**第十六章 舵机实验**

**一、实验内容**

基于远程云端硬件实验平台编程实现：控制舵机实验。

**二、实验目的**

学习用KEIL软件编写程序。

学习远程云端硬件实验平台基础器件电路搭建。

掌握单片机I/O口的应用。

了解单片机控制舵机模块的运行原理。

**三、实验原理**

实验器件简介

**舵机**

配合MCU器件使用通过调节pwm占空比来控制，也就是一个周期中高电平所占的百分比来控制舵机的转角的。根据舵机控制的原理，舵机的控制一般需要一个20ms左右的时基脉冲<周期>，该脉冲的高电平部分一般为0.5ms-2.5ms<占空比2.5%~12.5%>范围内的角度控制脉冲部分，总间隔为2ms。

即：

0.5ms--------------0度；

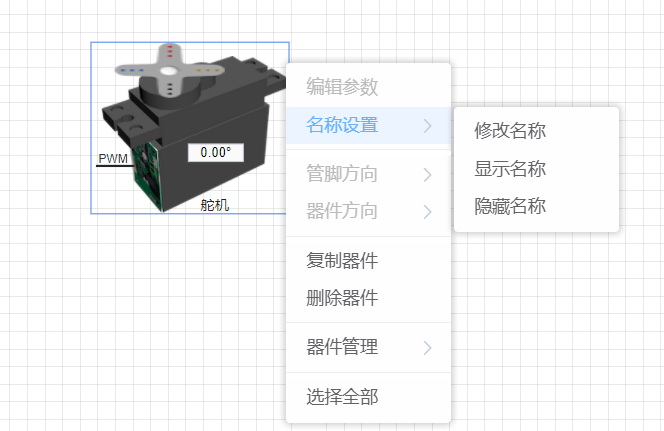
1.0ms------------45度；

1.5ms------------90度

2.0ms-----------135度；

2.5ms-----------180度；

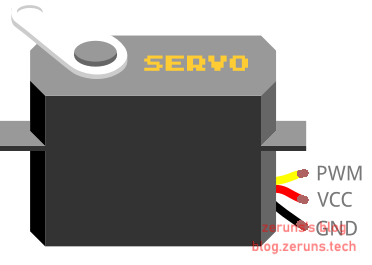
页面选中舵机器件可以通过鼠标右键单击器件调出任务窗口修改器件名称、修改器件显示层级、复制器件、删除器件。



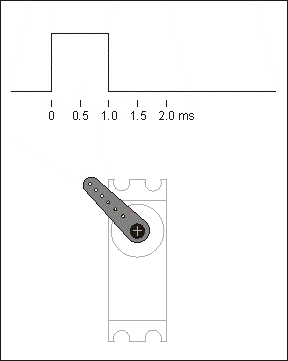
实物参考电路分析

PWM，英文名Pulse Width Modulation，是脉冲宽度调制缩写，它是通过对一系列脉冲的宽度进行调制，等效出所需要的波形（包含形状以及幅值），对模拟信号电平进行数字编码，也就是说通过调节占空比的变化来调节信号、能量等的变化，占空比就是指在一个周期内，信号处于高电平的时间占据整个信号周期的百分比，例如方波的占空比就是50%.

我们常说的舵机，它的学名叫做伺服电机，它是一种带有输出轴的小装置。当我们向伺服器发送一个控制信号时，输出轴就可以转到特定的位置。只要控制信号持续不变，伺服机构就会保持轴的角度位置不改变。如果控制信号发生变化，输出轴的位置也会相应发生变化。



控制线用于传输角度控制信号。这个角度是由控制信号脉冲的持续时间决定的，这叫做脉冲编码调制（PCM）。舵机的控制一般需要一个20ms左右的时基脉冲，该脉冲的高电平部分一般为0.5ms-2.5ms范围，总间隔为2ms。脉冲的宽度将决定马达转动的距离。例如：1.5毫秒的脉冲，电机将转向90度的位置（通常称为中立位置，对于180°舵机来说，就是90°位置）。如果脉冲宽度小于1.5毫秒，那么电机轴向朝向0度方向。如果脉冲宽度大于1.5毫秒，轴向就朝向180度方向。



实验所用模块配置与分析

MCU：逻辑器件-自定义管脚（MCU）

输出：实物器件-舵机

基本模块-频率测量输出

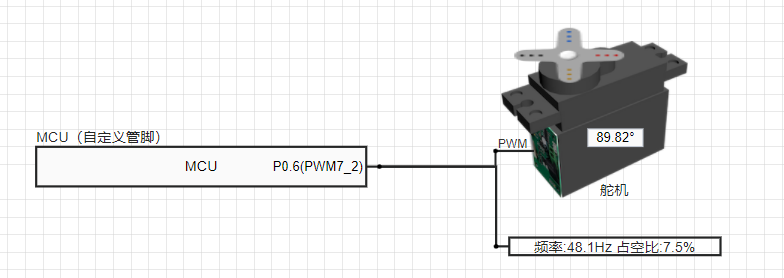
分析:PWM控制舵机转动

GPIO端口对应

频率测量输出对应单片机输出IO P0.6

控制舵机对应单片机输出IO P0.6

**四、实验参考仿真电路**



**五、实验步骤**

1. 按照参考仿真电路示图在平台实验面板里搭建仿真电路。

①在实验面板右侧，器件面板下，把自定义管脚（MCU)、舵机鼠标单击拖拽到面板里。

②拖拽自定义管脚（MCU）时，按照参考仿真电路选择输入、输出IO引脚。

③把自定义管脚（MCU)、舵机用线连接起来。

1. 在Keil uVision5软件里编写仿真电路程序。
2. 在平台实验面板里导入单片机程序.Hex文件。
3. 点击运行电路，查看电路现象。
4. 实验成功后，导出仿真电路图epl文件。

**六、参考代码与分析**

#define kServoPWMCountMax 2000

u16 ServoPWMCount = 0;

u16 ServoHighPulse = 50; //50-250

#define T1MS1 (65536-MAIN\_Fosc/100000) //1T模式

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数原型：void LED\_GPIO\_config(void)

函数功能：LED端口初始化函数

输入参数：无

输出参数：无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void REMOTO\_GPIO\_config(void)

{

GPIO\_InitTypeDef GPIO\_InitStructure; //结构定义

GPIO\_InitStructure.Mode = GPIO\_PullUp; //指定IO的输入或输出方式,GPIO\_PullUp,GPIO\_HighZ,GPIO\_OUT\_OD,GPIO\_OUT\_PP

GPIO\_InitStructure.Pin = GPIO\_Pin\_6; //指定要初始化的IO, GPIO\_Pin\_7, 或操作

GPIO\_Inilize(GPIO\_P0, &GPIO\_InitStructure); //初始化

}

void Time1\_16\_Config\_Init\_1(void)

{

AUXR |= 0x40;

TMOD = 0x00;

TL1 = T1MS1;

TH1 = T1MS1 >> 8;

TR1 = 1;

ET1 = 1;

EA = 1;

}

//50us一次中断

void tm1\_isr() interrupt 3

{

ServoPWMCount++;

if(ServoPWMCount>=kServoPWMCountMax)

{

ServoPWMCount=0;

}

if(ServoPWMCount>=ServoHighPulse)

{

P06 = 0;

}

else

{

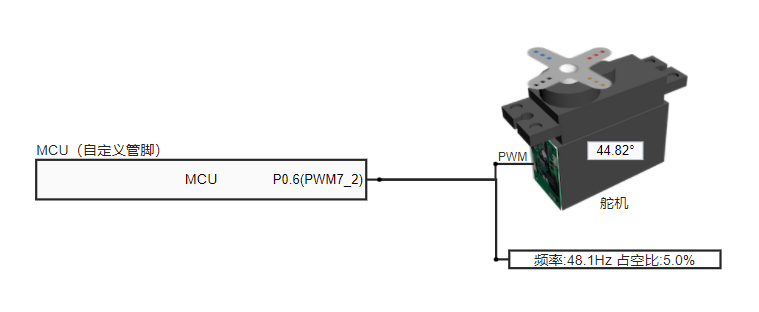
P06 = 1;

}

}

**七、实验现象**

1. 当点击程序运行按钮时，舵机开始以4S的时间转动角度（0-45-90-135-180）。



**八、拓展思考**

可以显示其他角度吗？