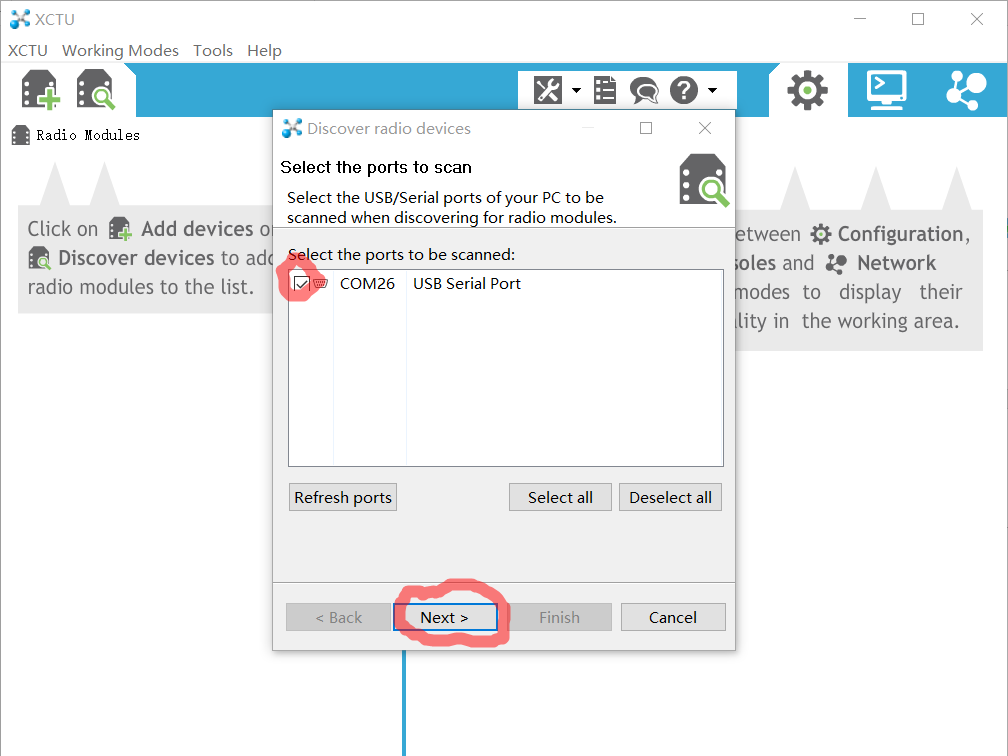
}

1. **XBee**
2. **硬件：**

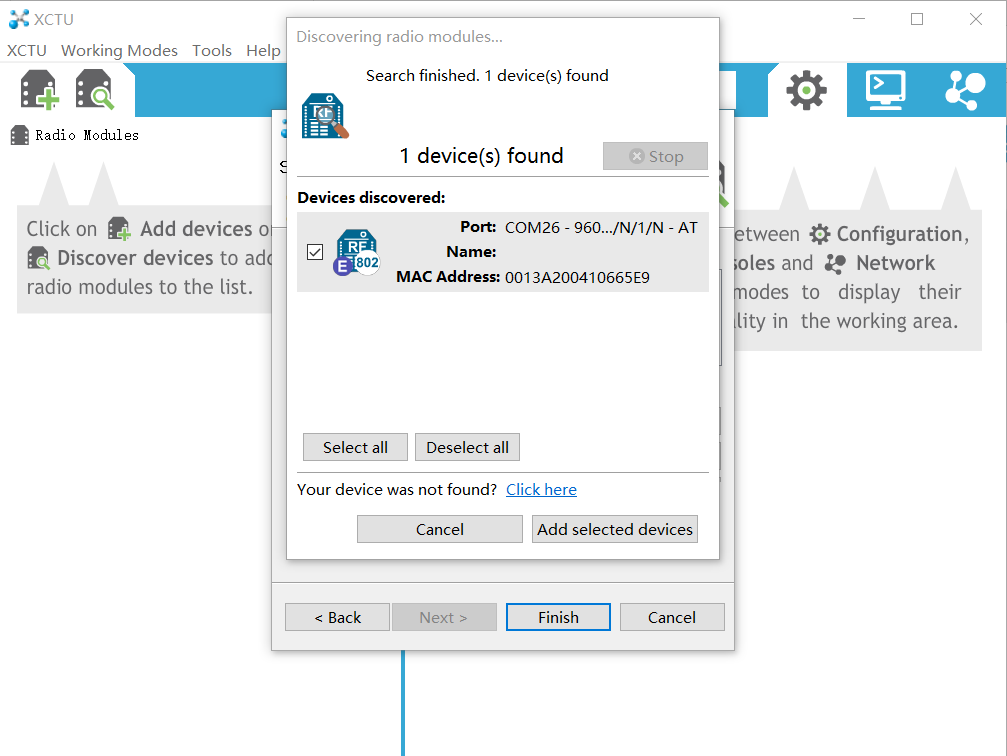
**打开界面**



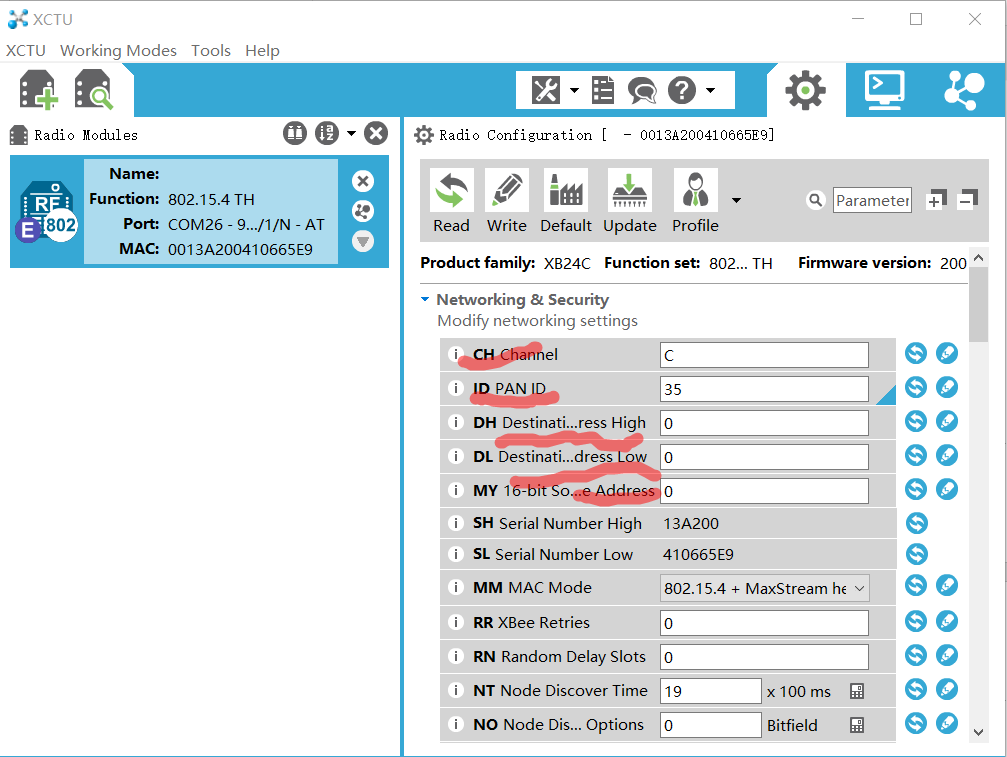
**点击搜索**



**勾选端口，next，finished**



**添加服务**

****

**将两个频道、id调为一致，然后选择广播状态**

1. **代码：**

**#include <SoftwareSerial.h>**

**SoftwareSerial BT(10, 11);//rx tx**

**char val;**

**int led=12;**

**void setup() {**

**Serial.begin(9600);**

**BT.begin(9600);**

**pinMode(led,OUTPUT);**

**}**

**void loop() {**

**if (BT.available())**

**{**

**val = BT.read();**

**Serial.print(val);**

**}**

**if(val=='a')**

**{**

**digitalWrite(led,HIGH);**

**}**

**if(val=='b')**

**{**

**digitalWrite(led,LOW);**

**}**

**}**

1. **按键开关的读值**
2. **硬件介绍：**



**正常状态下1、2导通，3、4导通当按键按下时1、2、3、4导通**

1. **电路连接：**

**本电路需要利用面包板，且按键开关中间的凹槽要和面包板中间的凹槽垂直（两凹槽要垂直）。1号引脚的一端接VCC，4号引脚的一端接一个电阻，电阻的另一端需要和GND相连，同时，4号引脚和一个数字端口相连接。**

1. **代码：**

**int BUTTON=7;//声明变量，按钮接在第7针脚**

**int val=0;//用来存储按钮返回值**

**void setup()**

**{**

**pinMode(BUTTON,INPUT);//设置7号针脚为输入模式**

**}**

**void loop()**

**{**

**val=digitalRead(BUTTON);**

**Serial.println(val);**

**}**

1. **按钮开关控制LED灯**
2. **电路连接：**

**LED正极接数字引脚，负极接GND，按钮开关接法同8（b）中**

1. **代码：**

**int LED=12; //声明变量 LED灯接在13针脚；**

**int BUTTON=7; //声明变量，按钮接在第7针脚；**

**int val=0; //声明变量，用来存储按钮返回值HIGH或者LOW；**

**int val\_old=0;//声明变量，存储上一次按钮返回值；比对val和val\_old的值，来判断按钮是按下还是抬起**

**//val==LOW && val\_old==HIGHT 抬起状态；val==HIGH && val\_old==LOW 按下状态；**

**int state=0;//0 off,1 on 声明变量，起到中转的作用，小技巧；**

**void setup() //初始化；**

**{**

**pinMode(LED,OUTPUT); //定义LED灯输入模式；**

**pinMode(BUTTON,INPUT);//定义BUTTON为输出模式；**

**Serial.begin(9600);**

**}**

**void loop()**

**{**

**val=digitalRead(BUTTON);//读取数字针脚的值；**

**Serial.println(val);**

**if((val==HIGH) && (val\_old==LOW))//比较前一次和现在的按钮返回的值，**

**//只有现在按下按钮这个状态下，才改变state的值**

**{**

**state=1-state;**

**}**

**if((val==LOW && val\_old==HIGH)) //去抖的作用，一般10-15ms；**

**{**

**delay(10);**

**}**

**val\_old=val;//把现在按钮返回的值赋予val\_old；**

**if(state==1)//控制灯开关；**

**{**

**digitalWrite(LED,HIGH);**

**}**

**else**

**{**

**digitalWrite(LED,LOW);**

**}**

**}**

1. **气敏传感器（烟雾及可燃气体检测模块MQ-2）**
2. **硬件介绍：**

**MQ-2有四个引脚，分别为VCC、GND、AO、DO，VCC和GND用于供电。输出有两种方式：AO输出当前特殊气体含量的参考值（0-1023），基本上100以下的示数为正常，DO根据预先设定的参考值的阈值输出或低电平信号。**

1. **电路连接：**

**VCC和GND分别对应开发板上的VCC和GND，A0连接A7，用于读出模拟值**

1. **代码：**

**int AD=A7;**

**int val=0;**

**void setup()**

**{**

**pinMode(AD,INTPUT);**

**Serial.begin(9600);**

**}**

**void loop()**

**{**

**val=analogRead(AD);**

**Serial.println(val,DEC);**

**delay(1000);**

**}**

1. **光照传感器(以GY-30模块为例)**
2. **硬件介绍：**

**GY-30模块有五个引脚，分别为VCC、SCL、SDA、ADDR、GND，其中VCC和GND用于供电。该模块采用IIC通信方式，SCL为IIC总线时钟线，SDA为IIC总线数据线，ADDR为IIC地址引脚。**

1. **电路连接：**

**VCC、GND、SCL、SDA分别和对应开发板上面的针脚相连接即可。**

1. **代码：**
2. #include <Wire.h> //IIC//20sda 21scl  
   #include <math.h>   
   int BH1750address = 0x23;   
   byte buff[2];  
   void setup()  
   {  
     Wire.begin();// 开启 I2C 总线(主设备)  
     Serial.begin(9600);  
   }  
   void loop()  
   {  
    int i;  
    uint16\_t val=0;  
    BH1750\_Init(BH1750address);  
    delay(200);  
    if(2==BH1750\_Read(BH1750address))  
     {  
      val=((buff[0]<<8)|buff[1])/1.2;  
      Serial.print(val,DEC);       
      Serial.println("[lx]");   
     }  
    delay(150);  
   }  
   int BH1750\_Read(int address) //  
   {  
     int i=0;  
     Wire.beginTransmission(address);  
     Wire.requestFrom(address, 2);// 请求gy-30 两个字节  
     while(Wire.available()) //  
     {  
       buff[i] = Wire.read(); //接收一个字节//读取DS1307秒  
       i++;  
     }  
     Wire.endTransmission(); //  结束发送  
     return i;  
   }  
   void BH1750\_Init(int address)   
   {  
     Wire.beginTransmission(address);// 开启发送  
     Wire.write(0x10) // 写入 DS1307 秒地址  
     Wire.endTransmission();//  结束发送  
   }

**12．继电器**

int JDQ = 8;  
  
void setup() {  
pinMode(JDQ , OUTPUT);  
}  
  
void loop() {  
digitalWrite(JDQ,HIGH);  
delay(3000);  
digitalWrite(JDQ,LOW);  
delay(3000);  
}

