一、摘要

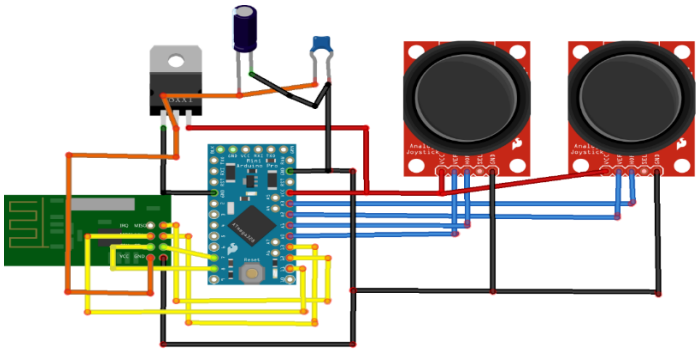
我们小组制作的船模为虾艇，驱动部分使用无刷电机和无刷电调，舵机控制转向；遥控器与接收器均基于 Arduino pro mini 制作，为 4 通道（可扩展），其中遥控器（左手转向，右手油门）通过杜邦线连接，接收器为自行焊接；通信模块使用的是 NRF24L01P+PA+LNA；使用了 AMS1117 3.3V 稳压芯片给通信模块提供安全的电压；设置了滤波电容保证信号良好；接收机上设置了 LED 灯以表示是否有信号；同时也对电机和电调设置了水冷保护。

二、部件选择

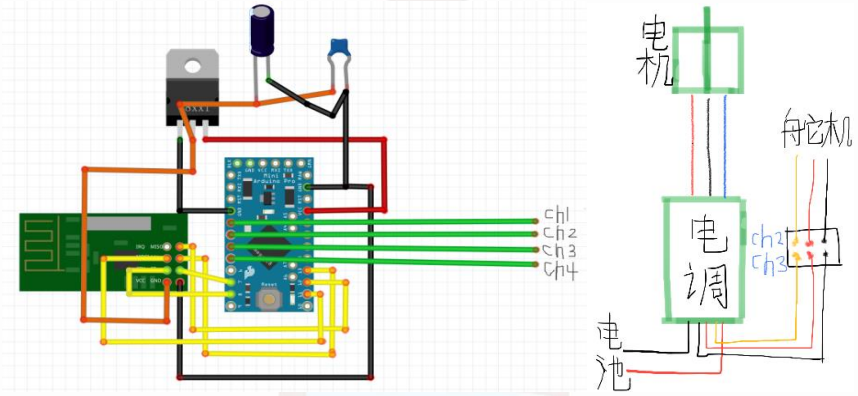
部件	型号	选择原因
电池	25c 2200mah 3s	电压为11.2V，重量相对较小
电机	2440无刷 300W 4500KV	相比同规格的2磁极电机发热更低扭力更强低速线性更好更省电
舵机	SG90	控制方向舵转向
电调	好盈海王40A 无刷	防水，高效散热，大电流
主控	Arduino pro mini	体积小，接线烧录方便
通信模块	NRF24L01P+PA+LNA	此2.4G 无线模块通信距离远，抗干扰能力强，丢包率低
稳压模块	AMS1117 3.3V	给通信模块供电，保证电压稳定
操纵杆	Joystick 模块 360度旋转	操作灵敏，收集4个通道的信号
电容	电解电容25V 100uF	滤除通信模块供电电压信号中的交流分量，以保证信号稳定
	104电容	

三、硬件框图

(1)遥控器接线

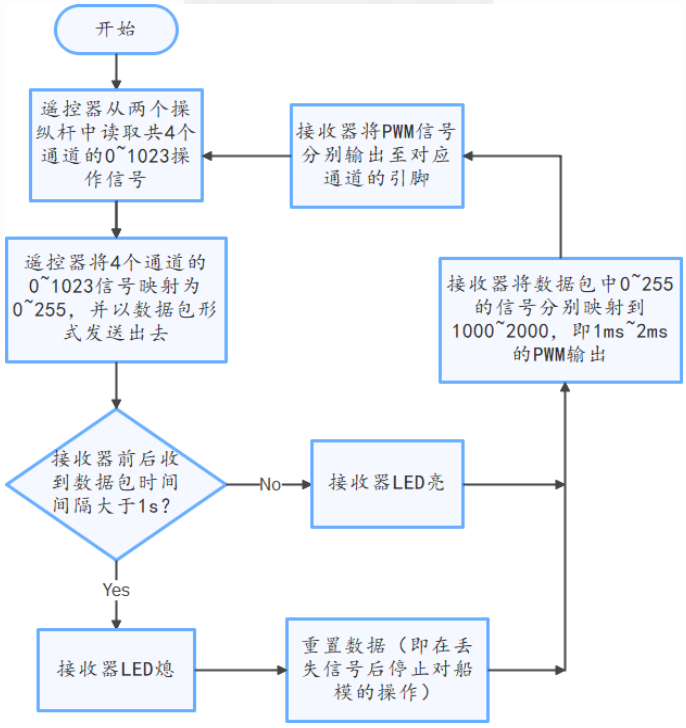


(2)接收器及船内驱动装置接线



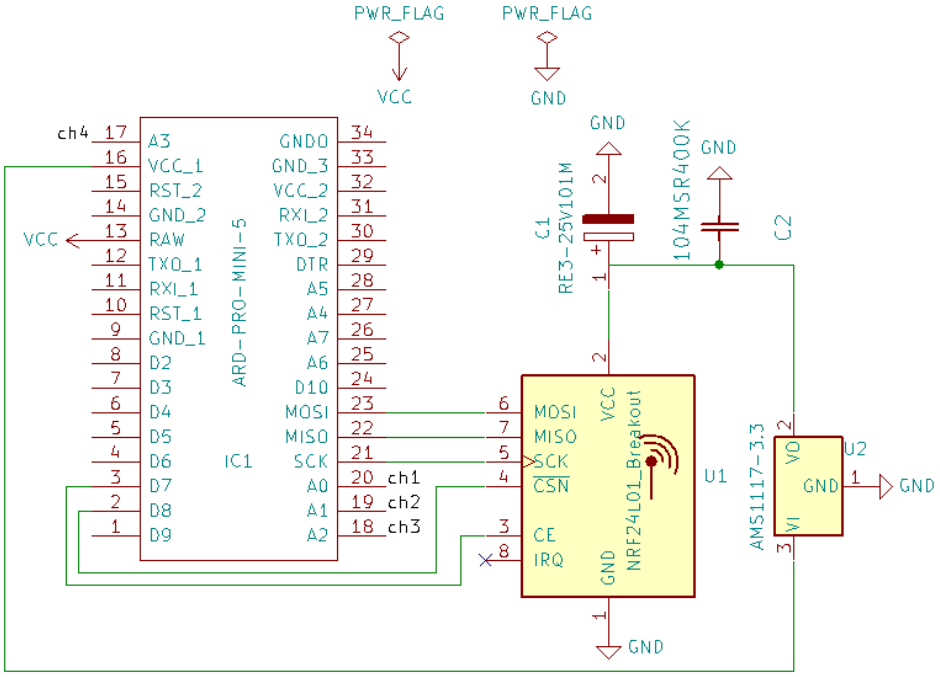
注：CH2接舵机、CH3接电调；使用 fritzing 绘制

四、软件流程

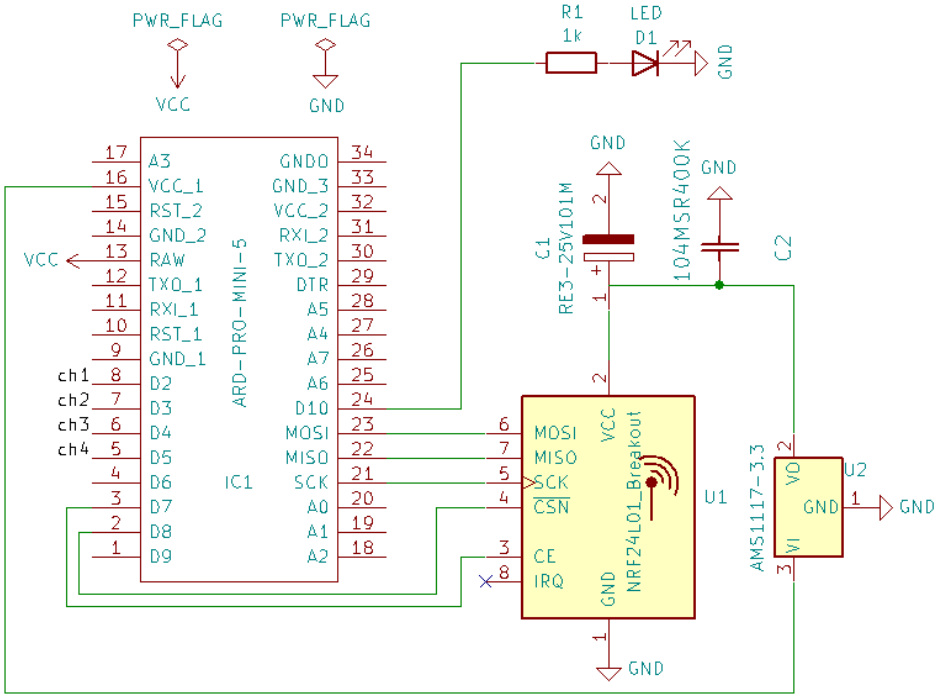


附录一 电路图

(1)遥控器



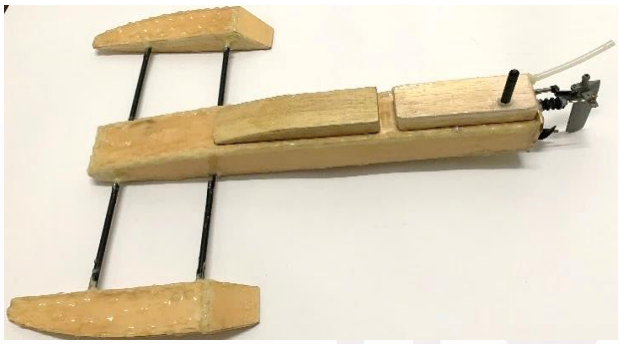
(2)接收器



以上原理图使用 KICAD 绘制

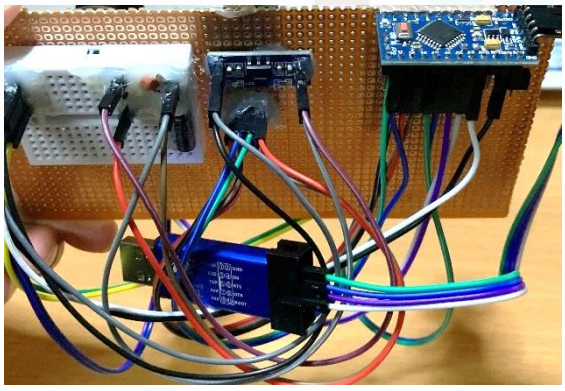
附录二 制作过程及实物展示

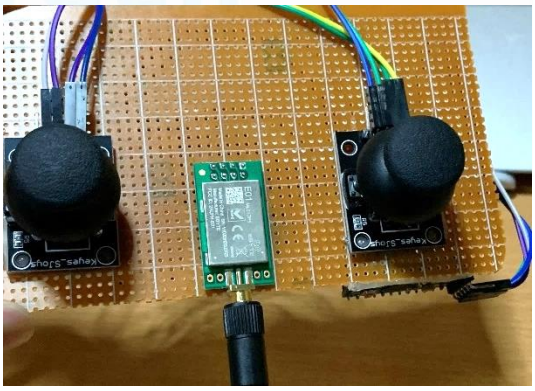
(1)船体



(2)遥控器与接收器

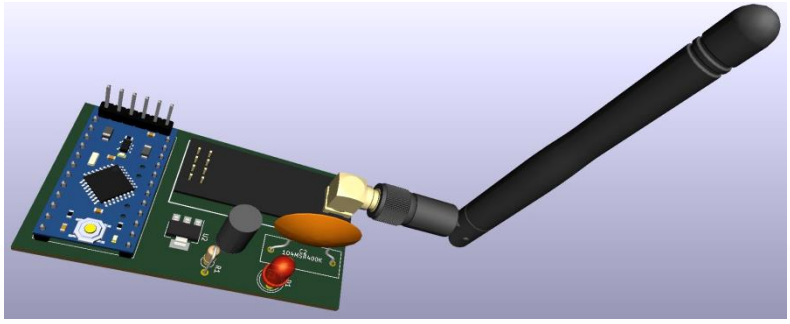
1、遥控器





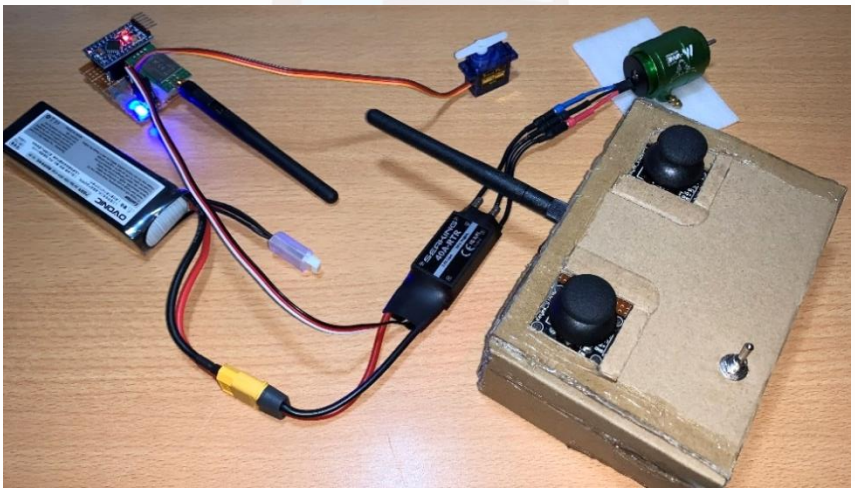
2、接收器

模型图：



实物图：

3、综合



附录三 源代码

(1)遥控器

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
const uint64_t pipeOut = 0x141450196; //与接收器中相同的地址进行通信
RF24 radio(7, 8);                     //SPI通信，引脚对应关系：CE ->7,CSN ->8
struct Signal
{
  byte joystick1_x;
  byte joystick1_y;
  byte joystick2_x;
  byte joystick2_y;
};
Signal data;

void ResetData()
{
  data.joystick1_x = 127;
  data.joystick1_y = 127;
  data.joystick2_x = 127;
  data.joystick2_y = 127;
}

void setup()
{
  radio.begin();
  //radio.setAutoAck(false);
  //radio.setDataRate(RF24_250KBPS);
  radio.setPALevel(RF24_PA_LOW);
  //radio.setChannel(108);
  radio.openWritingPipe(pipeOut); //pipeOut通信地址
  radio.stopListening();          //发射模式
  ResetData();                    //初始化6个通道值
  Serial.begin(115200);

  pinMode(2, INPUT); //正反通道开关为数字输入
  pinMode(3, INPUT);
  pinMode(4, INPUT);
  pinMode(5, INPUT);
}

// 将ADC获取的0~1023转换到0~255
int chValue(int val, int lower, int middle, int upper, bool reverse)
{
  val = constrain(val, lower, upper); //将val限制在lower~upper范围内
  if (val < middle)
    val = map(val, lower, middle, 0, 128);
  else
    val = map(val, middle, upper, 128, 255);
  //return (reverse ? 255 - val : val);
  return (0 ? 255 - val : val);
}

void loop()
{
  // 需要对摇杆的最值、中值进行设置
  data.joystick1_x = chValue(analogRead(A0), 4, 510, 1009, digitalRead(2))
  data.joystick1_y = chValue(analogRead(A1), 6, 530, 1020, digitalRead(3))
  data.joystick2_x = chValue(analogRead(A2), 0, 508, 1020, digitalRead(4))
  data.joystick2_y = chValue(analogRead(A3), 0, 500, 1017, digitalRead(5))

  //将数据发送出去
  radio.write(&data, sizeof(Signal));
}
```

(2)接收器

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <Servo.h>
int ch_1 = 0, ch_2 = 0, ch_3 = 0, ch_4 = 0;
Servo ch1;
Servo ch2;
Servo ch3;
Servo ch4;
struct Signal
{
    byte joystick1_x;
    byte joystick1_y;
    byte joystick2_x;
    byte joystick2_y;
};
Signal data;
const uint64_t pipeIn = 0x141450196; //对频地址，与发射端地址相同
RF24 radio(7, 8);

void ResetData() // 信号丢失时，关闭操作
{
    data.joystick1_x = 127;
    data.joystick1_y = 127;
    data.joystick2_x = 127;
    data.joystick2_y = 127;
}
void setup()
{
    ch1.attach(2);//设置PWM信号输出引脚
    ch2.attach(3);
    ch3.attach(4);
    ch4.attach(5);
    ResetData();//配置NRF24L01模块
    radio.begin();
    radio.setPALevel(RF24_PA_LOW);
    radio.openReadingPipe(1, pipeIn); //与发射端地址相同
    radio.startListening();           //接收模式
    pinMode(10, OUTPUT);              //LED输出
    digitalWrite(10, HIGH);
    Serial.begin(115200);
}
unsigned long lastRecvTime = 0;
void recvData()
{
    while (radio.available())
    {
        radio.read(&data, sizeof(Signal)); //接收数据
        lastRecvTime = millis();           //当前时间ms
    }
}
void loop()
{
    recvData();
    unsigned long now = millis();
    if (now - lastRecvTime > 1000)
    {
        ResetData(); //两次接收超过1s表示失去信号，输出reset值
        //Serial.print("无信号");
        digitalWrite(10, LOW);
    }
    else
    {
        //Serial.print("有信号");
        digitalWrite(10, HIGH);
    }
    ch_1 = map(data.joystick1_x, 0, 255, 1000, 2000); // 将0~255映射到1000~2000，即PWM输出
    ch_2 = map(data.joystick1_y, 0, 255, 600, 2400);
    ch_3 = map(data.joystick2_x, 0, 255, 1000, 2000);
    ch_4 = map(data.joystick2_y, 0, 255, 1000, 2000);
    ch1.writeMicroseconds(ch_1);// 将PWM信号输出至引脚
    ch2.writeMicroseconds(ch_2);
    ch3.writeMicroseconds(ch_3);
    ch4.writeMicroseconds(ch_4);
}
```

附录四 参考博客

① https://howtomechatronics.com/projects/diy-arduino-rc-transmitter/
② https://howtomechatronics.com/projects/diy-arduino-rc-receiver/
③ https://blog.csdn.net/weixin_42268054/article/details/105536264