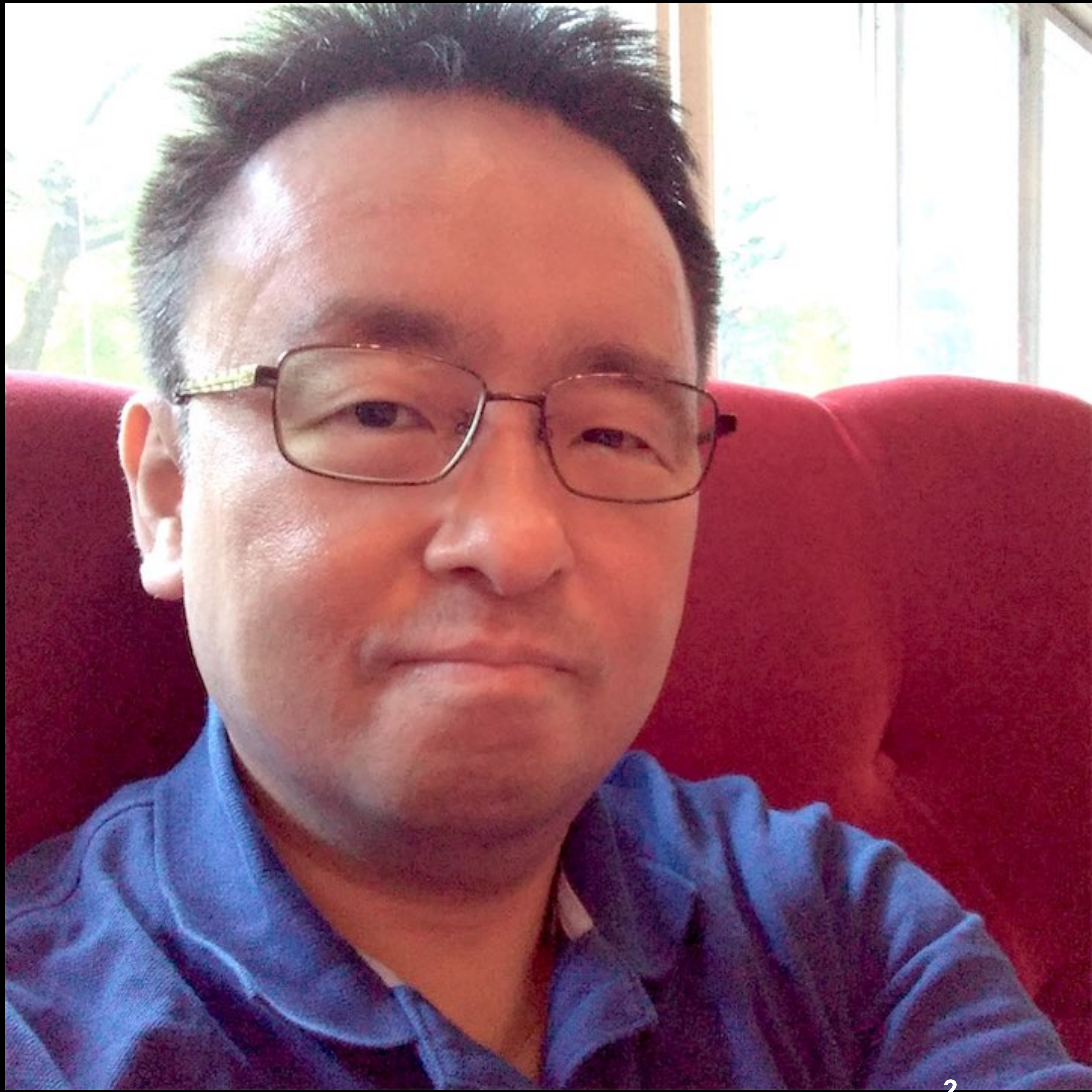
# How I discover a working implementation of `clock_nanosleep()` for macOS in CPAN `Time::Hires`

# Kenji Rikitake

# りきたけ けんじ

# 力武 健次

2-AUG-2019
kichijoji.pm 19
Musashino Public Hall
Musashino City, Tokyo, Japan
@jj1bdx



技術士（情報工学部門）
力武健次技術士事務所 所長
情報処理安全確保支援士
Guest Researcher at Pepabo R&D Institute
ペパボ研究所 客員研究員

# Time+sleeping are hard

# `sleep()`, `usleep()`, `nanosleep()`

## `gettimeofday()`, `gmtime()`, `localtime()`

Please don't mention all those NTP kludges and Erlang monotonic time implementation complexities for maintaining the monotonic increase characteristics of the internal time measurement. Thank you.

# Newer functions in Linux

`clock_gettime()`

`clock_nanosleep()`

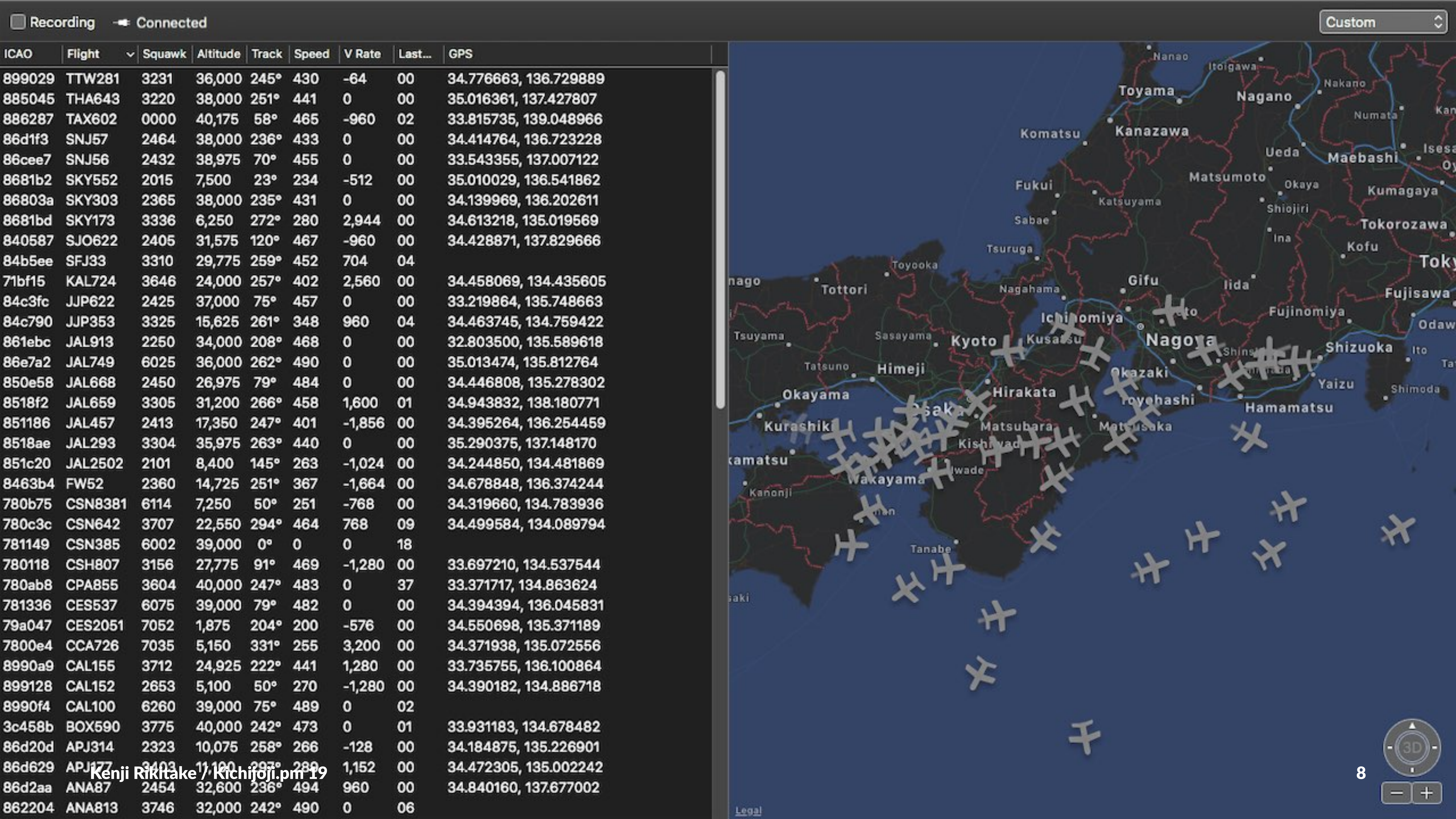
They are even defined in POSIX!

# Status on macOS Mojave 10.14.6

- `clock_gettime()`: defined since macOS Sierra 10.12 SDK
- `clock_nanosleep()`: still *undefined*
- So I need to build my own equivalent function to substitute `clock_nanosleep()`
- macOS uses `mach_absolute_time()`, `mach_wait_until()`, `mach_timebase_unfo()`, etc., defined in <`mach/mach_time.h`>

# Why I needed `clock_nanosleep()`?

- For dump1090, an ADS-B decoding software
- ADS-B: See FlightRadar24, FlightAware, etc.
- When dump1090 retrieves radiowave signals from a recorded file, it uses `clock_gettime()` and `clock_nanosleep()`



| ICAO | Flight | Squawk | Altitude | Track | Speed | V Rate | Last... | GPS |
|---|---|---|---|---|---|---|---|---|
| 899029 | TTW281 | 3231 | 36,000 | 245° | 430 | -64 | 00 | 34.776663, 136.729889 |
| 885045 | THA643 | 3220 | 38,000 | 251° | 441 | 0 | 00 | 35.016361, 137.427807 |
| 886287 | TAX602 | 0000 | 40,175 | 58° | 465 | -960 | 02 | 33.815735, 139.048966 |
| 86d1f3 | SNJ57 | 2464 | 38,000 | 236° | 433 | 0 | 00 | 34.414764, 136.723228 |
| 86cee7 | SNJ56 | 2432 | 38,975 | 70° | 455 | 0 | 00 | 33.543355, 137.007122 |
| 8681b2 | SKY552 | 2015 | 7,500 | 23° | 234 | -512 | 00 | 35.010029, 136.541862 |
| 86803a | SKY303 | 2365 | 38,000 | 235° | 431 | 0 | 00 | 34.139969, 136.202611 |
| 8681bd | SKY173 | 3336 | 6,250 | 272° | 280 | 2,944 | 00 | 34.613218, 135.019569 |
| 840587 | SJO622 | 2405 | 31,575 | 120° | 467 | -960 | 00 | 34.428871, 137.829666 |
| 84b5ee | SFJ33 | 3310 | 29,775 | 259° | 452 | 704 | 04 | |
| 71bf15 | KAL724 | 3646 | 24,000 | 257° | 402 | 2,560 | 00 | 34.458069, 134.435605 |
| 84c3fc | JJP622 | 2425 | 37,000 | 75° | 457 | 0 | 00 | 33.219864, 135.748663 |
| 84c790 | JJP353 | 3325 | 15,625 | 261° | 348 | 960 | 04 | 34.463745, 134.759422 |
| 861ebc | JAL913 | 2250 | 34,000 | 208° | 468 | 0 | 00 | 32.803500, 135.589618 |
| 86e7a2 | JAL749 | 6025 | 36,000 | 262° | 490 | 0 | 00 | 35.013474, 135.812764 |
| 850e58 | JAL668 | 2450 | 26,975 | 79° | 484 | 0 | 00 | 34.446808, 135.278302 |
| 8518f2 | JAL659 | 3305 | 31,200 | 266° | 458 | 1,600 | 01 | 34.943832, 138.180771 |
| 851186 | JAL457 | 2413 | 17,350 | 247° | 401 | -1,856 | 00 | 34.395264, 136.254459 |
| 8518ae | JAL293 | 3304 | 35,975 | 263° | 440 | 0 | 00 | 35.290375, 137.148170 |
| 851c20 | JAL2502 | 2101 | 8,400 | 145° | 263 | -1,024 | 00 | 34.244850, 134.481869 |
| 8463b4 | FW52 | 2360 | 14,725 | 251° | 367 | -1,664 | 00 | 34.678848, 136.374244 |
| 780b75 | CSN8381 | 6114 | 7,250 | 50° | 251 | -768 | 00 | 34.319660, 134.783936 |
| 780c3c | CSN642 | 3707 | 22,550 | 294° | 464 | 768 | 09 | 34.499584, 134.089794 |
| 781149 | CSN385 | 6002 | 39,000 | 0° | 0 | 0 | 18 | |
| 780118 | CSH807 | 3156 | 27,775 | 91° | 469 | -1,280 | 00 | 33.697210, 134.537544 |
| 780ab8 | CPA855 | 3604 | 40,000 | 247° | 483 | 0 | 37 | 33.371717, 134.863624 |
| 781336 | CES537 | 6075 | 39,000 | 79° | 482 | 0 | 00 | 34.394394, 136.045831 |
| 79a047 | CES2051 | 7052 | 1,875 | 204° | 200 | -576 | 00 | 34.550698, 135.371189 |
| 7800e4 | CCA726 | 7035 | 5,150 | 331° | 255 | 3,200 | 00 | 34.371938, 135.072556 |
| 8990a9 | CAL155 | 3712 | 24,925 | 222° | 441 | 1,280 | 00 | 33.735755, 136.100864 |
| 899128 | CAL152 | 2653 | 5,100 | 50° | 270 | -1,280 | 00 | 34.390182, 134.886718 |
| 8990f4 | CAL100 | 6260 | 39,000 | 75° | 489 | 0 | 02 | |
| 3c458b | BOX590 | 3775 | 40,000 | 242° | 473 | 0 | 01 | 33.931183, 134.678482 |
| 86d20d | APJ314 | 2323 | 10,075 | 258° | 266 | -128 | 00 | 34.184875, 135.226901 |
| 86d629 | APJ177 | 3403 | 11,100 | 297° | 289 | 1,152 | 00 | 34.472305, 135.002242 |
| 86d2aa | ANA87 | 2454 | 32,600 | 236° | 494 | 960 | 00 | 34.840160, 137.677002 |
| 862204 | ANA813 | 3746 | 32,000 | 242° | 490 | 0 | 06 | |

I found a reference implementation of *emulated* `clock_nanosleep()` in `Time::Hires` of Perl CPAN

# CPAN Time::Hires

```
# High resolution alarm, sleep, gettimeofday, interval timers
# which consist of the system call and library function wrappers

use Time::HiRes qw( clock_gettime clock_getres clock_nanosleep
                    ITIMER_REAL ITIMER_VIRTUAL ITIMER_PROF
                    ITIMER_REALPROF );

$realtime   = clock_gettime(CLOCK_REALTIME);
$resolution = clock_getres(CLOCK_REALTIME);

clock_nanosleep(CLOCK_REALTIME, 1.5e9);
clock_nanosleep(CLOCK_REALTIME, time()*1e9 + 10e9, TIMER_ABSTIME);
```

# Time::Hires has the C code for emulating clock_nanosleep() for macOS! [1]

```
#ifdef TIME_HIRES_CLOCK_NANOSLEEP_EMULATION
static int th_clock_nanosleep(clockid_t clock_id, int flags,
               const struct timespec *rqtp,
               struct timespec *rmtp) {
  if (darwin_time_init()) {
    switch (clock_id) {
    case CLOCK_REALTIME:
    case CLOCK_MONOTONIC:
// Refer to the original code for the further information
```

[1] https://metacpan.org/source/ATOOMIC/Time-HiRes-1.9760/HiRes.xs

# Porting the `Time::Hires` code to dump1090 worked! [2]

- Tested by `dump1090 --ifile test.bin`
- Worked the same as in Linux

[2] https://github.com/jj1bdx/dump1090/commit/ba0b63ee1eab28e42c61d9005eadd036503d2bd7

# Lessons learned

*Standards change*

Learn new system calls and library functions

Perl CPAN has many valuable code jewels

# Thank you
# Questions?