

# MG4 赤丸追跡ロボット の ソフトウェア解説

2018年12月

S. Ushimaru

# 動作概要

- フィールド内の赤丸を認識し、それを追いかける。
- 追尾中に赤丸を見失った場合は、再び探索行動を行う。
- 指定距離以内に物体（壁や人）があれば、回避動作を行う。（探索中のみ）
- 動作状態を音声で知らせる。
- いくつかのスレッドを生成して、マルチタスクで動作させる。
- 機体が停止してしまった場合は、脱出行動を行う。（未実装）

# MIRS、フィールド、赤丸

## • MIRS

- 機体の前後にカメラ（Logicool E270）を取り付ける。
- 機体前方の左右に超音波センサを取り付ける。
- Raspberry Pi のオーディオポートにスピーカーを接続する。  
（電源はポータブルバッテリーのUSBポートより取る）
- モータはMIRS MG4と同じ、ギア比は27:1のものを使用

## • フィールド

- フィールドの広さは任意
- フィールドは高さ30cmの壁で囲う。

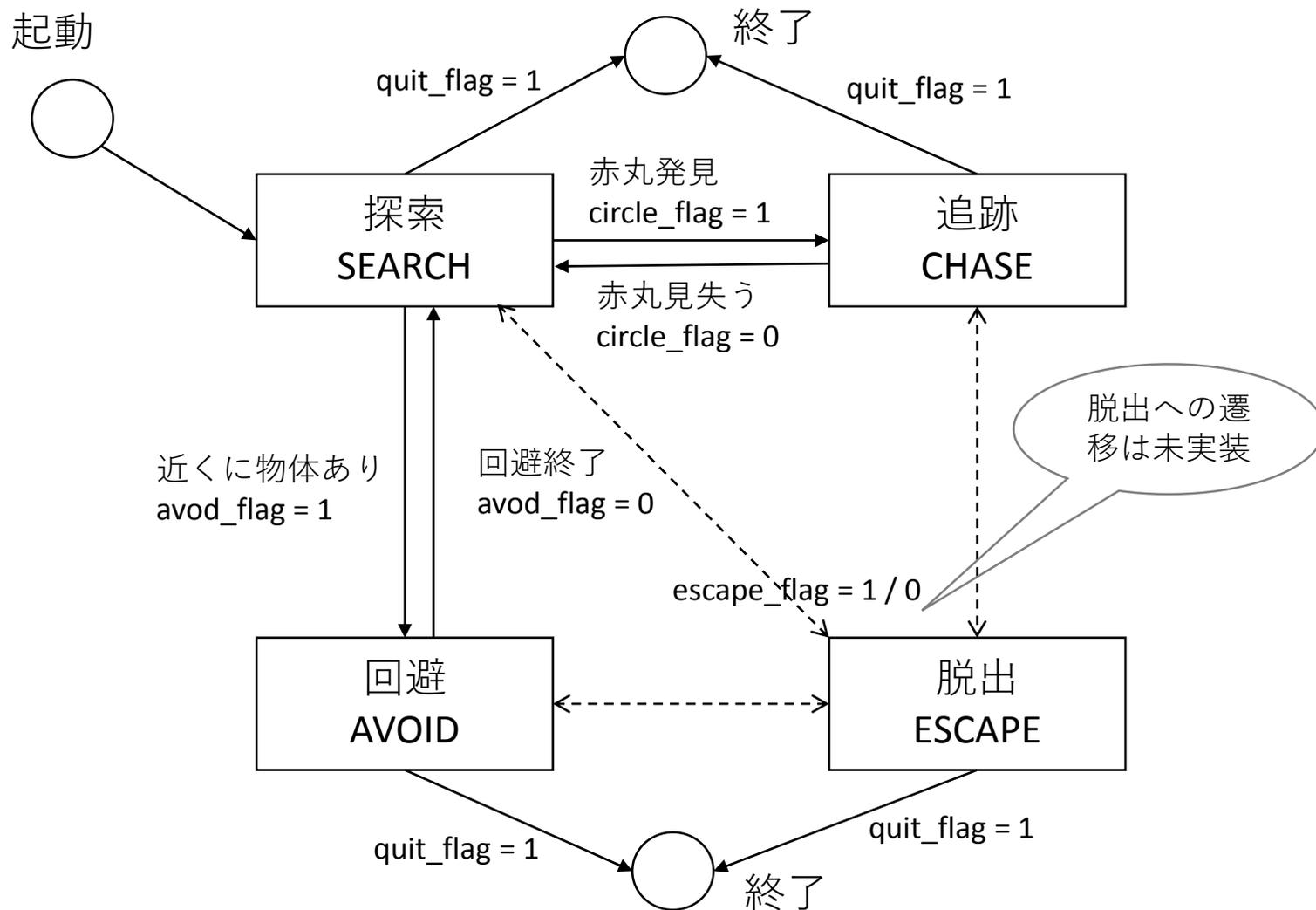
## • 赤丸

- 直径約10cmの白いボール紙を赤く塗って、長さ1mのアルミ製の棒の先端に取り付ける。
- この棒を機体の前にかざして、MIRSに発見・追尾させる。

# 動作モードの定義

動作モード	モード名	動作の概要
探索モード	SEARCH	赤丸を探す。 回転・直進を繰り返して赤丸を見つける。
追跡モード	CHASE	赤丸を追いかける。 赤丸方向に向きを変えて直進する。
回避モード	AVOID	物体から遠ざかる。 180度回転して、指定距離直進する。
脱出モード	ESCAPE	停止状態から脱出する。 後退と回転動作を組み合わせ、停止状態を抜ける。

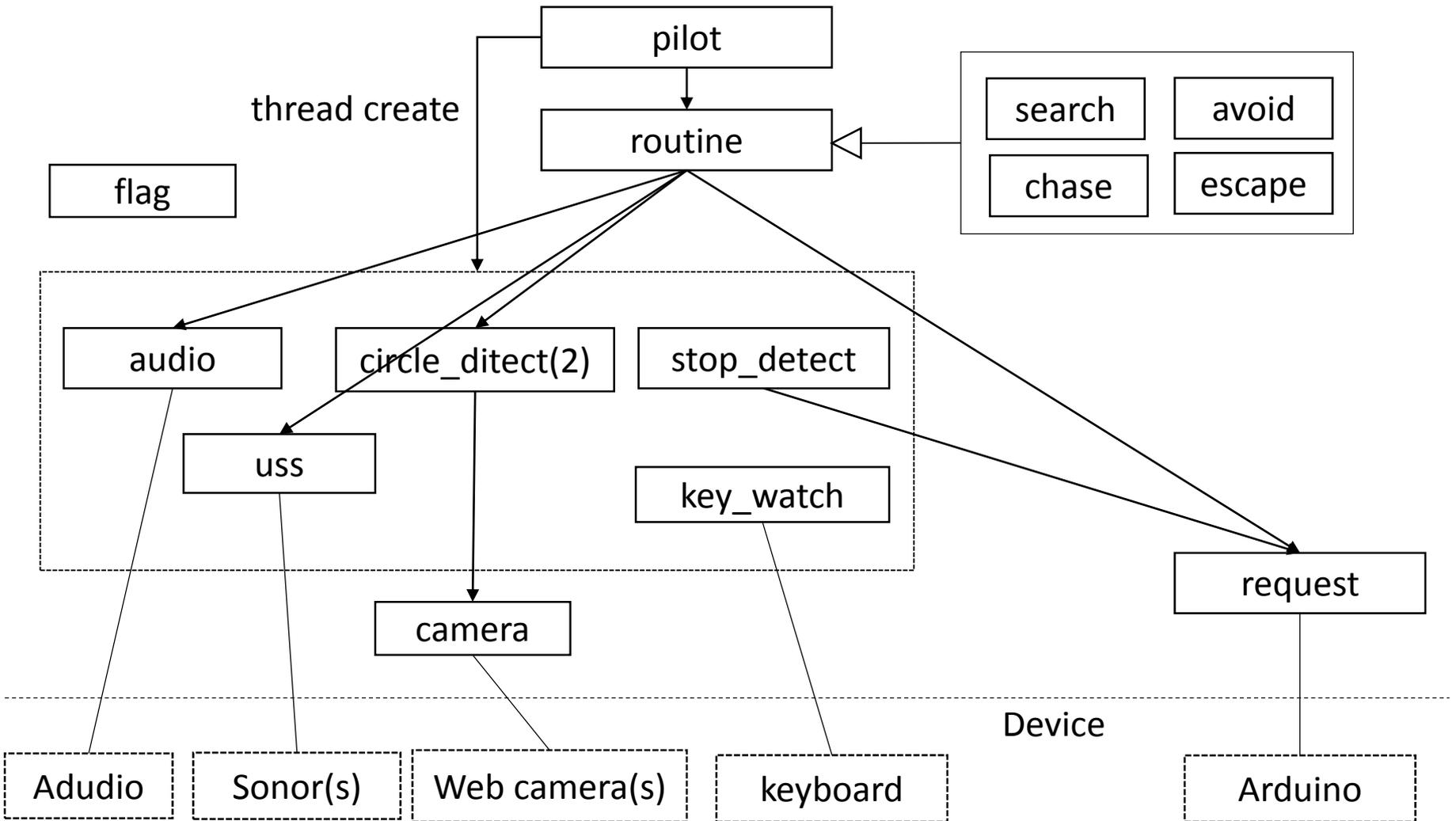
# 動作モードの遷移



# スレッド構成

スレッド名	概要	実装ファイル	備考
赤丸認識	前後のカメラ画像を処理し、赤丸エリアを特定する	circle_detect.c circle_detect2.c	ファイル名・関数名に2が付いてものが後方カメラ用、付いていなものが前方カメラ用
超音波計測	一定周期で前方左右の超音波計測を行う	uss.c	計測周期は100[ms] 標準プログラムと同じ
音声出力	指定された番号の音声データを出力する	audio.c	動作中に音声出力を繰り返す場合に使用する
キー入力監視	キー入力があるかを監視する	key_watch.c	“q”の入力で quit_flag を上げる
停止検知	タイヤが動いていないことを検知する	stop_detect.c	停止と判定されたら、stop_flag を上げる

# モジュール構造図



# モジュール一覧

モジュール名	概要	
pilot	最初に起動動作を行う。 動作モードを切り替える。	新規
routine_x	動作モードを定義したモジュール モード x は search（探索）、chase（追跡）、avoid（回避）、 escape（脱出）の4つがある。	新規
audio	Audio デバイスに音声を出力する。 thread として動作する。	新規
circle_detect	カメラ画像から赤丸エリアを特定する。 前後のカメラそれぞれについてモジュールを定義し、それぞれthread として動作する。 後方カメラ用のモジュールには 2 を付けている。	改変 後方カメラ用は 新規
camera	カメラ画像のキャプチャを行う。	
key_watch	キーボード入力を監視する。thread として動作する。 終了キー（q）が入力されたかを判断する。	新規
stop_detect	左右モータのエンコーダ値を監視し、停止状態にあるかを判断する。	新規
request	Arduino との通信を行う。thread として動作する。	改良
flag	フラグ変数の定義と初期化	新規

# フラッグ

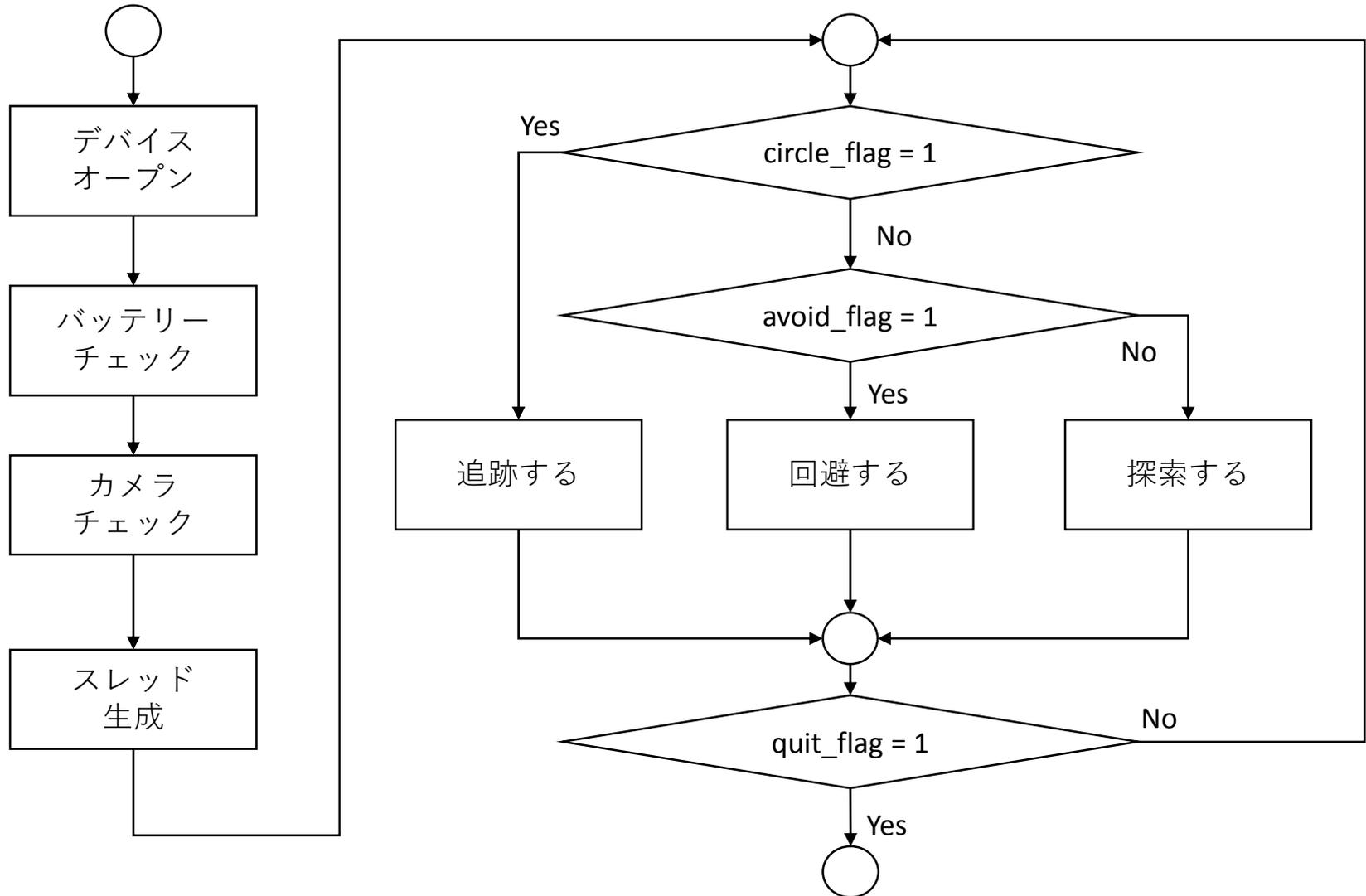
- スレッド間のデータ共有はフラッグを用いて行う。
  - 共有変数としてフラッグ変数を用意する。
  - フラッグ変数は `flag.c` で定義、`flag.h` で `extern` 宣言している。
  - フラッグ変数は `int` 型で定義し、値は 0 または 1

フラッグ変数	フラッグ値を 1 とする状況	備考
<code>forward_flag</code>	前進動作のとき 1	
<code>backward_flag</code>	後退動作のとき 1	
<code>rotate_flag</code>	回転動作のとき 1	
<code>stop_flag</code>	機体停止と認識したとき 1	
<code>circle_flag</code>	赤丸を認識したとき 1	
<code>avoid_flag</code>	回避が必要と判断したとき 1	
<code>escape_flag</code>	脱出が必要と判断したとき 1	
<code>quit_flag</code>	キーボードから停止入力があったとき 1	

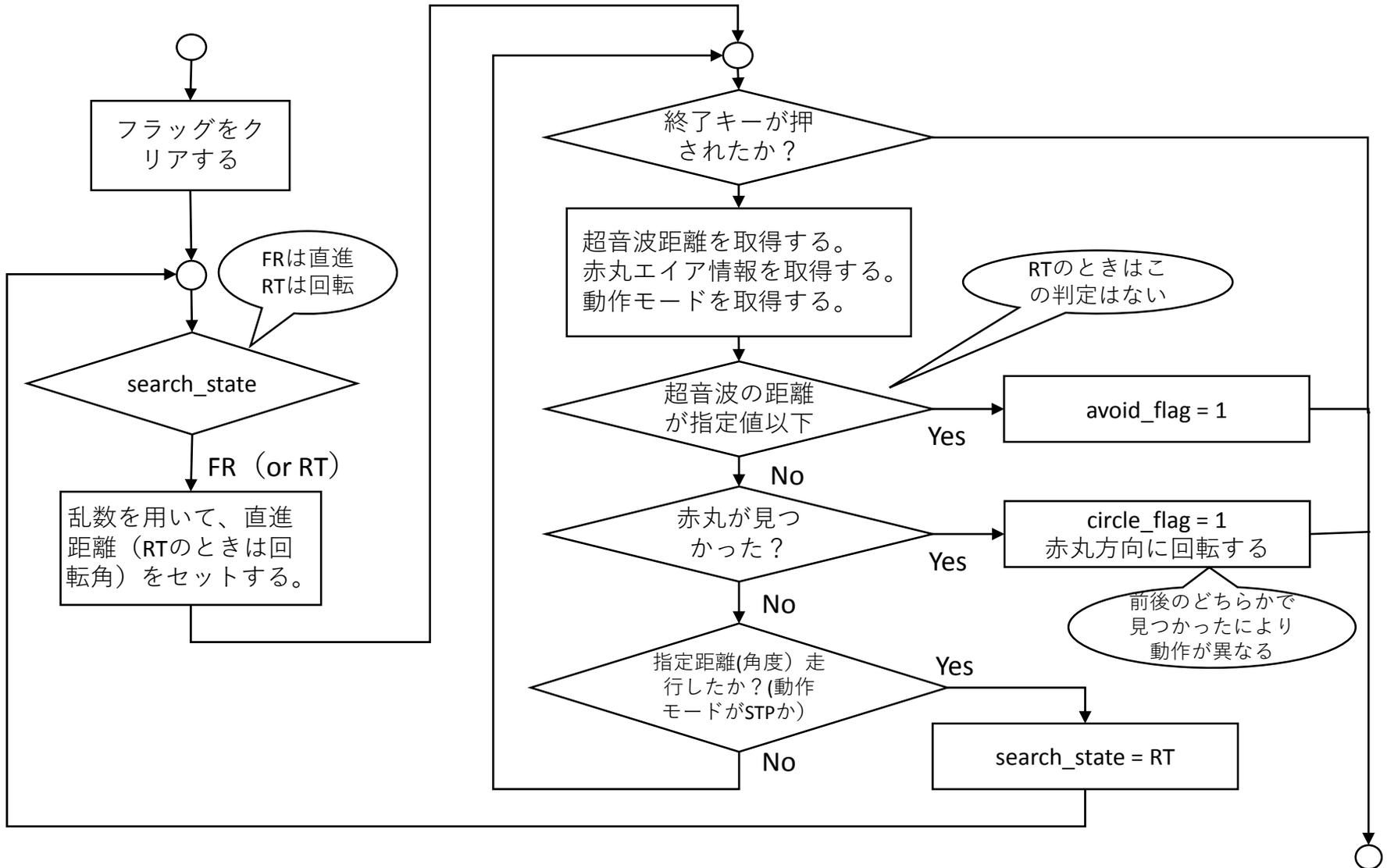
# 音声出力

知らせること	音声ファイル	備考
検索中	search.wav, search2.wav	検索中に繰り返し
赤丸を発見	find.wav	後ろカメラで発見時に向き直るまで繰り返し（音声スレッド）
追跡中	chase.wav	追跡中に繰り返し（音声スレッド）
赤丸を見失った	lost.wav	見失ったときに1度だけ
物体に近づき過ぎた	otto.wav	回避中に繰り返し（音声スレッド）
バッテリーなし	batt_off.wav	起動時に1度だけ
バッテリー電圧が低下	batt_low.wav	起動時に1度だけ
起動準備完了	ready.wav	起動時に1度だけ

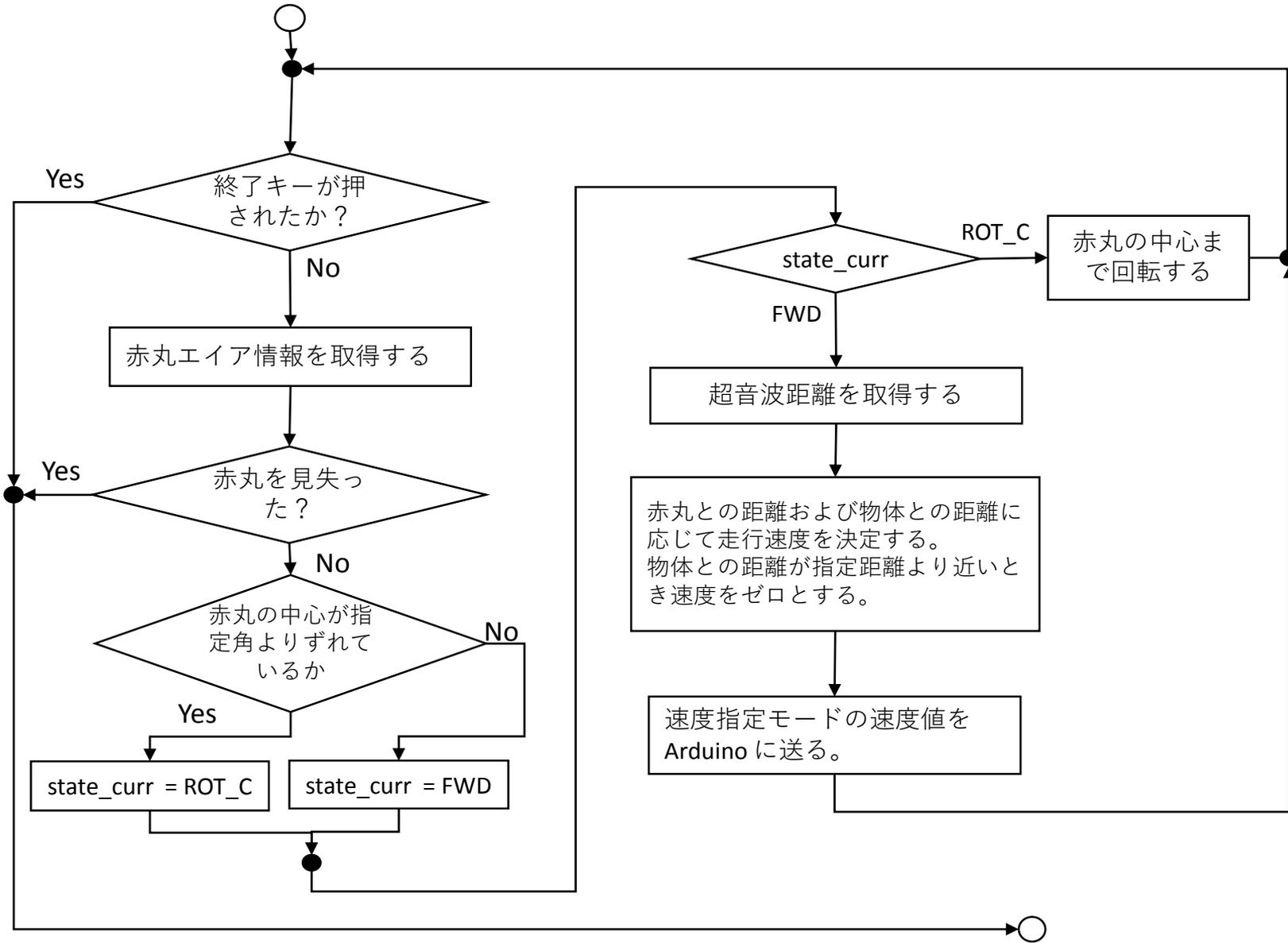
# pilotモジュールの動作フロー



# 探索の動作フロー



# 追跡の動作フロー



# 起動時の動作

- バッテリー状態をチェックし、電圧が基準値以下の場合、その旨を音声で伝え、プログラムを終了する。
- 2つのカメラの認識ができなかった際は、その旨を音声で伝え、プログラムを終了する。
- デバイスファイルを開く。
  - 開くことが出来なかった場合は、プログラムを終了する。
- スレッドを create する。

# 走行モードへの速度指定モード(VEL)の追加に伴うモジュールの変更

- 概要

- 追跡動作を行う際に、機体の速度を指定して走行するようにさせる。そのために、走行モードに速度指定モード VEL を追加した。

- RaspberryPi の request モジュールの変更

ファイル	関数	変更内容
request.c	request_set_run_mode()	command_data に run_state に VEL を加える。
request.h		run_state_t に VEL を加える。
test_request.c	test_request()	case に VEL モードを加える。

## • Arduino プログラムの変更

ファイル	変更内容
define.h	run_state_t に VEL を加える。
slave.ino	switch 文に VEL に対応する case 4 のエントリを加える。
run_control.ino	stwitch 文に case VEL のエントリを加える。 run_control_set(), run_control_execute() の両方を変更

ソースコード

Raspberry Pi プログラム

[http://www2.denshi.Numazu-ct.ac.jp/mirsdcoc2/mirsmg4d/syst/num0010a/mg4\\_festival\\_demo\\_program.tar.gz](http://www2.denshi.Numazu-ct.ac.jp/mirsdcoc2/mirsmg4d/syst/num0010a/mg4_festival_demo_program.tar.gz)

Arduino プログラム

[http://www2.denshi.Numazu-ct.ac.jp/mirsdoc2/mirsmg4d/syst/num0010a/mg4\\_arduino\\_v321.zip](http://www2.denshi.Numazu-ct.ac.jp/mirsdoc2/mirsmg4d/syst/num0010a/mg4_arduino_v321.zip)