

中国·珠海		文档编号							
全志科技股份有限公司		文档名称		AWPilot 飞控开发说明					
All Winner Tech. CO. LTD		版 本		V1.1.5	密 级			共 1 页	
<div>AWPilot 飞控开发说明</div>									
文档作者				创建日期		2017-04-18			
拟 制				日 期		2017-04-18			
审 核				日 期					
批 准				日 期					
<div></div>									
All Winner Technology									
CopyRight©2017 All Winner Technology, All Right Reserved									

版本历史

	修改人	时间	备注
V1.0.0		2017-04-18	建立初始版本
V1.0.4		2017-11-17	性能提升：提升定高效果 减小起飞时偏移程度 减小定位模式下抖动 缺陷修复： 修复电量检测问题 修复碰撞检测过于灵敏问题
V1.0.07		2018-01-12	修改默认编译选项为 EVAB2 硬件板
V1.0.10	吕元宙	2018-03-20	1. 添加飞控的部分文档（在飞控 doc 目录下） 2. 更改日志记录格式，需要搭配 AWGcs 使用（在飞控 tools 目录下） 3. 添加 AWGcs 的通信支持 4. 优化 Makefile 多线程编译，提高编译速度 5. 提高参数在飞控中的使用率 6. 添加 SPL06 气压计的支持 7. 优化代码架构，更为模块化，更简洁 8. 优化电机动力分布计算方法 9. 调整电机顺序，电机 2 和 3 互换，使得动力输入顺序顺时针递增 10. 添加无头模式功能
V1.1.1	吕元宙	2018-5-23	1. 优化飞控架构 2. 优化 poshold 的刹车逻辑 3. 添加空翻功能
V1.1.2	吕元宙	2018-9-6	飞控： 1. 增加 TCP 接口 2. 修正 poshold 的 bug 3. 增加内存锁定功能，禁止系统对飞控内存进行 swap 操作 4. 增加飞行模式的退出接口 5. 增加热点环绕模式 6. 增加测距仪的定高功能 地面站： 1. 增强稳定性，优化经常退出的问题 2. 增加飞控的 IP 地址选择 3. 增加文件传输的功能 4. 增加参数导出为 .h 文件的功能 5. 增加参数的分类显示功能
V1.1.3	吕元宙	2018-11-27	飞控： 1. 修正内存锁定功能 2. 匹配新光流优化 poshold 刹车逻辑 3. 变更库的编译器版本为 uclib 4. 添加 PID 的 I 项默认值 5. 优化代码架构 地面站： 1. 变更实时状态曲线为双窗口模式

			<ul style="list-style-type: none"> 2. 优化手柄的初始化 3. 优化参数设置的逻辑，增加同步速度 4. 优化稳定性 5. 增加日志回顾时的飞行模式和 ERROR 信息显示
V1.1.4	吕元宙	2018-12-6	飞控： <ul style="list-style-type: none"> 1. 修复姿态初始化时异常 ACC 值导致的姿态为 NAN 值异常问题 2. 根据新 SDK 更新 make install 路径 3. 根据光流的置信度变更融合值
V1.1.5	吕元宙	2019-05-20	飞控： <ul style="list-style-type: none"> 1. 更新编译器为 musl 库 2. 增加 TF 卡拔卡处理 3. 增加 PID 的低通滤波预置接口 4. 更新气压计解算参数匹配新流程

Confidential

目录

1. 版本说明.....	5
1.1. 概述.....	5
1.1.1. 编写目的.....	5
1.1.2. 适用范围.....	5
1.1.3. 相关人员.....	5
2. AWPilot 简介.....	6
2.1. 概述.....	6
2.2. 硬件平台.....	6
2.3. 软件平台.....	7
2.4. RT Linux 系统.....	7
2.5. 飞控软件结构.....	7
3. 开发步骤.....	9
3.1. 获取 SDK.....	9
3.2. SDK 简介.....	9
3.3. 编译下载.....	9
3.3.1. 编译飞控.....	9
3.3.2. 编译内核、拷贝飞控.....	9
3.3.3. 打包固件.....	9
3.4. 硬件配置.....	10
3.5. 代码开发.....	10
3.5.1. 程序启动.....	10
3.5.2. 飞控参数调试.....	10
3.6. 调试工具的使用.....	10
3.6.1. Adb 调试.....	11
3.6.2. 串口调试.....	11
3.6.3. AWGcs 地面站.....	11
4. 开发文档说明.....	12

1.版本说明

1.1. 概述

1.1.1. 编写目的

介绍 AWPilot 的组成和编译方法

1.1.2. 适用范围

MR100 Linux 飞控项目的应用开发

1.1.3. 相关人员

MR100 应用开发人员。

Confidential

2. AWPilot 简介

2.1. 概述

AWPilot 是一套由 Allwinner 开发的基于实时 Linux 系统的多旋翼飞控软件平台。该软件平台整合了多旋翼飞控所需的姿态，速度，位置等状态估计算法和相应的控制算法，同时平台已具备常用的控制逻辑，基于该平台，开发者可快速实现丰富多彩的飞行功能，而无需担心复杂算法的实现难度和开发风险。

目前该平台有丰富的飞行功能支持，包括：

- 姿态自稳
- 气压计定高
- GPS 悬停
- 光流悬停
- Local360
- 一键起飞
- 一键降落
- 一键空翻

AWPilot 代码简洁高效，层次清晰。所需底层支持全部使用标准 Posix 接口实现。目前版本直接支持实时 Linux 系统，可与 Linux 图传应用等同时运行，互不干扰。有效提高系统集成度，降低系统成本。

2.2. 硬件平台

AWPilot 目前支持的硬件平台为 MR100。MR100 为我司开发的一颗专用于无人机系统的高性能一体化飞行芯片，通过搭载 Cortex-A7 和专有 H.264 编解码器，可同时实现带 GPS、光流的飞控和 1080P WIFI 图传。

目前 AWPilot 可支持的传感器列表如下：

类型	型号	备注
IMU	MPU6050	
	ICM20689	
罗盘	IST8310	
	HMC5883	
	QMC5883	
	MMC5883	
	LIS3MDL	
气压计	SPL06	
	DPS280	
GPS	ublox	

2.3. 软件平台

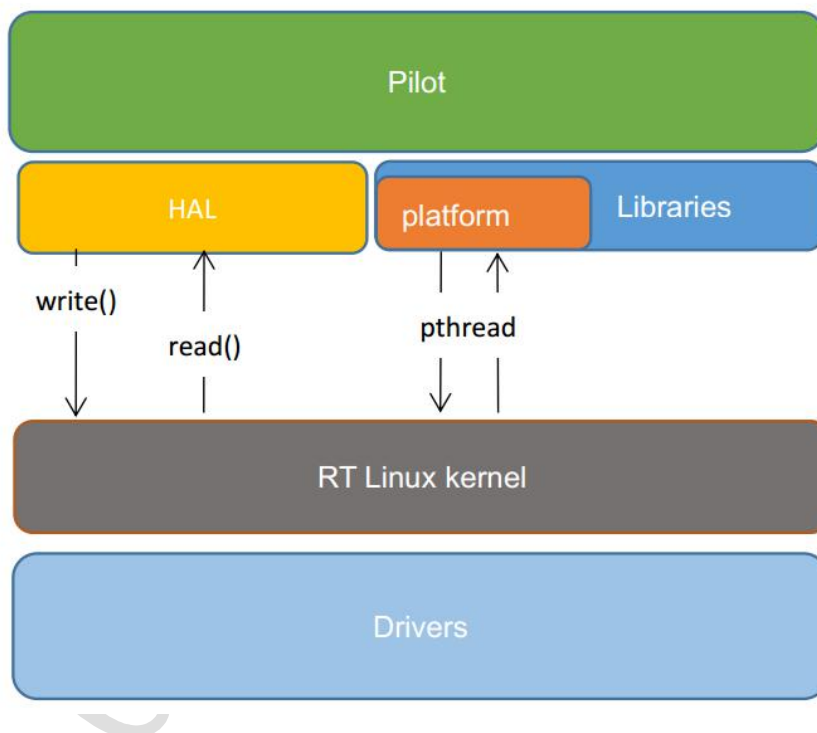
AWPilot 飞控平台采用层次化，模块化的设计方式实现。具有维护性高，移植性好，兼容性强等特点，运行于我司悉心优化的 RT Linux 之上，取得了良好的实时性表现。

2.4. RT Linux 系统

众所周知，标准 Linux 内核只能满足软实时应用，并不能满足飞控这样的硬实时应用。因此为保证 AWPilot 在能够取得良好的飞行性能和稳定性，安全性，我们对标准 Linux 系统进行改造，使得标准内核具有了硬实时应用保障能力。同时我们重新优化了飞控系统所需的每一个外设驱动，使其具有硬实时保障能力。最终使得 AWPilot 具有了优异的性能和稳定性表现。

2.5. 飞控软件结构

AWPilot 飞控软件结构如下图：

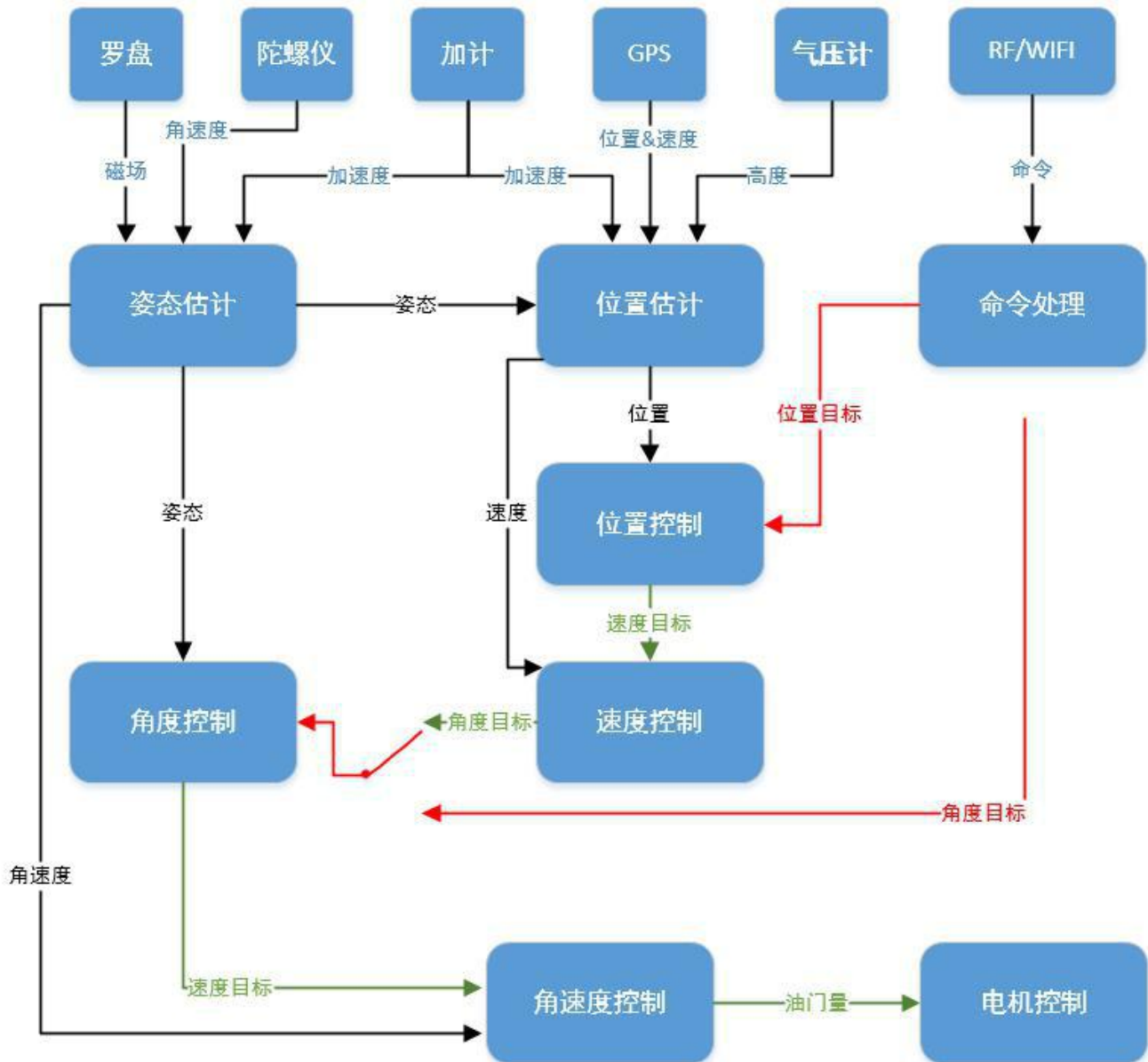


AWPilot 飞控软件按作用可分为 4 个层次：

- 硬件驱动层：该层由各个传感器和接口的驱动构成，形式为实时 Linux 标准驱动，为上层代码提供传感器和接口数据访问服务。
- 实时 Linux 内核：为飞控系统提供基本的系统服务，如硬件访问，文件访问，网络，进程调度等。提供的接口同一为 read/write, socket, pthread 这类的标准接口，因此具有很好的移植性。
- 适配层：适配层包含 HAL 和 Libraries。
HAL 对硬件层进行抽象和封装，对上提供同一的接口，对下调用操作系统提供的接口访问硬件。
Libraries 包含飞控所需的各类算法和依赖库等，其中与底层系统有关的部分被组织到一个名为 Platform.c 的文件中，底层的改动只需修改 Platform.c 即可。
- 飞控：飞控即为具体的算法和功能逻辑处理。Pilot 基于 HAL 和 Libraries，实现飞控所需的姿态，速度，

位置等状态估计和对应状态的控制，同时实现的具体功能，如起飞，降落，Flip 等相关的逻辑处理。

AWPilot 飞控具有一套完善的姿态，位置控制机制，保证丰富的飞控功能能够顺利实现。其主要内容可分为状态估计，状态控制和命令处理三大部分，下图为 AWPilot 飞控内部各个状态估计模块、状态控制模块、命令处理模块的组成和他们之间联系：



3. 开发步骤

3.1. 获取 SDK

请参考《MR100-FK SDK 下载说明.pdf》

3.2. SDK 简介

SDK 目录如下:

```
|---lichee
    |---buildroot
    |---build.sh
    |---img_trans_app
    |---linux-3.4
    |---pilot
    |---tools
```

其中:

Buildroot ---- 用于编译文件系统、 busybox, 编译工具链等

Build.sh ---- 编译脚本

Img_trans_app ---- 图传代码

Linux-3.4 ---- 内核

pilot ---- 飞控代码

Tools ---- 打包工具

3.3. 编译下载

SDK 的编译过程为:

- 编译飞控
- 将生成的可执行文件拷贝到打包目录
- 编译内核
- 打包固件

3.3.1. 编译飞控

```
cd pilot/
make ARCH=EVAB2
make install
```

注:如编译过程中遇到找不到交叉编译器,请执行./build.sh 配置交叉编译器环境

3.3.2. 编译内核、拷贝飞控

```
./build.sh
```

3.3.3. 打包固件

```
./build.sh pack
```

下载烧录前请先安装全志科技提供的 USB 驱动,使用全志科技提供的 PhoenixSuit, 在一键刷机界面选择打包

生成的固件文件:

(lichee/tools/pack/sun8iw8p1_linux_mr100-evb-perf_uart0.img)

首先使用 USB 连接硬件板, 按住烧录键 (如果没有烧录键, 请使用导电金属短接 Flash 1, 2 脚) 上电, 等待 PhoenixSuit 自动跳转到正在烧录界面时, 立刻放开所有按键。等待烧录完成即可。

3.4. 硬件配置

- (1) 在 linux-3.4 下执行指令 `make ARCH=arm menuconfig`
- (2) Device driver->UAV sensor support 选择对应 IMU 和气压计
- (3) 修改 IMU 和气压计 I2C 地址:
 - a、打开对应 IMU 和气压计的驱动代码, 驱动代码在 linux-3.4\drivers\uav_sensors 目录下。
 - b、更改 I2C 地址的定义, 例如 `#define SENSOR_I2C_SLAVE_ADDRESS 0x77`

3.5. 代码开发

3.5.1. 程序启动

MR100 程序启动过程为 Linux 系统标准启动过程。飞控和图传的启动通过 Linux 启动脚本启动。启动脚本位于 `lichee/tools/MR100/autorun.sh`。如需定制启动过程, 请修改该文件。

3.5.2. 飞控参数调试

参数的调试有三种方式:

- 通过代码修改: `pilot/config/param_x.h` 文件 (x 代表开发板型号)
- 通过参数文件修改: 使用系统串口修改 `/netprivate/param` 文件后重启飞控生效
- 通过地面站连接上飞控后在参数配置页面修改

3.6. 调试工具的使用

利用使用调试工具可以提高调试效率。当前调试效率最高的方法是使用 AWGcs 地面站。使用以下方法需要运行 `pmonitor` 小程序。Pmonitor 的主要作用是查询是否有 `pilot` 程序正在运行, 如果没有则运行 `"/tmp/pilot &"`。

将 `pmonitor` 设置为自启动:

- (1) 将 `linux_pilot/tools/pmonitor/pmonitor` 复制到 `lichee/tools/ELF/`
- (2) 在 `autorun.sh` 中添加 `pmonitor &`

注: 运行新上传的 `pilot` 需要将 MR100 的 `/netPrivate/param` 删除, 或将 `main.c` 中的 `param_load()` 注释掉, 如下图所示。

```
void module_param_init()
{
    param_init();
    param_calib_init();

    log_param_init();
    nav_param_init();
    imu_param_init();
    attitude_param_init();
    mag_param_init();
    control_param_init();
    att_control_param_init();
    pos_control_param_init();
    motor_param_init();
    rc_param_init();
    failsafe_param_init();
    system_param_init();
    flow_param_init();
    rangefinder_param_init();

    // param_load();
    param_calib_load();
} ? end module_param_init ?
```

3.6.1. Adb 调试

Adb 登陆到 MR100 终端, pilot 及其他程序的打印信息都不会在 adb 终端显示 (在 adb 端重新执行指令“pilot &”运行 pilot 才会有打印信息), 所以 adb 调试效率最低。

(1) 在 windows 双击 push_pilot.bat 文件, 主要作用是上传和杀掉 pilot, 然后 pmonitor 会主动运行刚上传的 pilot 程序, 将执行以下命令:

```
adb push F:\lichee\tools\img_app\pilot\pilot /tmp/ //路径根据实际情况进行修改。
```

```
adb shell "killall pilot"
```

```
adb shell "chmod 777 /tmp/pilot"
```

(2) 在 windows 双击 updae_img.bat 文件, 将自动进入烧写模式, 无需手动短接 flash 引脚。将执行以下命令:

```
adb shell "echo 0x01c20508 0x5aa5a55a > /sys/class/sunxi_dump/write"
```

```
adb shell "reboot"
```

(3) 在 windows 双击 pull_log.bat 文件, 将下载 tmp/log.bin 文件, 将执行以下命令。

```
adb pull /tmp/log.bin F:\log //路径根据实际情况进行修改。
```

3.6.2. 串口调试

Linux 指令编辑运行和终端实时显示。

3.6.3. AWGcs 地面站

主要有以下功能, 具体使用方法请查阅《AWGcs 使用说明》。

- (1) 支持在线调试参数, 无需重新编译 pilot
- (2) 支持实时显示数据曲线
- (3) 支持上传 pilot 和下载 log
- (4) 支持遥控器操作飞行

4. 开发文档说明

在飞控 pilot_linux 下的 doc 目录有以下开发文档

- <AWGcs 使用说明>：飞控配套地面站的使用说明
- <AWPilot 悬停逻辑及调试手册>：描述飞控的悬停逻辑及常见的优化方法
- <AWLink 协议说明>：飞控通信协议的说明文档
- <AWPilot 日志协议>：飞控日志协议的说明文档

Confidential