

图1 左转效果图

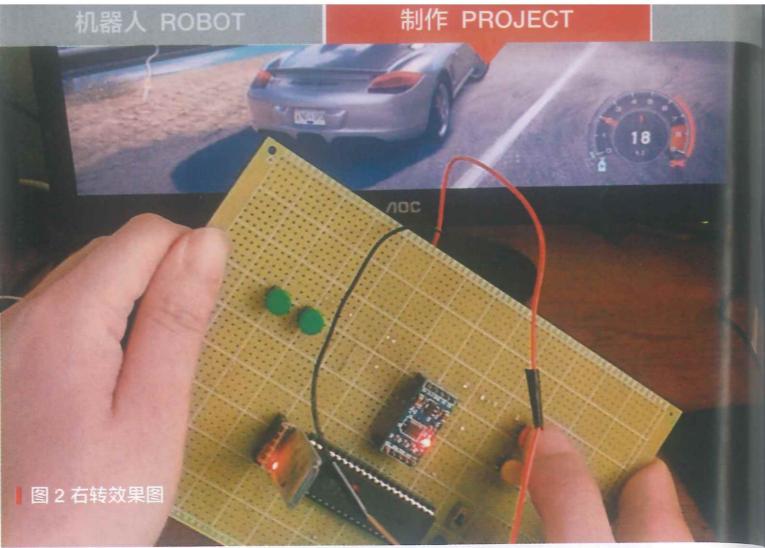


图2 右转效果图

视频观看地址, Chinked-out 工作室优酷视频站:
<http://i.youku.com/chinkedout>



无线方向盘

◇ Chinked-out 工作室

赛车竞速类游戏有着广泛的爱好者群体, 在不同的游戏平台, 游戏厂商们不遗余力地开发赛车类游戏, 较为知名的游戏如 Xbox 360 平台的《极限竞速》系列、Wii 平台下的《马里奥赛车》系列, 以及国内赛车游戏爱好者耳熟能详的 PC 平台《极品飞车》系列。赛车游戏的硬件外设也在不断推陈出新。其中最具代表性的就是 Xbox360 的无线方向盘和 Wii 游戏机的 Wii Remote 控制器, 两种外设均可以通过感应人体动作, 让玩家以近似操控方向盘的形式操控赛车方向, 带来丰富的游戏体验。而在 PC 端, 主流的赛车游戏外设仍

然是游戏手柄或有线方向盘。为尝试给 PC 端赛车游戏爱好者带来无线方向盘的操作体验, 我设计了一款基于 STC12C5A60S2 单片机的无线方向盘, 可实现《极品飞车 14》的游戏操作。

实际效果

游戏中, 赛车将随着“方向盘”的左转或右转进行转向, 如图 1、图 2 所示。同时, 在方向盘上配 5 个按钮, 如图 3 所示。其中, 右侧 3 个按钮功能分别是: 油门(红色)、氮气(黄色)、刹车(黑色)。左侧两个按钮为左、右方向控制, 与手势

方向并无冲突。

硬件框架

本设计由发送端(方向盘)、接收端、无线传输三大模块组成, 其中发送端的核心为 ADXL345 加速度的传感器, 接收端的核心为 CH375B 模块(或 CH376S)。单片机对加速度传感器数据读取、处理后, 不断向接收端发送控制字符, 接收端实际上是一个键盘, 收到的控制字符就是游戏中对应的键盘按键值, 根据收到的按键值操作游戏中的赛车。硬件框架如图 4 所示。

原理图分为发送端电路和接收端电路,

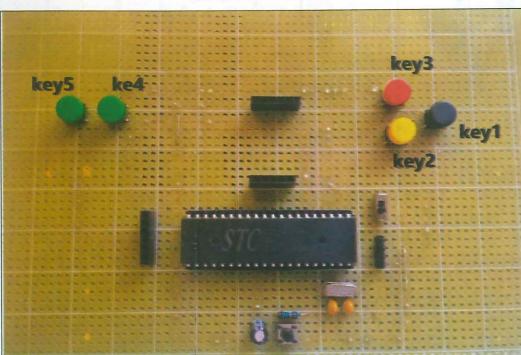


图3 按键布局图

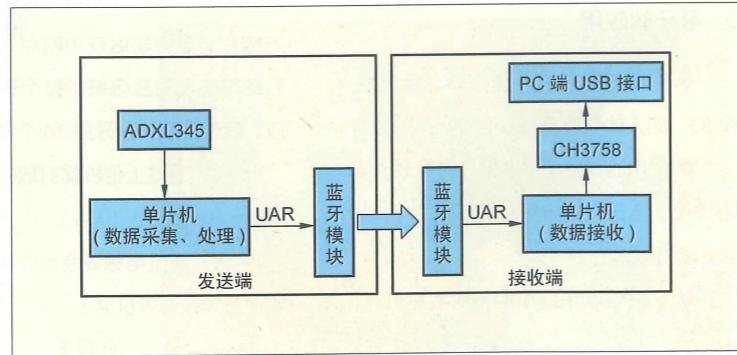


图4 无线方向盘系统框架图

表1 元器件材料清单

名称	规格	数量
万能板	9cm × 15cm	2
单片机: STC12C5A60S2	DIP40	2
40Pin 单片机母座		2
晶体振荡器	11.0592MHz	2
陶瓷电容	22pF	4
电解电容	22 μF	2
电阻	10kΩ	2
拨动开关		2
排针	1 组 40Pin	1
杜邦线	1 组 40Pin	1
6mm × 6mm × 7.3mm 方头按键		7
A44 圆形按键帽(按键帽颜色自选与按键需匹配)		5
5Pin 排针母座(用于固定 ADXL345)		2
4Pin 排针母座(用于固定蓝牙模块)		2
ADXL345 传感器		1
USB 转 TTL 下载器	CH340 系列	1
USB 线	母对母	1
CH376S 模块		1
蓝牙串口模块	HC-05	2

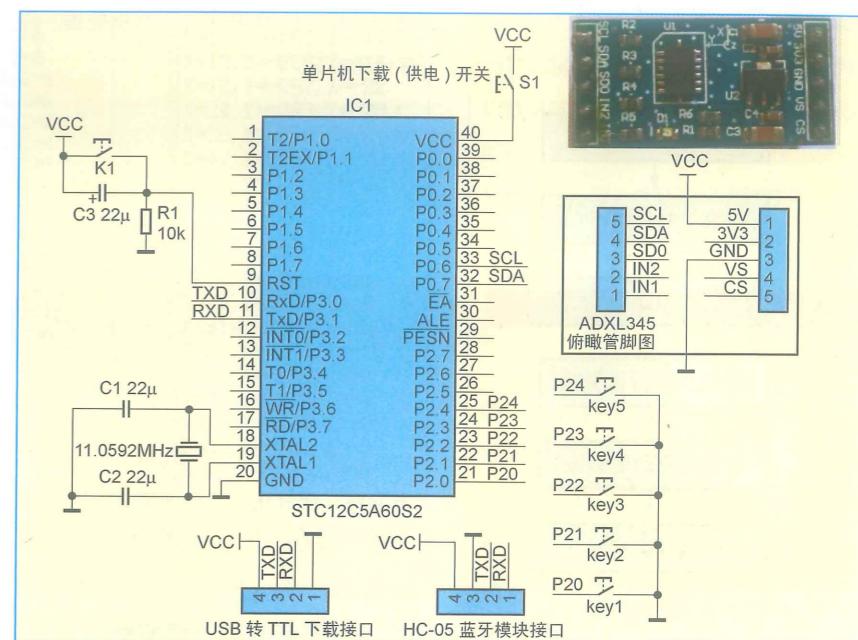


图5 发送端电路图

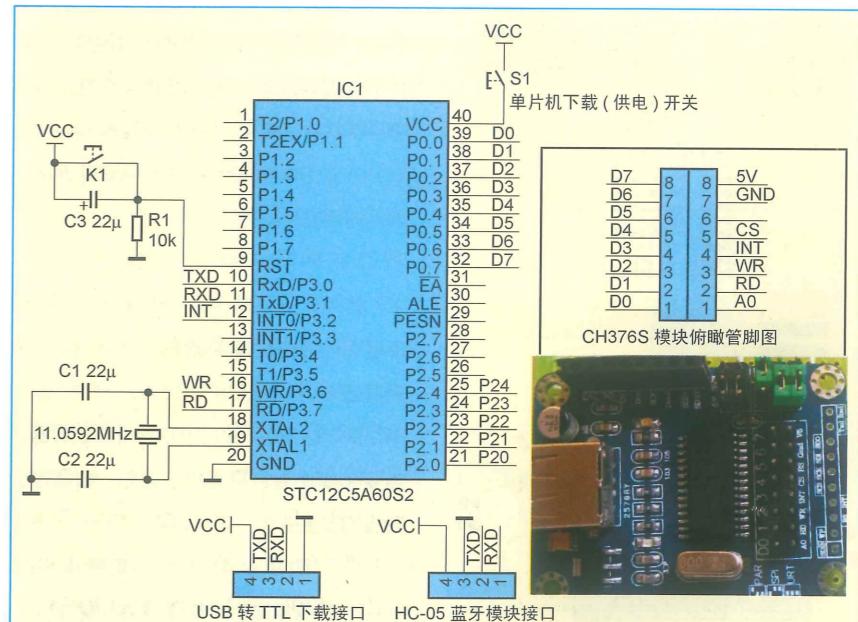


图6 接收端电路图

1. 手势识别原理

方向盘可以识别人手的动作, 依靠的是 ADXL345 加速度传感器模块。ADXL345 是 ADI 公司采用 MEMS 技术具有 SPI 和 I²C 数字输出功能的三轴加速度计。为了降低程序设计难度, 此处使用 ADXL345 的倾角测量功能, 通信采用 I²C 协议。

当人手改变 ADXL345 模块的位置时, 必然影响模块与 X、Y、Z 三轴的倾角数据, 单片机可以将这些数据采集, 但并不知道人手动作与三轴数据的对应关系。这就需要把 ADXL345 模块的数据读取出来, 再根据与人手动作的对应关系确定程序算法。那么如何观察 ADXL345 模块的实时数据呢? 需先设计 ADXL345 模块测试程序。

功能是每间隔 0.3s 将采集到的 ADXL345 模块数据发送到 PC 端的串口助手。

2. ADXL345 测试程序解读(本测试程序源代码见 AXDL345-test.zip)

ADXL345 测试程序流程图如图 7 所示。ADXL345 模块初始化程序、数据读取

程序、单片机串口发送程序可分别参考传感器和单片机官方手册, 本文不再赘述。

程序中, Multiple_Read_AXDL345() 子函数功能为: 连续 6 次读取 ADXL345 模块数据, 读取到的数据存在 BUF[6] 数组中, BUF[0] ~ BUF[5] 依次存储数据内容为:

X 轴高 8 位, X 轴低 8 位, Y 轴高 8 位, Y

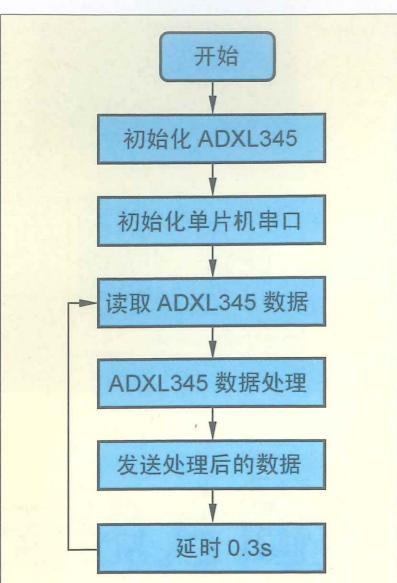


图 7 ADXL345 测试程序流程图

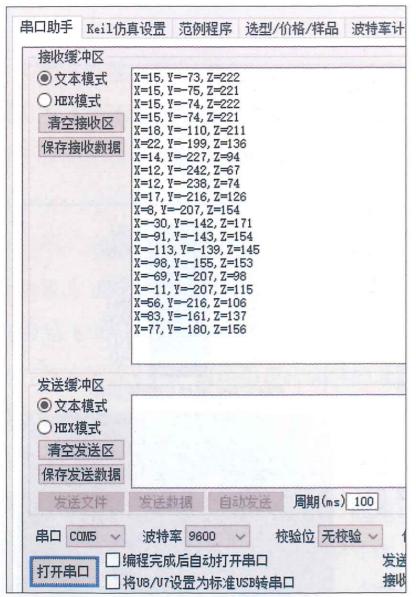


图 8

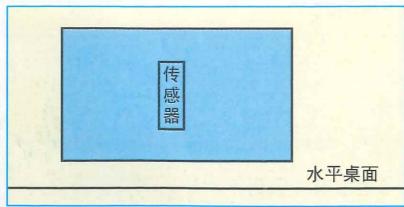


图 9 垂直摆放示意图

轴低 8 位, Z 轴高 8 位, Z 轴低 8 位。此时的三轴数据尽管已经读取到单片机中, 但还需要进一步处理。Data_Process() 子

函数功能为处理 ADXL345 数据处理。程序代码如下。

```

void Data_Process()
{
    X_data=BUF[1]<<8|BUF[0];
    Y_data=BUF[3]<<8|BUF[2];
    Z_data=BUF[5]<<8|BUF[4];
    sprintf(Sbuf, "X=%d, Y=%d, Z=%d\n", X_
    data, Y_data, Z_data);
}
  
```

X、Y、Z 三轴数据既有大小, 也有方向性(通过正负区分方向), 因此 X_data、Y_data、Z_data 3 个变量为有符号整型变量, Data_Process() 函数中, 先将每个轴的高 8 位和低 8 位数据合并为一个 16 位数据, 合并后的数据为二进制补码格式。若将这些数据直接发送到串口通过 PC 显示, 得到的是补码格式的 16 进制数据,

并不能直观观察数据。因此, 需要将数据进行字符串格式化, 这里使用 sprintf 函数。X、Y、Z 三轴数据经 sprintf 函数处理后, 得到的数据格式为:

X=“ ”, Y=“ ”, Z=“ ”

其中“ ” 内容为十进制数据, 字符串格式化后的数据存储在 Sbuf 数组中, 最后通过单片机 UART 串口发送到 PC 端的串口助手软件后, 即可观察到十进制数据, 当变化模块与水平面的位置时, 数据也会对应的发生变化。效果如图 10 所示。

串口助手参数设定: 波特率选择 9600, 接收数据格式选择“文本模式”。

3. ADXL345 模块数据分析

可以直观地观察 ADXL345 模块数据后, 需对数据进行分析, 判断出哪些数据与人手的动作相关联。分析方法如下。

(1) 将焊接好的电路平板垂直于水平桌面摆放, 如图 9 所示。此状态下观察串口助手接收到的数据, 如图 10 所示。

在此测试过程中, 考虑人手操作及环

境误差, X 轴数据变化范围为 -10 ~ 10, Y 轴数据变化范围为 -260 ~ -240, Y 轴数据变化范围为 -20 ~ 20。这里可理解为, 此状态下, X 轴数据接近 0, Y 轴数据接近 -255, Z 轴数据接近 0。

(2) 将焊接好的电路平板垂直于水平桌面摆放后, 逐渐向左侧倾斜一定角度。

此状态下观察串口助手接收到的数据, 如图 11 所示。

在模块逐渐向左倾斜的过程中, 可以观察到 X 轴的数据明显发生变化: 由趋近于 0 逐渐向负值变化, 随着向左倾斜角度的增大, 负数值的绝对值也逐渐变大。而 Y 轴、Z 轴的数据变化并不明显。

(3) 将焊接好的电路平板垂直于水平桌面摆放后, 逐渐向右侧倾斜一定角度。

此状态下观察串口助手接收到的数据, 如图 12 所示。

在模块逐渐向右倾斜的过程中, 与左倾斜测试中相似, 可以观察到 X 轴的数据明显发生变化: 由趋近于 0 逐渐向正值变化, 随着向右倾斜角度的增大, 正数值的绝对值也逐渐变大。而 Y 轴、Z 轴的数据变化并不明显。

通过 3 次测试表明, 传感器的左倾斜和右倾斜, 将直接影响到 X 轴的数据变化, 而对其他轴的数据影响并不明显。方向盘的左转和右转实际上就是识别传感器的左倾斜数据和右倾斜数据, 体现在程序中, 就是根据 X 轴的数据变化来做程序处理。

4. 发送端单片机数据处理

无线方向盘除了可以通过手势识别控制游戏赛车, 也可通过按钮对游戏操作, 本设计以游戏《极品飞车 14》为测试模型, 此游戏中, 除了需要左、右方向的控制, 还需要油门、刹车(倒车)和氮气键。单片机将采集到的 ADXL345 传感器数据、按键数据处理后, 发送出游戏可以识别的键盘代码(keycodes), 即可控制游戏了。

```

X=13, Y=-257, Z=-17
X=12, Y=-256, Z=-12
X=12, Y=-257, Z=-13
X=14, Y=-257, Z=-11
X=12, Y=-257, Z=-13
X=13, Y=-257, Z=-11
X=12, Y=-257, Z=-8
X=13, Y=-257, Z=-6
X=12, Y=-257, Z=-11
X=13, Y=-257, Z=-10
X=13, Y=-258, Z=-10
X=13, Y=-257, Z=-7
X=12, Y=-257, Z=-10
X=13, Y=-257, Z=-14
X=13, Y=-257, Z=-12
X=14, Y=-257, Z=-12
X=13, Y=-258, Z=-11
X=13, Y=-257, Z=-13
X=12, Y=-257, Z=-13
X=13, Y=-257, Z=-13
X=12, Y=-257, Z=-13
X=13, Y=-258, Z=-11
X=13, Y=-257, Z=-10
X=13, Y=-257, Z=-7
X=12, Y=-257, Z=-10
X=13, Y=-257, Z=-13
X=12, Y=-257, Z=-13
X=13, Y=-258, Z=-11
X=13, Y=-257, Z=-10
X=14, Y=-258, Z=-9
  
```

图 10 垂直状态下的数据

```

X=12, Y=-258, Z=6
X=3, Y=-256, Z=4
X=16, Y=-256, Z=2
X=25, Y=-254, Z=3
X=29, Y=-254, Z=2
X=32, Y=-255, Z=4
X=36, Y=-254, Z=7
X=44, Y=-251, Z=4
X=45, Y=-251, Z=7
X=58, Y=-246, Z=11
X=63, Y=-245, Z=9
X=65, Y=-244, Z=15
X=70, Y=-250, Z=-12
X=74, Y=-252, Z=-9
X=79, Y=-251, Z=-8
X=81, Y=-249, Z=-9
X=82, Y=-248, Z=-10
X=84, Y=-250, Z=-8
X=87, Y=-248, Z=-7
X=90, Y=-248, Z=-9
X=89, Y=-245, Z=-9
X=92, Y=-246, Z=-5
  
```

图 11 左倾状态下的数据

```

X=13, Y=-257, Z=17
X=12, Y=-258, Z=16
X=12, Y=-257, Z=22
X=29, Y=-263, Z=28
X=38, Y=-256, Z=27
X=47, Y=-255, Z=20
X=55, Y=-255, Z=20
X=62, Y=-252, Z=11
X=65, Y=-253, Z=17
X=70, Y=-250, Z=12
X=74, Y=-252, Z=9
X=78, Y=-239, Z=13
X=76, Y=-240, Z=13
X=77, Y=-241, Z=13
X=78, Y=-238, Z=13
X=76, Y=-240, Z=9
X=74, Y=-243, Z=9
  
```

图 12 右倾状态下的数据

采样、发送周期, 受 8 位单片机限制, 当数据采集、发送周期过快, 将影响到串口数据的有效性。延时也不宜过长, 过长的延时虽然降低了发送周期, 但也使得游戏操控不灵敏。此处建议延时时间为 50 ~ 80ms。

接收端设计

由于发送端发送的数据内容是键盘代码, 接收端的数据自然也是键盘代码, 但键盘代码只能用于键盘外设, 接收端的单片机只能接收数据, 并不能实现键盘功能。那么如何让 PC 把单片机识别为键盘呢? 这里通过 CH376S 模块(或 CH375B 模块)来实现。

表 2 游戏主要操作键与键盘代码对照

游戏功能	对应键盘按键	键盘代码(16 进制)
左转	左方向键	4F
右转	右方向键	50
油门	A	04
刹车(倒车)	Z	1D
氮气	左 Shift	E1

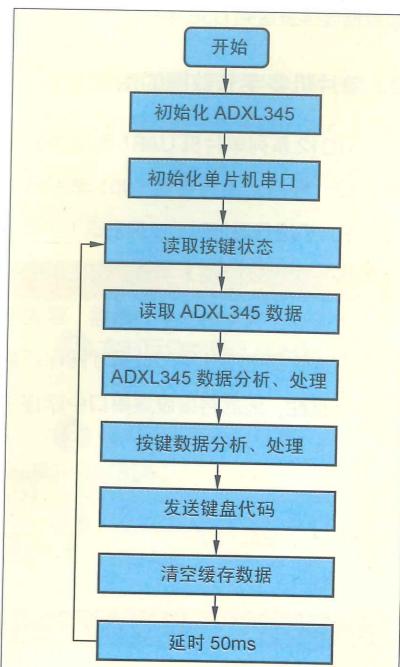


图 13 发送端程序流程图

CH376S 芯片是一款专门连接单片机和 PC、其他外设的芯片，单片机可以通过它读取 U 盘内容、SD 卡内容；或让键盘、鼠标、打印机等外设形式与 PC 进行通信。本设计使用 CH376S 键盘功能，将单片机模拟为 USB 键盘，从而实现游戏控制。

1. CH376S 数据发送格式

CH376S 芯片的初始化代码、功能代码可参考官方手册。这里只介绍 CH376S 的数据格式及接收端的数据接收方式。接收端程序完整程序代码见文件 Receive.zip。

单片机每次向 CH376S 连续发送 8 字节数据，程序中，8 字节数据存储在数组 buf[] 中，在这 8 字节的数据中，前两位是功能代码，后 6 位才是键盘代码，因此必须将接收到的键盘代码存储在数组的后 6 个单元中。接收端单片机一次接收到 5 字节数据，5 字节的数据从 buf[2] 开始存储，存储到 buf[6] 结束。程序中，通过语句

```
"for(i=0;i!=8;i++)buf[i+2]=Recbuf[i];"
```

实现。最后，执行语句 "for(i=0;i!=8;i++)CH375_WR_DAT_PORT(buf[i]);" 将 8 字节数据连续发送到 USB 端口。

2. 单片机多字节数据的接收

STC12 系列单片机 UART 串口为全双工串口，程序上兼容传统 8051 单片机。通常单片机进行通信时一次只发送 1 字节，接收端一次也只接收 1 字节。但在本设计中，发送端一次发送 5 字节数据，接收端也必须接收到全部 5 字节数据才能保证数据的完整性。因此将接收端串口中断程序设计如下。

```
void uart_receive(void) interrupt 4
{
    unsigned char ch;
    ES=0;
    if(RI)
    {
        ch=SBUF;
        Recbuf[Rec_count]=ch;
    }
}
```

```
Rec_count++;
if(Rec_count>4)Rec_count=0;
}
RI=0;
ES=1;
}
```

单片机 UART 串口有数据接收后，向 CPU 请求中断，单片机响应中断后进入该中断程序。每当接收完 1 字节数据后，硬件自动将 RI 置 1（需由软件清零），接收到的数据存储在 SBUF 寄存器中。51 单片机一次只能接收一个 1 字节数据，为使其可以立即接收下 1 字节数据，须将当前接收到的数据保存。程序中，将接收到的数据保存到 Recbuf 数组中，通过变量 Rec_count 自增操作可实现联系存储数据。当 Rec_count>4 时，单片机已经接收到 5 字节数据，完成数据接收任务。最后将 Rec_count 变量清零，为下一次数据接收做好准备。

无线传输端

发送端和接收端使用 UART 串口通信协议，为使其可以进行无线通信，使用满足 UART 通信协议的无线模块即可。市面上较为流行的模块有 Wi-UART 串口模块、Zigbee 模块、蓝牙 UART 模块。这些无线模块尽管使用不同的无线传输方案，但它们只是将 UART 有线传输方式改为无线传输，并不影响用户的 UART 程序。用户只需将一对无线设备进行配对即可使用。这里以市面上常见的 HC-05 主从一体蓝牙模块作简要说明。

通常购买到的 HC-05 蓝牙串口模块出厂设置波特率为 38400，而单片机 UART 串口波特率为 9600，且蓝牙模块必须是一主模式和一从模式相结合，方可配对使用。因此，需对蓝牙串口模块的主从模式和波特率进行配置即可。所需工具和软件为：计算机、串口助手软件、USB 转 TTL 模块

(CH340 系列)。

步骤 1：连接好蓝牙模块和 USB 转 TTL 模块，使蓝牙模块进入到绝对 AT 模式（长按配置按钮进入 AT 模式。不同厂家生产的模块配置方案可能不同，需注意厂家提供的手册说明）。USB 转 TTL 模块连接到 PC 的 USB 端口，PC 安装好 CH340 芯片驱动。

步骤 2：打开串口助手软件，找到计算机识别的虚拟串口号，打开串口，波特率设置为 38400，发送和接收端均使用文本模式。

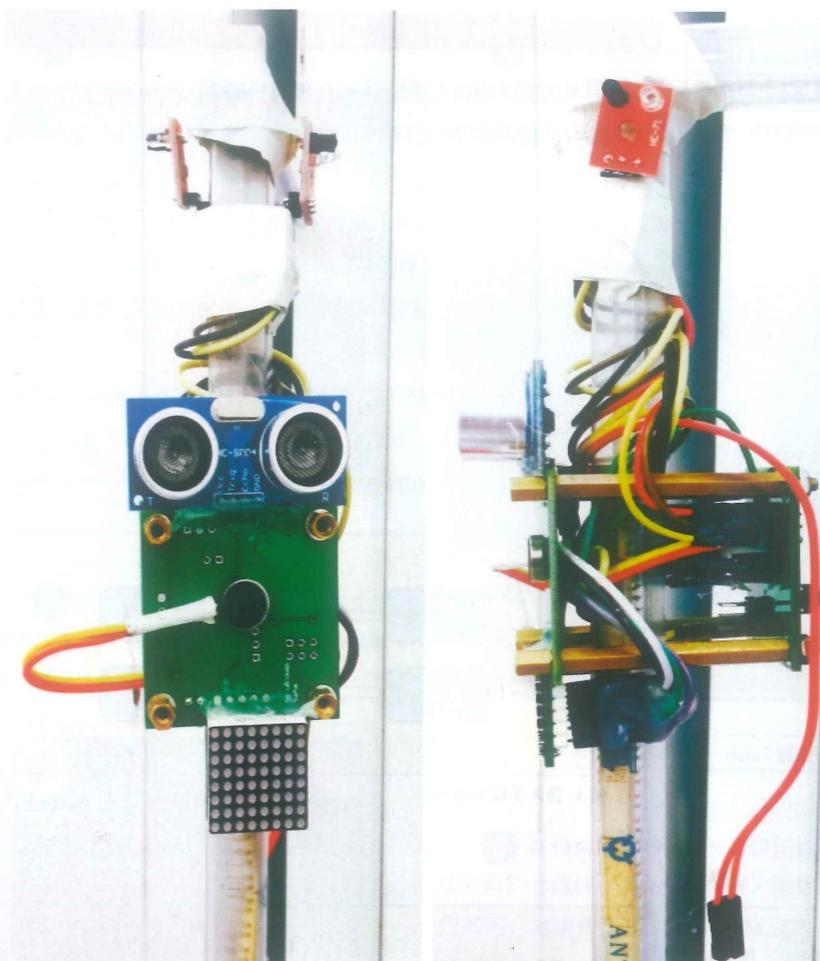
步骤 3：发送端发送 AT 后，接收端返回 OK，表示连接成功。

步骤 4：(1) 发送 "AT+ROLE=0"，返回 OK，代表设置从模式成功。再发送 "AT+UART=9600, 0, 0"。设置模块波特率为 9600，采用 1 位停止位、无奇偶校验位的通信格式。返回 OK 代表设置成功。(2) 用同样的方法，设置第二个蓝牙串口模块为主模式（发送指令为 AT+ROLE=1），通信格式与从模式完全一致。

设置好后，两只蓝牙模块可分别连接到发送端和接收端使用，与单片机连接时，不区分主从模式，可以随意连接。模块 TXD 连接单片机 RXD (Pin30 脚)，模块 RXD 连接单片机 TXD (Pin31 脚)。VCC 和 GND 与系统电路的 VCC、GND 对应相连。

使用中应注意：(1) 当蓝牙模块连接到电路中后，会影响到单片机的程序下载，在下载程序之前，必须移除蓝牙串口模块。

(2) 蓝牙串口模块在上电后会自行配对，在配对过程中，蓝牙模块会自动溢出数据，从而影响到接收端单片机，进而导致单片机向 PC 机发送乱码。为避免这种情况发生，可上电后等待蓝牙模块配对成功，再打开单片机电源开关（原理图中已经是这样设计）。或在已经出现发送乱码的情况下，将单片机复位。☒



制作背景

据统计，全球盲人的百分比约占世界人口的 5%，5% 看似渺小，但总人数并不少，所以盲人的安全问题不容忽视。其中比较集中的问题是盲人过马路以及正常行走，虽然路上铺设有盲人专用道，但大多被人们忽视，有的也已经破损。如果盲人没有及时躲避障碍或者车辆，便会对自身和对方造成难以想象的伤害。基于以上原因，我决定制作一个超声波盲人手杖。

设计思路

考虑到人们对日常手杖的使用习惯，我设计的盲人手杖也采用单手持握的方式，

操作也更加方便。同时考虑到重量、体积、系统稳定性等因素，所以采用了单独的 PCB 设计。这款盲人手杖的目的是为了让路人更容易了解盲人的意图，同时也可以让盲人更快地了解前方行人的位置。

设计方案

此盲人手杖主要具有两个功能：第一是将盲人的意图展现给行人；第二是将使用者前方的位置信息及时反馈回来。为了防止误触，就在手指感应环节采用了红外感应装置，这样可以防止误触，并且方便操作，还能展现给路人一个 8×8 的红色点阵，在光照强烈时依然可以清晰显示，在信息反馈方面，采用了精准度高、抗干扰

基于超声波技术的盲人手杖

◇
杨帆

能力强、传播距离长的超声波模块，它能够实时将前方信息进行采集，当近距离内有障碍时，可以通过迷你电机振动把信息反馈给盲人。

产品功能

① 实时反馈给路人向左转或向右转的信号。

② 超声波采集前方信息，并通过振动模块实时反馈。

③ 无论在黑夜或白天，8×8 的红色点阵都能清晰显示。

产品搭建

涉及的芯片及用料见附表。