

摘要

在无人机实训中，我们使用 C 语言编程实现对 LED、电机转速、USB 通信控制，无人机姿态和遥控器摇杆状态的读取，并使用示波器调试程序。最后在大作业综合这些内容用遥控器实现控制无人机的解锁、变速、上锁效果。除了规定的功能，我们也加入了一些**创新性想法**。**防误操作功能**是让无人机在解锁的一秒内的转速慢慢向无人机读取到的瞬时值靠近，防止因遥控器误操作导致在无人机解锁的瞬间转速过快而造成的意外。**转速过低自动停转功能**是当转速落入阈值内时，停止电机的转动，这弥补了原本无法实现电机停转的缺陷。**上锁后转速慢慢下降功能**是防止无人机在高空因突然上锁导致的直接坠落。

关键词 LED 电机转速 USB 通信 防误操作 转速过低自动停转 上锁后转速慢慢下降

1 引言

1.1 研究背景

随着当今时代工程领域全球化、现代化和技术化发展，促进了新型飞行器—无人机的诞生。该领域的新技术近年来不断突破，以及对低成本、多功能军、民用飞行器需求不断增长，使得载人飞行器已经无法满足繁多的任务需求，尤其是长航时侦察和情报收集。在实际运用中，部分载人机已经被性价比较高的无人机所取代。无人机能执行各种危险的任务而使飞行员无需面临生命危险。因此，对军、民用飞行器制造商和各类学术科研机构来说，无人机在航空器中已逐渐发展成炙手可热的研究内容 [1]。无人机是在上世纪二十年代开始应用发展，当时的人们还没有认识到其价值所在，而且受限于当时飞机在很多技术上发展都不成熟，比如定位、电子设备、飞行控制等方面，这就导致了无人机一直处在发展缓慢的阶段。直到上世纪 90 年代末，无人机开始引起了全世界的广泛关注。

1.2 国内外研究现状及发展

目前国外四旋翼无人机的研究正在紧锣密鼓的进行，世界范围内有很多 Quad-Rotor 项目在研，具体如下：

斯坦福大学的 Mesicopter 计划。Mesicopter 是斯坦福大学 Ian Kroo 和 Fritz 教授的研究小组在 NASA 的支持下，为研究微型旋翼技术设计的一款超微型旋翼飞行器。机身 16×16mm 的方形框架，旋翼直径 1.5cm，电机直径 3cm，重 325 克。斯坦福大学的 STARMAC (Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control) 项目 [2]。STARMAC 是由 Gabriel Hoffmann 等人设计的多智能体控制自主旋翼无人机平台，它包含一套惯性测量单元、两个控制器、一个超声波传感器和一个差分 GPS 设备，并通过蓝牙设备（目前已更新为 Wi-Fi）与地面工作站实现通讯。宾西法尼亚大学的 Quad-Rotor 项目，该项目初期利用 HMX-4 作为 Quad-rotor 研究平台。HMX4 最大长度 76cm，重约 700g，在机体下方印有 5 个标志。采用地面摄像头测量跟踪被标记的点的面积与位置，从

而计算获得无人机的位置信息和姿态角信息，角速率则由 3 轴陀螺仪测量获得，主要用于无人机姿态增稳控制。HMX4 的开发团队又研制了一套基于地面和机载双摄像头的视觉定姿与定位系统，进一步提高了姿态测量和位置定位精度。目前（2008 至今）宾夕法尼亚大学 Grasp 实验室 Daniel Mellinger 等人，在 Ascending Technology 的平台上进行研究 [3]。

虽然国际上对四旋翼无人机的研究已经取得了丰硕的成果，并已将研究重点转入智能飞行，并投入商业应用。而目前国内的研究较少，多处于起步阶段。已经有文献报道的研究单位包括南京航空航天大学、北京航空航天大学、上海交通大学、国防科技大学。纵观国内研究现状，国内的四旋翼无人机的研究多集中于建模和控制算法的仿真上，少数有实际飞行实验。少数科研机构虽制作出样机，但因缺乏多旋翼无人机控制上的核心技术与经验，无法得到满意的飞行性能。由于多旋翼飞行器具有广阔的应用前景，目前国内各界人士都提出了对多旋翼无人机的期待，希望未来几年，国内对旋翼无人机的研究能够取得突破，研发出高性能的旋翼无人机。

1.3 微型四旋翼无人机简介

微型四旋翼无人机作为多旋翼无人机最简单最流行的一种，它是一种能够实现原地垂直起飞和悬停、无人驾驶、重复使用、自主飞行、能携带多种设备、执行多种任务的一种飞行器。四旋翼无人机其实是融合了固定翼无人机和单旋翼无人机的特点，它的结构类似于固定翼无人机，具有四个旋翼相互对称的结构。而四旋翼无人机也可以和普通无人直升机一样，可以从水平状态直接切换到垂直状态，从而实现原地垂直起飞和降落。除了上述的这些特点，微型四旋翼无人机还有以下特点：

1. 它能够实现原地垂直起落、空中悬停、原地任意旋转、低空飞行等飞行状态 [4]。
2. 具有飞行速度快、噪声小、飞行效率高、飞行距离远、升限高等特点。
3. 可以在消防灭火、石油开发、灾后人员搜救等恶劣环境下进行全天候工作，可以减少人类生命的伤亡并可以提高其工作效率，从而产生更高的经济效益。
4. 其自身具有四个旋翼的对称结构，旋翼之间不会产生较大的空气动力干扰。还因其自身结构简单紧凑，操作控制简单，易于自学。

由于微型四旋翼无人机本身的机体结构是属于多耦合、非线性、具有干扰敏感特性的欠驱动系统 [5]，所以目前只能通过自动控制器来协调平衡四个旋翼之间的升力来控制其飞行轨迹和飞行姿态。一般情况下，微型四旋翼无人机在空中飞行都会受到各种物理效应和的影响和各种气流的干扰 [6]，从而影响各种测量检测单元的测量精度，进而就会对自动控制器的性能产生影响。这些都使无人机控制器的设计变的更加困难。

近几十年来，微型四旋翼无人机因其具有广泛的应用前景而受到了越来越多人的喜爱和研究，进而掀起了全世界各个高校、研究所及商业公司的研究浪潮。[7]

在民用方面，特别是消费级的微型无人机，其因廉价、操作简单、携带便捷等优点正逐渐进入每个普通家庭。现在四旋翼无人机商机无限，这股热潮还可以看做是载人四旋翼无人机大规模发展的前奏。随着嵌入式技术、智能化和信息化技术的快速发展，与大数据、互联网的结合使微型四旋翼无人机逐渐趋向于功能多样、快速、控制灵活、微型化等方向日新月异地发展，并悄然无息地改变大众生活。无人机的以上发展方向决定了它未来的用途和空间，所以对无人机控制技术的研究是具有重要的现实意义和经济意义。[8]

四旋翼无人机是一种由 4 个输入力产生 6 个自由度方向运动的欠驱动旋翼式无人机[9]。四个旋翼对称地分布在机身 4 角，同一条角线上的两个旋翼以相同的速度、相同的方向旋转，两条对角线上的两对旋翼转动速度相同，方向相反，因此抵消了彼此旋转所产生的扭力矩，提高了旋翼的利用效率。通过控制 4 个旋翼的转速可使无人机实现各种飞行动作。

2 实践内容

本次工程设计训练，主要目标是对无人机技术有一定的理解并能够通过编写程序完成指定的动作，进行了以下几个内容。

1. 程序的安装（略）
2. 无人机和遥控器程序的烧写和激活（略）
3. 简单的工程创建（略）
4. **LED 的控制**

用引脚编写程序来控制 LED 的亮灭状态，加入延迟函数后控制 LED 灯的亮灭时间，从而设计出一个跑马灯的效果。LED 只有亮和灭两个状态，利用人眼视觉残余，在极短的时间内通过设计 LED 的开闭时间实现 LED 灯的亮度控制。在亮度控制的基础上实现从暗到亮、亮到暗的呼吸灯效果。再进一步结合 USB 通信，实现地面站控制 LED 的亮灭。最后在遥控器上用左右按键，以长按短按的形式分别实现控制 LED 的亮灭。

5. **示波器的使用**

延迟函数在无人机控制的编程中经常使用，要实现精确的延迟需要借助软件自身携带的示波器功能，不断调试并在示波器中读取数据，最终使延迟函数接近 0.5 秒。

6. **电机转速的控制**

通过调用已编写的电机转速函数对无人机四个电机转速进行控制，实现 4 个电机的转速不同。加入延迟函数后，实现电机转速的加速减速效果。最后结合 USB 通信实现地面站对电机转速的控制。

7. USB 通讯

在接收到地面站信号后，通过分析信号来显示相应的信息内容。另外结合 LED、无人机电机转速模块实现地面站对 LED 的亮灭，无人机转速的控制。

8. 无人机、遥控器姿态的读取

用 USB 将无人机与地面站连接后，改变无人机物理状态，读取相应传感器信息后获取无人机状态，同样读取遥控器摇杆的状态。

9. 大作业

将左摇杆拉到左下角一秒实现无人机的解锁，解锁后无人机电机缓慢转动，通过左摇杆的上下移动来控制无人机电机的转速。最后长按右摇杆 0.5 秒实现无人机的上锁，电机的停转，再次解锁仍能保持上述功能的使用。

3 研究方案

3.1 STM32 无人机 LED 灯操作

实验一：无人机中有 3 个 LED 灯，分别用 3 个引脚 GPIOC 13 GPIOC 14 GPIOA 8。写程序设置 2 个 LED 灯亮 (GPIOC 13 GPIOA 8)，1 个 LED 灯灭 (GPIOC 14)。

思路 先分别连接 3 个 LED 的引脚，再分别创建 LED 开闭函数，最后在主函数中调用上述函数使其满足题目要求。

列表 1: LED 灯简单控制

```
1 //LED引脚的连接//
2 GPIO_InitStruct.GPIO_Pin = GPIO_Pin_13|GPIO_Pin_14
3 ;
4 GPIO_Init(GPIOC,&GPIO_InitStruct);
5 //LED开闭函数的创建（部分代码）//
6 void led1_on(void)
7 {
8     GPIO_ResetBits(GPIOC,
9     GPIO_Pin_13);
10 }
11 void led1_off(void)
12 {
13     GPIO_SetBits(GPIOC,
14     GPIO_Pin_13);
```

```

14     }
15     //主函数中调用LED灯函数//
16     led1_on();
17     led2_off();
18     led3_on();

```

实验二：调整 delay_ms 函数，使时间接近于 0.5s，设计一个函数 delay_us 函数通过调节参数，使时间不断接近于 0.5s。

思路 创建延时函数，并用 for 语句粗略的定时 1ms。调用延时函数，并使其输入值为 500。因电脑性能的差异以及其他代码也参与运算，因此需要不断更改延时函数中 for 语句的循环次数，直至最接近 0.5ms。

列表 2: 延时函数的创建

```

1     //创建延时1毫秒的函数//
2     void delay_ms(int ms)
3     {
4         int i ,j;
5         for( i=0;i<ms;i++)
6             {
7                 for(j=0;j <10000;j++){ }
8             }
9     }
10    //调用延时函数//
11    delay_ms(500);
12    //根据示波器调试延时函数//
13    void delay_ms(int ms)
14    {
15        int i ,j;
16        for( i=0;i<ms;i++)
17            {
18                for(j=0;j <12053;j++){ }
19            }
20    }

```

实验三：编程跑马灯，或使 3 组 LED 等循环亮或设计自己想要的方式

思路 基于实验一、实验二我们调用 led 函数和延迟函数，实现跑马灯的效果。

列表 3: 跑马灯设计（部分代码）

```

1      for ( i=0;i <2;i++)
2      {
3          led1_on();
4          led2_on();
5          led3_on();
6          delay_ms(1000);
7          led1_off();
8          led2_off();
9          led3_off();
10         delay_ms(1000);
11     }

```

实验四：设计一个其中一个灯为呼吸灯。

思路 LED 灯只有亮和灭两种状态，为了改变亮度可以通过控制 PWM 占空比来实现。使用 For 语句实现 LED 灯从灭到亮，从亮到灭的过程，同时调用延迟函数来延长这个过程，实现缓缓变亮、缓缓变暗的效果。

列表 4: 呼吸灯设计（部分代码）

```

1      for ( i=0;i <100;i++)
2      {
3          led3_on();
4          delay_us(100*i);
5          led3_off();
6          delay_us(100*(100-i));
7      }

```

3.2 无人机 PWM-电机转速控制

实验一：对无人机四个电机转速进行控制，实现 4 个电机的转速不同

思路 直接调用已给的电机转速函数。

列表 5: 电机转速控制

```

1      TIM_SetCompare1(TIM4,50);
2      TIM_SetCompare2(TIM4,100);
3      TIM_SetCompare3(TIM4,150);

```

3.3 USB 与上位机通信

实验一：地面站发送 0xaa,0xaa（通信协议，用于判断）,0x18（数据的作用）时，地面站显示 ABC 地面站发送 0xaa,0xaa,0x19 时，地面站显示 HuiKe。

思路 仿照已给的 ANO_DT_Send_User 函数创建地面站传送 Huike 函数，再加入 if 语句来判断地面站应该显示哪个信息。

列表 6: 发送 huike

```

1      //caption=发送huike//
2      void ANO_DT_Send_Huike()
3      {
4          u8 _cnt=0; //u8=0-255
5          vs16 _temp;
6          u8 sum = 0;
7          data_to_send[_cnt++]=0xAA; //0=0xAA
8          data_to_send[_cnt++]=0xAA; //1=0xAA
9          data_to_send[_cnt++]=0xf1; //2=0xf1
10         data_to_send[_cnt++]=0; //3=0
11         data_to_send[_cnt++]='h';
12         data_to_send[_cnt++]='u';
13         data_to_send[_cnt++]='i';
14         data_to_send[_cnt++]='k';
15         data_to_send[_cnt++]='e';
16         data_to_send[3] = _cnt-4;
17         for( u8 i=0;i<_cnt;i++)
18             sum+= data_to_send[i];
19         data_to_send[_cnt++]=sum;
20         ANO_DT_Send_Data(data_to_send, _cnt); //校验
           和;
21     }
22     //分析接收信号内容//
23     if (data[1]==0xaa && data[2]==0x18)
24     {

```

```

25         ANO_DT_Send_User();
26     }
27     else if (data[1]==0xaa && data[2]==0x19)
28     {
29         ANO_DT_Send_Huikue();
30     }

```

实验二：地面站发送 0xaa,0xaf,0x01 时，无人机的 LED3 亮地面站发送 0xaa,0xaf,0x00 时，无人机的 LED3 灭。

思路 基于实验一，同样加入 if 语句判断 LED 灯是亮是灭。

列表 7: 根据信号内容控制 LED 开闭

```

1         if (data[1]==0xaf &&
2             data[2]==0x01)
3         {
4             led1_on();
5             led2_on();
6             led3_on();
7         }
8         else if (data[1]==0xaf &&
9                 data[2]==0x00)
10        {
11            led1_off();
12            led2_off();
13            led3_off();
14        }

```

实验三：地面站发送 0xaa,0xaf,0x20,0x01,0x00,0x01,0x50,0x00,0x00,0x02,0x00 时，无人机的电机转速分别为 100,150,0,200。

思路 分析发送的信息，根据协议内容给电机不同转速。

列表 8: 地面站控制电机转速（部分代码）

```

1         if (data[1]==0xaf && data[2]==0x20)
2         {
3             for (i=0;i<5;i++)
4             {

```



```

5         ANO_DT_Send_User();
6     }
7 }
8 else if (data[1]==0x01 && data[2]==0x00)
9 {
10     TIM_SetCompare1(TIM4,100);
11     TIM_SetCompare2(TIM4,100);
12     TIM_SetCompare3(TIM4,100);
13     TIM_SetCompare4(TIM4,100);
14 }
15 else if (data[1]==0x01 && data[2]==0x50)
16 {
17     TIM_SetCompare1(TIM4,150);
18     TIM_SetCompare2(TIM4,150);
19     TIM_SetCompare3(TIM4,150);
20     TIM_SetCompare4(TIM4,150);
21 }

```

3.4 IMU

实验一：获取无人机状态

思路 读取无人机传感器信息后传送给地面站并显示。

列表 9: 无人机状态的获取

```

1     while (1)
2     {
3         MPU_Data_Prepare(0.003f);
4         AK8975_Read(0.02f);
5         delay_ms(100);
6         ANO_DT_Send_Senser(sensor.Acc.x,sensor.Acc.y,
7                             sensor.Acc.z,sensor.Gyro.x,sensor.Gyro.y,
8                             sensor.Gyro.z, 0, 0, 0);
9         My_IMU_update(0.006f);
10        ANO_DT_Send_Status(imu_data.roll,imu_data.pit,
11                            imu_data.fus_yaw,wcz_h_fus.out*100,2,

```

```

9         fly_ready);
        ANO_Send_Status(imu_data.rol,imu_data.pit,
10        imu_data.fus_yaw,0,0,0);
    }

```

实验二：获取遥控器摇杆状态

思路 通过通信建立地面站和遥控器的连接，同时在主函数中对遥控器中的按键和 ADC1 模块进行初始化，调用 DT_Send_RC_Data() 即可。

列表 10: 遥控器摇杆状态的获取

```

1     int main()
2     {
3         led_init();
4         KEY_Init();
5         ADC1_Init();
6         NVIC_PriorityGroupConfig(NVIC_PriorityGroup_4)
7         ;
8         Stick_Scan();
9         DT_Send_RC_Data(Rc.THR,Rc.YAW,Rc.ROL,Rc.PIT,Rc
        .AUX1,Rc.AUX2,Rc.AUX3, Rc.AUX4, Rc.AUX5, Rc
        .AUX6);
    }

```

3.5 遥控器的控制

实验一：短按实现左右键控制左右灯

思路 先连接遥控器 LED 灯的引脚，再创建控制 LED 开闭的函数，最后用 if 语句判断遥控器是左键按还是右键按，并调用相应的 LED 灯函数。

列表 11: 短按遥控器按键

```

1     //遥控器LED灯引脚的连接//
2     GPIO_InitStruct.GPIO_Pin = GPIO_Pin_4 | GPIO_Pin_5
3     ;
4     GPIO_Init(GPIOB,&GPIO_InitStruct);
5     //遥控器LED开闭函数的创建（仅部分）//
    void led_1_on(void)

```

```

6      {
7      GPIO_ResetBits(GPIOB, GPIO_Pin_4);
8      }
9      //判断遥控器左右按键状态//
10     while(1)
11     {
12         if (Mode_Read == RESET)
13         {
14             led_1_on();
15             led_2_off();
16         }
17         else if(Fun_Read == RESET)
18         {
19             led_2_on();
20             led_1_off();
21         }
22         else {}
23     }

```

实验二：长按实现左右键控制左右灯

思路 在短按的基础上，加入 for 循环语句，在三秒内不断查询这期间按键是否有变化。

列表 12: 长按遥控器按键（长按左键代码）

```

1      //不断查询遥控器状态是否更改//
2      if (Mode_Read == RESET)
3      {
4          int flag1 = 0, flag2 = 0;
5          for(i=0; i<3000; i++)
6          {
7              for(j=0; j<1000; j++)
8              {
9                  if (Mode_Read != RESET)
10                 {
11                     flag1 = 1;

```

```

12         }
13     }
14 }
15 if (flag1 == 1)
16     continue;
17 led_1_on();
18 led_2_off();
19 }

```

3.6 大作业

功能一：把左摇杆拉到无人机的左下角 1 秒发送指令给无人机，无人机收到指令后，解锁并把 4 个电机缓慢转动

思路 读取左摇杆数据，当读出左摇杆拉到左下角后就在 1 秒内不断判断是否一直在左下角，若满足则发送解锁信号，无人机收到信号后，分析信号内容后慢慢转动。

列表 13: 功能一

```

1 // 查询是否满足解锁条件 //
2 if (Rc.THR<1200&&Rc.YAW<1200)
3 {
4     my_delay_ms(50);
5     flag++;
6     if (flag==20)
7     {
8         if (Rc.THR<1200&&Rc.YAW<1200)
9         {
10             DT_Send_isAutofly(1);
11             tep=1;
12             flag = 0;
13         }
14 // 无人机接收到解锁信号后，电机缓慢转动 //
15 if (flag_unlock == 1)
16 {
17     u8 i = 0;
18     for (i = 0; i < 4; i++)

```

```

19         motor_speed[i] = THR/10;
20     }

```

功能二：左摇杆往上推无人机转速会加快不至于飞起来，左摇杆往下推无人机转速会变慢

思路 无人机解锁后能持续接收遥控器左摇杆信息，并根据信息的变化改变电机转速。

列表 14: 功能二

```

1      // 遥控器持续向无人机发送左摇杆的信息 //
2      while{
3          DT_Send_RC_Data(Rc.THR);
4      }
5      // 在解锁状态下，无人机将根据接受信息实时改变电机转速 //
6      if(flag_unlock == 1)
7      {
8          u8 i = 0;
9          for(i = 0; i < 4; i++)
10             motor_speed[i] = THR/10;
11     }

```

功能三：按住右键无人机上锁，电机关闭

思路 当遥控器读取到右摇杆被按下后，发送上锁信号给无人机，无人机收到并且分析后，上锁无人机，并关闭电机。

列表 15: 功能三

```

1      // 查询到遥控器右键长按发送上锁信号 //
2      if(Mode_Read==RESET)
3      {
4          my_delay_ms(50);
5          flag1++;
6          if(flag1==1)
7          {
8              if(Mode_Read==RESET)
9              {
10                 DT_Send_isAutofly(0);

```

```

11         DT_Send_isUnlock(0);
12         tep=0;
13         flag1=0;
14     }
15 }
16 }
17 //无人机收到上锁信号//
18 if(flag_autofly == 0 && flag_unlock == 0)
19 {
20     led2_off();
21     u8 i = 0;
22     for(i = 0; i < 4; i++)
23         motor_speed[i] = 0;
24     set_motor_speed(motor_speed);
25 }

```

4 仿真或实验结果

4.1 LED 灯实现跑马灯和呼吸灯

在主函数中，通过调用延迟函数和 for 循环语句，实现一灯亮两灯灭、两灯亮一个灯灭三灯循环亮灭的效果，从而实现 LED 灯闪烁和跑马灯功能。利用人眼视觉残余，在极短的时间内通过对 LED 灯亮灭时间进行控制，从而实现对 led 灯的亮度控制，实现 LED 灯从暗慢慢变亮、亮慢慢变暗的呼吸灯效果。

4.2 示波器波形结果图

借助软件自身的示波器，在 delay_ms 函数中通过不断调整参数 j 改变循环次数，使时间接近 0.5s。最终结果 j=12053 时，0.5s 约等于 0.502221s。

4.3 电机转速控制

通过配置定时器可直接输出 PWM 波，在主函数中调用 TIM_SetCompare() 函数实现对电机转速的控制，通过示波器仿真成功输出目标的 PWM 波，可同时输出四路不同占空比的 PWM 波，分别代表飞行器上四个螺旋桨的转速，可用来调整占空比而改变螺旋桨

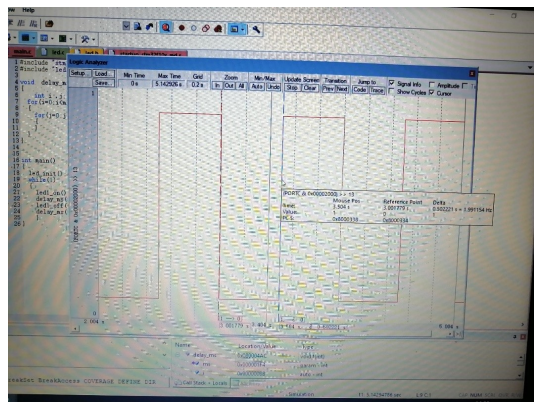


图 4.1: 示波器波形结果图

转速，实现无人机的平稳起飞。当设置 4 个电机的转速分别为 50、100、150、200 时，对应的 4 路不通占空比的 PWM 波如下：

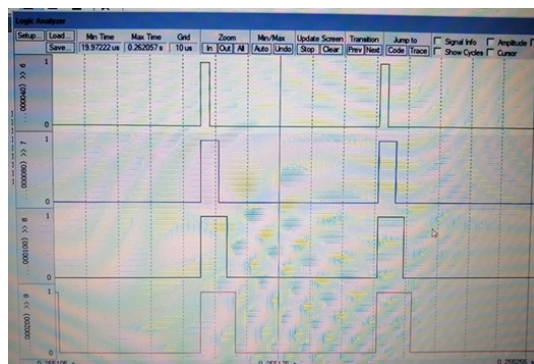


图 4.2: 电机转速控制 PWM 波形图

4.4 USB 通信

通过 USB 通信可以将飞机和遥控器与地面站进行连接，使用上位机在地面站中查看无人机姿态和遥控器摇杆数值等数据。同时也可通过通信实现上位机发送指令控制无人机。在无人机主函数中调用接收和解析遥控数据的函数，通过上位机发送不同的指令成功实现地面站显示 ABC、HuiKe、点亮和关闭无人机的 LED3 小灯、启停无人机和改变无人机电机转速的功能。为之后通过无人机和遥控器进行通信从而实现改变摇杆数值达到控制无人机转速的功能打下基础。

4.5 无人机、遥控器姿态的获取

无人机姿态的获取是通过读取无人机上陀螺仪和电子罗盘两个传感器的数据而得到的。将无人机、遥控器分别与地面站建立连接，通过改变无人机姿态和对遥控器摇杆进行

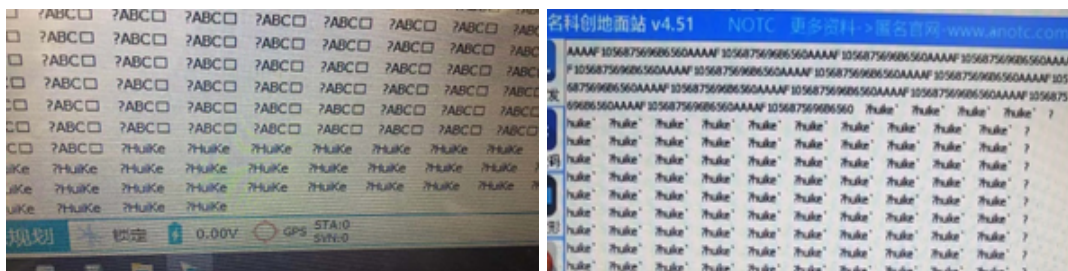


图 4.3: 地面站发送 ABC、HuKe

控制，从而可以从地面站上的飞控状态上读取相应的数值。



图 4.4: 无人机、遥控器姿态的获取

4.6 大作业

设置信道，在主函数中，将无人机和遥控器的信道设置为 34，实现遥控器与无人机信道上的连接；在通信模块上，实现无人机和遥控器功能上的对接。飞行器解析遥控器通过无线通信传来的数据，在遥控器的 data_transfer 文件中写入将要发送出的自定义的指令，并在飞行器的 data_transfer 文件中写入对接收到自定义指令时做出的相应动作。实验结果为遥控器左摇杆拉到左下方一秒，无人机收到指令并亮灯解锁，电机慢慢转动。同时通过改变左摇杆上下坐标位置对电机转速进行控制。最后长按右键实现无人机的上锁，电机停转。上锁后需再次解锁才能控制无人机电机的转速。

5 结论

通过这次工程设计训练，我了解了无人机多种多机器人系统控制以及有限时间控制的方法，目前关于有限时间控制技术的研 究正处在快速发展中，但由于有限时间控制系统本身的非连续性，导致有限时间控制系统的分析与综合问题比较困难，所以在之后的毕业论文中，我也会更加深入地去探索关于有限时间控制系统的方法。

参考文献

- [1] 伍赛特. 无人机技术特点研究及未来发展趋势展望 [J]. 现代制造技术与装备, 2019, 43(5): 678-682.
- [2] Gabe Hoffmann,Dec Gorur Rajnarayan,Steven L.Waslander.The Stanford Testbed of autonomous Rotorcraft for Multi Agent Control(STARMAC). Digital Avionics Systems Conference,2004,12(02):110 121.
- [3] Daniel Mellinger,Nathan Michael, Michael Shomin,Vijay Kumar. Recent advances in quadrotorcapabilities, IEEEinternational conferernce onRobitics and Automation,Shanghai,China,2011: 2964 2965.
- [4] 王福超. 四旋翼无人飞行器控制系统设计与实现 [D]. 哈尔滨工程大学,2013.
- [5] 宁鑫. 四旋翼无人机控制方法研究 [D]. 哈尔滨工业大学, 2017.
- [6] 胡琼. 无人机驾驶航空飞行器飞行控制方法研究 [D]. 北京理工大学,2015.
- [7] 单海燕. 四旋翼无人直升机飞行控制技术研究,[硕士学位论文]. 南京: 南京航空航天大学,2008.
- [8] 宿敬亚, 樊鹏辉, 蔡开元. 四旋翼飞行器的非线性 PID 姿态控制. 北京航空航天大学学报,2011,(09):1054 1058.
- [9] Robert Mahony, Vijay Kumar, Peter Corke. Multirotor Aerial Vehicles: Modeling,Estimation, and Control of Quadrotor[J]. Proceedings of IEEE ROBOTICS & AUTOMATION MAGZINE, 2012 19(3):20-32.
- [10] 岳基隆, 张庆杰, 朱华勇. 微小型四旋翼人机研究进展及关键技术浅析 [J]. 电光与控制, 2010, 10:46-52.
- [11] Euston M, Coote P, Mahony R, et al. A complementary filter for attitude estimation of a fixed-wing UAV[C]//2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2008: 340-345.
- [12] Mauricio Alejandro Lotufo,Luigi Colangelo,Carlos Perez-Montenegro,Enrico Canuto,Carlo Novara. UAV quadrotor attitude control: An ADRC-EMC combined approach[J]. Control Engineering Practice,2019,84.
- [13] Bristeau P J, Callou F, Vissiere D, et al. The navigation and control technology inside the ar. drone micro uav[J]. IFAC Proceedings Volumes, 2011, 44(1): 1477-1484..