

# 舞いあがれ 人力飛行機

## 第3回 フライト・ロガーのソフトウェア

樋田 啓

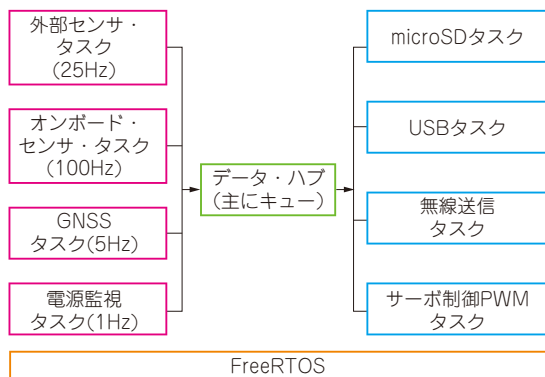


図1 フライト・ロガーのソフトウェアの概略図

人力飛行機にはさまざまなセンサが搭載されており、それらで取得した飛行データをフライト・ロガーにより記録します。サンプリング・レートの異なる複数のセンサから情報が集まるわけです。

データの送信先も、データ記録用のmicroSDやデータ表示端末など複数存在します。

これらをマイコンの限られたCPUパワーで効率良く処理するためには、フライト・ロガーのソフトウェアもハードウェア同様に重要です。

今回は人力飛行機のコンテストに15年ほど参加してきた筆者が、自作したフライト・ロガーのソフトウェアを紹介します。実際に取得したフライト・ログも併せて掲載します。

## フライト・ロガーのソフトウェア

### ● FreeRTOSのタスクに分けて実行

フライト・ロガーのソフトウェアはリアルタイムOSのFreeRTOS<sup>(1)</sup>をベースに構成しました。概要を図1に示します。

FreeRTOS上では、小さな機能単位に対応する複数のタスクを動かしています。それらはデータ・ハブ(キュー)を介してセンサによる測定値をやり取りします。



写真1 製作したフライト・ロガーを搭載した機体の試験飛行。湖で行われるコンテストでは飛行機にボートで伴走する。試験飛行では伴走者は自転車で追いかける。

キューはmicroSDやUSB、無線モジュールといった出力先の数だけ用意し、センサ・データを取得するタスク(図1の左側)から送られたデータを格納します。

マイコンのA-Dコンバータやカウンタなどに接続する外部センサ、SPI接続のオンボード・センサ、GNSSモジュールに対応するデータを入力するタスクは、センサのサンプリング・レートに合わせた周期(図1)で実行されます。

例えば、加速度/ジャイロ/地磁気/気圧センサは、10ms間隔(100Hz)で実行されるオンボード・センサ・タスクの中でセンサ・データを取得します。

これらのセンサ・データは、GNSS時刻に基づくタイム・スタンプを付与した上、バイナリ形式の32バイトのブロック<sup>(2)</sup>に整形してキューに入れます。

また、microSD、USBなどに対応するデータを出力するタスクは、処理すべきデータがキューにあるかどうかを定期的にチェックし、データがある場合は出力処理を行います。

### ● センサ・データ取得と通信処理を両立させる工夫

USBや無線モジュールでデータを出力する場合は、