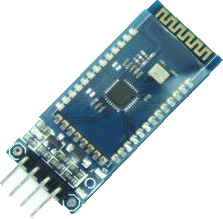
**Arduino课堂笔记**

1.复习蓝牙模块

蓝牙模块（Bluetooth module）是指集成蓝牙功能的芯片基本电路集合，用于短距离2.4G的无线通讯模块。

VCC 电源正极，接（3.3v～5v）

GND 电源负极，接GND

TXD 模块串口发送脚，接Arduino的RX引脚。

RXD 模块串口接收脚，接Arduino的TX引脚。

Arduino UNO的0和1引脚为串口通信专用引脚，也是程序下载的专用引脚，如果蓝牙模块占用此引脚，那么下载程序前可能需要先拔掉蓝牙模块的接线，下载完程序后再接上。

2.软串口通信

在Arduino UNO上只有0(RX)、1(TX)两个引脚是串口通信引脚，我们把这组串口引脚称为硬件串口，而这组串口常用于与计算机进行通信，如果还想连接其他串口设备，则会影响其与计算机的正常通信。此时，可以使用软件模拟串口。软串口是由程序模拟生成的，使用起来不如硬串口稳定，并且与硬串口一样，波特率越高越不稳定。

注意：在ArduinoUNO上所有的引脚都可以设置为RX和TX引脚，但是其他型号的Arduino板上只有特定引脚可以设置为RX引脚。

#include<SoftwareSerial.h> //导入软串口库文件

SoftwareSerial mySerial = SoftwareSerial(rxPin,txPin);//实例化软串口对象，并指定软串口的RX和TX引脚

rxPin：表示软串口接收引脚

txPin：表示软串口发送引脚

mySerial.begin(9600);//初始化软串口，并设置波特率

mySerial.available(); //判断软串口缓冲器是否有数据装入

mySerial.read();    //读取软串口数据

mySerial.readString();  //读取软串口数据，以字符串形式读取

mySerial.write(); //软串口输出数据，以字节形式输出到串口

mySerial.print();    //软串口输出数据，不包括换行，数据以ASCII码形式输出

mySerial.println();   //软串口输出数据，包括换行，数据以ASCII码形式输出

3.编程任务

(1)利用手机App通过蓝牙控制LED的亮灭。

当通过App发送指令给蓝牙模块时，可以控制Arduino端的LED灯亮灭状态。

注意：Arduino端要求使用软串口接收和发送数据。

#include<SoftwareSerial.h>

//实例化软串口对象，括号中参数1是RX引脚号，参数2是TX引脚号

SoftwareSerial mySerial = SoftwareSerial(10, 11);

char val;

void setup()

{

mySerial.begin(115200); //设置软串口的波特率

pinMode(13, OUTPUT);

}

void loop()

{

val = mySerial.read();//读取蓝牙串口发送来的数据

if (val == 'n')

{

digitalWrite(13, HIGH);

mySerial.println("LED ON!");

}

else if (val == 'f') {

digitalWrite(13, LOW);

mySerial.println("LED OFF!");

}

}

(2) 修改蓝牙遥控小车，使用软串口实现

#include<SoftwareSerial.h>

SoftwareSerial mySerial = SoftwareSerial(10, 11);

#define IN1 2

#define IN2 4

#define ENA 3

#define IN3 7

#define IN4 6

#define ENB 5

char val;

void setup() {

pinMode(IN3, OUTPUT);

pinMode(IN4, OUTPUT);

pinMode(ENB, OUTPUT);

pinMode(IN1, OUTPUT);

pinMode(IN2, OUTPUT);

pinMode(ENA, OUTPUT);

mySerial.begin(115200);

delay(1000);

}

int a;

void loop() {

val = mySerial.read();

if (val == 'f')

{

mForward(120, 120);

mySerial.println("forward!");

} else if (val == 'b') {

mBackward(120, 120);

mySerial.println("backward!");

} else if (val == 'l') {

mLeft(120, 120);

mySerial.println("turnLeft!");

} else if (val == 'r') {

mRight(120, 120);

mySerial.println("turnRight!");

} else if (val == 's') {

mStop(0, 0);

mySerial.println("stop!");

}

delay(10);

}

void mForward(int spl, int spr) {

//右电机正转

digitalWrite(IN1, HIGH);

digitalWrite(IN2, LOW);

analogWrite(ENA, spr);

//左电机正转

digitalWrite(IN3, HIGH);

digitalWrite(IN4, LOW);

analogWrite(ENB, spl);

}

void mBackward(int spl, int spr) {

//右电机反转

digitalWrite(IN1, LOW);

digitalWrite(IN2, HIGH);

analogWrite(ENA, spr);

//左电机反转

digitalWrite(IN3, LOW);

digitalWrite(IN4, HIGH);

analogWrite(ENB, spl);

}

void mLeft(int spl, int spr) {

//右电机正转

digitalWrite(IN1, HIGH);

digitalWrite(IN2, LOW);

analogWrite(ENA, spr);

//左电机反转

digitalWrite(IN3, LOW);

digitalWrite(IN4, HIGH);

analogWrite(ENB, spl);

}

void mRight(int spl, int spr) {

//右电机反转

digitalWrite(IN1, LOW);

digitalWrite(IN2, HIGH);

analogWrite(ENA, spr);

//左电机正转

digitalWrite(IN3, HIGH);

digitalWrite(IN4, LOW);

analogWrite(ENB, spl);

}

void mStop(int spl, int spr) {

//右电机停止

digitalWrite(IN1, LOW);

digitalWrite(IN2, LOW);

analogWrite(ENA, spr);

//左上角电机停止

digitalWrite(IN3, LOW);

digitalWrite(IN4, LOW);

analogWrite(ENB, spl);

}

4.AT指令

AT指令是用于终端设备与PC应用之间的连接与通信的指令。AT 即Attention。每个AT命令行中只能包含一条AT指令；对于AT指令的发送，除AT两个字符外，最多可以接收1056个字符的长度（包括最后的空字符）。

实际上通过AT指令我们可以设置蓝牙模块的名称、配对密码、工作模式等。

注：蓝牙模块有很多种类，由于蓝牙版本和生产厂商不同，其进入AT指令的方式也不同。我们现在用的BT05低功耗蓝牙模块，发送AT指令时要有回车换行符。





进入AT指令的方法：使用官方提供的串口助手软件或者编写程序，使用Arduino提供的串口助手，本节课使用Arduino提供的串口监视器。

(1)编写程序，注意BT05蓝牙模块的默认波特率是115200，所以程序中将软串口的波特率设置为115200，与蓝牙模块相同。

(2)先将蓝牙模块连接到Arduino板上，其中蓝牙的TXD引脚连接到10号引脚，蓝牙的RXD引脚连接到11号引脚。只要没有其他蓝牙设备与该蓝牙模块配对连接，则该蓝牙模块的状态就是AT指令模式。

(3)将程序下载到Arduino板，然后打开串口监视器，在串口监视器中设置好波特率以及回车和换行符，在发送框发送相应的AT指令即可。

示例代码：

#include<SoftwareSerial.h>

SoftwareSerial mySerial = SoftwareSerial(10, 11);

//实例化软串口对象，括号中参数1是RX引脚号，参数2是TX引脚号

char v;

String s;

void setup() {

Serial.begin(9600);//设置硬串口波特率

mySerial.begin(115200);//设置软串口的波特率

}

void loop() {

if (Serial.available() > 0) //判断串口监视器是否有数据发送

{

v = Serial.read();//读取串口监视器发送的数据

mySerial.write(v);//把数据再发送给蓝牙模块

}

while(mySerial.available()>0) //判断蓝牙模块是否发送了数据

{

s=mySerial.readString();//以字符串形式读取蓝牙模块发送的数据

Serial.println(s);//把数据再发给串口监视器便于观察

}

}

5.传统蓝牙模块的AT指令(以HC-05为例)





(1)编写程序，注意程序中设置与蓝牙模块通信的串口波特率为38400,然后下载到Arduino中，然后断开电源。

#include<SoftwareSerial.h>

SoftwareSerial softSerial = SoftwareSerial(10, 11);

String s="";

void setup() {

Serial.begin(9600);

softSerial.begin(38400);

}

void loop() {

if (Serial.available() > 0) {

softSerial.write(Serial.read());

}

while (softSerial.available() > 0) {

s = softSerial.readString();

Serial.println(s);

}

}

(2)我们使用的传统蓝牙模块型号是HC-05，其进入AT指令的方法是先按住蓝牙模块上的小按钮，然后供电，当蓝牙模块的指示灯慢速闪烁时(亮灭间隔大约1秒)再松开小按钮，此时蓝牙模块进入AT指令模式。

(3)在串口监视器中发送相应的AT指令即可。

注意：使用AT指令修改/查询蓝牙模块的名称时，需要将模块的EN引脚接入3.3v电压，使蓝牙模块进入完全AT模式。只有HC-05的蓝牙模块是这样的，其他蓝牙模块要查数据手册。

6.蓝牙模块的主从机模式

蓝牙模块通信是指两个蓝牙模块或蓝牙设备之间进行通信，进行数据通信的双方一个是主机，一个是从机。

主设备模式：工作在主设备模式,可以与一个从设备进行连接。在此模式下可以对周围设备进行搜索并选择需要连接的从设备进行连接。一个蓝牙设备以主模式发起呼叫时，需要知道对方的蓝牙地址，配对密码等信息，配对完成后，可直接发起呼叫。

从设备模式：工作在从机模式下的蓝牙模块只能被主机搜索，不能主动搜索。从设备跟主机连接以后，也可以和主机设备进行发送和接收数据。