Taking KDE to the Skies
— Making the drone ground control Kirogi —

Eike Hein
Milan, Italy · September 8th, 2019
"Kirogi is a ground control application for drones."
Drones: Supported Hardware
Drone software: Ground Control & Pilot

Running the *Pilot*

Using a *Ground Control*
Basic Drone Piloting 101

- Yaw
- Roll
- Pitch
Demo
Demo 1/5: Vehicle Screen

Select drone model · Vehicle status
Demo 2/5: Flight Controls

Direct control · Flight & vehicle status · Flight parameters
Demo 3/5: Navigation Map

Device & vehicle location • Fly by map
Demo 4/5: Settings

- Allow Kirogi to request the location of this device
- Show D-pads when a gamepad controller is connected

Instant apply
Demo 5/5: Application About

Kirogi 0.1-dev
A ground control application for drones.

Copyright
(c) 2019 The Kirogi Team
https://www.kirogi.org/
License: GPL v2

Libraries in use
KDE Frameworks 5.63.0
The xcb windowing system
Qt 5.13.0 (built against 5.13.0)

Authors
- Eike Hein <hein@kde.org>
- Rafael Brandmaier <rafael.brandmaier@kdemail.net>
- L. 'AsmoArael' C. <clc.jarryh99@outlook.fr>

Contributor credits
Demo Reel

Video available here.
Why make this?
Reasons for Kirogi

- It's a classic free software use case
  - Existing proprietary apps are usually single-vendor or single-drone
  - Vendors often abandon apps when releasing new products
  - Open source can serve users better here

- Learning to make a KDE app for mobile devices

- Drones are fun!
Kirogi From Above

- Written in C++14 and QML
- It uses KDE's Kirigami GUI library
  - Runs well on desktop and mobile
- Only non-Qt/KF5 dependencies: GStreamer, QtZeroConf
  - GStreamer works well with Qt across desktop and Android (gstqmlgl)
  - KDNSSD could use an Android port
- Using the latest Qt 5.12+ features
  - e.g. Input Handlers
Software Architecture

- Application
  - main.qml
  - Vehicle.qml
  - FlightControls.qml
  - ...

- Backend Library (libkirogicore)
  - VehicleSupportPluginModel
  - VehicleSupportPlugin
  - AbstractVehicle
  - ...

- Parrot Plugin
  - ParrotPlugin
  - ParrotVehicle
  - ParrotConnection
  - ParrotProtocol

- Tello Plugin
  - RyzeTelloPlugin
  - RyzeTelloVehicle
  - RyzeTelloConnection

- Your Plugin Here

- Written mostly in Qt Quick / QML
- C++ helpers to integrate GStreamer, Android permissions

- Usable by third parties
- Written in C++14
- Public APIs in C++ and QML

- Enables third-party extensibility
- Responsible for vehicle discovery and communication
- No restrictions on implementation strategy
The KDE mobile application stack compares favorably: Kirogi comes in smaller than single-vendor (and sometimes single-drone) applications.

1 v4.3.20 apkmirror.com / Google Play: 259 MB  
2 v6.5.2 apkmirror.com / Google Play: 274 MB
The Story Behind the Mascot

*Wild* geese are known as expert flyers. Kirogi's mascot is a farm goose with an adventurous spirit, who flies just as well thanks to the use of superior technology.

*Kirogi* is the Korean word for wild geese!
The Mascot: Making Of

- Posted the idea on Krita's *Artists Wanted* forum on February 11th, 2019.
- Selecting among several submissions, decided to work with French art student L. 'AsmoArael' C. on February 13th.
- Several poses, and palettes / lighting moods were explored.
- Approved the final commissioned drawing on February 25th. **Made in KDE's Krita!**
- Later on, commisioned additional head pose sketches for the KDE VDG's Rafael Brandmaier to base the app icon on.
The Mascot: Sketches 1/2
The Mascot: Sketches 2/2
The Application Icon
Kirogi So Far
What Went Well

• The KDE community is amazing
  • The project received a huge amount of help from SDK, Sysadmin, WWW, Promo
  • Enthusiasm & positive feedback make it easy to stay motivated

• The tooling is comprehensive
  • KDE Android SDK Docker image & helper scripts
  • Content creation: Krita & Kdenlive

• The technology for KDE mobile apps is there
What Went Wrong

- ... but Android is a bit frustrating
  - Steep learning curve
  - Lots of gotchas (library naming, ...)
  - Lack of asset compatibility (no SVG icons as-is, ...)
  - Poor debug experience

- More KDE Frameworks modules need Android ports
Future Plans

• Make a release!

• Mission planning on the Navigation Map
  • Use Marble for mapping?

• Better video and still capture
  • Gimbal control
  • Media download
  • Media gallery
  • Local stream recording

• Open protocols: MAVLink & MSP

• FPV
Resources

Browse [https://kirogi.org/](https://kirogi.org/) 😊