湖北工業大學

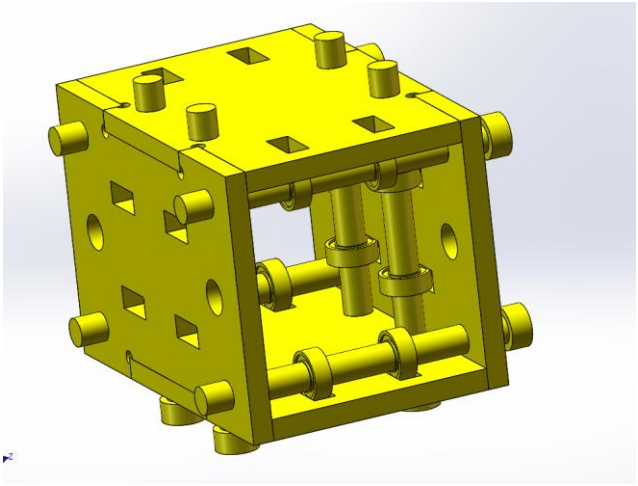
ROBOCON 2019 MR2

技术报告

目 录

一 结构

1.1 滑车装置



滑车相邻两片板子采用槽式楔合连接，每个相对的面用 4 个 M4×45 的长螺栓相连接，整体尺寸 37×37（不考虑螺栓尺寸）。

滑动功能的实现依靠 16 个每面 4 个外径 6mm、内径 4.5mm 的小轴承。

四面的面板一般情况下可以采用碳素纤维板，铝制板，雕刻定制。

二 硬件（重点、难点技术）

2.1 遥控器及其遥感急停

2.1.1 PS2 手柄的协议发送及遥停

PS2 手柄是一个很好用的东西，但是由于开发商提供的是 F1 的寄存器版本的代码，移植到 F4 费时费力。

因此用一块 F1 的开发板，使用开发商提供的寄存器例程不断读取数据，用串口发送出去。

发送协议说明；

波特率 115200 8 N 1

通信协议格式 FA FF KEY LX LY RX RY CHECKSUM

校验和算法 将所有的数据位相加 末八位为 FF 则为真

再此基础上，我们使用了 f1 定义一个按键为遥停，使用一个 IO 口控制继电器的通断

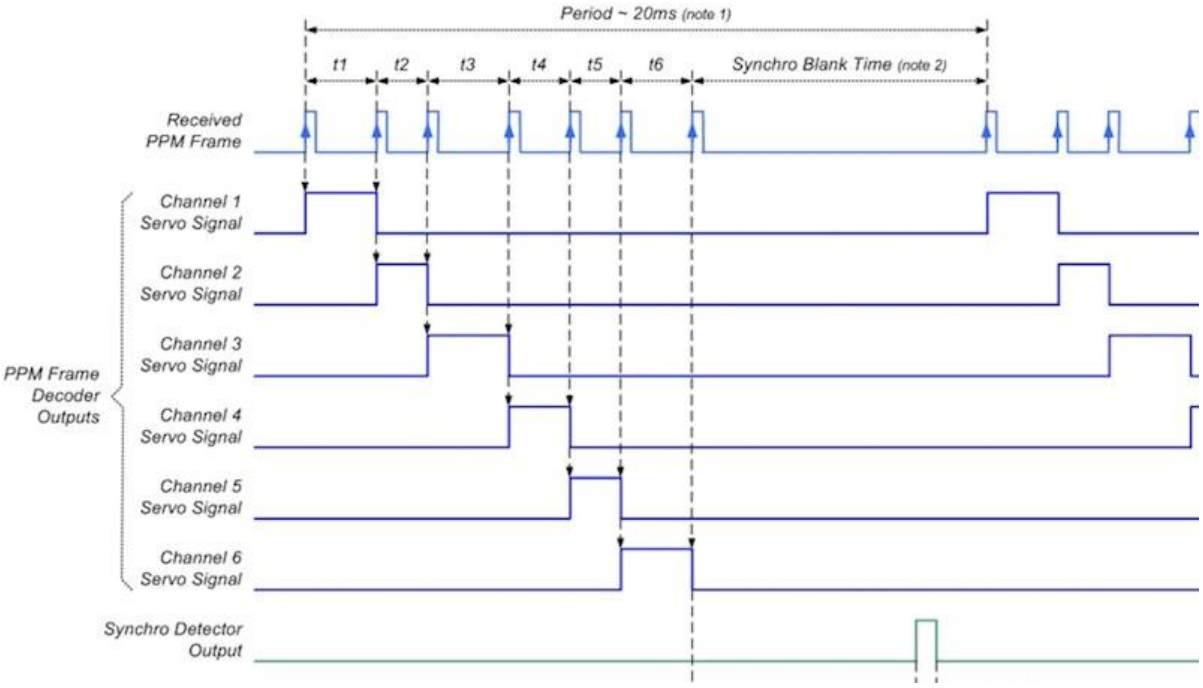
控制主电路的电源，而此遥控电路和主电路互为独立电路。比较可靠，防止在调试过程中出现意外。

```
while(1)
{
  KEY=PS2_DataKey();
  PSS_LX_VALUE=PS2_AnologData( PSS_LX);
  PSS_LY_VALUE=PS2_AnologData( PSS_LY);
  PSS_RX_VALUE=PS2_AnologData( PSS_RX);
  PSS_RY_VALUE=PS2_AnologData( PSS_RY);

  checksum=KEY+PSS_LX_VALUE+PSS_LY_VALUE+PSS_RX_VALUE+PSS_RY_VALUE;
  checksum = ~checksum;

  if(KEY==PSB_R2)
    GPIO_SetBits(GPIOB,GPIO_Pin_5);

  USART1_Send_Byte(0xfa);
  USART1_Send_Byte(0xff);
  USART1_Send_Byte(KEY);
  USART1_Send_Byte(PSS_LX_VALUE);
  USART1_Send_Byte(PSS_LY_VALUE);
  USART1_Send_Byte(PSS_RX_VALUE);
  USART1_Send_Byte(PSS_RY_VALUE);
  USART1_Send_Byte(checksum);
```

```
23.}
```

2.1.2 航模遥控

航模遥控一般使用的是 PPM 模式，PPM 就是将多路的 PWM 波融合起来。



每一帧为 20ms，再将 20ms 划分为每 2ms 一小帧，则一共有 10 个小帧，也即 10 个 channel。但由于需要加入同步帧，则最多有 9 个 channel。每一 channel 有 2ms，这 2ms 由固定的 0.5ms 再加上可调节的 0.5ms~1.5ms 构成。

解码程序的写法，通过不断的捕获高电平时间，一直到捕获到同步帧，再将一个完整的帧从数组拿出来解算的的出具体的数值。

```
1. if(TIM2CH3_CAPTURE_STA&0X80)        //成功捕获到了一次高电平
2. {
3.     if(ppm_rx_sta==1) {
4.         ppm_rx[ppm_rx_num+1]=TIM2CH3_CAPTURE_VAL;
5.         ppm_rx_num++;
```

```
    }

    if(4>TIM2CH3_CAPTURE_STA&0X3F>0||TIM2CH3_CAPTURE_VAL>3000) ppm_rx_sta++;//低电平时间大于 3000us 为起始帧

    if(ppm_rx_sta==2) {
        ppm_rx_sta=0;    // 1 表示接收到同步帧/ 2 接收到到下一起始帧 ppm 数据接收完毕
        ppm_rx[0]=1;
        ppm_rx_num=0;
    }

    TIM2CH3_CAPTURE_STA=0;          //开启下一次捕获

}
```

三 软件

3.1 HAL 库和 CUBEMX 的使用

从 15、16 年开始，ST 逐渐停止了对标准外设库的更新和维护，转向了 HAL 和 LL 库，目前已有的库有三种。

SPL：Standard Peripheral Library 标准外设库

HAL：Hardware Abstraction Layer 硬件抽象层库

LL：Low-layer 底层库

STM32CubeMX 这个工具是 ST 目前重点打造的工具。

STM32CubeMX 最重要的特性：

1.直观的选择 STM32 微控制器（MCU）和微处理器（MPU）。

2.丰富易用的图形化界面:

3.直接生成硬件初始化代码，节省时间

3.2 串口无线调参

在空闲中断的基础上使用串口处理逻辑。(bsp_usart.c)

```
1. uint8_t   usart2_buf[USART2_BUFLEN];
2. static void uart_rx_idle_callback(UART_HandleTypeDef* huart)
3. {
4.     /* clear idle it flag avoid idle interrupt all the time */
5.     __HAL_UART_CLEAR_IDLEFLAG(huart);
6.
7.     if (huart == &USART2_HUART)
8.     {
9.         __HAL_DMA_DISABLE(huart->hdmarx);
10.
11.        if ((USART2_MAX_LEN - dma_current_data_counter(huart->hdmarx->Inst
   ance)) == USART2_BUFLEN)
12.            usart2_callback_handler(&usart2info, usart2_buf);
13.
14.        /* restart dma transmission */
15.        __HAL_DMA_SET_COUNTER(huart->hdmarx, IMU_MAX_LEN);
16.        __HAL_DMA_ENABLE(huart->hdmarx);
17.
18.    }
19.}
```

这里，由于数据位数定义为一位，只要有一位数据过来，就会触发串口空闲中断，将接收到的数据放入存入 usart2_buf。

在 DEBUG.C 里,开了一个任务，每隔 500ms 判断一次接收到的数据。使用 switch case 选择执行的语句。

目前实现一个选择的功能，只判断一位，由于使用的是 16 进制，一个字节的上限有大约 256 种，所以目前使用的数量是没有问题的。

下面列举了几种不同的使用场景 如在 MR2 中改变步态参数、改变寄存器的值控制舵机位置。

```
1. void InterpretCommand(void)
2. {
3.     switch(usart2_buf[0])
4.     {
5.     case 0x01:
6.         state = WALK;
7.         printf("\r\n**********state = WALK**********\r\n");
8.         break;
9.     case 0x02:
10.        state_detached_params[state].detached_params_0.stance_height+=1;
11.    case 0x03:
12.        TIM4->CCR1+=20;
13.    }
14.}
```

由于时间限制，没有能够做出上位机，此处的工作已经为上位机留下了基础，已经做好了协议数据的传输，只需要再定义一套通信协议。

就可以实现上位机的数据互发，实时调参，监控。

3.3 虚拟示波器

根据山外上位机的通信协议，写了串口 DMA 发送的代码

```
1. uint8_t VcanTxBuff[20]= {0};
2. short  wave_form_data[8] = {0};
3.
4. void vcan_send_byte(uint8_t date);
5. void Vcan_Send_Wave_Data(void);
6.
7.
```

```
8. void VcanGC_task(void *pvParameters)
9. {
10.     extern int count;
11.     for(;;)
12.     {
13.         wave_form_data[0] =(char)imuinfo.ActVal[2];
14.         wave_form_data[1] =(char)imuinfo.ActVal[1];
15.         wave_form_data[2] =(char)imuinfo.ActVal[0];
16.         wave_form_data[3] =10;
17.         Vcan_Send_Wave_Data();
18.         vTaskDelay(200);
19.     }
20.
21.}
22.
23.void Vcan_Send_Wave_Data(void)
24.{
25.     VcanTxBuff[0]=0x03;
26.     VcanTxBuff[1]=0xfc;
27.
28.for(int i =0;i<12;i++)
29.   VcanTxBuff[i+2]=imuinfo.data[11-i];
30.
31.
32.     VcanTxBuff[14]=0xfc;
33.     VcanTxBuff[15]=0x03;
34.
35.
36.     HAL_UART_Transmit_DMA(&huart8, (uint8_t*)&VcanTxBuff, sizeof(VcanTxBuf
   f));
37.
38.}
```

山外上位机的功能很强大，这里只用了他的虚拟示波器的功能。

但是由于山外上位机是为智能车所开发的，很多功能我们用不上，自己的上位机还是很有必要。