## 【RTT设计大赛】基于AB32VG1遥控小车

**前言**

   小车是电子大赛常用的装备，采用舵机转向，两个电机调速驱动的四轮小车，以前都是用ti’或stm32f1的板子做的，用RISC-V架构的芯片做一个小车的控制器应该没问题。这里利用AB32VG1开发板做核心控制元件，用无线串口通信，实现小车的遥控。

**一、硬件说明**

硬件部分主要包括以下三部分：

1. 主控装置：AB32VG1开发板
2. 驱动装置：四路（两路）电机驱动模块

（3）遥控通信：HC-HID

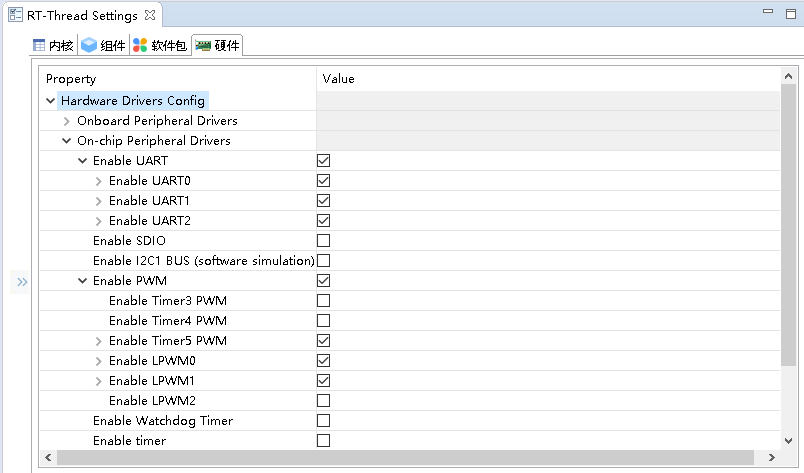
**二、软件说明**

1.AB32VG1开发板

（1）开发平台：RT-Thread Studio

（2）组态配置

比例阀主要是采用PWM电流驱动，采用闭环控制可以提高控制精度，所以组态配置中使能PWM信号和ADC0通道。此外，在PWM信号上叠加震颤能提高比例阀的动态响应，降低滞环，所以使能了硬件定时器timer。



（3）主要代码

**①舵机控制**

**int** **sg**(**int** argc, **char** \*argv[])

{

period1 = 10000000; /\* 周期为10ms，单位为纳秒ns \*/

pulse1 = 0; /\* PWM脉冲宽度值，单位为纳秒ns \*/

/\* 设置LED引脚脚模式为输出 \*/

rt\_pin\_mode(SG\_PIN\_NUM, PIN\_MODE\_OUTPUT);

/\* 拉高LED引脚 \*/

rt\_pin\_write(SG\_PIN\_NUM, PIN\_HIGH);

/\* 查找设备 \*/

pwm\_dev = (**struct** rt\_device\_pwm \*)rt\_device\_find(PWM\_DEV\_NAME);

**if** (pwm\_dev == RT\_NULL)

{

rt\_kprintf("steering gear control run failed! can't find %s device!\n", PWM\_DEV\_NAME);

**return** RT\_ERROR;

}

/\* 设置PWM周期和脉冲宽度默认值 \*/

rt\_pwm\_set(pwm\_dev, PWM\_DEV\_CHANNEL, period1, pulse1);

/\* 使能设备 \*/

rt\_pwm\_enable(pwm\_dev, PWM\_DEV\_CHANNEL);

**if**(argc <2)

{

/\* 由0度到180度，然后由180度到0度，循环运行5个周期 \*/

t1 = 50;

**while** (--t1)

{

rt\_thread\_mdelay(1000);

rt\_pwm\_set(pwm\_dev, PWM\_DEV\_CHANNEL, period1, period1-500000);

rt\_thread\_mdelay(5000);

rt\_pwm\_set(pwm\_dev, PWM\_DEV\_CHANNEL, period1, period1-2500000);

}

}

**else**

{

/\*将msh输入的字符转换为数值\*/

pulse1 = Ascii2Hex(argv[1],**strlen**(argv[1]));

/\* 计算舵机目标位置所需要的高电平脉冲宽度 \*/

t1 = 2000000/180\*pulse1;

pulse1 = period1-500000-t1; //pwm设定的占空比为低电平宽度，舵机控制使用高电平宽度控制旋转角度

/\* 设置PWM周期和脉冲宽度 \*/

rt\_pwm\_set(pwm\_dev, PWM\_DEV\_CHANNEL, period1, pulse1);

}

}

**void** **set\_ang**(uint8\_t ang) //period-500000 对应0° ， period-2500000 对应180°， 90°对中，

{**static** rt\_uint32\_t ang\_pulse;

ang\_pulse=1500000+ang/90\*100000;

pulse1=period1-ang\_pulse;

rt\_pwm\_set(pwm\_dev, PWM\_DEV\_CHANNEL, period1, pulse1);

}

②行走电机驱动

**int** **main**(**void**)

{

uint8\_t pin = rt\_pin\_get("PE.1");

rt\_pin\_mode(pin, PIN\_MODE\_OUTPUT);

uint8\_t pin\_dir1 = rt\_pin\_get("PA.0"); //电机方向控制

uint8\_t pin\_dir2 = rt\_pin\_get("PA.1"); //电机方向控制

rt\_pin\_mode(pin\_dir1, PIN\_MODE\_OUTPUT);

rt\_pin\_mode(pin\_dir2, PIN\_MODE\_OUTPUT);

rt\_kprintf("Hello, world\n");

thread\_serial1();

**while** (1)

{

rt\_pin\_write(pin, PIN\_LOW);

rt\_thread\_mdelay(500);

rt\_pin\_write(pin, PIN\_HIGH);

rt\_thread\_mdelay(500);

sg(0,"100000");

rt\_thread\_mdelay(1000);

tr(0,"2500000");

**if**(rec\_ok==1) //收到命令

{

rec\_ok=0;

**if**(rec\_buf[1]=='L')// 左转角度

{ang\_l=(rec\_buf[2]-0x30)\*10 +(rec\_buf[3]-0x30);}

rt\_kprintf("ang\_l%d",ang\_l);

**if**(rec\_buf[4]=='R')// 右转角度

{ang\_r=(rec\_buf[5]-0x30)\*10+(rec\_buf[6]-0x30);}

rt\_kprintf("ang\_r%d",ang\_r);

**if**(rec\_buf[7]=='F')// 前进

{

rt\_pin\_write(pin\_dir1, PIN\_LOW);

rt\_pin\_write(pin\_dir2, PIN\_LOW);

}

**if**(rec\_buf[7]=='N')// 空挡

{

rt\_pin\_write(pin\_dir1, PIN\_HIGH);

rt\_pin\_write(pin\_dir2, PIN\_HIGH);

}

**if**(rec\_buf[7]=='R')// 后退

{

rt\_pin\_write(pin\_dir1, PIN\_LOW);

rt\_pin\_write(pin\_dir2, PIN\_HIGH);

}

**if**(rec\_buf[7]=='T')// 刹车

{speed=0;

rt\_pin\_write(pin\_dir1, PIN\_HIGH);

rt\_pin\_write(pin\_dir2, PIN\_LOW);

}

speed=(rec\_buf[8]-0x30)\*10+(rec\_buf[9]-0x30);

rt\_kprintf("apeedl%d",speed);

clear\_buf();

pwm\_out(speed,speed);

**if** (ang\_l>ang\_r)

set\_ang(-ang\_l);

**else** {

set\_ang(-ang\_r);

}

}

}

}

**三、测试效果**

通过串口调试助手发送小车控制命令，小车能实现前进、后退、左转、右转动作，也能控制小车的速度。

**总结**

  通过测试，可以实现小车速度，转向的遥控控制，参照例程很容易实现PWM速度控制和舵机控制，串口实现起来也比较容易，下一步可以增加串口通讯的效验码进一步提高可靠性，并将小车的实际速度、实际方向、实际转角等参数回传给遥控端。

