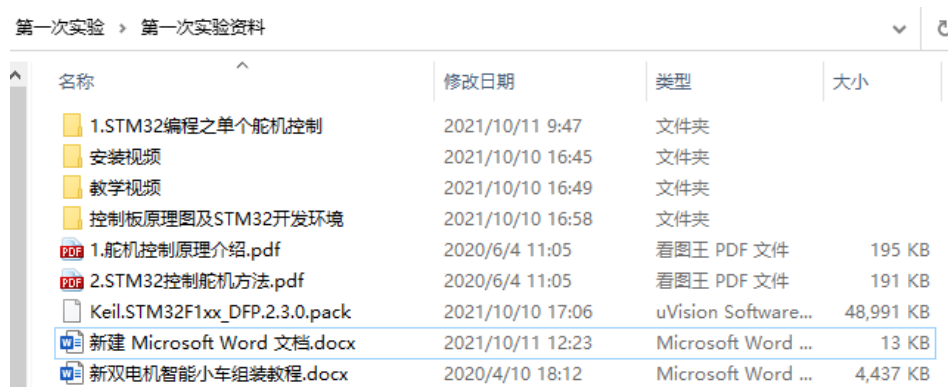


《机器人导论实验》报告

	日期：2022. 10. 16	得分：
学号：	姓名：	专业：

一、实验准备

1. 点开“第一次实验资料”，安装“Keil.STM32F1xx_DFP.2.3.0.pack”功能包。



第一次实验 > 第一次实验资料						
名称	修改日期	类型	大小			
1.STM32编程之单个舵机控制	2021/10/11 9:47	文件夹				
安装视频	2021/10/10 16:45	文件夹				
教学视频	2021/10/10 16:49	文件夹				
控制板原理图及STM32开发环境	2021/10/10 16:58	文件夹				
1.舵机控制原理介绍.pdf	2020/6/4 11:05	看图王 PDF 文件	195 KB			
2.STM32控制舵机方法.pdf	2020/6/4 11:05	看图王 PDF 文件	191 KB			
Keil.STM32F1xx_DFP.2.3.0.pack	2021/10/10 17:06	uVision Software...	48,991 KB			
新建 Microsoft Word 文档.docx	2021/10/11 12:23	Microsoft Word ...	13 KB			
新双电机智能小车组装教程.docx	2020/4/10 18:12	Microsoft Word ...	4,437 KB			

安装完成后即可对小车代码进行编辑。

2. 准备好 STM32 环境下的串口下载软件 mcuisp，以实现与开发板的通信和代码烧录。

二、实验原理

本次实验主要练习、测试如何通过调整代码来对应改变舵机的转动角度、转动周期等参数。













舵机是一个微型的伺服控制系统，需要一个外部控制器产生脉宽调制信号以控制舵机转动角度，脉冲宽度是舵机控制器所需编码信息，对于一个确定的舵机，其控制脉冲周期(PWM 周期)是确定的（通常是 20ms），此时脉冲宽度和舵机输出轴转角一一对应。

三、实验任务

1. 下载源代码

根据“第一次实验资料”里的“教学视频”，将源代码“CAR_STM32F103C6”下载到开发板上。（下载时拨到“下载”方向，并看情况按一次旁边的“复位键”；下载完后拨到“工作”方向，按一次旁边的“复位键”才能让程序跑起来）

工程文件的路径分别为“.\CAR_STM32F103C6\MDK-ARM\CAR_STM32F103C6.uvprojx”和“.\CAR_STM32F103C6\MDK-ARM\CAR_STM32F103C6\CAR_STM32F103C6.hex”。

<input type="checkbox"/> 名称	修改日期	类型	大小
 CAR_STM32F103C6	2022/10/10 11:23	文件夹	
 DebugConfig	2022/9/25 23:44	文件夹	
 RTE	2022/9/26 9:55	文件夹	
 Task	2022/9/25 23:44	文件夹	
 CAR_STM32F103C6.uvguix...	2021/10/18 12:00	Microsoft...	73 KB
 CAR_STM32F103C6.uvguix...	2022/10/10 13:18	STEVEN ...	89 KB
 CAR_STM32F103C6.uvoptx	2022/10/10 11:22	UVOPTX ...	20 KB
<input checked="" type="checkbox"/>  CAR_STM32F103C6.uvprojx	2022/10/10 13:18	磳ision5 P...	22 KB
 startup stm32f103x6.lst	2022/10/10 11:22	LST 文件	36 KB
 CAR_STM32F103C6.axf	2022/10/10 11:23	AXF 文件	648 KB
 CAR_STM32F103C6.bu...	2022/10/10 11:23	Microsoft...	4 KB
<input checked="" type="checkbox"/>  CAR_STM32F103C6.hex	2022/10/10 11:23	HEX 文件	23 KB

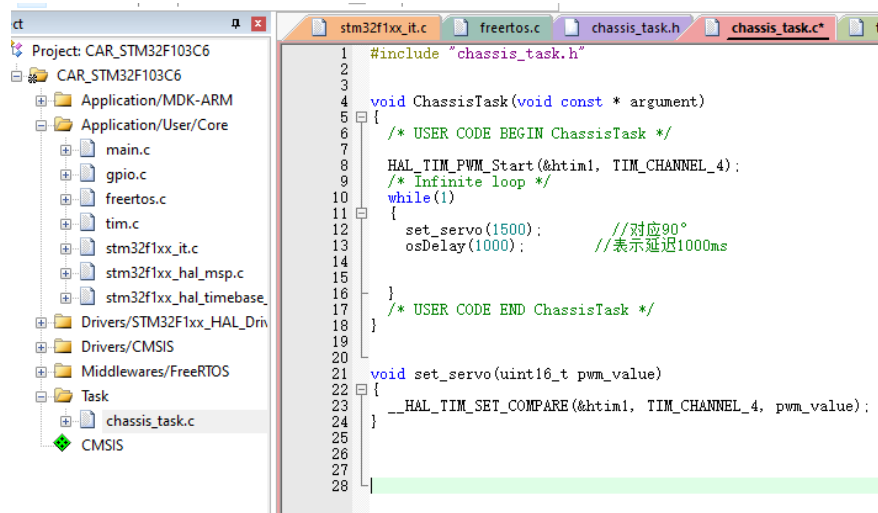
2. 改写源代码

根据网址

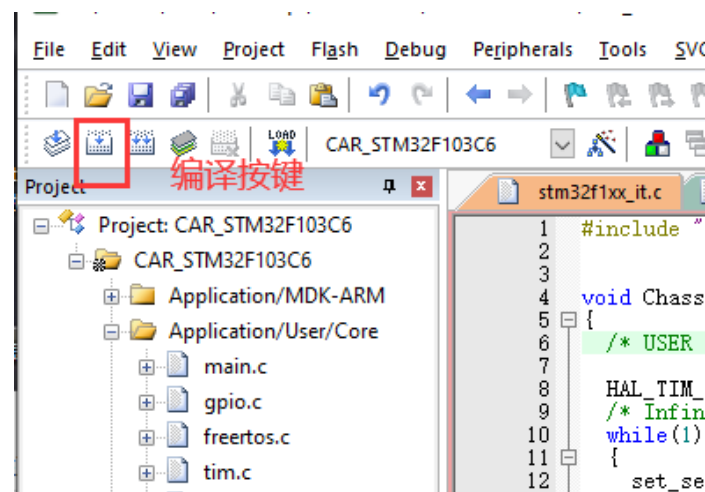
https://blog.csdn.net/qq_40499719/article/details/81267895 理解舵

机代码，如下图所示，打开“Task”里的“chassis_task.c”文件，改写 while

（1）里的舵机转动代码（比如改写成按 $90^{\circ} \rightarrow 50^{\circ} \rightarrow 140^{\circ} \rightarrow \dots$ 的设定转动，转动角度随意，但注意由于机械结构的限制，实际可能只能够在 $30^{\circ} \rightarrow 150^{\circ}$ 的范围里转动，需要提前试出转动范围，防止损坏电机），并拍照记录舵机的实际转动效果。

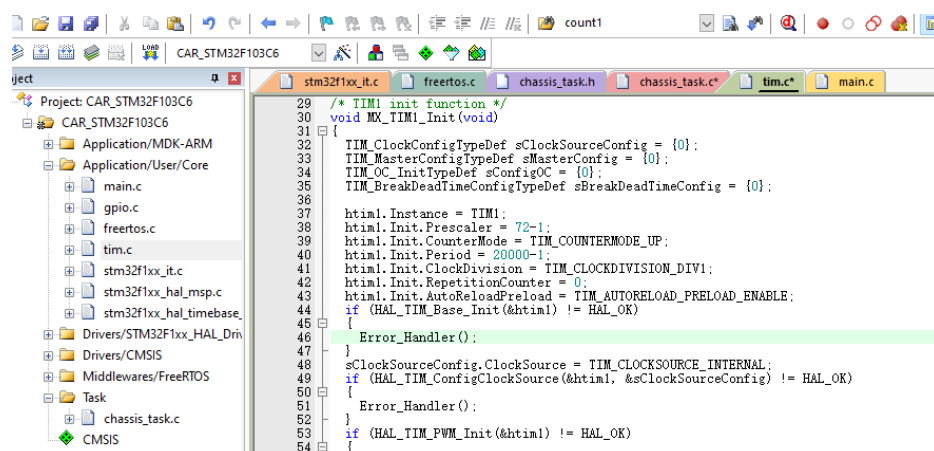


更改完程序后需要按一下 F7 重新编译一下，才能将 hex 文件更新，如果是笔记本有冲突点击左上角的编译按钮。



3.强化对舵机的理解（选做）

打开“Application/User/Core”里的“tim.c”文件



将“htim1.Init.Prescaler”与“htim1.Init.Period”的值分别改为“720-1”与“2000-

1” ,寻找此时的 90° 、 60° 对应的 “chassis_task.c” 程序里的 “pwm_value”大概是多少（此时 1500 已经 100%不对应 90° 了）

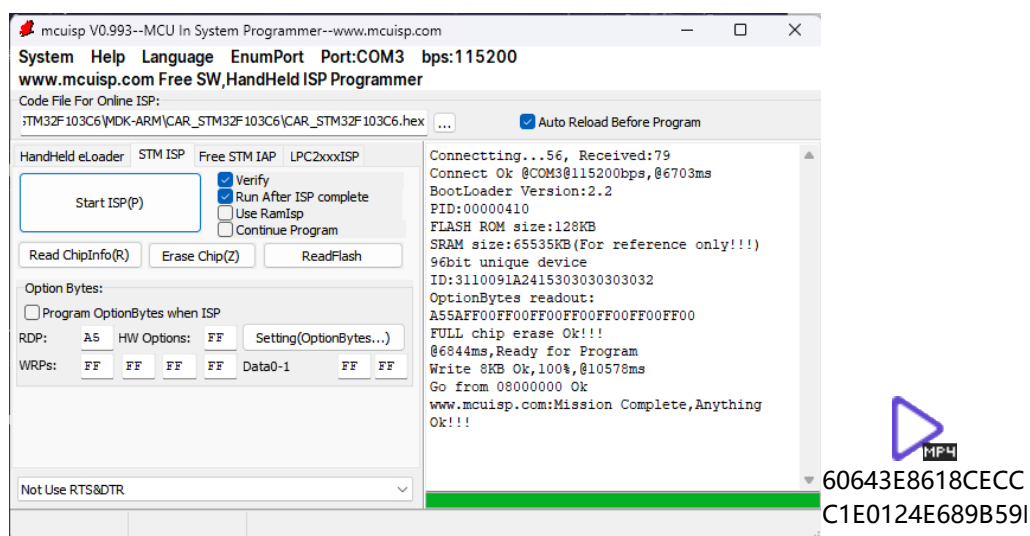
```
while(1)
{
    set_servo(1500);          //对应90°
    osDelay(1000);           //表示延迟1000ms
}
/* USER CODE END ChassisTask */
}

void set_servo(uint16_t pwm_value)
{
    __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_4, pwm_value);
}
```

再将 “htim1.Init.Prescaler” 与 “htim1.Init.Period”的值分别改为 “7200-1”与 “200-1”，重新寻找 90° 、 60° 对应的 “pwm_value”的数值。

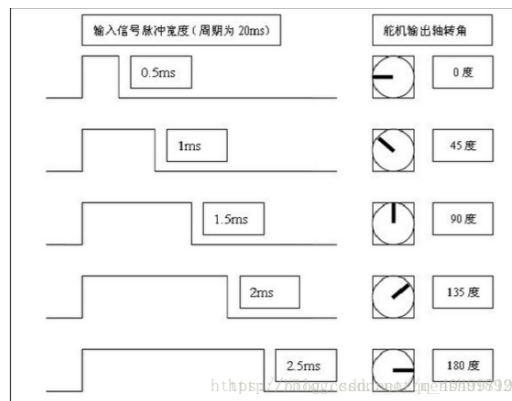
四、 实验过程与实验结果

1. 由于原始代码中对应的舵机角度就是 90° ，看不出效果，所以对其稍微调整后，其效果变为旋转两个角度，间隔 2s，效果如下：



2. 此时要修改 chassis_task.c 文件中 set_servo(1500);的值，以使舵机转到不同的角度。但是为什么 1500 对应的是 90° 呢？

根据这篇 [csdn](#) 博客可知，在 PWM 信号频率为 50Hz 时，脉冲宽度 1500μs，其占空比就对应 90° 。



因此按照该表,即可知 45° 和 135° 对应的脉宽分别为 $1000\ \mu s$ 和 $2000\ \mu s$ 。

因此我们通过以下的代码段获得了如下视频的效果:

```
while(1)
{
    set_servo(1000);    //对应45°
    osDelay(2000);      //表示延迟2000ms
    set_servo(2000);    //对应135°
    osDelay(2000);      //表示延迟2000ms
}
```

ED297E3330D13
DC5C84CAB8A62{

3. 根据 https://blog.csdn.net/qq_40499719/article/details/81267895 和 https://blog.csdn.net/Iron_vik/article/details/117083345 这两篇博客总结可知:

(1) 舵机接收的 PWM 信号频率是固定的,通常为 50Hz ,即周期 20ms 。根据 PWM 周期计算公式

$$T_{PWM} = (arr + 1) \times (psc + 1) \div CLK$$

代入计算得 $T_{PWM} = 0.02s = 20ms$,是一致的。也正是因为频率恒定这一要求, arr 和 psc 都是以对应倍数增大和减小,以保证 $(arr + 1) \times (psc + 1)$ 值恒定。

(2) 在第二篇文章中提到“设 arr (自动重装载值) 为 x 时。转动角度所对应的值 $= (x+1) \times (1 - \text{占空比})$ ”,其实就是给出了要修改的值(设为“ pwm_value ”)的计算公式(设脉宽为“ $pulse_width$ ”):

$$pwm_value = (arr + 1) \times \left(\frac{pulse_width}{T_{PWM}} \right)$$

可见 pwm_value 只与 arr 有关。

因此当 $arr = 2000 - 1$ 时, pwm_value 对应缩小 10 倍, 90° 对应 150, 60° 约为 116.7;

$arr = 200 - 1$ 时, pwm_value 对应缩小 100 倍, 90° 对应 15, 60° 约为 11.7;



69E11C3E9D4894
6F0E153D0F0A1B4

实验效果如视频所示：

五、 实验心得

在本次实验中，由于默认的代码小车的轮子就是 90° 的，导致我们最初写入代码后小车没有明显反应，同时我们也没有仔细检查代码的含义，以为小车坏了，反复检查车体花费了不少时间。

除此之外别的过程都很顺利，在解决这一问题之后，我们的进展就非常顺利了。