

Study on Obstacle Avoidance Flight System of Multi-Rotor UAV Based on Pixhawk

Hang Zhao, Lifeng Wang

Field Bus Technology & Automation Lab, North China University of Technology, Beijing
Email: zh_ncut@163.com, lf_wang@ncut.edu.cn

Received: May 2nd, 2017; accepted: Jun. 10th, 2017; published: Jun. 13th, 2017

Abstract

Autonomous obstacle avoidance is an important guarantee for the successful completion of the mission, and it reflects the intelligence and security of the aircraft. This article aims to study the obstacle avoidance technology of multi-rotor UAV based on the cross type quadrotor and Pixhawk which is the open source flight control system. A simple and efficient obstacle avoidance system is developed. The flight test shows that the obstacle avoidance system can realize the obstacle avoidance of UAV.

Keywords

Pixhawk, Quadrotor, Obstacle Avoidance

基于Pixhawk的多旋翼无人机避障飞行系统研发

赵航, 王立峰

北方工业大学现场总线技术及自动化实验室, 北京
Email: zh_ncut@163.com, lf_wang@ncut.edu.cn

收稿日期: 2017年5月2日; 录用日期: 2017年6月10日; 发布日期: 2017年6月13日

摘要

飞行器的自主避障是顺利完成飞行任务的重要保证, 同时在很大程度上体现了飞行器的智能性和安全性。本文以Pixhawk开源飞控系统为基础, 以“X”型四旋翼飞行器为平台, 对多旋翼飞行器自主避障技术进行研究, 开发出一款简单高效避障系统。飞行试验表明该避障系统可实现飞行器避障飞行。

关键词

Pixhawk, 四旋翼飞行器, 避障

Copyright © 2017 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

近年来, 多旋翼无人机因其机动灵活、垂直起降、自主飞行等特性, 在军用领域、农业领域和工业领域等方面得到了广泛的应用。然而, 由于飞行环境与飞行任务的日趋复杂化, 无人机在执行飞行任务中遇到障碍物时, 如何实现快速高效的避障, 保证无人机的安全性, 提高无人机的智能性, 正成为无人机发展的趋势。

国内外很多学者和无人机公司对多旋翼飞行器避障系统进行了研究, 例如文献[1]基于 Arduino UNO 采集超声波数据和遥控器信号, 并根据超声波测量的数据对遥控信号进行逻辑判断后, 发送给飞控系统进行避障, 该避障系统具备多适配性的特点。文献[2]对无人机编队避障方法进行研究, 首先通过构建 Voronoi 图, 利用 K 路径算法为无人机找到多条备选路径, 然后建立协同函数, 为无人机规划出最优的障碍物规避航迹。文献[3] [4]针对无人机电力巡线过程中避障和路径规划问题进行研究。文献[5]基于 Pixhawk 和改进人工势场法(IAPF), 提出适用于桥梁检测的无人机自动避障路径规划和实现。文献[6] [7]采用多传感器数据融合, 检测飞行器周围环境信息, 对飞行器避障飞行以及航迹规划进行研究。另外, 国内外各大无人机公司相继推出具备避障功能的飞行器, 如零度“Xplorer 2”、DJI“Guidance”、昊翔“Typhoon H RealSense”等。

本文以 Pixhawk 开源飞控系统为基础, 并以“X”型四旋翼飞行器为平台, 对多旋翼飞行器如何实现自主避障进行研究。

2. 系统架构整体设计

本文无人机避障系统整体实现流程如图 1 所示, 其中, 测距传感器采用 HC-SR04 超声波传感器, 实时检测飞行器周围环境信息; 基于 Arduino Mega2560 开发板, 设计出一款带有 CAN 总线和串口两种通讯接口的数据采集模块, 用于采集超声波数据和实现与 Pixhawk 飞控的通讯; 在搭载 APM 飞行控制栈的 Pixhawk 飞控系统中完成上层避障应用的开发。

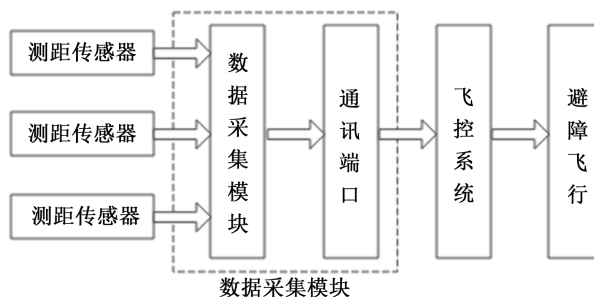


Figure 1. Flow chart of obstacle avoidance

图 1. 避障实现流程图

3. 硬件设计

本文基于 Arduino Mega2560 开发板, 设计出一款带有 CAN 总线接口和串口的数据采集模块, 主要完成两方面的工作: 1. 采集超声波数据; 2. 通过 CAN 总线或串口的方式与 Pixhawk 飞控进行通讯, 完成数据交换。图 2 为主芯片 Atmega2560 外围电路、串口和超声接口原理图。

CAN 总线接口实现如图 3 所示, 其中, CAN 控制器采用 MCP2515, 通讯速率可达 1 Mb/s, 通过 SPI 的通讯方式与 MCU 连接; CAN 收发器采用 TJA1050, 可为总线提供差动发送性能, 为 CAN 控制器提供差动接收性能。

模块还挂载有 ATmega16u2 芯片用于虚拟一个 USB-Serial 转换器, 可实现模块与计算机的通信和程序下载。

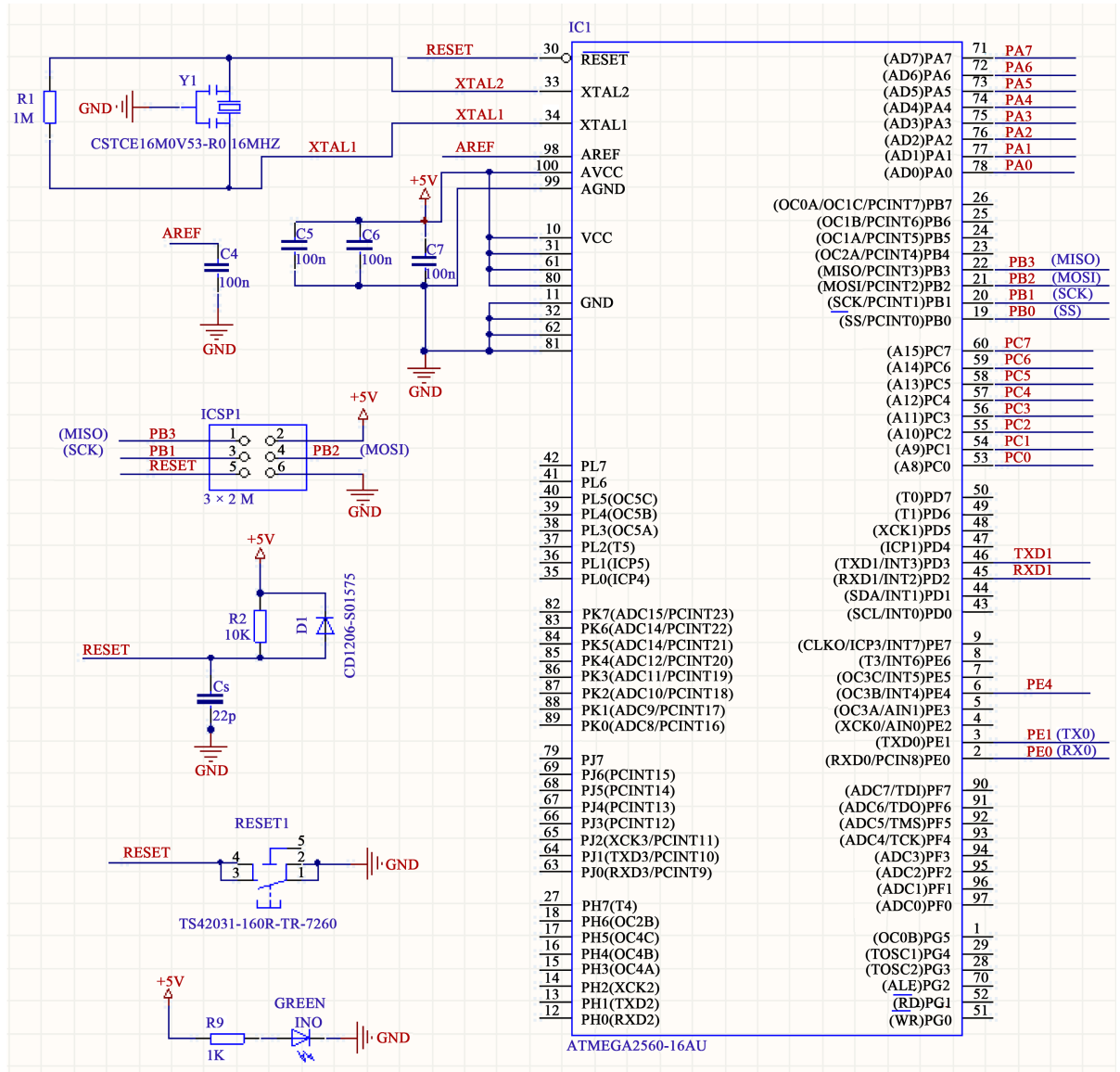


Figure 2. Serial port and ultrasonic interface diagram

图 2. 串口和超声波接口图

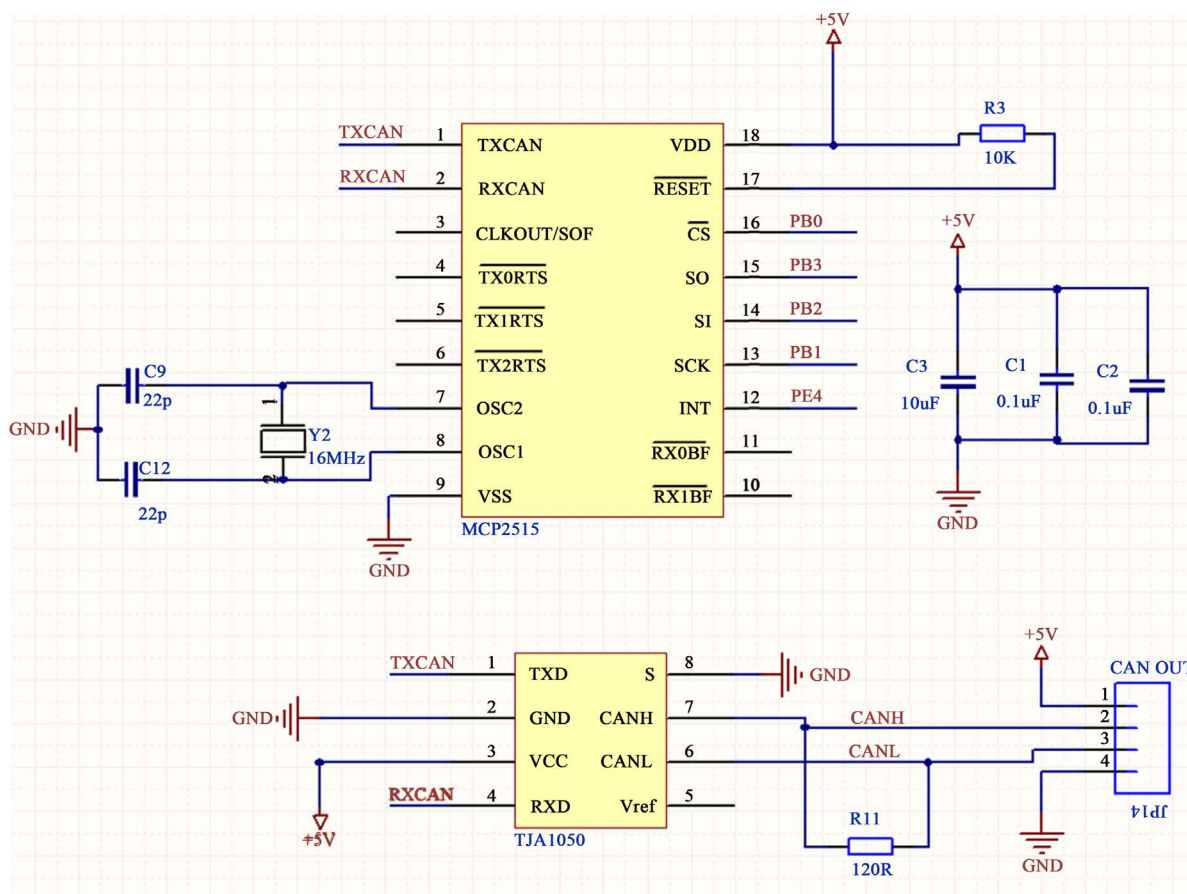


Figure 3. CAN interface diagram
图 3. CAN 总线接口图

4. 软件设计

4.1. 超声波数据采集与发送

本文使用 Arduino 进行超声波数据采集和发送。选用的超声波模块为 HC-SR04，可实现对 2 cm~450 cm 距离内平面相对比较平整的物体进行检测。该模块有四个引脚，分别为 Trig、Echo、VCC、GND。首先定义 Trig、Echo 端连接的引脚；进行串口初始化，设置串口通信波特率；配置 Trig 引脚为输出，Echo 引脚为输入，并在 loop() 循环中对 trig 引脚持续输出 10 us 高电平，获得 echo 引脚高电平持续的时间；计算超声波传感器测量的距离；调用滤波器，对测量的数据进行滤波处理；串口输出滤波后的数据，具体实现流程如图 4 所示。

4.2. Pixhawk 串口接收

1. 自定义主题

Pixhawk 飞控系统中封装了一个跨平台无锁 publish/subscribe 模式的对象请求代理(uORB)，用于完成进程间通信和数据交换。uORB 将进程间通信的数据命名为“主题”，“主题”中定义了通信的数据结构，通过“主题”的发布和订阅完成进程间的数据交换，如图 5 所示。Pixhawk 飞控系统中所有通用接口标准主题全部定义在 ardupilot/modules/PX4Firmware/msg 目录下的.msg 文件中，编译飞控程序过程中，

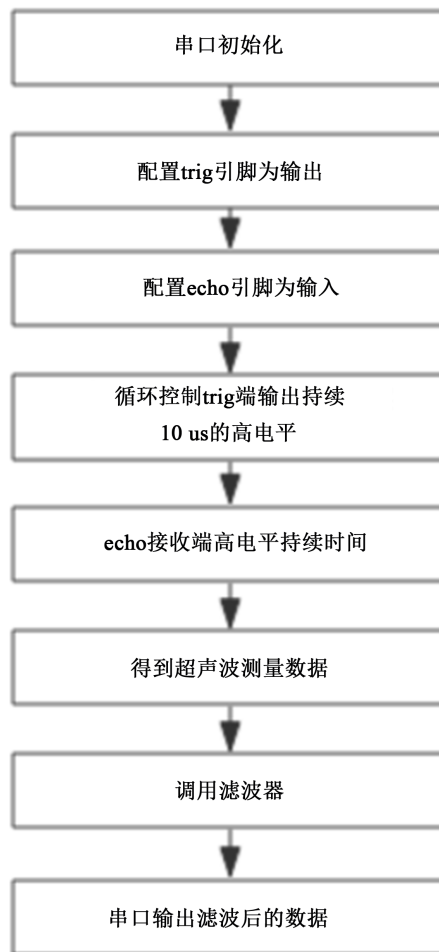


Figure 4. Flow chart of distance measurement
图 4. 测距流程图



Figure 5. Topic publish/subscribe
图 5. 主题发布/订阅

通过调用工具 Python, 检索该目录下的 .msg 文件, 并以 msg.h.template 文件中定义的模板, 将所有 .msg 文件转换生成 .h 文件, 并保存在目录 PX4Firmware/src/modules/uORB/topics 下, 该 msg.h 实现对主题的数据结构定义, 通过调用 ORB_DECLARE() 完成了主题的声明。

本文使用的是自定义主题, 则需要在 ardupilot/modules/PX4Firmware/msg 目录下新建一个名为 serial_sonar.msg 文件, 在该文件中定义 Pixhawk 串口接收到所有超声波数据。

2. 定义主题

主题的发布需完成三个独立又相关联的动作:调用 `ORB_DEFINE()`定义主题;调用 `orb_advertise()`公告主题;调用 `orb_publish()`发布主题。

Pixhawk 中所有标准主题的定义均封装在 `ardupilot/modules/PX4Firmware/src/modules/uORB/objects_common.cpp` 文件中,则本文需要在该文件中完成 `serial_sonar` 主题定义。

3. 创建新的任务进程

Pixhawk 飞控系统使用具有多任务特性的 NuttX 实时操作系统,在飞控上电之后, NuttX 完成底层任务、设备等初始化工作。Pixhawk 飞控系统所有的任务以模块化的形式进行封装,均在 `ardupilot/modules/PX4Firmware/src/module` 目录下。本文需要在该目录下创建一个新的任务模块文件夹,并在该文件夹中定义任务实现文件和 `.mk` 文件。任务实现文件中通过调用 `px4_task_spawn_cmd()`函数创建新的任务进程。其中,在入口函数中完成了串口配置、波特率设置、主题公告和发布等工作,具体实现流程如图 6 所示。而在 `.mk` 文件中需要定义该模块编译指令 `MODULE_COMMAND` 和依赖文件,以便在编译飞控程序时进行调用和执行。

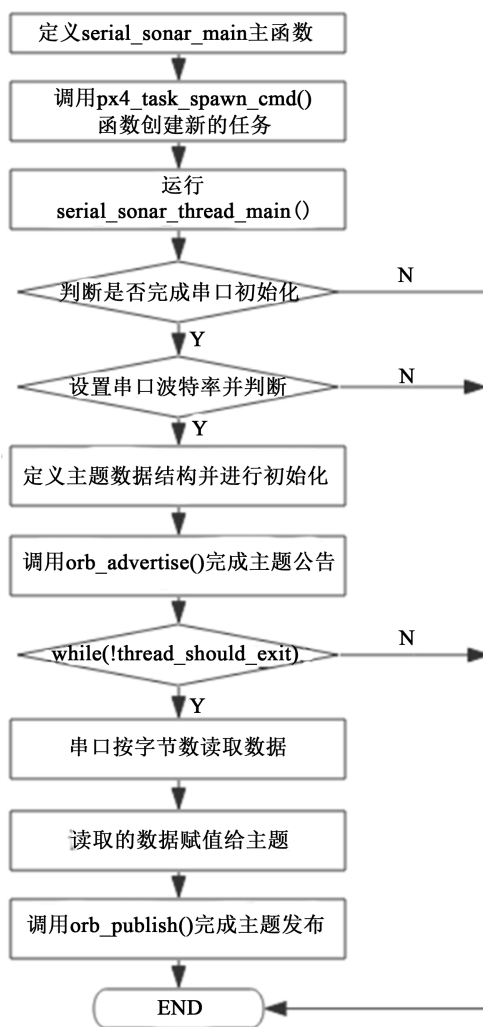


Figure 6. Flow chart of Pixhawk serial port program

图 6. Pixhawk 串口接收程序流程图

4. 加入 Pixhawk 编译系统

Pixhawk 借助 make 工具实现对飞控程序的编译。Makefile 文件中定义了程序编译规则、文件包含和调用关系等。在 px4_targets.mk 文件中, 有以下这样几行程序, 规定了飞行固件编译需要哪些底层驱动及任务模块。

```
# we have different config files for V1 and V2
PX4_V1_CONFIG_FILE=$(MK_DIR)/PX4/config_px4fmu-v1_APM.mk
PX4_V2_CONFIG_FILE=$(MK_DIR)/PX4/config_px4fmu-v2_APM.mk
PX4_V4_CONFIG_FILE=$(MK_DIR)/PX4/config_px4fmu-v4_APM.mk
```

本文以“X”型四旋翼飞行器为飞行平台, 其固件编译指令为 make px4-v2, 在解析编译指令的过程中确定了飞控板类型为 px4fmu-v2, 则上述程序最终只执行 config_px4fmu-v2_APM.mk 文件中包含的内容。则本文需要在该 mk 文件中, 加入自定义的串口接收任务模块。

4.3. 上层避障实现

上层避障指的是如何在悬停模式(Loiter)下实现飞行器避障飞行, 需要解决的问题有: 1. 如何将接收到的超声波数据传输至 ArduPilot 应用中; 2. 如何在 Loiter 飞行模式下编写具体的避障控制逻辑。

ArduPilot 应用程序接收超声波数据具体实现流程如图 7 所示。

1. Pixhawk 飞控系统依赖的库可以分为三类: 核心库, 如 AP_AHRS 捷联惯导库、AP_Motors 电机控制信号解算库、AC_AttitudeControl 姿态位置控制库等; 传感器库, 如 AP_InertialSensor 陀螺仪和加速度计数据读取库、AP_Baro 气压计接口库等; 其他库, 如 AP_Mission 用于 EEPROM 数据存储库、AP_Camera 摄像机控制库等, 这些库文件集中在 ardupilot/libraries 目录下。而在 ardupilot/ArduCopter 目录下有 make.inc 这样一个文件, 该文件中定义了飞控系统所依赖的所有库文件, 同时在 sketch_sources.mk 文件中有如下程序, 完成 make.inc 文件中库列表解析, 并将所有库的源文件包含进编译系统中。

```
# get list of libraries from make.inc
include$(MAKE_INC)
LIBTOKENS:= $(LIBRARIES)
# Find sketchbook libraries referenced by the sketch.
# Include paths for sketch libraries
SKETCHLIBS:= $(wildcard $(addprefix
$(SKETCHBOOK)/libraries/, $(LIBTOKENS)))
SKETCHLIBNAMES:= $(notdir $(SKETCHLIBS))

SKETCHLIBSRCDIRS := $(SKETCHLIBS)$(addsuffix
/utility, $(SKETCHLIBS))
SKETCHLIBSRCS:=$(wildcard $(foreach suf-
fix, $(SRCSUFFIXES), $(addsuffix /$(suffix), $(SKETCHLIBSRCDIRS))))
SKETCHLIBOBS:= $(addsuffix .o, $(basename $(subst
$(SKETCHBOOK), $(BUILDROOT), $(SKETCHLIBSRCS))))
SKETCHLIBINCLUDES := -I$(SKETCHBOOK)/libraries/
-I$(BUILDROOT)/libraries/ -I$(BUILDROOT)/libraries/GCS_MAVLink/
SKETCHLIBSRCSRELATIVE:= $(subst $(SKETCHBOOK)/, , $(SKETCHLIBSRCS))
```

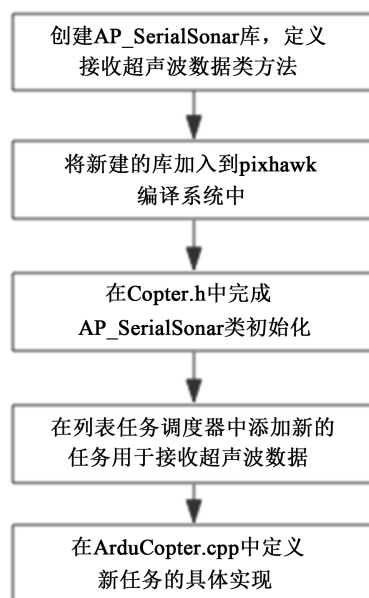


Figure 7. Flow chart
图 7. 流程图

本文首先需要在 `ardupilot/libraries` 目录下创建新的 `AP_SerialSonar` 超声波传感器接口库, 定义 `AP_SerialSonar` 类和 `get_sonar_data()` 接收超声波数据函数, 其实现流程如图 8 所示。首先, 在 `AP_SerialSonar` 类初始化时, 调用 `orb_subscribe()` 订阅主题, 通过 `px4_poll()` 监控 `px4_pollfd_struct_t` 结构体指定的文件描述符, 当出现有数据可读(POLLIN)合法事件时, 调用 `orb_copy()` 获取订阅的主题中的数据并将该数据保存到指定的 `buffer` 中。

其次, 将 `AP_SerialSonar` 库加入到飞控编译系统中, 即在 `make.inc` 文件中添加 `LIBRARIES+= AP_SerialSonar`。

2. Pixhawk 中所有类声明全部封装在 `ardupilot/ArduCopter/Copter.h`, 而飞行控制栈初始化工作在 `init_ardupilot()` 中完成, 为保证系统完整性和统一性, 本文需要在以上两个地方完成 `AP_SerialSonar` 类声明和初始化。

3. 在飞行控制栈的列表任务调度器中, 添加新任务用于更新接收的超声波数据。本文添加的新任务为 `SCHED_TASK(update_serial_sonar,10,100)`, 其中任务名为 `update_serial_sonar`, 调用频率为 10 Hz, 期望运行最长时间为 100 ms。

4. 添加完新任务之后, 需要在 `ArduCopter.cpp` 中定义任务的具体实现。调用 `AP_SerialSonar` 类中用于接收超声波数据的函数 `get_sonar_data()`, 并赋值给预先定义好的全局变量, 具体实现流程如图 9 所示。

本文旨在使飞行器具有快速高效的避障功能, 具体实现为飞行器采取向上逃逸模式进行避障, 即飞行器检测到障碍物时, 飞行器尽快处于悬停状态, 然后垂直向上飞行, 具体实现流程如图 10 所示。

Loiter 飞行模式下首先将 `roll`、`pitch` 通道输入转换成期望的加速度, `raw` 通道输入转换成偏航角速率, `throttle` 通道输入转化成飞行器的爬升速率, 然后调用水平位置控制器、高度控制器和姿态控制器完成对飞行器姿态解算和定点控制。本文飞行避障实现流程首先是在获取滚转和俯仰通道输入之前, 对飞行器周围环境进行检测, 当检测到障碍物时, 进一步对飞行器飞行速度进行判断, 如果飞行速度不为零, 则将滚转和俯仰通道输入强制置零, 使飞行器尽快处于悬停状态; 如果飞行器本身就处于悬停状态, 则增加油门通道输入值, 提高飞行器飞行高度, 从而实现飞行避障。

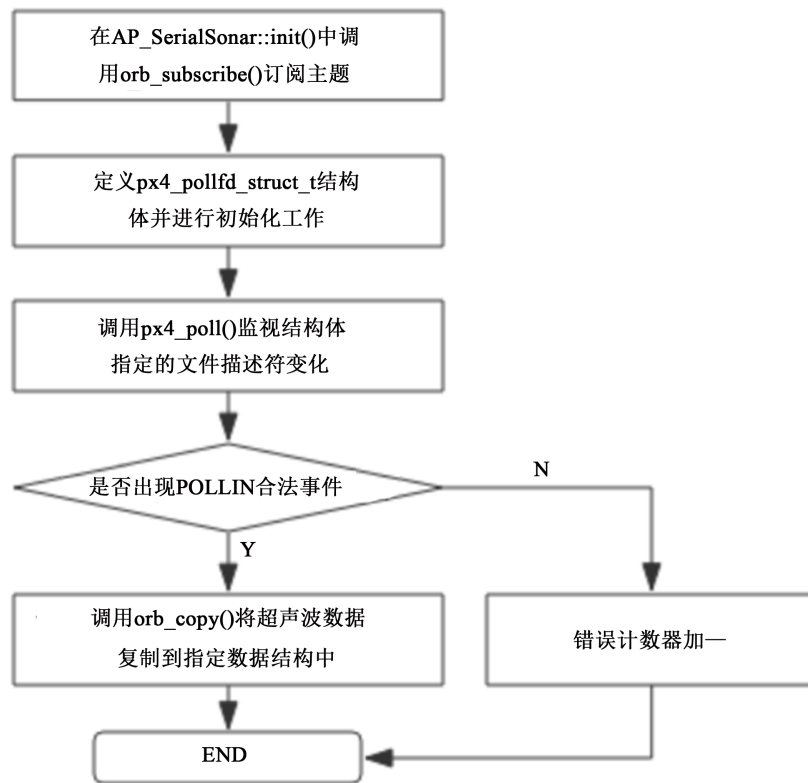


Figure 8. get_sonar_data() flow chart

图 8. get_sonar_data()流程图

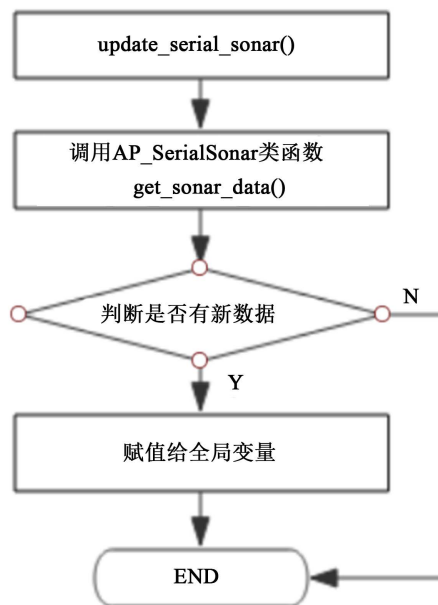


Figure 9. flow chart

图 9. 流程图

开发的避障系统相继在 Stablize、AltHold、Loiter 飞行模式下完成了避障测试, 通过 Mission Planner 观察当飞行器检测到障碍物时, 电机输出的情况, 图 11~图 13 为避障测试效果图。

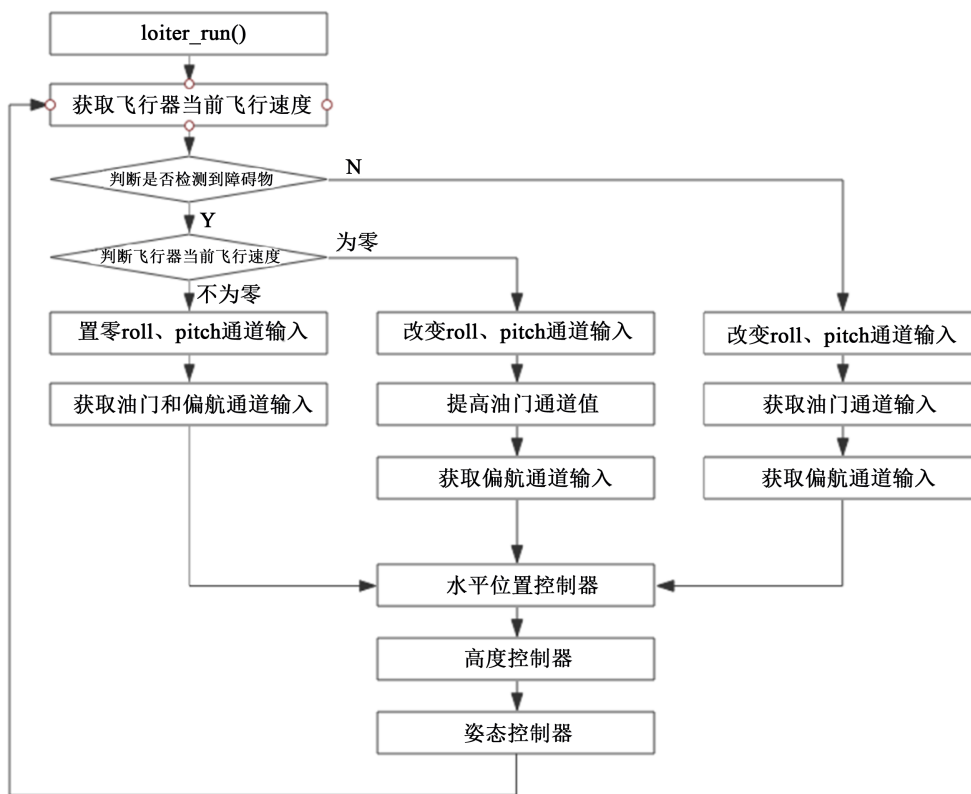


Figure 10. Obstacle avoidance flow chart

图 10. 避障流程图



Figure 11. Obstacle avoidance chart of Stabilize flight mode

图 11. Stabilize 模式下避障效果图



Figure 12. Obstacle avoidance chart of AltHold flight mode
图 12. AltHold 模式下避障效果图

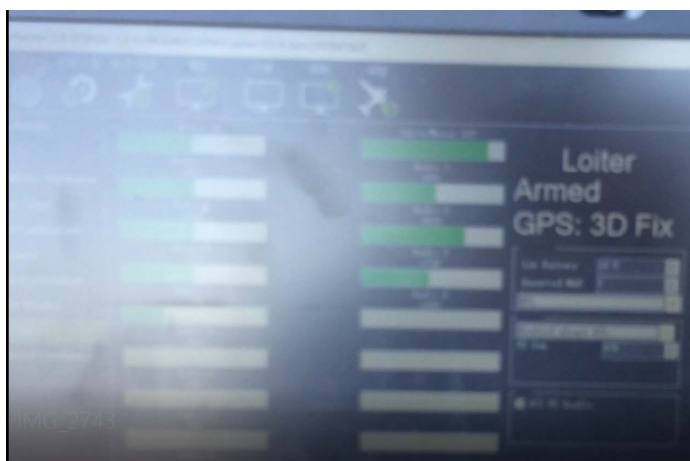


Figure 13. Obstacle avoidance chart of Loiterflight mode
图 13. Loiter 模式下避障效果图

5. 结论

本文基于 Pixhawk 飞控系统, 以“X”型四旋翼飞行器为平台, 经过多次飞行试验验证, 开发的避障系统可以实现飞行器避障功能, 具备一定的实用性。

参考文献 (References)

- [1] 王力群, 林朝辉. 基于 Arduino UNO 平台的多适配性无人机避障技术[J]. 科技创新导报, 2016(9): 18-19.
- [2] 周炜, 魏瑞轩, 董志兴. 基于层次分解策略无人机编队避障方法[J]. 系统工程与电子技术, 2009, 31(5): 1152-1157.
- [3] 徐华东. 无人机电力巡线智能避障方法研究[J]. 南京航空航天大学, 2014.
- [4] 郑天茹, 王滨海, 刘浪, 王骞, 张晶晶. 电力巡线无人直升机障碍规避系统[J]. 山东电力技术, 2012(1): 14-17.
- [5] 袁栋, 陈旭芳, 郁乐乐. 基于 Pixhawk 和 IAPF 法桥检无人机避障[J]. 低温建筑技术, 2017, 39(2): 40-42.
- [6] 任耀庭. 基于超声波测距与图像信息相融合的旋翼无人机避障算法研究[J]. 电子科技大学, 2016.
- [7] 杨淑媛. 旋翼无人机室内自主避障飞行研究[J]. 山西工程职业技术学院, 2016(6): 86-88.

期刊投稿者将享受如下服务：

1. 投稿前咨询服务 (QQ、微信、邮箱皆可)
2. 为您匹配最合适的期刊
3. 24 小时以内解答您的所有疑问
4. 友好的在线投稿界面
5. 专业的同行评审
6. 知网检索
7. 全网络覆盖式推广您的研究

投稿请点击：<http://www.hanspub.org/Submission.aspx>

期刊邮箱：dsc@hanspub.org