

# AK9754 approach and departure detection Sample source code



# 目次

- はじめに
- ソースコードファイルについて
- 動作フロー
- パラメータ設定
- 立去り検知アルゴリズム詳細



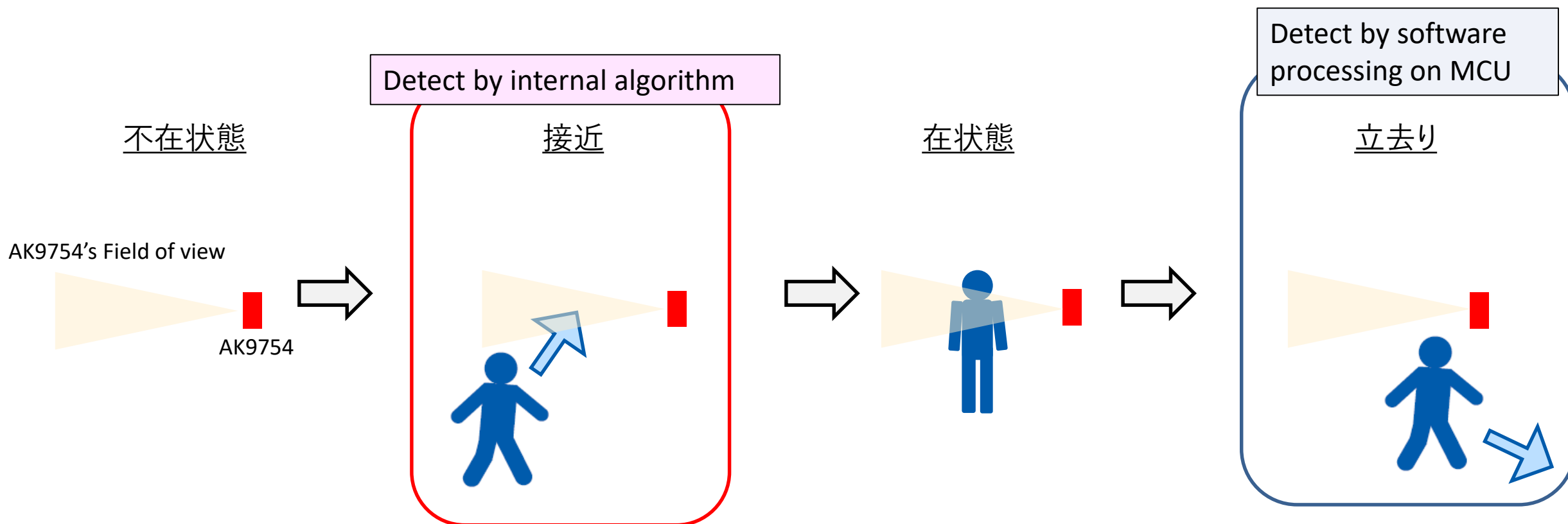
はじめに

# 用語定義

- **在状態**
  - センサ視野内に人がいる状態
- **不在状態**
  - センサ視野内に人が全くいない状態
- **接近**
  - 不在状態から在状態に遷移するイベント
- **立去り**
  - 在状態から不在状態に遷移するイベント

# アルゴリズムのコンセプト

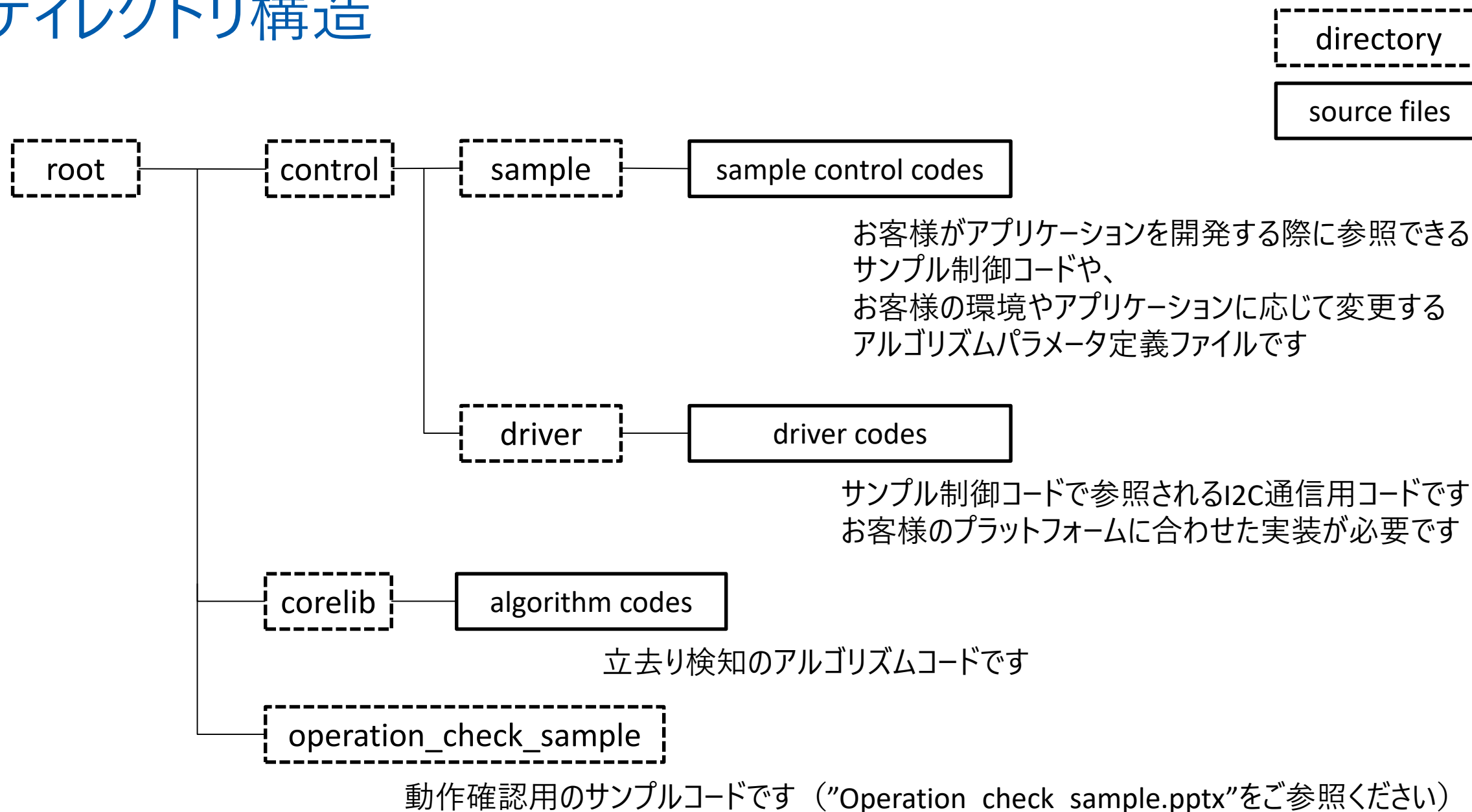
- ソースコードでは、**接近**をAK9754の**内蔵アルゴリズム**により検知し、**立去り**をMCU上の**ソフトウェア処理**により検知することで、現在の状態を識別します。





## ソースコードファイルについて

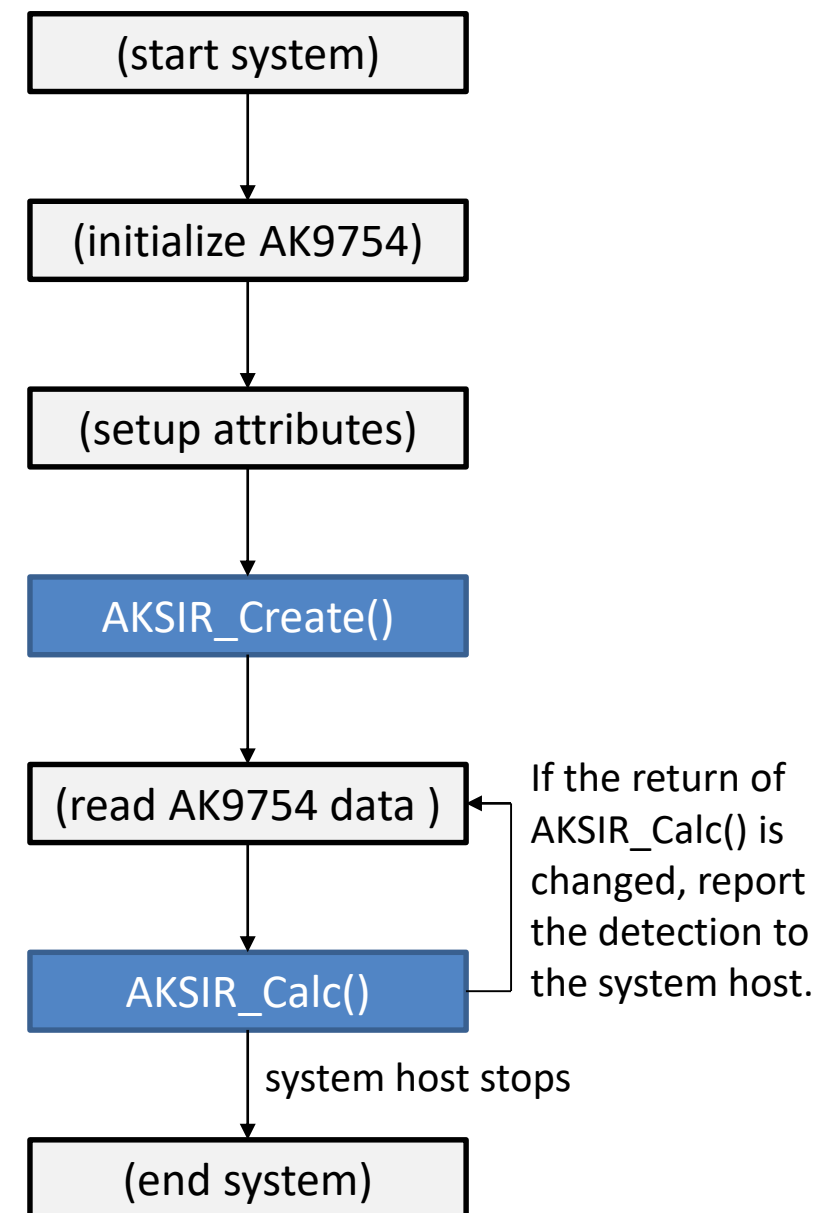
# ディレクトリ構造



# corelibについて

- ❑ 立去り検知アルゴリズム本体のソースコードです。
- ❑ corelib内のソースコードは変更していただく必要はありません。

- corelib/src/AKSIR.c
  - お客様のシステムから呼び出すAPIを定義しています。
    - AKSIR\_Create()とAKSIR\_Calc()の二つのAPI関数があります。
  - 右図の順番でAPIを使用してください。
  - 後述する制御コード例のmain.cに具体的な記述例があります。





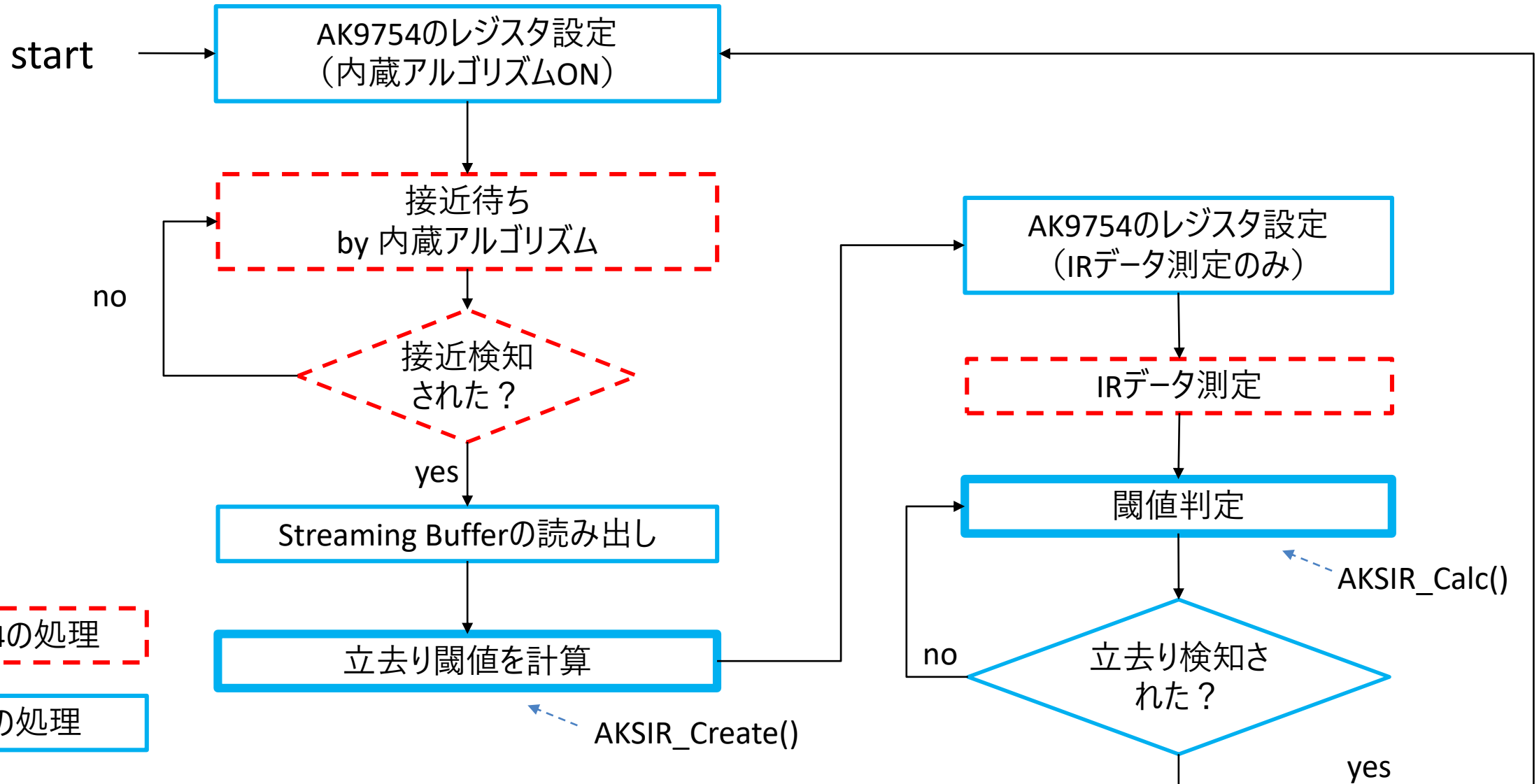
# 制御コード例について

- control/sample/src/main.c
  - お客様のシステムによる制御の例です。
  - 本ソースコードを参考にして組み込み実装を行ってください。
- control/sample/include/AK9754\_param.h
  - アルゴリズムやAK9754の動作に関するパラメータが記述されています。
  - お客様の環境などに合わせて変更してください。
  - （パラメータについての説明はスライド“パラメータについて”を参照してください）
- control/driver/drivers.c
  - I2C通信などの関数が、中身のない状態で実装されています。
  - お客様のプラットフォームに合わせて実装を行ってください。



## 動作フロー

## 動作フロー





# パラメータ設定

# パラメータ設定方法

- ❑ control/sample/include/AK9754\_param.h 内にパラメータ設定が記述されています。
- ❑ 必要に応じて設定値を書き換えてください。

```
/* Algorithm parameters */  
#define NUMBER_OF_DEPARTURE_COUNTS (3)
```



```
/* Algorithm parameters */  
#define NUMBER_OF_DEPARTURE_COUNTS (5)
```

(例) NUMBER\_OF\_DEPARTURE\_COUNTSの値を3から5に変更する場合

# パラメータについて（全般）

## ■AK9754\_SLAVE\_ADR

MCUからAK9754へI2Cアクセスを行う際のスレーブアドレスです。  
AK9754のCAD pin の設定に合わせて変更してください。

| CAD1          | CAD0          | Slave Address |
|---------------|---------------|---------------|
| VSS           | VSS           | 60H           |
| VSS           | non-connected | 61H           |
| VSS           | VDD           | 62H           |
| non-connected | VSS           | 64H           |
| non-connected | non-connected | 65H           |
| non-connected | VDD           | 66H           |
| VDD           | VSS           | 68H           |
| VDD           | non-connected | 69H           |
| VDD           | VDD           | Do Not Use    |

# パラメータについて（接近検知）

## □AK9754\_HBDTH\_VAL

内蔵アルゴリズムの閾値です。

お客様の環境、窓材などに合わせて変更してください。

# パラメータについて（立去り検知）

## ■AK9754\_ODR\_VAL\_DEP

立去り検知中の、AK9754のデータ測定周波数です。

周波数を下げるほど、立去りを検知するまでの時間が長くなります。

## ■NUMBER\_OF\_DEPARTURE\_COUNTS

IR測定値が立去り閾値を下回ってから、立去りと判定するまでのサンプル数です。

この設定値の数だけ連続でIR測定値が閾値を下回った場合に、立去りと判定されます。

この数値を大きくするほど、立去りを検知するまでの時間が長くなります。

## ■MIN\_DELTA

立去り閾値を決定する時に用いる定数です。

大きくするほど立去りを検知しやすくなりますが、

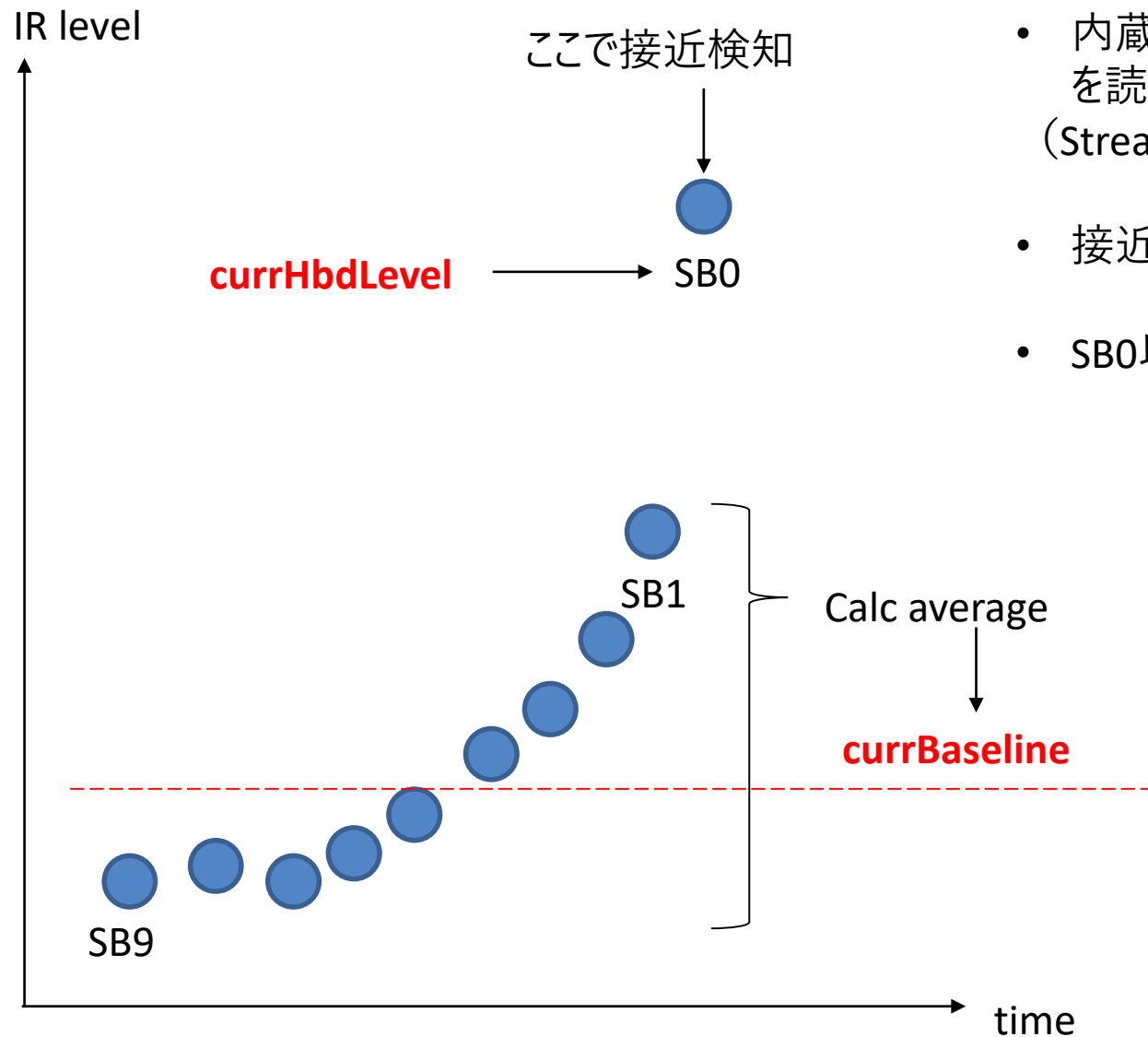
反面、在状態の時に誤って立去りを検知するケースが増える可能性があります。





## 立去り検知アルゴリズム詳細

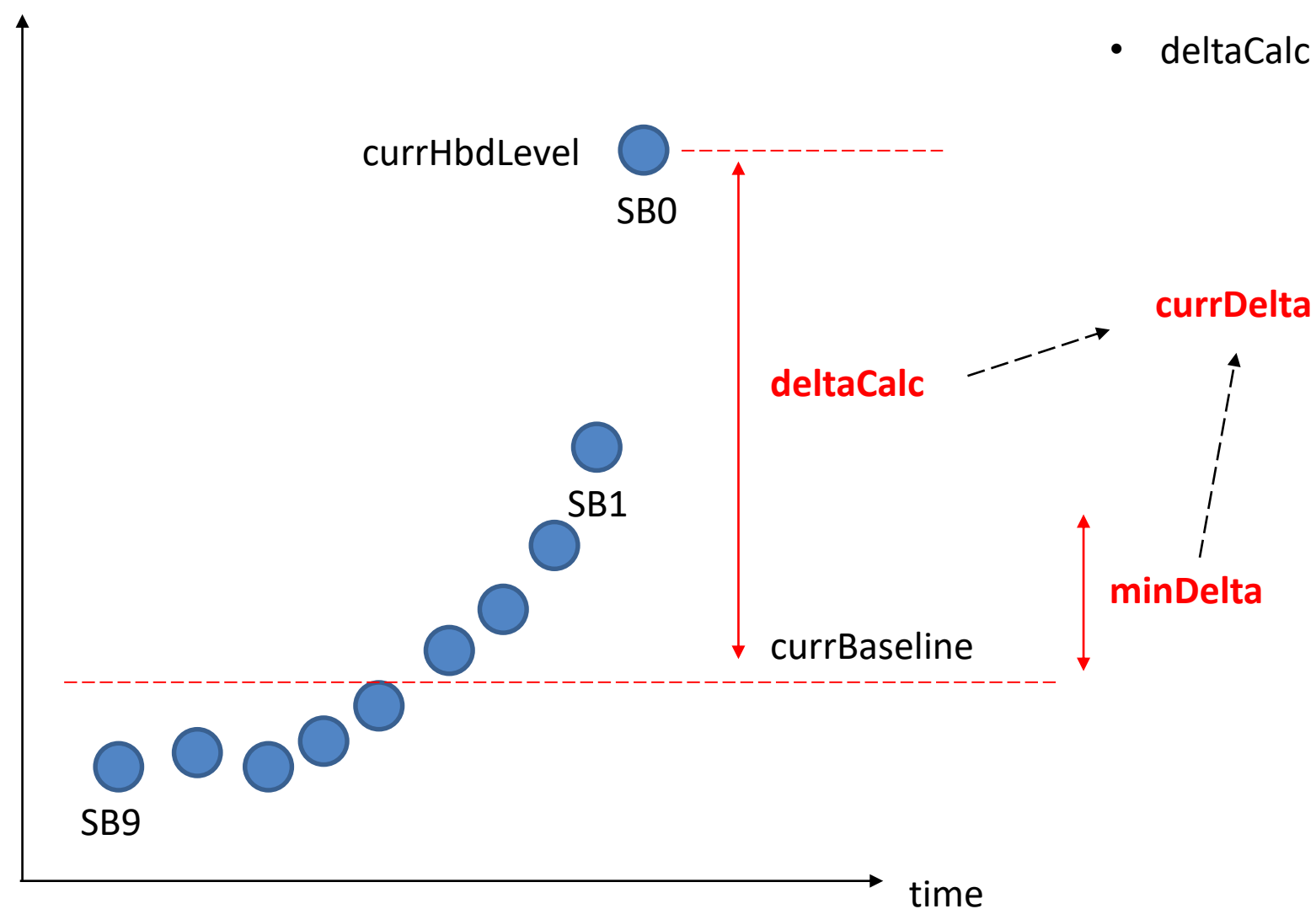
# 立去り閾値設定



- 内蔵アルゴリズムが接近を検知したら、Streaming Buffer内のIR測定値を読み出します (SB0~SB9)  
(Streaming Bufferについては、AK9754のデータシートを参照してください)
- 接近を検知した時のIR測定値 (SB0) を **currHbdLevel** に設定します
- SB0以外のIR測定値の平均を計算して **currBaseline** とします

# 立去り閾値設定

IR level

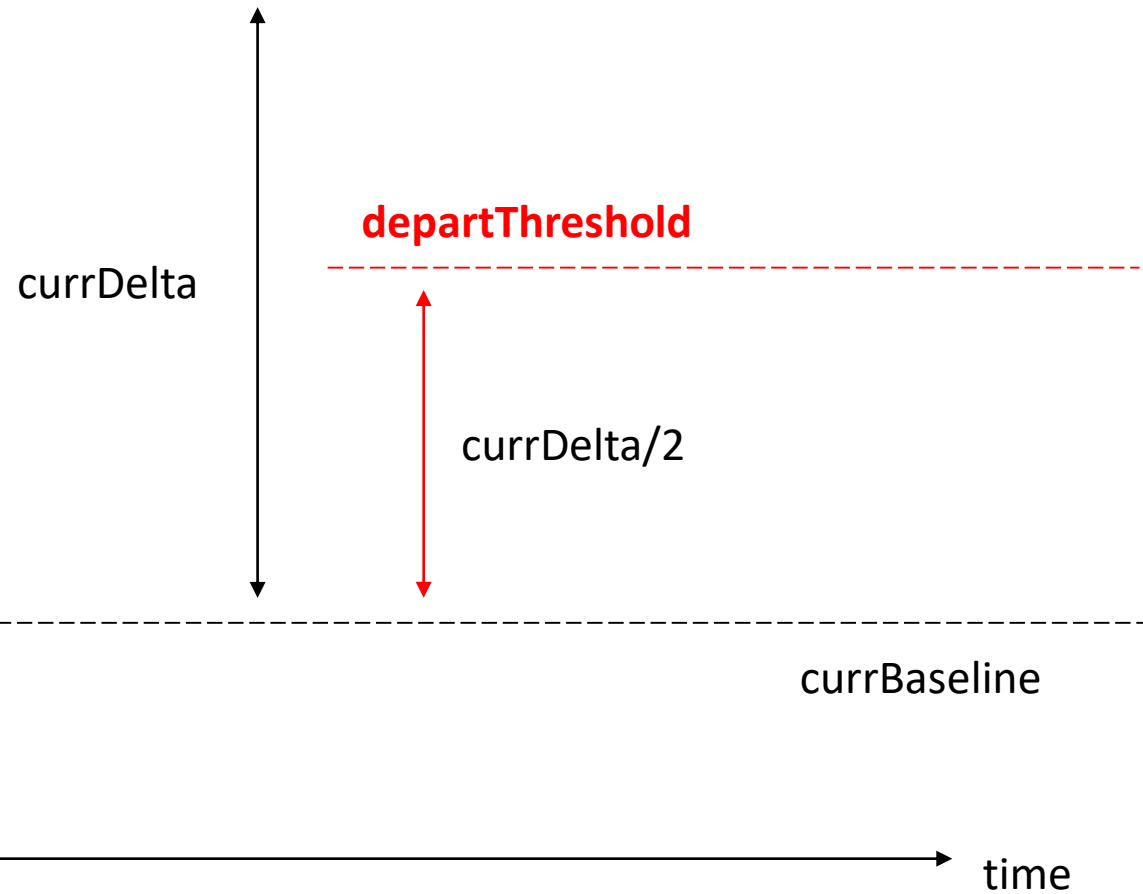


- currHbdLevel と currBaseline の差分を deltaCalc とします
- deltaCalc と MIN\_DELTA の大きい方を currDelta とします

# 立去り閾値設定

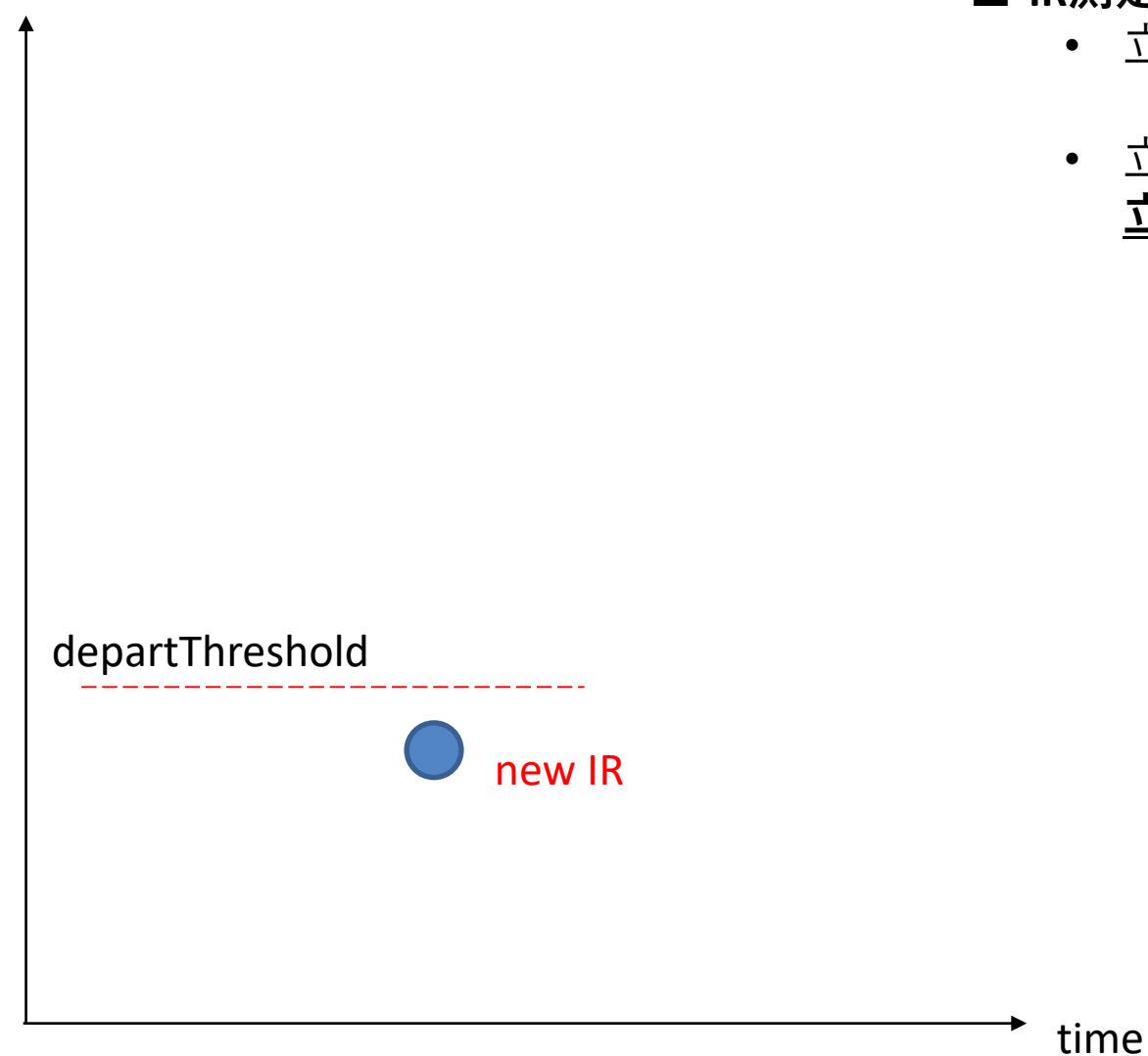
IR level

- currBaselineにcurrDelta/2を足したものを立去り閾値とします



# 立去り判定

IR level



## □ IR測定値が立去り閾値以下の場合

- 立去りカウンタを1増やします
- 立去りカウンタがNUMBER\_OF\_DEPARTURE\_COUNTS以上になったら立ち去ったと判定します

# 立去り判定

IR level

□ IR測定値が立去り閾値より高い場合

- 立去りカウンタを0に戻します

new IR

departThreshold

time



## 注意事項

# 注意事項

- 本ソースコードは、AK9754とソフトウェア処理を組み合わせ、人の接近・立去りを検知する機能を実現する手段の一例を示したものです。
- センサ視野内に人がいる状態からアルゴリズムを開始した場合は、最初の検知が正しく行われない場合があります。一度センサ視野内に人がいない状況になったのちは正しく動作します。
- 在状態中に環境温度が大きく上昇した場合、不在状態でのIR出力信号が閾値より高い値となる場合があります、立去り検知を誤る可能性があります。
- センサ視野内で、人体以外の熱源の温度変動がある場合は、誤検知する可能性があります。
- 誤検知によりフェイタルエラーが発生する可能性がある場合は、タイマー制御などによるフェールセーフ機能をご準備ください。



# *Creating for Tomorrow*

昨日まで世界になかったものを。

私たち旭化成グループの使命。

それは、いつの時代でも世界の人びとが“いのち”を育み、  
より豊かな“暮らし”を実現できるよう、最善を尽くすこと。

創業以来変わらぬ人類貢献への想いを胸に、  
次の時代へ大胆に伝えていくために一。

私たちは、“昨日まで世界になかったものを”創造し続けます。

**AsahiKASEI**