
ArduPilotの概要

アーキテクチャー、開発事例

July, 29, 2017

Yuichiro Shibata / Ruby Development Inc.



Agenda

- 自己紹介
- 概要
- アーキテクチャー
- 開発事例
- 開発事例デモ (動画)

自己紹介



柴田 有一郎

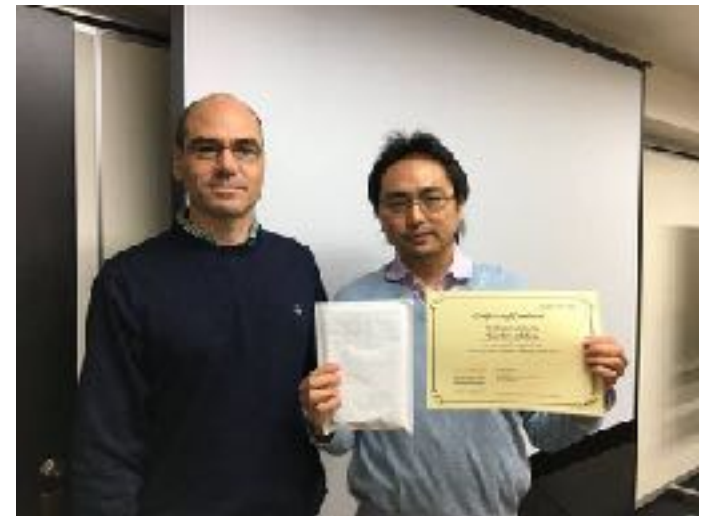
(株)Ruby 開発 エンジニア 兼 執行役員
ベトナム子会社 代表取締役

言語

C / C++ / Ruby

- 20年間近くソフトウェア開発（組込系、Web系）に従事
- 2015年
 - 新規事業立上げ
 - Ruby on Railsをコア技術とし、アジャイルとソーシャルコーディングをベースとするリモート開発を前提とした開発スタイルです。
- 2017年
 - MQTT for Rubyと、AWS IoT Device SDK for RubyをOSSとして開発し、公開しています。

- Drone Japanさんが開催する「ソフトウェア・エンジニア養成塾第2期」に参加し、MVP受賞



ArduPilotの概要

- ArduPilotの名前の "Ardu" の部分はArduinoから
 - 当初、APM1およびAPM2ボードで、Arduino開発環境とAVR CPUをベースにしていたため

- オートパイロットシステム
- マルチプラットフォーム
- オープンソース
- 拡張性が高い
- 信頼性が高い

- Stabilize（GPS不要）
 - 完全マニュアルモード
- AltHold（GPS不要）
 - 高度を自動で維持
- Loiter（GPS必要）
 - 高度、位置、姿勢を自動で維持
- Guided（GPS必要）
 - GCSなどから操作
- Auto（GPS必要）
 - 自律航行。ウェイポイントなどのミッションを作成し、ミッション通りに自動航行します
- RTL（GPS必要）
 - 離陸した場所（Home）に、一定の高度を保って直線的に戻ってくる

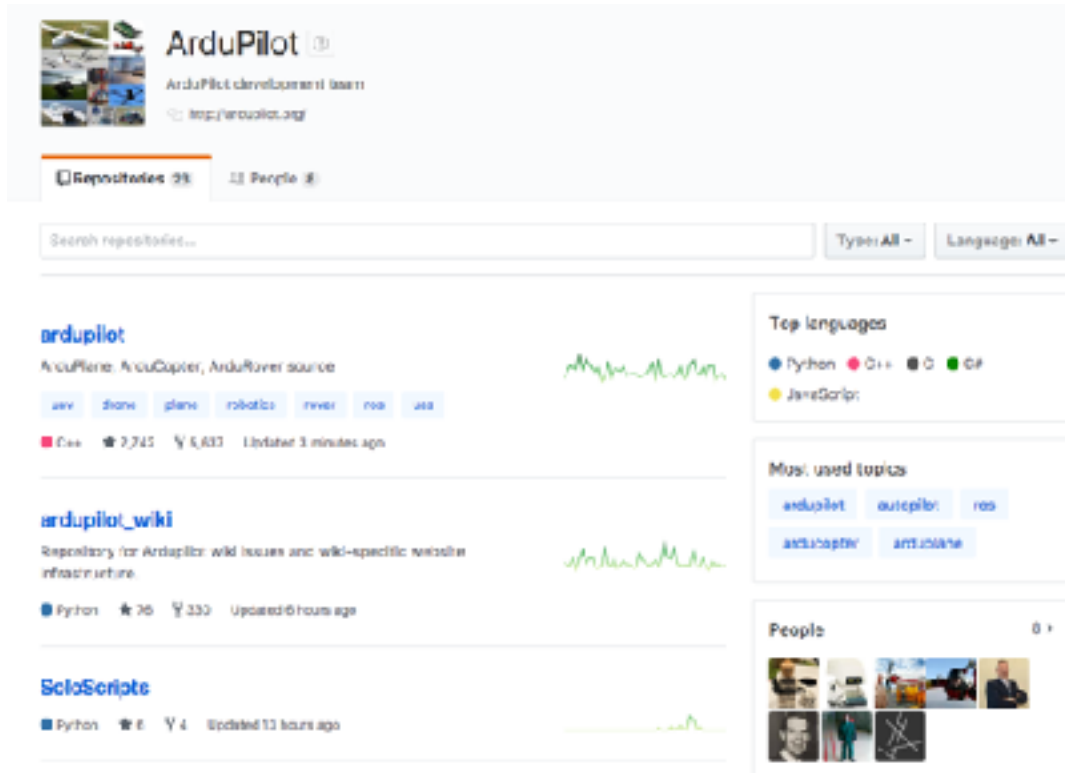
- GPSを使用した自律航行（ドローン）
 - <https://youtu.be/7rj2Wr3bmhk?t=66>
 - <https://youtu.be/7rj2Wr3bmhk?t=380>
- Non-GPS（Pozyxを使用）の自律航行
 - <https://youtu.be/MKWIPsBmAyo?t=2>
 - <https://youtu.be/MKWIPsBmAyo?t=132>

- マルチコプター (ドローン)
- プレーン (飛行機型)
 - <https://youtu.be/hDG-KIYyYDU?t=0>
 - <https://youtu.be/hDG-KIYyYDU?t=56>
- ローバー (陸上、水上ボート)
 - <https://youtu.be/ALXEe75JrII?t=13>
- サブマリン
 - <https://youtu.be/gGHNgvnkq40?t=108>
- アンテナトラッカー
 - <https://youtu.be/K48xzl9-igA?t=83>

• GitHubで公開

• <https://github.com/ArduPilot/ardupilot>

- Pull Request数：158（2017/09/27時点）
- Fork数：5,627（2017/09/27時点）
- コントリビューータ数：329名（2017/09/27時点）



ArduPilot ^①
ArduPilot development team
<http://ardupilot.org>

Repositories: 23 | People: 3

Search repositories... | Type: All - | Language: All -

ardupilot
ArduFrame, ArduCopter, ArduRover source
avr drone plane robotics rover ros uav
C++ ★ 2,745 🍴 5,633 Updated 3 minutes ago

ardupilot_wiki
Repository for ArduPilot: wiki issues and wiki-specific website infrastructure.
Python ★ 76 🍴 330 Updated 6 hours ago

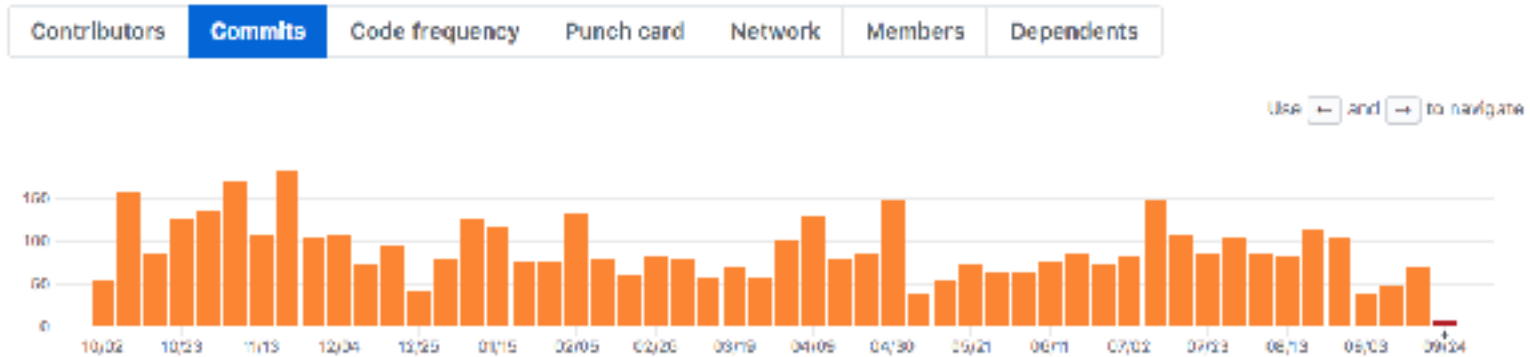
ScioScripts
Python ★ 6 🍴 4 Updated 13 hours ago

Top languages
Python C++ C C# JavaScript

Most used topics
ardupilot autopilot ros ardupilot articlans

People 0 +

- 常にバグ対応や機能追加が行われている



- ソースコードが公開されているため、セキュリティーや機密性に関して、ソースコードを確認することが出来る
- 世界中1,000,000台以上の車両に搭載されている実績
- ユーザー数が、200,000人~300,000人とされています。

- オープンソースのため、自由に機能追加することが可能
- 追加した機能は、Pull RequestでMasterソースへ追加されます。

- 様々はハードウェアにインストール可能
- ARM Cortex 32-bit
 - Pixhawk



- PixRacer



- 様々はハードウェアにインストール可能
- Linux Computers
 - NAVIO2(on Raspberry Pi)



- Intel Aero



- Bebop2

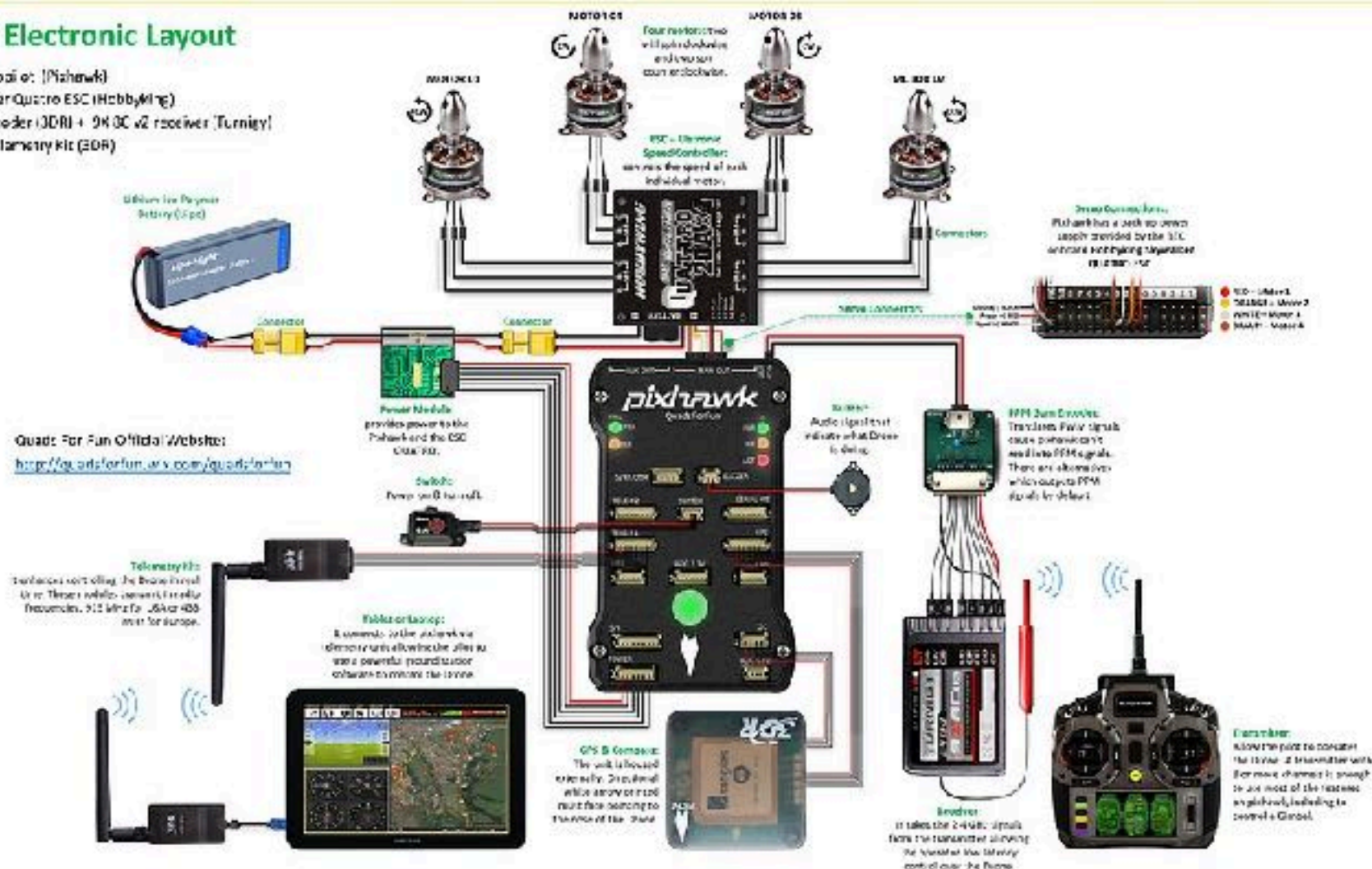


- Pixhawk
 - 3DR社製
 - CPU : 32bit ARM Cortex M4 18-Mhz
 - センサー
 - 加速度&ジャイロ : MPU6000
 - 気圧 : MS5611
 - コンパス
 - GPS (オプション、外付け)
 - UbloxM8

•Pixhawk

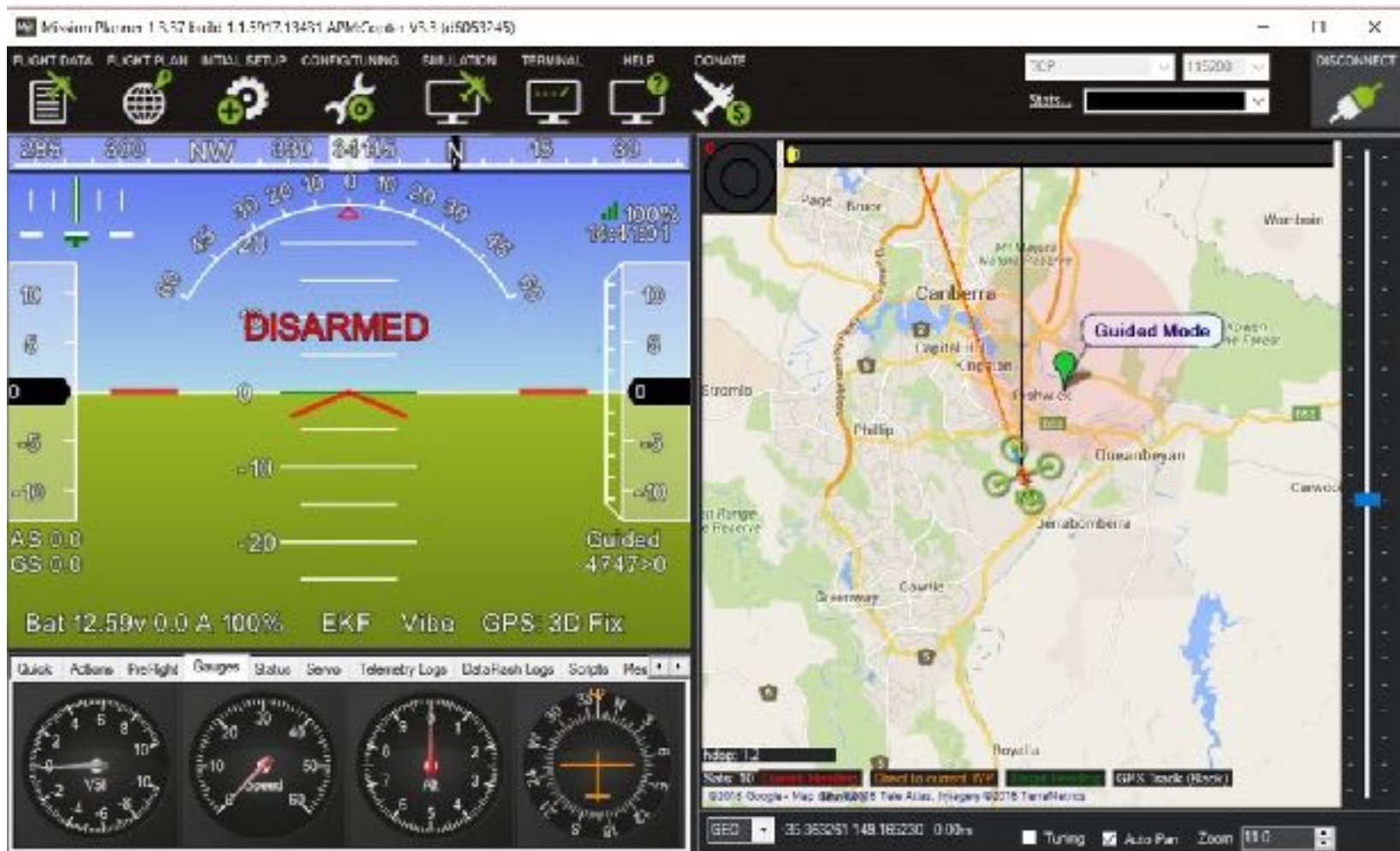
Typical Electronic Layout

- PWM Autopilot: (Pixhawk)
- SkywalkerQuadro ESC (HobbyKing)
- PWM encoder (DDR) + 9K DC v2 receiver (Turnigy)
- Radio Telemetry kit (EPR)





•Mission Plannerの画面



- Ground Control Station (GCS)
 - 機能概要
 - ファームウェア更新
 - 機体の初期設定
 - フレームタイプの設定
 - キャリブレーション（機体調整）
 - ECS、加速度センサー、コンパス、
 - プロポチャンネル、フライトモード
 - フェールセーフ、etc…
 - チューニング
 - 操作の感度（反応の良さ）などの調整
 - 機体の設定
 - パラメータ設定
 - 機体情報の表示
 - 位置、高度、向き、各種センサー情報

- 機能概要
 - 自律航行
 - ウェイポイント
 - 経路上の位置情報
 - ミッションコマンド
 - 離陸、着陸、 etc..
 - 飛行ログの取得・解析

•Mission Plannerのウェイポイントの設定

The screenshot shows the Mission Planner interface. At the top, there are menu options: FLIGHT DATA, FLIGHT PLAN, INITIAL SETUP, CONFIGURING, EMULATION, TERMINAL, HELP, and DOWNLOAD. The main map area displays a mission plan with 5 waypoints (1-5) and a 'Home' point. The waypoints are connected by yellow lines. The 'Home' point is located at the center of the mission area. The waypoints are numbered 1 through 5, starting from the Home point and following a path around the area.

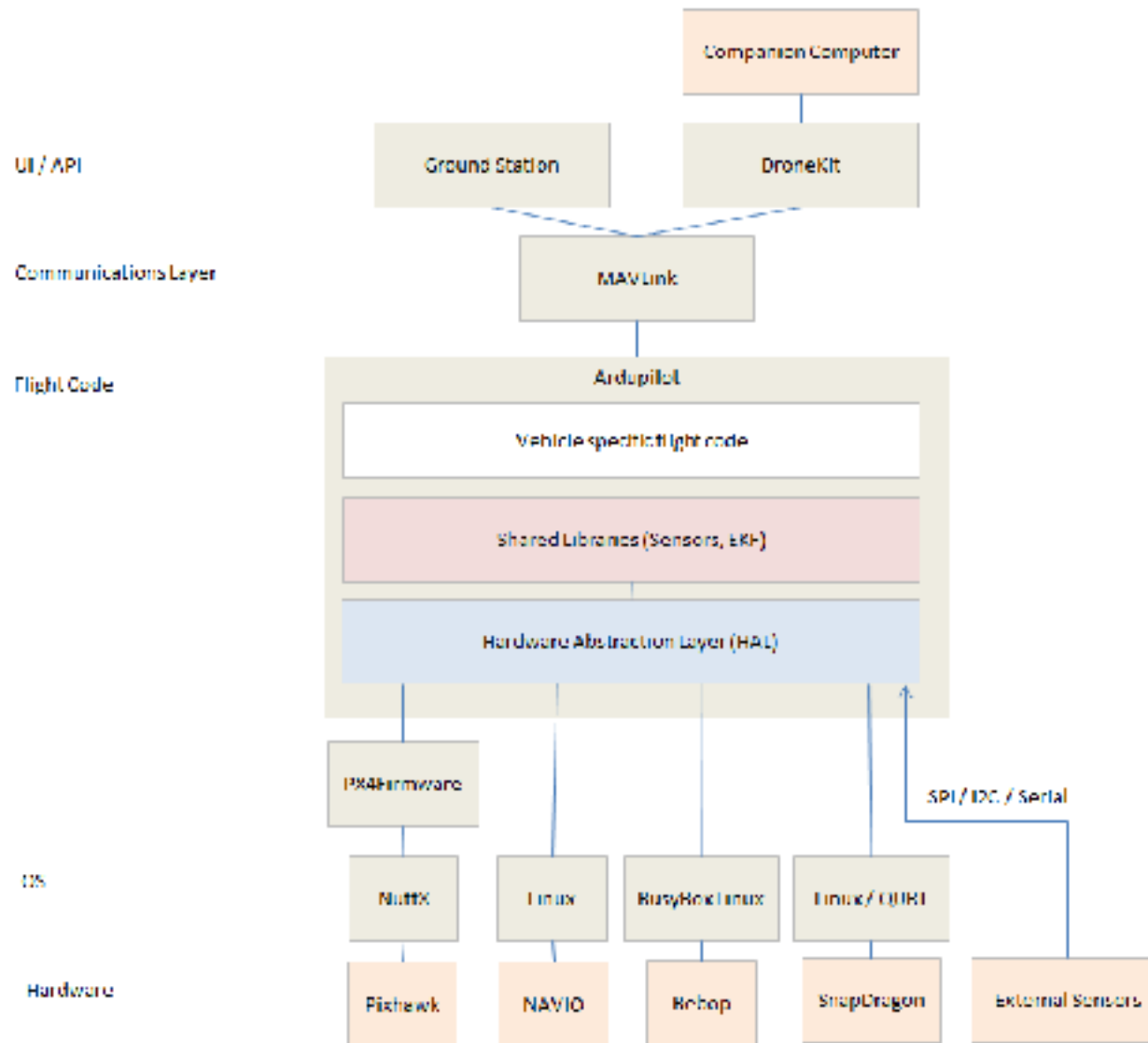
Below the map, there is a 'Waypoints' table with the following data:

Waypoint	Label	Alt	Spd	Yaws	Roll	Pitch	Lat	Long	Alt	Arms	Up	Down	Land F	Dir	AV
1	WAYPOINT	0	0	0	0	0	-35.0407920	117.8277050	00	X	🏠	🏠	0.0	104.5	
2	WAYPOINT	0	0	0	0	0	-35.0406756	117.8259110	00	X	🏠	🏠	0.0	109.7	270
3	WAYPOINT	0	0	0	0	0	-35.0410248	117.8241612	100	X	🏠	🏠	0.0	141.2	270
4	WAYPOINT	0	0	0	0	0	-35.0428335	117.8259873	100	X	🏠	🏠	0.0	145.1	149
5	WAYPOINT	0	0	0	0	0	-35.0427105	117.8274572	00	X	🏠	🏠	0.0	134.5	84

On the right side of the interface, there is an 'Action' panel with various controls, including a 'Home Location' section showing coordinates: Lat: -35.0410248, Long: 117.8277050, Alt (m): 0.0.

- 代表的なGCS
 - MissionPlanner
 - 対応OS:Windows、Ubuntu(mono上での動作)
但し、Ubuntuでは正常に動作しない
 - APM Planner 2
 - 対応OS:Windows、Mac OS、Ubuntu
 - QGroundControl
 - 対応OS:Windows、Mac OS、Ubuntu、Android、iOS

アーキテクチャ



<http://ardupilot.org/dev/docs/learning-ardupilot-introduction.html>

- 各車両タイプの最上位ディレクトリ

29,076 commits 25 branches 168 releases 329 contributors GPL-3.0

Branch: master New pull request Create new file Upload files Find file Clone or download

lucasedemarchi AP_HAL_Linux: ToneAlarm: !def out bogus message Latest commit 5ebc8c2 a day ago

.github	github: Add Sub vehicle to Issues template	a month ago
APMrover2	Rover: Report GPS health	a day ago
AntennaTracker	global: use static method to construct AP_AIRSPY_FCN	3 days ago
ArduCopter	Copter: Check GPS health	a day ago
ArduPlane	Plane: Report GPS health	a day ago
ArduSub	Sub: Check GPS health	a day ago
Tools	Tools: LogAnalyzer: TestGPSGlitch: flake8 compliance	23 hours ago
benchmarks	waf: add benchmark Waf tool	2 years ago
docs	docs: Fix typos	a year ago
libraries	AP_HAL_Linux: ToneAlarm: !def out bogus message	8 hours ago
mk	build: remove Linux boards from make	17 days ago
modules	PX4NuttX: submodule update	2 days ago
tests	tests: add macro for printing test parameter	a year ago
.dir-locals.el	emacs: add a .dir-locals.el	11 months ago

• APMrover2 : ロバー
• AntennaTracker : アンテナトラッカー
• ArduPlane : プレーン
• ArduSub : サブマリン

•各車両タイプで共有されるライブラリー

29,076 commits 25 branches 168 releases 329 contributors GPL-3.0

Branch: master New pull request Create new file Upload files Find file Clone or download

lucademarchi AP_HAL_Linux: ToneAlarm: ifdef out bogus message Latest commit 5ebc8c2 a day ago

.github	github: Add Sub vehicle to issues template	a month ago
APMrover2	Rover: Report GPS health	a day ago
AntennaTracker	global: use static method to construct AP_AHRS_DCM	3 days ago
ArduCopter	Copter: Check GPS health	a day ago
ArduPlane	Plane: Report GPS health	a day ago
ArduSub	Sub: Check GPS health	a day ago
Tools	Tools: LogAnalyzer: TestGPSGlitch: flake8 compliance	23 hours ago
benchmarks	waf: add gbenchmark Waf tool	2 years ago
docs	docs: Fix typos	a year ago
libraries	AP_HAL_Linux: ToneAlarm: ifdef out bogus message	8 hours ago
mk	build: 共通で使用されるライブラリー	17 days ago
modules	PX4NuttX: 各種センサーなど	2 days ago
tests	tests: add macro for printing test parameter	a year ago
.dir-locals.el	emacs: add a .dir-locals.el	11 months ago

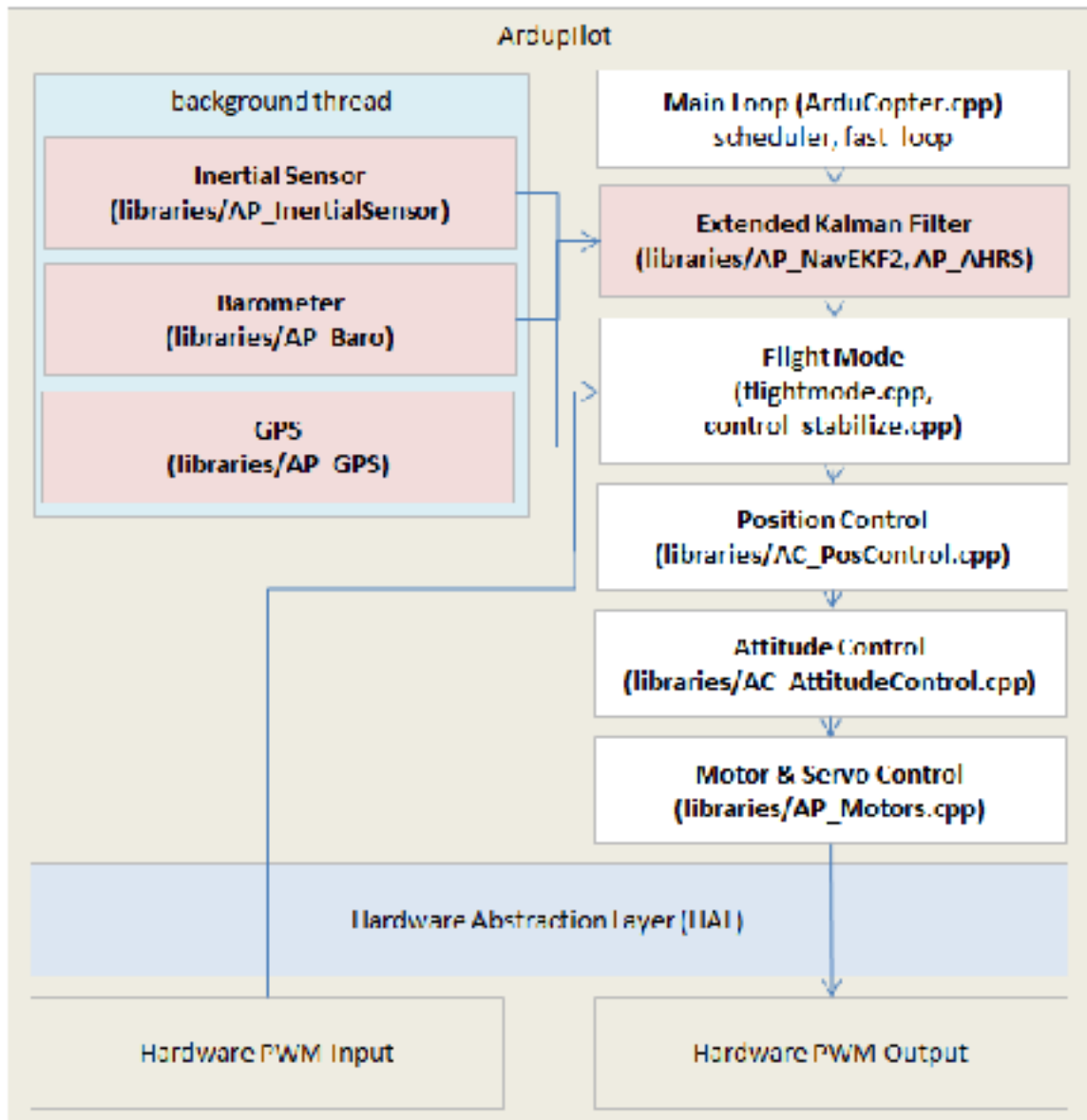
- AP_InertialSensor
 - ジャイロと加速度計のデータを読み取り、キャリブレーションを実行し、メインコードや他のライブラリにデータを提供します
- AP_Baro
 - 気圧計ライブラリ
- AP_GPS
 - gpsライブラリ
- AP_Compass
 - 3軸コンパスライブラリ

- ハードウェアを抽象化し、ハードウェアプラットフォームの差異を吸収

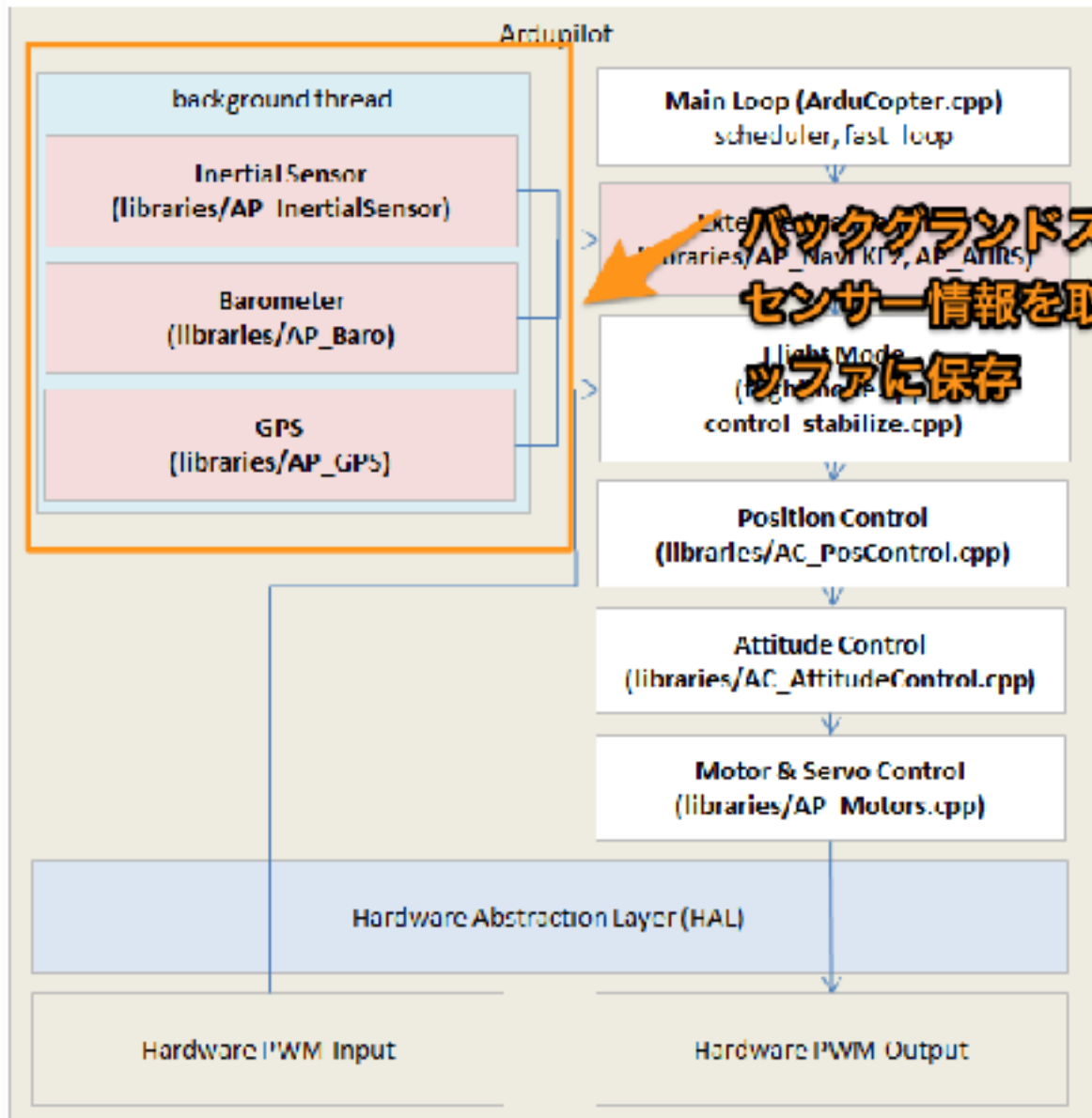
インタフェースを定義

AP_HAL	AP_HAL: RCOOutput_Tap: make LEDs blink on non-Aero ESCs
AP_HAL_AVR	AP_HAL_AVR: remove examples
AP_HAL_Empty	global: remove AP_HAL::in_timerprocess()
AP_HAL_FLYMAPLE	AP_HAL_FLYMAPLE: remove hal
AP_HAL_Linux	AP_HAL_Linux: ToneAlarm: ifdef out bogus message
AP_HAL_PX4	AP_HAL: move in tap esc implementation
AP_HAL_QURT	AP_HAL_QURT: make in_main_thread const and override
AP_HAL_SITL	SITL: add parameter to simulate reversed pitot tube order
AP_HAL_VRBRAIN	AP_HAL_VRBRAIN: make in_main_thread const and override

各ハード事のインタフェース実装

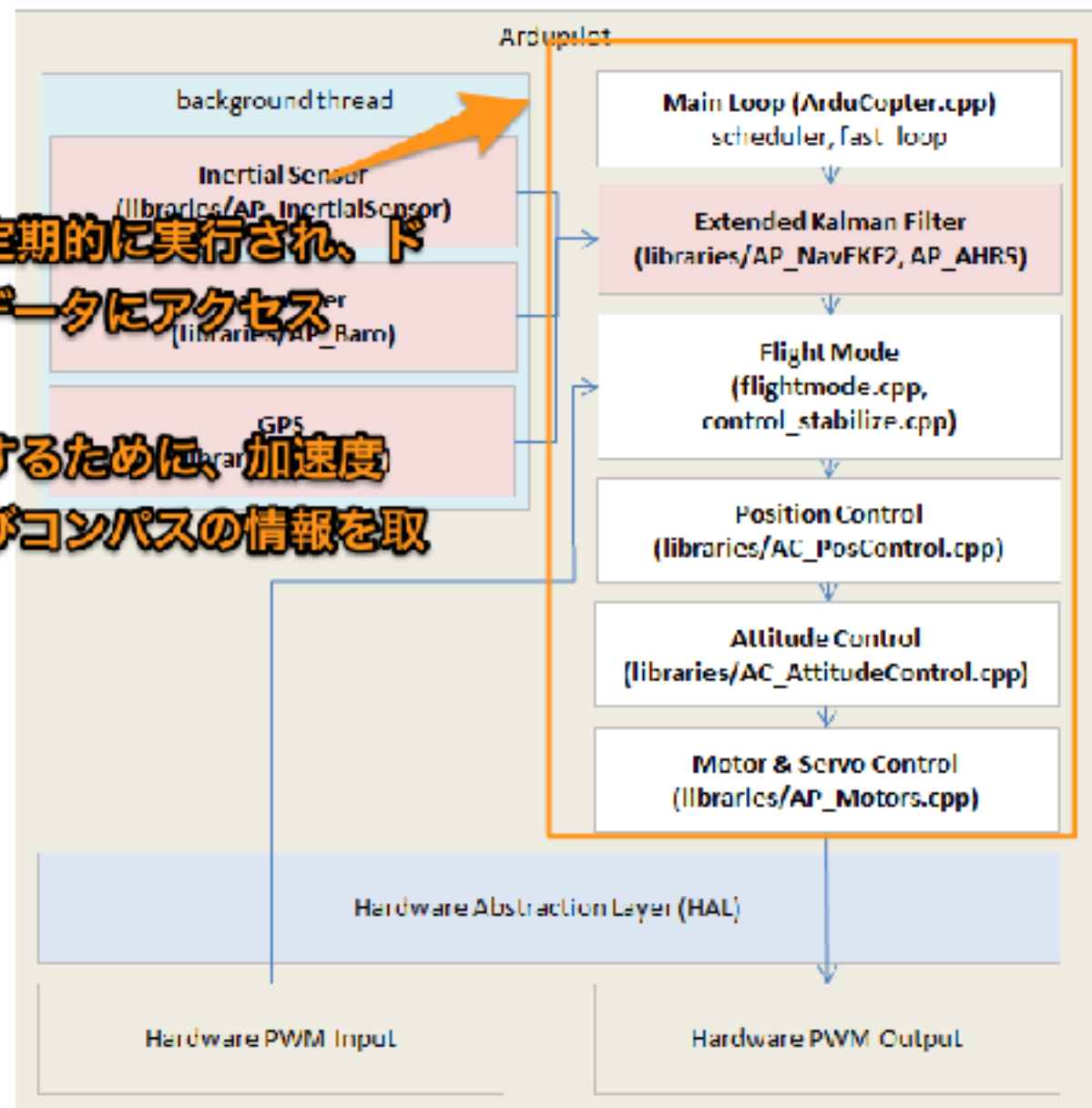


<http://ardupilot.org/dev/docs/code-overview-sensor-drivers.html>



<http://ardupilot.org/dev/docs/code-overview-sensor-drivers.html>

メインスレッドは定期的に行われ、ドライバーの最新のデータにアクセス
 姿勢推定値を計算するために、加速度計、ジャイロおよびコンパスの情報を取得する。



- Linux、Windows、Macで開発可能
 - Linux (Ubuntu)
 - 必要なソフトを入れればそのまま開発可能
 - Windows
 - Cygwinを使用
 - 仮想マシンにUbuntuを入れて開発
 - Mac
 - 必要なソフトを入れればそのまま開発可能
 - 仮想マシンにUbuntuを入れて開発

- 基本は、Makeを使用してビルド
- 今後は、Wafを使用していく方向
 - Waf
 - Python製のビルドツール

開発事例

1. MQTTによるDroneのIoT化
2. インターネットからのDroneの遠隔制御
3. フライトデータのリアルタイム監視

MQTTによるDroneのIoT化

IoT用の通信プロトコルMQTTの機能を追加し、IoT Platformへ接続

MQTT プロトコル

- IoTやM2M (Machine-to-Machine) 向けの軽量プロトコル
- TCP/IPネットワークで利用でき、セキュリティを考慮した「ユーザ認証」と「通信の暗号化」が可能
- 1つのサイズ小さいデータを、大量に、低遅延かつ、双方向で通信可能

1. インターネットを使用したDrone制御

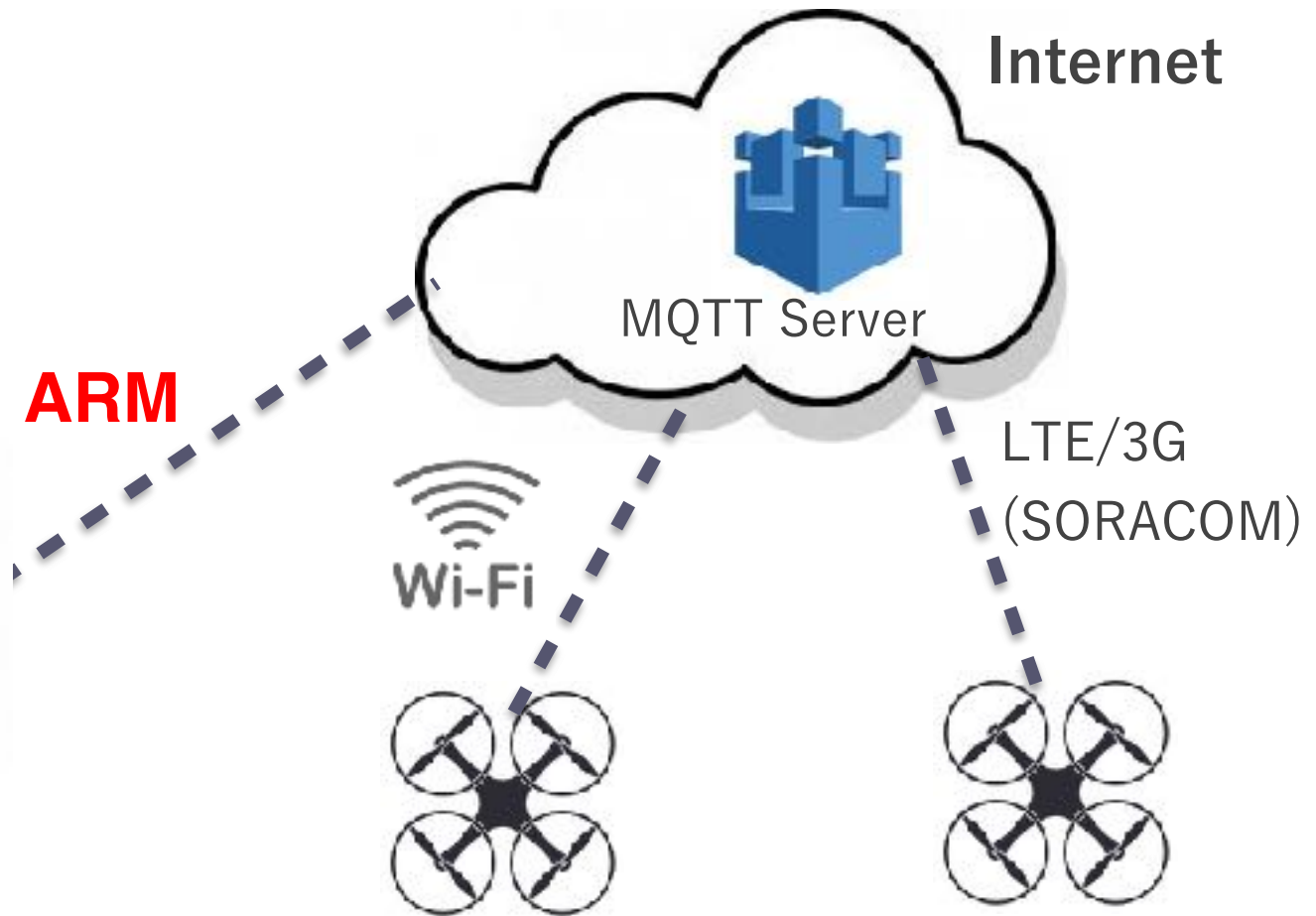
Command :

ARM

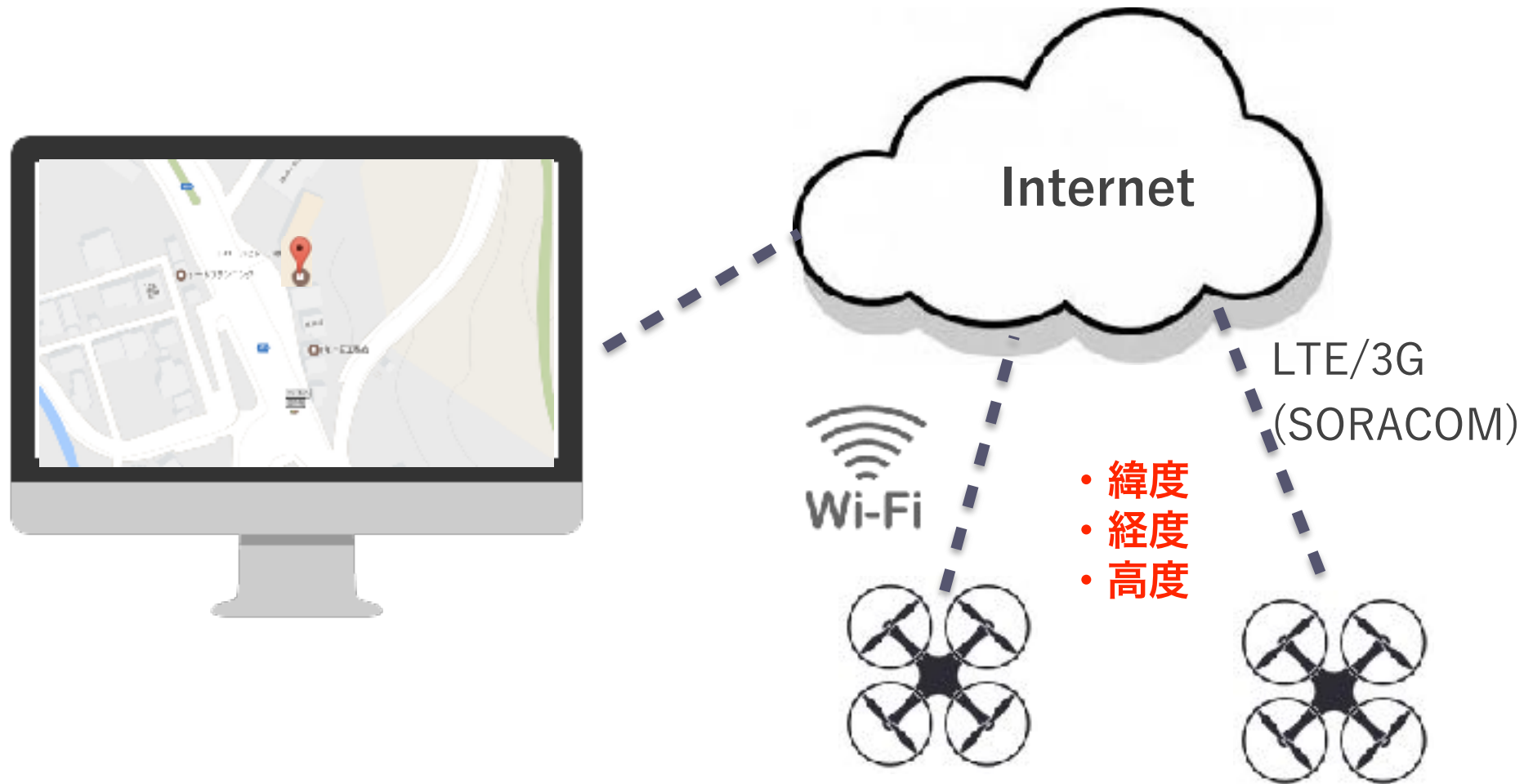
Take off 2m

Land

DISARM



2. フライトデータのリアルタイム監視



WiFiでのデモ内容

- 福岡県のパソコンから、神奈川県Droneを操縦します。
 - コマンド
 - ARM (飛行可能な状態)
 - Take Off 2m (2m上空へ飛行)
 - ロイターモードへの変更
 - Land (着陸)
 - DisArm (飛行可能状態を解除)。

Wi-Fiによる遠隔制御

https://youtu.be/xd4703d2Z_Q?t=16

SORACOMでのデモ内容

- 福岡県のパソコンから、神奈川県Droneを操縦します。
 - コマンド
 - ARM (飛行可能な状態)
 - DisArm (飛行可能状態を解除)
- Droneを車で移動し、位置情報がリアルタイムで監視できていることを確認します。

SORACOMによる遠隔制御

<https://youtu.be/iztbzkxXXwl?t=10>

<https://youtu.be/iztbzkxXXwl?t=70>

最後に

- ArduPilotを勉強したい方は、是非、Drone Japanさんの「ドローンソフトウェアエンジニア養成塾」に参加されてください。

ドローンソフトウェアエンジニア養成塾第4期

—



ドローンソフトウェアエンジニア養成塾
第4期全体概要

9月16日～12月16日の第1&第3土曜日

Demo & Overview	Indoor Seminar	DIY Workshop [任意参加]	Flight Experience	修了 認定
1 day x 4 hours @ 難波大 9/16 (土)	4 days x 1.5 hours @ 東京/京都 10/7(土) ~ 12/16 (土)	1 day x 6 hours @ 中広 10/21 (土)	1 day x 4 hours @ 東京近郊 / 京都近郊 12/2 (土)	

お問い合わせ先

<https://www.drone-j.com/>

株式会社 Ruby開発

(Ruby Development Inc.)



Ruby(on Rails)での
自社内 開発を行う会社です

Thank you !