

实验3 外部设备基本控制

一、实验目的

1. 巩固编写和调试C语言程序的方法；
2. 掌握外部设备接口电路基本控制方法。

二、实验任务

1. 无源蜂鸣器的检测和控制
2. 小车的结构、检测和控制
3. 红外避障模块的检测及应用

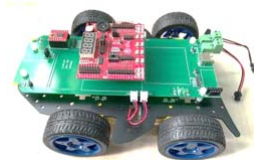
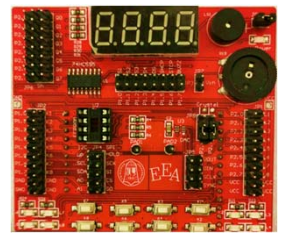
(线上同学用LED代替小车的转动和控制，
用一个按键代替红外避障模块)

三、完成实验作业

提交任务1、2、3程序代码；
任务2小车转动演示视频。



图3-3 无源蜂鸣器



小车



红外避障模块

实验3 实验任务

1. 无源蜂鸣器的检测及控制

- 2) 设计无源蜂鸣器与单片机的连接，编程让无源蜂鸣器发出高、中、低三种不同的声音。
- 3) 加入4个按键控制蜂鸣器的发声：按下K1键，发出高音；K2键发出中音；K3键发出低音；K4键，停止发声。

2. 小车的结构、检测和控制

- 1) 参照小车安装和调试课件的步骤1~5，了解小车的结构,完成小车的转动检测。
- 2) 设计单片机与小车的连线，利用单片机的基本I/O功能，控制小车运动，包括前进、后退、左转、右转等；
- 3) (提高) 加入按键切换小车的转动，比如按下K1键前进；K2后退等，可自行设计按键与小车控制的关系。

3. 红外避障模块检测的应用

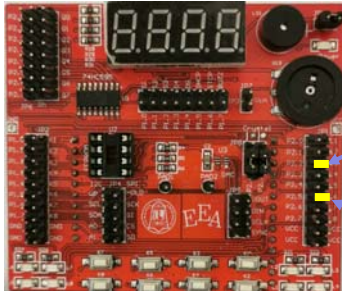
- 1) 了解红外避障模块工作原理，完成避障模块检测；
- 2) 设计避障模块与单片机的连接，当检测到前方有障碍物时，蜂鸣器发出一声警报。
- 3) (提高) 在任务1的基础上，添加检测功能，当小车在转动过程中检测到有障碍物时，停止转动。

1. 无源蜂鸣器的检测及控制

- 1) 蜂鸣器的原理和检测
- 2) 设计无源蜂鸣器与单片机的连接，编程让无源蜂鸣器发出高、中、低三种不同的声音。
- 3) 加入4个按键控制蜂鸣器的发声：
按下 K1键，发出高音；
K2键，发出中音；
K3键，发出低音；
K4键，停止发声。

■ 问题讨论：

如何理解局部变量*i* 定义的不同，
程序执行速度的变化？



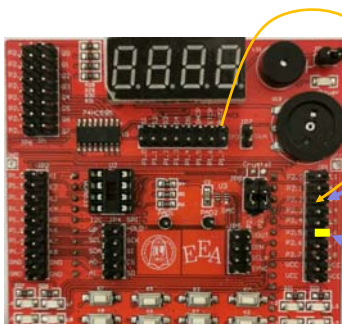
P2.2 与 L3连接

P2.5 与 L6连接

```
//可根据练习的需要，灵活取消或保持各种方式变量i定义前的注释符，
//以便观察不容类型变量对程序执行的影响
#include "msp430.h"
unsigned int j;           //定义全局int型变量
//unsigned long i;        //定义全局long型变量
int main ( void )
{ //unsigned int i;         //定义局部int型变量
  //unsigned long i;        //定义局部long型变量
  BUZZ                    //定义局部long型变量
  //register unsigned int i; //定义寄存器int型局部变量
  //register unsigned long i; //定义寄存器long型局部变量
  WDTCTL = WDTPW + WDTHOLD; //关闭看门狗
  P2SEL &=~(BIT2+BIT5);    //设置引脚P2.2和P2.5为基本输入输出功能
  P2SEL2 &=~(BIT2+BIT5);
  P2OUT |=BIT2+BIT5;       //设置引脚P2.2和P2.5输出的初值为1
  P2DIR |=BIT2+BIT5;       //设置端口P2.2和P2.5为输出方向
  for (;;) //主循环
  { P2OUT ^=(BIT2+BIT5);    //将P2.2和P2.5的值取反后输出
    for (i=0xFFFF; i>0; i--); //延时
  };
}
```

■ 问题讨论：

如何理解局部变量*i* 定义的不同，
程序执行速度的变化？



P2.2 与 Buzz连接

P2.5 与 L6连接

```
//可根据练习的需要，灵活取消或保持各种方式变量i定义前的注释符，
//以便观察不容类型变量对程序执行的影响
#include "msp430.h"
unsigned int j;           //定义全局int型变量
//unsigned long i;        //定义全局long型变量
int main ( void )
{ //unsigned int i;         //定义局部int型变量
  //unsigned long i;        //定义局部long型变量
  BUZZ                    //定义局部long型变量
  //register unsigned int i; //定义寄存器int型局部变量
  //register unsigned long i; //定义寄存器long型局部变量
  WDTCTL = WDTPW + WDTHOLD; //关闭看门狗
  P2SEL &=~(BIT2+BIT5);    //设置引脚P2.2和P2.5为基本输入输出功能
  P2SEL2 &=~(BIT2+BIT5);
  P2OUT |=BIT2+BIT5;       //设置引脚P2.2和P2.5输出的初值为1
  P2DIR |=BIT2+BIT5;       //设置端口P2.2和P2.5为输出方向
  for (;;) //主循环
  { P2OUT ^=(BIT2+BIT5);    //将P2.2和P2.5的值取反后输出
    for (i=0xFFFF; i>0; i--); //延时
  };
}
```

有关蜂鸣器

- 有源蜂鸣器和无源蜂鸣器
- 有源蜂鸣器，**内部有振荡源**，在它两个引脚接上电压就会发出固定频率的声音；
- 无源蜂鸣器，内部没有振荡源，接上电源后不会发声，使用时需要用**不同频率的方波信号来驱动使其发声**。
- 改变输入的方波信号频率，可以改变蜂鸣器发出声音的音调高低；输出的方波个数控制发声的音长。
- 调节音调和音长，可以使蜂鸣器发出不同旋律的音乐。

C大调各音符参考频率

唱名	简谱	C调(Hz)
do	1	262
re	2	294
mi	3	330
fa	4	349
sol	5	392
la	6	440
si	7	494
do(高)	1 [•]	523

任务1 无源蜂鸣器的检测和控制

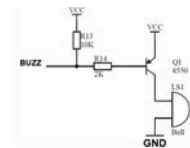
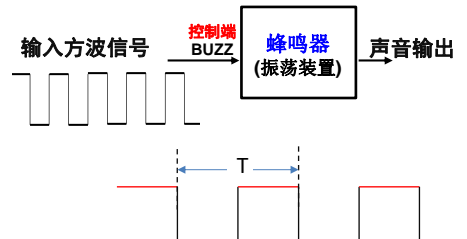


图3-3 无源蜂鸣器



周期 T : 1个方波所占时间

频率 f : 1秒内的方波(脉冲)个数

周期 $T = 1/f$

占空比: 高电平在一个周期中的比例

无源蜂鸣器的控制及应用

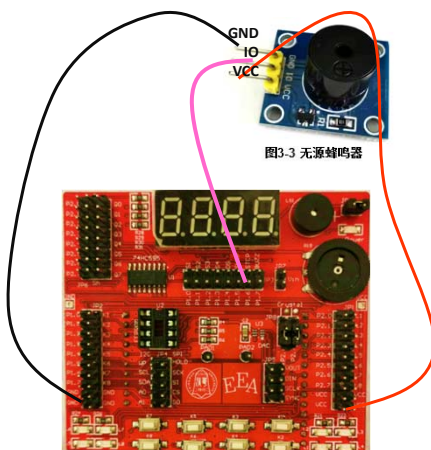
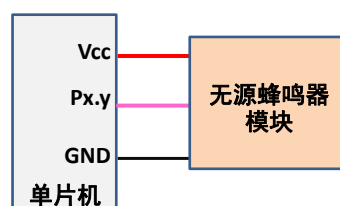


图3-3 无源蜂鸣器



步骤

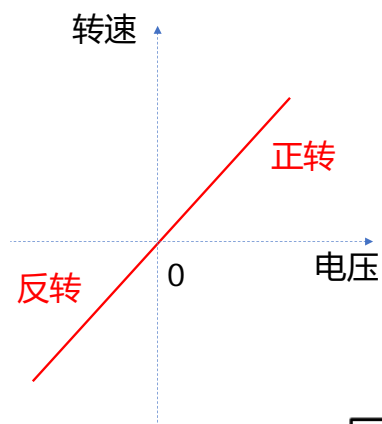
- 掌握外部设备(电路)原理
- 检测外部设备
- 设计与单片机的连接
- 编写程序
- 调试程序

程序的编写

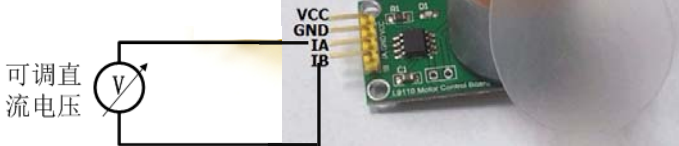
- 对用到的引脚进行相应的初始化设置
- 主循环中完成控制功能

```
#include "msp430.h" //包含头文件
//全局变量定义
int main( void )
{
    //局部变量定义
    WDTCTL = WDTPW + WDTHOLD; //关闭看门狗
    初始化部分; //对用到的模块进行初始设置
    while(1) //主循环部分
    {
        MCU实时处理部分; //主循环部分
    }
}
```

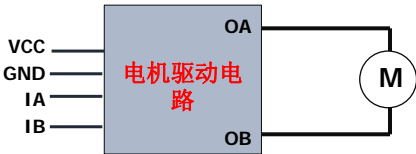
直流电机的调速调向



调节直流电机：
直流电压大小：调速
电流方向：调向



直流电机及其驱动电路

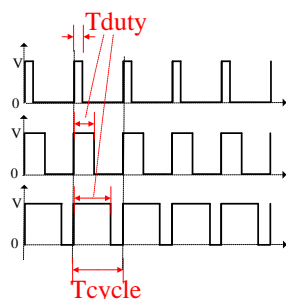


调节直流电机：
直流电压大小：调速
电流方向：调向

IA	IB	方向	调速
PWM	0	正转	PWM的占空比
0	PWM	反转	PWM的占空比



脉冲宽度调制PWM: Pulse Width Modulation



等效直流电压(周期平均)为: $V_o = \frac{T_{duty}}{T_{cycle}} V$

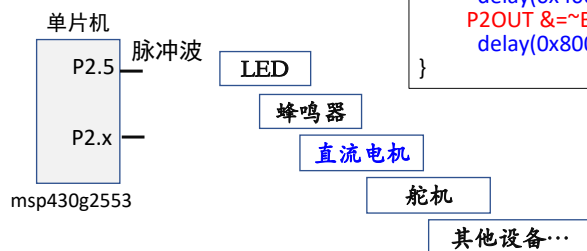
Tcycle: 脉冲周期

Tduty: 高电平所占时间

占空比: 高电平占一周期的比例

编程通过P2.5输出脉冲波:

```
for (i=0; i<5000; i++)
{
    P2OUT |=BIT5; //置1
    delay(0x4000);
    P2OUT &=~BIT5; //置0
    delay(0x8000);
}
```



思考: 输出的波形

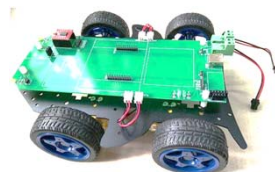
- 占空比是多少?
- 周期怎么改?
- 占空比怎么改?
- 脉冲的个数怎么改?
- 电流方向怎么改?

小车安装和调试

一、小车套件简介

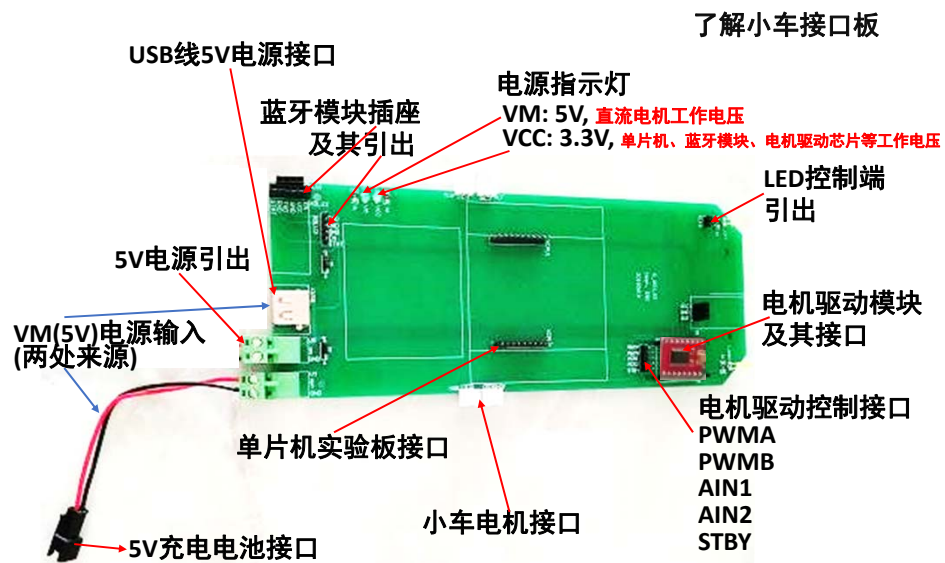
二、基本安装与调试步骤

- 步骤1: 在4个电机上分别安装固定件
- 步骤2: 将电机安装到底板上
- 步骤3: 将接口板和单片机板安装到底板上
- 步骤4: 与电脑连接进行调试
- 步骤5: 检测电机工作状态
- 步骤6: (提高)遥控小车设计与调试
- 步骤7: (提高)无线小车接线



(大家领到的小车已安装完毕,
步骤1~3供大家了解)

步骤3: 将接口板和单片机板安装到底板上



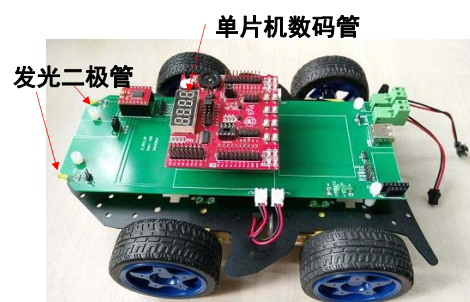
步骤3: 将接口板和单片机板安装到底板上(续)

① 拧上4个固定塑料螺柱

② 安装4个电机电源线



③ 插上单片机实验板



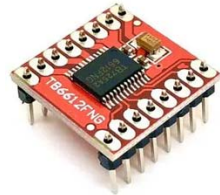
注意插接方向:

单片机数码管和驱动板的2个发光二极管同侧

TB6612电机驱动模块小板

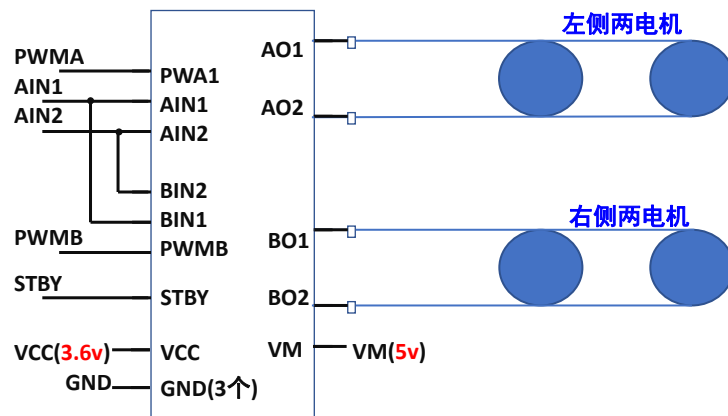
VM	1	VM	PWM	16	PWMA
VCC	2	VCC	AIN2	15	AIN2
GND	3	GND	AIN1	14	AIN1
AO1	4	AO1	STBY	13	STBY
AO2	5	AO2	BIN1	12	BIN1
BO2	6	BO2	BIN2	11	BIN2
BO1	7	BO1	PWMB	10	PWMB
GND	8	GND	GND	9	GND

可以驱动两路电机

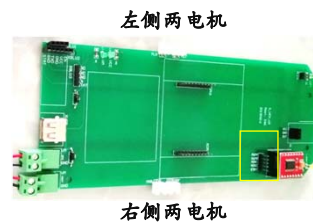
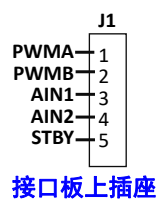


引脚号	引脚名称	引脚功能
1	VM	电机工作电源(5~12V)
2	VCC	芯片工作电源(3.3~5V)
3	GND	芯片地信号
4	AO1	A电机端子1
5	AO2	A电机端子2
6	BO2	A电机端子2
7	BO1	A电机端子1
8	GND	芯片地信号
9	PWMA	A电机速度控制
10	AIN2	A电机方向控制2
11	AIN1	A电机方向控制1
12	STBY	电机暂停控制(A、 B电机共用)
13	BIN1	B电机方向控制1
14	BIN2	B电机方向控制2
15	PWMB	B电机速度控制
16	GND	芯片地信号

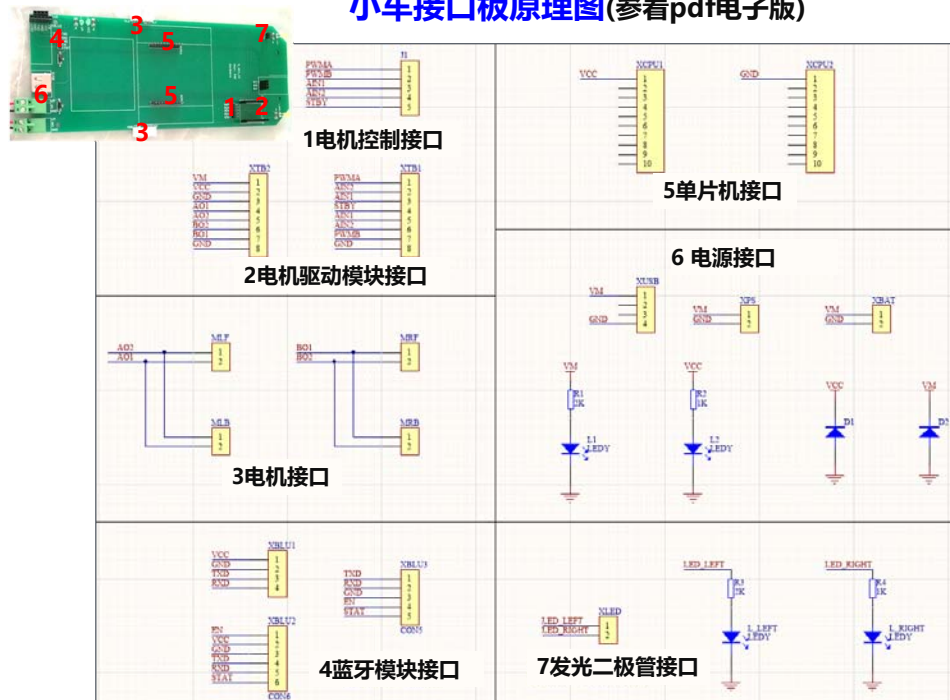
TB6612电机驱动芯片



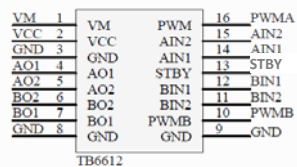
两路电机的方向
由相同的信号AIN1、AIN2控制



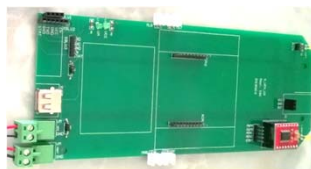
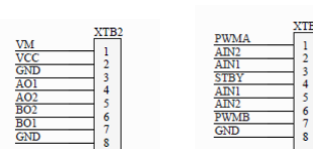
小车接口板原理图(参看pdf电子版)



TB6612电机驱动模块小板



TB6612电机驱动模块插座



注意:

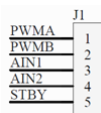
小车控制板设计时, 为简化连线:

- 两个左轮同用AO1、AO2输出驱动
两个右轮同用BO1、BO2输出驱动
故: 两个左轮同用PWMA控制
两个右轮同用PWMB控制
- 两侧轮子方向控制信号用的都是AIN1、AIN2,
故: 两侧轮同进同退, 转弯有转速差控制

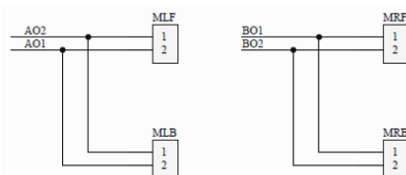
电机控制接口 (需自己设计与单片机的连接)

说明:

- PWMA 左轮转速控制
- PWMB 右轮转速控制
- AIN1、AIN2 两轮转动方向控制
- STBY 待机控制

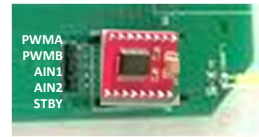
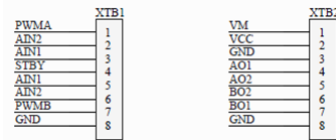


电机接口

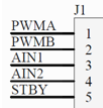


TB6612FNG驱动模块相关信号

■ TB6612FNG驱动模块排针信号



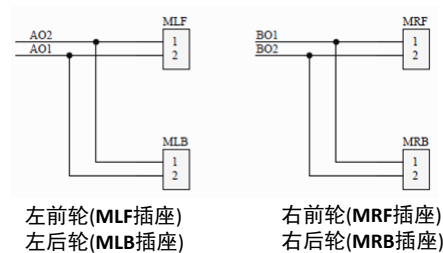
■ 电机控制信号引脚



STBY 待机控制
 PWMA 左轮转速控制
 PWMB 右轮转速控制
 AIN1、AIN2 两轮转动方向控制
 (实验板上左、右轮共用这组信号)

■ 连接电机的输出信号

左轮驱动输出端AO2、AO1，
 右轮驱动输出端BO2、BO1，
 通过转接板连接至电机



TB6612模块与MSP430实验板的连接

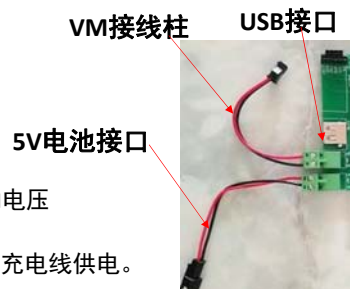
- 左侧两个电机共用A路(以相同方式转动)
 - 电机连接到TB6612模块的AO1和AO2
 - 由PWMA信号控制转速
- 右侧两个电机共用B路(以相同方式转动)
 - 电机连接到TB6612模块的BO1和BO2
 - 由PWMB信号控制转速
- 两路电机的方向控制信号相同(两侧电机转动方向相同)
 - 转接板上，AIN1与BIN1短接， AIN2与BIN2短接；
 - 由AIN1、AIN2控制电机向前、或向后转动；
 - 控制两侧电机转速不同，控制小车左右转向。
 - 如控制一侧电机转动，另一侧电机停转，则小车在原地打转。
- 如不用控制STBY信号，可直接将STBY信号接至实验板上的VCC。



小车内部有两个主要电源

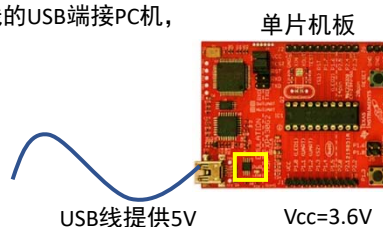
■ VM电源

- 电机工作电源，
- 电压约为5V左右，由转接板上L1灯(VM)指示
- 可由电池供电、电脑USB口、或USB充电器提供VM电压
- 通过VM接线柱引出板上的VM电源
- 因电池续航能力有限，建议调试电机时尽量用USB充电线供电。



■ VCC电源

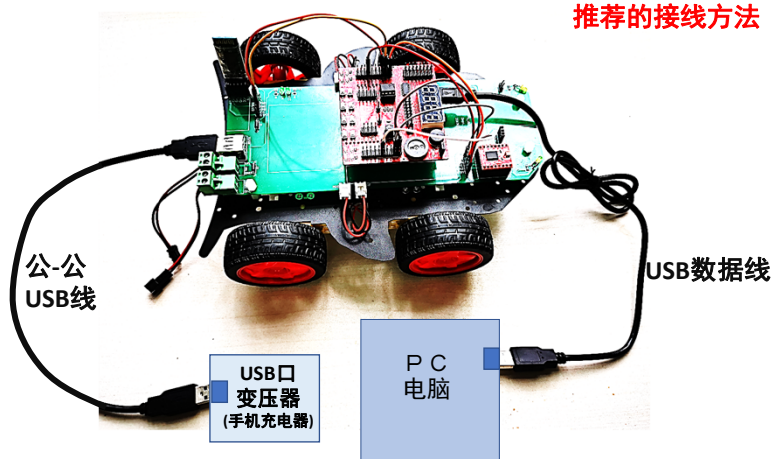
- msp430实验小板的USB线引入5v，经单片机板上电平转化芯片转成Vcc，约3.3V，提供单片机、单片机扩展板、和电机驱动模块的工作电源；
- 由小车接口板上L2灯指示Vcc电源状态；
- 在用CCS对单片机编程调试时，miniUSB数据线的USB端接PC机，由PC机提供单片机工作电源；
- 在小车完成调试，要放置地面转动时，mini数据线的USB端接转接板上USB口，由电池供电单片机实验板的5v电源。



步骤4: 小车与电脑连接进行调试(接线方法1)

由变压器给电机供电

推荐的接线方法

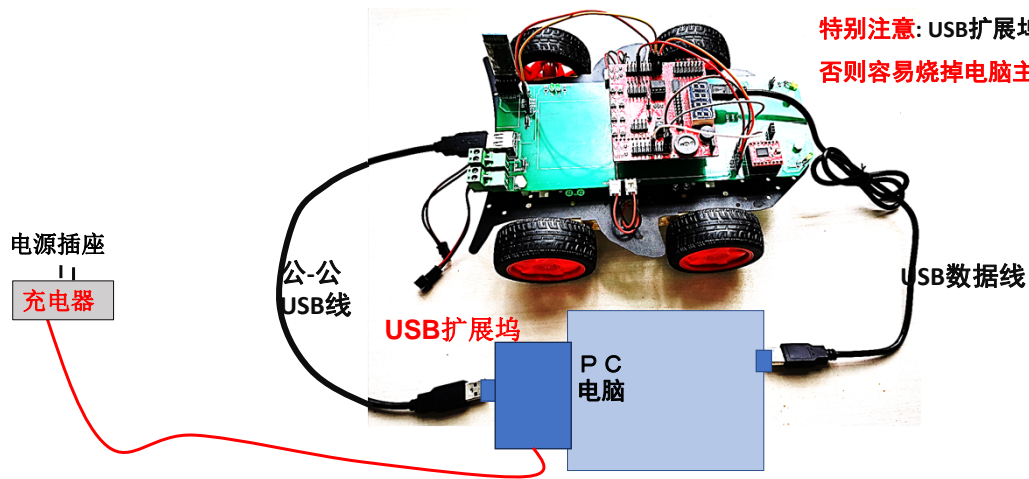


- 由变压器或充电器通过公-公USB线提供电机工作电源VM
- 由电脑USB口通过USB数据线提供单片机工作电源VCC

步骤4: 小车与电脑连接进行调试 (接线方法2)

由USB扩展坞给电机供电

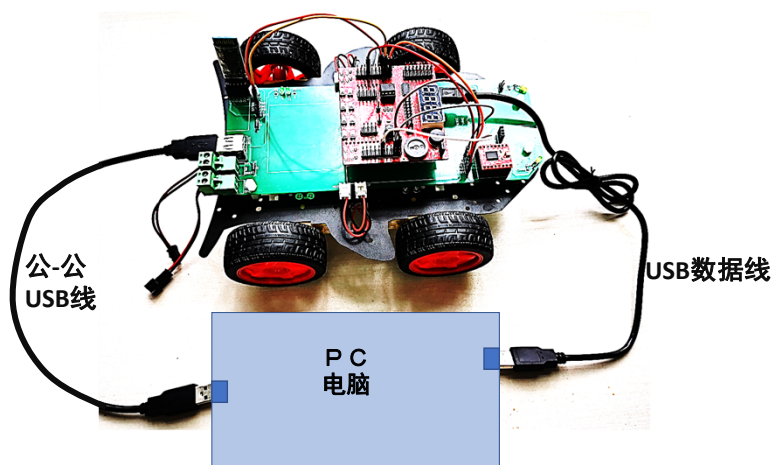
特别注意: USB扩展坞应该供电,
否则容易烧掉电脑主板



- 由电脑USB口通过公-公USB线提供电机工作电源VM
- 由电脑USB口通过USB数据线提供单片机工作用电源VCC

步骤4: 小车与电脑连接进行调试 (接线方法3)

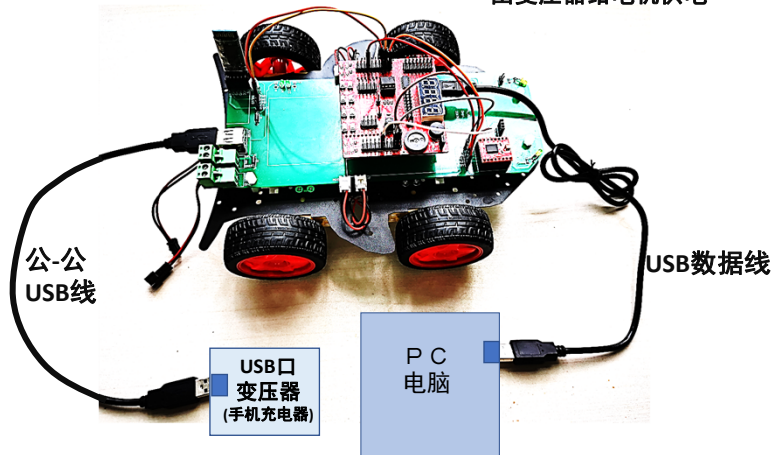
由电脑给电机供电



- 由电脑USB口通过公-公USB线提供电机工作电源VM
- 由电脑USB口通过USB数据线提供单片机工作用电源VCC

步骤4: 小车与电脑连接进行调试(接线方法2)

如果电脑可用的USB口较少
由变压器给电机供电



- 由变压器或充电器通过公-公USB线提供电机工作电源VM
- 由电脑USB口通过USB数据线提供单片机工作电源VCC

步骤5 检测电机工作状态



- 将小车放在实验盒上，置轮胎于空中，以便观察小车电机转动受控状态；
- 参照表1 和表2，了解 and 测试小车工作状态；
- 待测试小测功能均正常后，可进入步骤6

表1 单侧电机(左轮或右轮)的控制

输入引脚				输出引脚(接电机端)		
IN1	IN2	PWM	STBY	OUT1	OUT2	模式
H	H	x	H	L	L	制动
L	H	PWM波	H	L	H	逆时针转 (PWM占空比控制转速)
		L	H	L	L	制动
H	L	PWM波	H	H	L	顺时针转 (PWM占空比控制转速)
		L	H	L	L	制动
L	L	H	H	OFF(高阻抗)		停转
x	x	x	L	OFF(高阻抗)		待机

IN1和IN2同高或同低、或STBY为低, 电机制动、停转或待机

表中： H: 表示高电平
L: 表示低电平
x: 任意，可以是高或低电平
PWM: 频率、占空比可调的方波

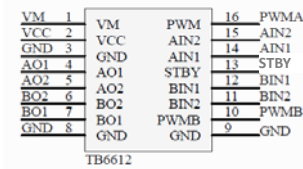
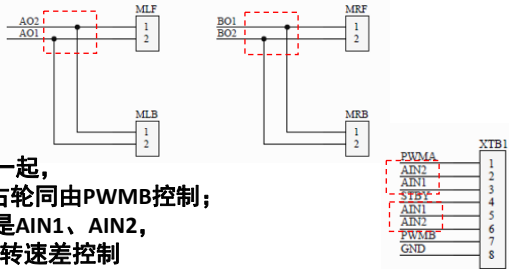


表2 双侧电机(左轮、右轮)的控制关系

STBY	AIN1	AIN2	PWMA	左轮	PWMB	右轮	整体动作
H	H	L	PWM波	顺时针转	PWM波	顺时针转	向前
			PWM波	顺时针转	L	制动	右转
			L	制动	PWM波	顺时针转	左转
			L	制动	L	制动	停
	L	H	PWM波	逆时针转	PWM波	逆时针转	后退
			PWM波	逆时针转	L	制动	左转
			L	制动	PWM波	逆时针转	右转
			L	制动	L	制动	停

■ STBY为H、且AIN1和AIN2不同时，两侧电机转动受各自的PWM控制



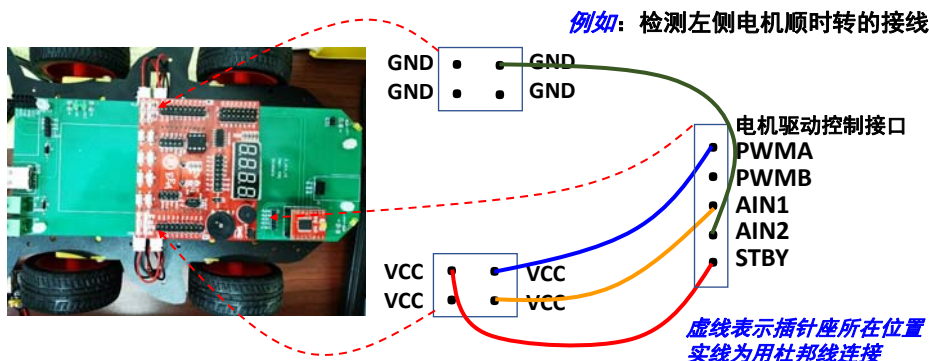
注意，从小车接口板原理图可知：

- 两侧左轮、两侧右轮分别各自连在了一起，故：两个左轮同由PWMA控制，两个右轮同由PWMB控制；
- 两侧左、右轮子方向控制信号用的都是AIN1、AIN2，故：左、右两侧轮同进同退，转弯由转速差控制

检测小车电机及电机驱动部分

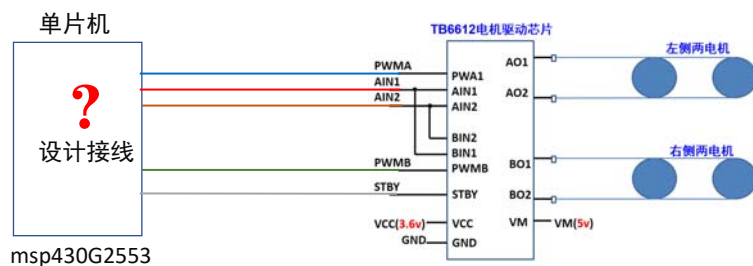
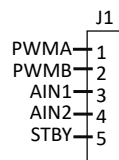
步骤:

- 在步骤4基础上, 按表2设置电机控制接口处引脚STBY、AIN1、AIN2、PWMA、PWMB等信号的电平, 检测小车, 并掌握小车控制方法:
用杜邦线将控制信号根据需要, H电平连到VCC; L电平连到GND, 观察左、右两侧电机状态。
其中, 当PWM=VCC时, 相当于PWM波占空比为100%;
当PWM=GND时, 相当于PWM波占空比为0%。
- 弄清小车动作规律后, 可编程控制PWM波占空比大小, 实现对小车电机转动速度做控制。



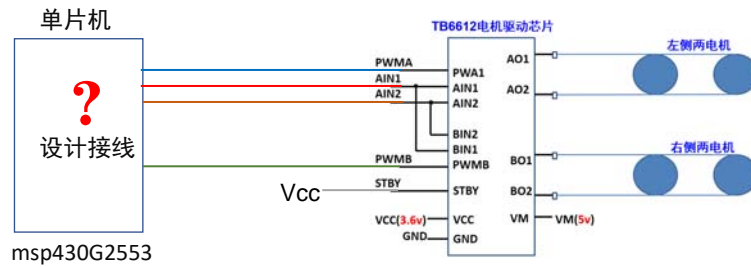
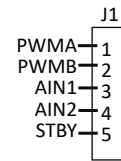
实验任务2(续)

- 设计单片机与小车的连线, 利用单片机的基本I/O功能, 编程控制小车运动, 包括前进、后退、左转、右转等;
(暂时不用调速, 成功了再加上高、低两档速度的变化)
- (提高)** 加入按键切换小车的转动,
比如按下K1键前进; K2后退等, 可自行设计按键控制小车的关系。



实验任务2(续)

- 2) 设计单片机与小车的连线，利用单片机的基本I/O功能，编程控制小车运动，包括前进、后退、左转、右转等；(暂时不用调速，成功了再加上高、低两档速度的变化)
- 3) **(提高)** 加入按键切换小车的转动，比如按下K1键前进；K2后退等，可自行设计按键控制小车的关系。

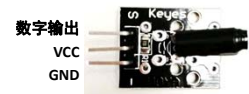
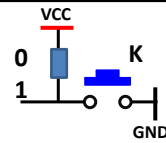


调试经验：

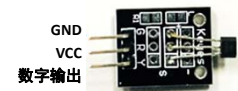
1. 控制电机的频率低时，比如10Hz以下，此时占空比应大于90%，否则电机转动不匀速，造成转一阵，停一阵的感觉；
2. 调试过程中，暂停程序查找原因，出现电机高速旋转时，可手动将STBY引脚信号抽出连到GND上，让电机停转；待取消暂停，继续运行程序时，再恢复STBY引脚的连接。

输出1位数字信号量的传感器

基本原理是当**外部环境变化**时，
比如**磁力大小、光线亮度、噪声大小**等发生变化，
使**传感器模块产生1位数字电压输出量高、低变化**，
通过这个数字电压量的检测，判断外部事件/环境的状态。

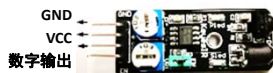
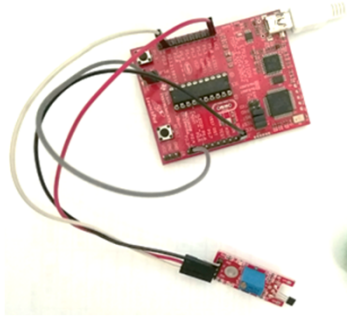
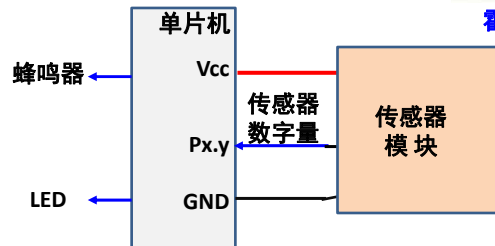


震动、敲击、倾斜



霍尔磁力

单片机控制系统



红外避障



人体红外



人体触摸

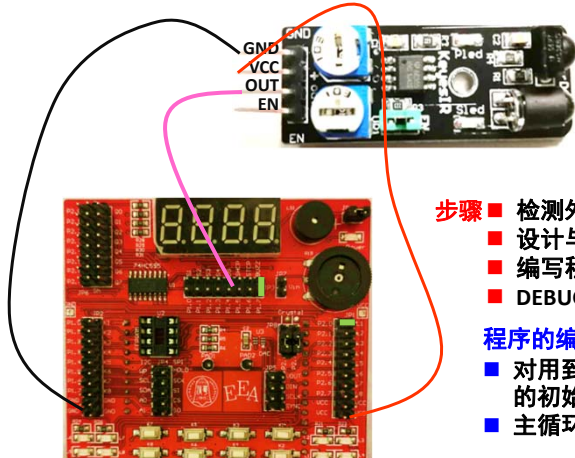
任务3 红外避障模块的检测及应用

- 1) 了解红外避障模块工作原理，
完成避障模块检测；



- 本模块有四个引脚，使用时用到其中的VCC、GND和OUT。EN不用，悬空即可。模块工作电压为3.3V~5V。当模块接好电源，板上Pled指示灯亮。
- 模块具有一对红外线发射与接收管，发射管发射出一定频率的红外线，当前方一定距离有障碍物时，红外线反射回来被接收管接收，此时板上指示灯Sled亮起，信号经过电路处理后，由OUT引脚输出一个低电平数字信号；无障碍物时，OUT引脚输出高电平。
- 通过检测OUT引脚电平状态，可以判断前方有无障碍物。
- 可通过电位器旋钮调节检测距离，有效距离可以在2~30cm。

任务3 红外避障模块的检测及应用

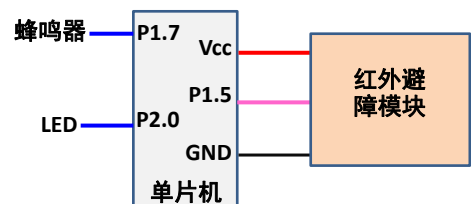


- 步骤**
- 检测外部设备
 - 设计与单片机的连接
 - 编写程序
 - DEBUG下检测外部设备

程序的编写

- 对用到的引脚进行相应的初始化设置
- 主循环中完成控制功能

```
#include "msp430.h" //包含头文件
//全局变量定义
int main( void )
{ //局部变量定义
    WDTCTL = WDTPW + WDTHOLD; //关闭看门狗
    初始化部分; //对用到的模块进行初始设置
    while(1) //主循环部分
    { //主循环部分
        MCU实时处理部分;
    }
}
```



实验任务3(续)

- 2) 设计避障模块与单片机的连接，当检测到前方有障碍物时，控制蜂鸣器发出3声警报。
- 3) (提高) 在任务1的基础上，添加检测功能，当小车在转动过程中检测到有障碍物时，停止转动。

