**Arduino课堂笔记**

1.复习蓝牙模块

(1)蓝牙模块接线方式：

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。

(2)软串口通信

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

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.write(); //软串口输出数据，以字节形式输出到串口

(3)传统蓝牙模块进入AT指令的方法

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。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。

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。

(4)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。

2.摇杆模块

双轴摇杆传感器模块采用摇杆电位器制作，具有两轴(x和y)模拟输出，1路按钮(z轴)数字输出。配合Arduino可以制作遥控器等互动作品。



引脚说明：

GND——负极

+5V——正极

VRX——X轴，横向推动摇杆的值，数据范围0~1023，中间位置默认为512，但实际有偏差，需要测试。

VRY——Y轴，纵向推动摇杆的值，数据范围0~1023，中间位置默认为512，但实际有偏差，需要测试。

SW——Z轴，垂直按压摇杆的值，值为0或1

引脚接线与Arduino连接：

GND——Arduino的GND

+5V——Arduino的+5v

VRX——Arduino模拟输入引脚(A0~A5)

VRY——Arduino模拟输入引脚(A0~A5)

SW——Arduino的数字引脚

程序示例：

#define VRX A0

#define VRY A1

#define SW A2

int jox, joy, jos;

void setup() {

pinMode(VRX, INPUT);

pinMode(VRY, INPUT);

pinMode(SW, INPUT);

Serial.begin(9600);

}

void loop() {

jox = analogRead(VRX);//读取x轴的数值，范围为0~1023

joy = analogRead(VRY);//读取y轴的数值，范围为0~1023

jos = digitalRead(SW); //读取按键的数值，为0和1

Serial.print("X=");

Serial.print(jox);

Serial.print(" |Y=");

Serial.print(joy);

Serial.print(" |Z=");

Serial.println(jos);

}

3.蓝牙遥控小车：制作蓝牙遥控小车，使用两个Arduino UNO和两个蓝牙模块，分别作为遥控端和被控端

遥控端的蓝牙模块为主机模式，Arduino读取摇杆的X轴和Y轴的值，并以此作为判断条件通过蓝牙模块将控制小车移动的指令发送出去；

被控端的蓝牙模块为从机模式，接收到遥控端的指令后就通过Arduino控制小车移动。

分析：

(1) 先把AT指令的程序上传到Arduino板上，再分别通过AT指令设置两个蓝牙模块为主机模式和从机模式：

**主机模式的蓝牙模块：**

AT -------------------进入AT指令,成功则返回OK

AT+NAME=HC05M ----------设置蓝牙模块的名字为HC05M，成功则返回OK

AT+UART=9600,0,0 ----------设置蓝牙模块的波特率和校验位等，成功则返回OK

AT+PSWD=1212 ------------设置蓝牙模块的配对密码，成功则返回OK

AT+ROLE=1 ------------设置蓝牙模块的主从机模式，主机为1，成功则返回OK

AT+CMODE=1 ------------设置蓝牙模块的连接模式为任意蓝牙地址连接模式，成功则返回OK。

**从机模式的蓝牙模块：**

AT -------------------进入AT指令,成功则返回OK

AT+NAME=HC05S ----------设置蓝牙模块的名字为HC05S，成功则返回OK

AT+UART=9600,0,0 ----------设置蓝牙模块的波特率和校验位等，成功则返回OK

AT+PSWD=1212 ------------设置蓝牙模块的配对密码，成功则返回OK

AT+ROLE=0 ------------设置蓝牙模块的主从机模式，从机为0，成功则返回OK

AT+CMODE=1 ------------设置蓝牙模块的连接模式为任意蓝牙地址连接模式，成功则返回OK。

(2)编写遥控端的程序，先通过判断摇杆的值确定前进和停止的指令

(3)编写被控端的程序，根据蓝牙模块接收到的指令控制小车前进和停止。

(4)确定以上两步没有问题再补全遥控小车前进、后退、左转、右转等指令

**遥控端程序示例(示例中的摇杆数值为实际测得，摇杆不同，数值有所不同)**：

注：摇杆的中间值分别为x=525和y=528，为了让摇杆在中间位置的时候更加稳定(避免数值跳变)，所以在判断时把摇杆的值各加了20，各减了20，在505~545和508~548之间时，判定为摇杆没有被推动，小车处于停止状态。

#include<SoftwareSerial.h>

#define VRX A0

#define VRY A1

SoftwareSerial softSerial = SoftwareSerial(10, 11);

char s;

int jox, joy, jos;

void setup() {

pinMode(VRX, INPUT);

pinMode(VRY, INPUT);

softSerial.begin(9600);

}

void loop() {

jox = analogRead(VRX);

joy = analogRead(VRY);

if ((jox > 505 and jox < 545) and joy < 508)

{ //前进

softSerial.write("F");

} else if (jox < 505 and (joy > 508 and joy < 548))

{ //左转

softSerial.write("L");

} else if (jox < 505 and joy < 508)

{ //左前转

softSerial.write("X");

} else if (jox > 545 and (joy > 508 and joy < 548))

{ //右转

softSerial.write("R");

}

} else if (jox > 545 and joy < 508)

{ //右前转

softSerial.write("Y");

} else if (joy > 548)

{ //后退

softSerial.write("B");

} else

{ //停止

softSerial.write("S");

}

delay(300);//延时300毫秒，避免发送指令太频繁

}

**被控端程序示例：**

void mForward(int spl, int spr) {

digitalWrite(IN1, HIGH);

digitalWrite(IN2, LOW);

analogWrite(ENA, spr);

digitalWrite(IN3, HIGH);

digitalWrite(IN4, LOW);

analogWrite(ENB, spl);

}

void mBackward(int spl, int spr) {

digitalWrite(IN1, LOW);

digitalWrite(IN2, HIGH);

analogWrite(ENA, spr);

digitalWrite(IN3, LOW);

digitalWrite(IN4, HIGH);

analogWrite(ENB, spl);

}

void mLeft(int spl, int spr) {

digitalWrite(IN1, HIGH);

digitalWrite(IN2, LOW);

analogWrite(ENA, spr);

digitalWrite(IN3, LOW);

digitalWrite(IN4, HIGH);

analogWrite(ENB, spl);

}

void mRight(int spl, int spr) {

digitalWrite(IN1, LOW);

digitalWrite(IN2, HIGH);

analogWrite(ENA, spr);

digitalWrite(IN3, HIGH);

digitalWrite(IN4, LOW);

analogWrite(ENB, spl);

}

void mStop(int spl, int spr) {

digitalWrite(IN1, LOW);

digitalWrite(IN2, LOW);

analogWrite(ENA, spr);

digitalWrite(IN3, LOW);

digitalWrite(IN4, LOW);

analogWrite(ENB, spl);

}

#include<SoftwareSerial.h>

//示例中小车右侧电机为IN1、IN2和ENA控制

//小车左侧电机为IN3、IN4、ENB控制

#define IN1 2

#define IN2 4

#define ENA 3

#define IN3 6

#define IN4 7

#define ENB 5

SoftwareSerial softSerial = SoftwareSerial(A0, A1);

char s;

void setup() {

pinMode(IN3, OUTPUT);

pinMode(IN4, OUTPUT);

pinMode(ENB, OUTPUT);

pinMode(IN1, OUTPUT);

pinMode(IN2, OUTPUT);

pinMode(ENA, OUTPUT);

softSerial.begin(9600);

}

void loop() {

if (softSerial.available() > 0) {

s = softSerial.read();

}

if (s == 'F')

{

mForward(120, 120);

//Serial.println("forward!");

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

mBackward(120, 120);

//Serial.println("backward!");

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

mLeft(120, 120);

//Serial.println("turnLeft!");

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

mRight(120, 120);

//Serial.println("turnRight!");

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

mStop(0, 0);

//Serial.println("stop!");

}else if(s == 'X')

{//左前转

mForward(90, 150);

//Serial.println("forward!");

}else if(s == 'Y')

{//右前转

mForward(150, 90);

//Serial.println("forward!");

}

}