

24.SG90 舵机模块使用实验例程

一、实验目的

使用盘古 PGX-Nano 开发板驱动 SG90 舵机模块左右旋转与归位

二、实验简介

24.1.1. SG90 舵机模块介绍

SG90 舵机是一种位置（角度）伺服的驱动器，适用于那些需要角度不断变化并可以保持的控制系统。在机器人机电控制系统中，舵机控制效果是性能的重要影响因素。舵机可以在微机电系统和航模中作为基本的输出执行机构，因为其简单的控制和输出机制使得其在嵌入式系统中应用十分广泛。

24.1.2. SG90 舵机工作原理

SG90 舵机是一种小型舵机，具有三条信号线：GND（接地）、VCC（电源正极、5V）和 SIG（控制信号）。它通过接收 SIG 信号线传入的脉宽调制信号（PWM），解析目标角度信息，并利用内部信号调制芯片将信号转化为直流偏置电压。舵机内部的电位器与输出轴机械连接，实时反馈当前角度并产生对应电压。信号调制芯片通过比较直流偏置电压和电位器反馈电压计算电压差，并将其传递给电机驱动芯片控制电机的转动方向。当电机转动时，电位器同步旋转，逐步减小电压差，直到达到目标角度时停止电机。通过减速齿轮组，电机的高速低扭矩输出被转化为低速高扭矩的精确控制，从而实现舵机的稳定动作和自校正功能。这里只需要大概了解其工作原理即可，我们主要重点关注的是 SG90 舵机的控制原理。

SG90 舵机通过脉宽调制信号（PWM）控制。其需要一个 20ms 的时基脉冲，高电平持续时间在 0.5ms~2.5ms。对于 180 度舵机而言。不同一个 PWM 信号的不同高电平持续时间与舵机旋转角度的对应关系如下：

高电平持续时间/ms	舵机角度/°
0.5	0
1.0	45
1.5	90
2.0	135
2.5	180

我们使用开发板输出高电平持续时间在 0.5ms~2.5ms 之间，周期为 20ms 的 PWM 信号就能够控制舵机任意旋转 0~180 度。

24.2. 程序设计

超声波驱动模块端口如下：

端口	I/O	位宽	描述
clk	input	1	50MHz 时钟
rst_n	input	1	复位信号
key_in	input	3	key_in[0]右转、key_in[1]左转、key_in[2]归位
pwm	output	1	舵机控制信号

驱动部分代码如下，完整源码请查看 demo 源文件

```
//20ms 计数器
always @(posedge clk or negedge rst_n)begin
    if(!rst_n)
        cnt_20ms <= 0;
    else if(cnt_20ms== TIME_20MS - 1)
        cnt_20ms <= 0;
    else
        cnt_20ms <= cnt_20ms + 1;
end

always @(posedge clk or negedge rst_n)begin
    if(!rst_n)
        cnt_3s <= 0;
    else if(cnt_3s== TIME_3S - 1)
        cnt_3s <= 0;
    else
        cnt_3s <= cnt_3s + 1;
end

//按键情况列出
always  @(posedge clk or negedge rst_n)begin
    if(rst_n==1'b0)
        pwm_level <= pwm_0d5ms;//默认归位
    else if((key_in[0] == 0)&&(cnt_3s== TIME_3S - 1)&&(pwm_level<pwm_2d5ms))
        pwm_level <= pwm_level + 1'b1;
    else if((key_in[1] == 0)&&(cnt_3s== TIME_3S - 1)&&(pwm_level>pwm_0d5ms))
        pwm_level <= pwm_level - 1'b1;
    else if(key_in[2] == 0)
        pwm_level <= pwm_0d5ms;//归位
    else
        pwm_level <= pwm_level;
```

```

end

//输出 pwm
always @(posedge clk or negedge rst_n)begin
    if(rst_n==1'b0)
        pwm <= 0;
    else if(cnt_20ms < pwm_level)
        pwm <= 1;
    else
        pwm <= 0;
end

```

模块的核心是如何产生受按键控制的可以调整的 pwm 信号，我们使用一个 20ms 计数器 cnt_20ms 用来产生 20ms 的时基脉冲。使用一个 pwm_level 信号决定输出的 pwm 信号高电平持续时间，只有当 cnt_20ms 的值小于 pwm_level，输出的 pwm 信号才为高电平，当 cnt_20ms 的值大于 pwm_level，输出的 pwm 信号为低电平。所以我们只需要通过按键逻辑更改 pwm_level 的值，即可控制输出 pwm 信号的高电平持续时间，pwm_level 默认为 pwm_0d5ms，也就是默认舵机旋转 0 度。

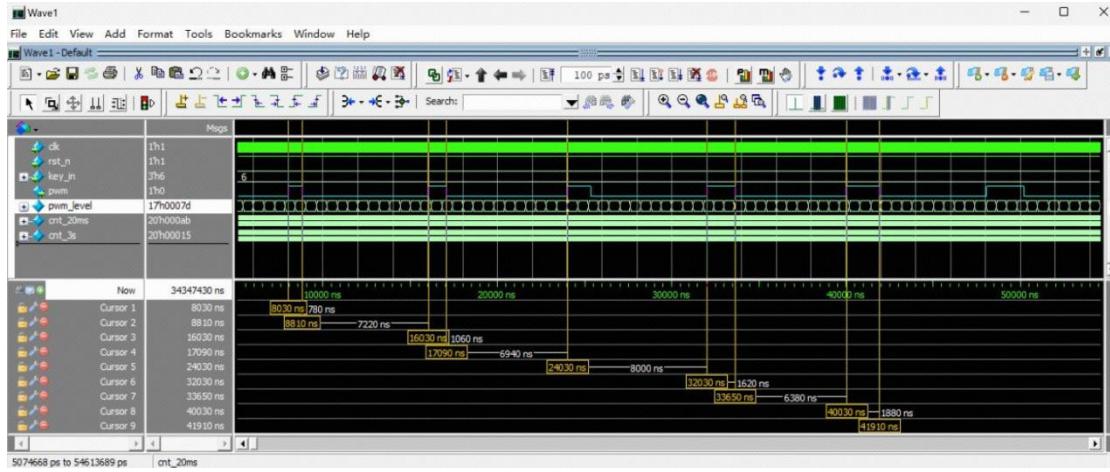
当按键 S0 被按下后，每当 cnt_3s 计数到 1000，并且 pwm_level 的值小于 pwm_2d5ms 时，pwm_level 才会加一，这是为了使得舵机转动得不至于过快和防止高电平持续时间大于 2.5ms 不满足 SG90 舵机的控制要求。

当按键 S1 被按下后，每当 cnt_3s 计数到 1000，并且 pwm_level 的值大于于 pwm_0d5ms 时，pwm_level 才会减一，这是为了使得舵机转动得不至于过快和防止高电平持续时间小于 0.5ms 不满足 SG90 舵机的控制要求。

当按键 S2 被按下后，pwm_level 的值被重新被设置为 pwm_0d5ms，使得舵机旋转到 0 度位置，舵机归位。

24.3. 代码仿真

舵机驱动模块输入输出逻辑较为简单，我们只需要模拟参数时钟与复位即可，为了便于观察，我们使用 defparam 语句对舵机驱动模块内的参数进行重定义，我们这里是模拟按下按键 S0 使得舵机向右旋转，期望输出的 pwm 信号应该是占空比逐渐增大的。Modelsim 与 PDS 进行联合仿真的操作在第二部分开发环境搭建章节已有介绍，我们这里启动 Modelsim 进行仿真，查看 pwm 信号是否如预期般占空比逐渐增大。



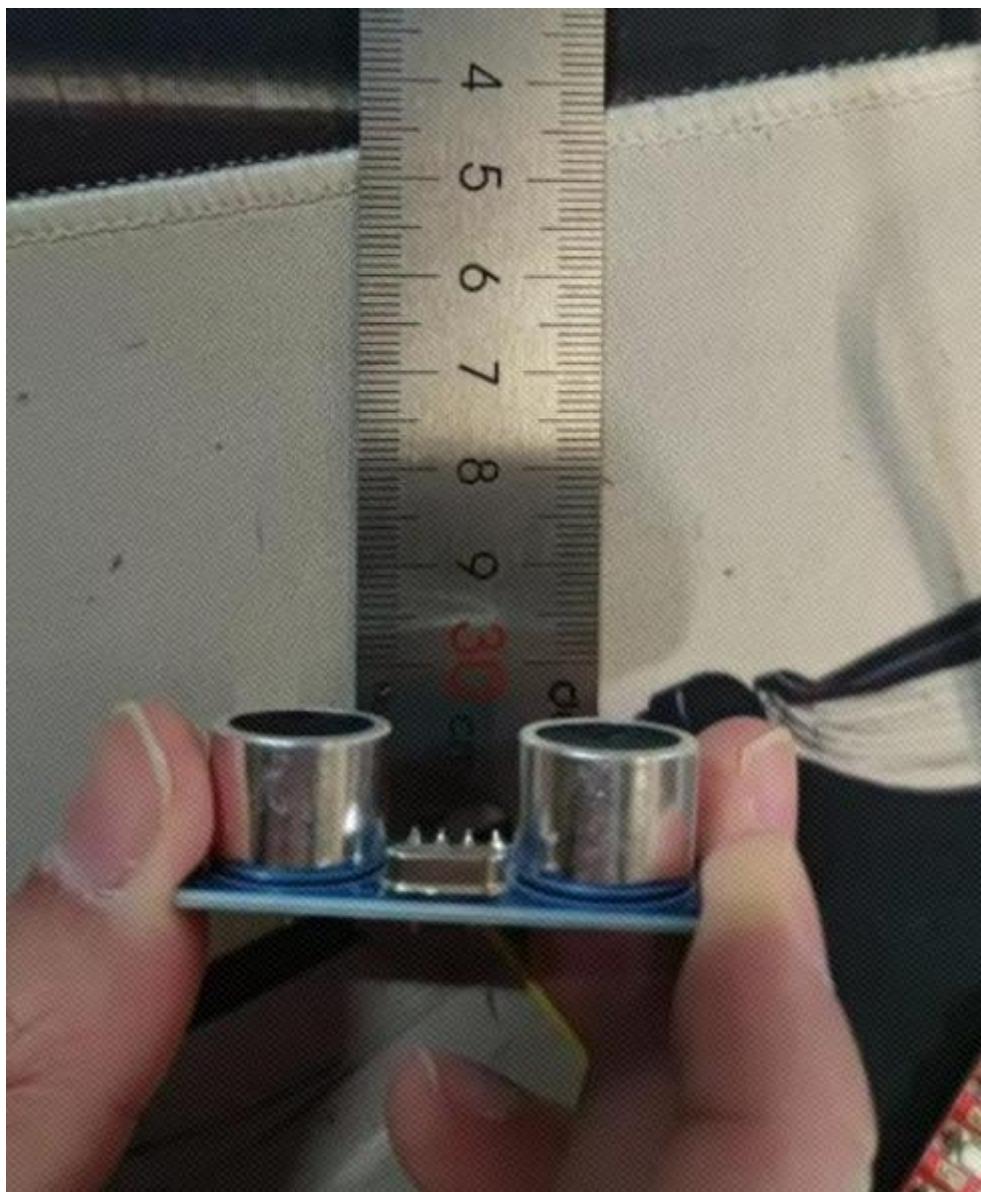
如上图所示，当按下按键 S0 之后（key_in 位宽为 3，按键按下后为低电平，未按下为高电平，按下 S0 后变为 3' b110, 也就是十进制的 6），可以发现输出的 pwm 信号高电平部分在逐渐增加，由于我们在 tb 代码中更改了参数，所以这里的 pwm 信号周期、高电平持续时间并不满足舵机控制的要求，只是为了直观的看出占空比的变换，对 pwm 信号进行了调整。在实际舵机控制过程中，为了舵机运行得更加平稳，不出现抽搐等现象，pwm 波形的变换过程会更加平缓。

同理如果按下 S1，舵机将会向左旋转，对应的 pwm 占空比应该会逐渐减小，按下 S2，舵机将会归位，回到 0 度位置，读者可以在 tb 文件中对输入按键信号进行修改，观察 pwm 信号的变换情况。

24.4. 实验现象

当按下开发板按键 S0 时，舵机向右旋转，当按下 S1 时，舵机向左旋转，当按下 S2 时舵机归位。





串口打印信息如下：

