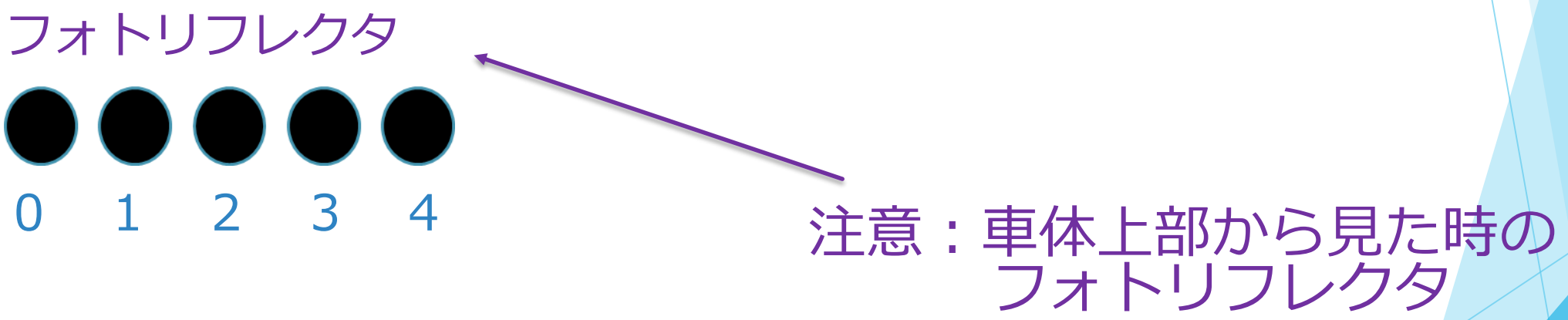
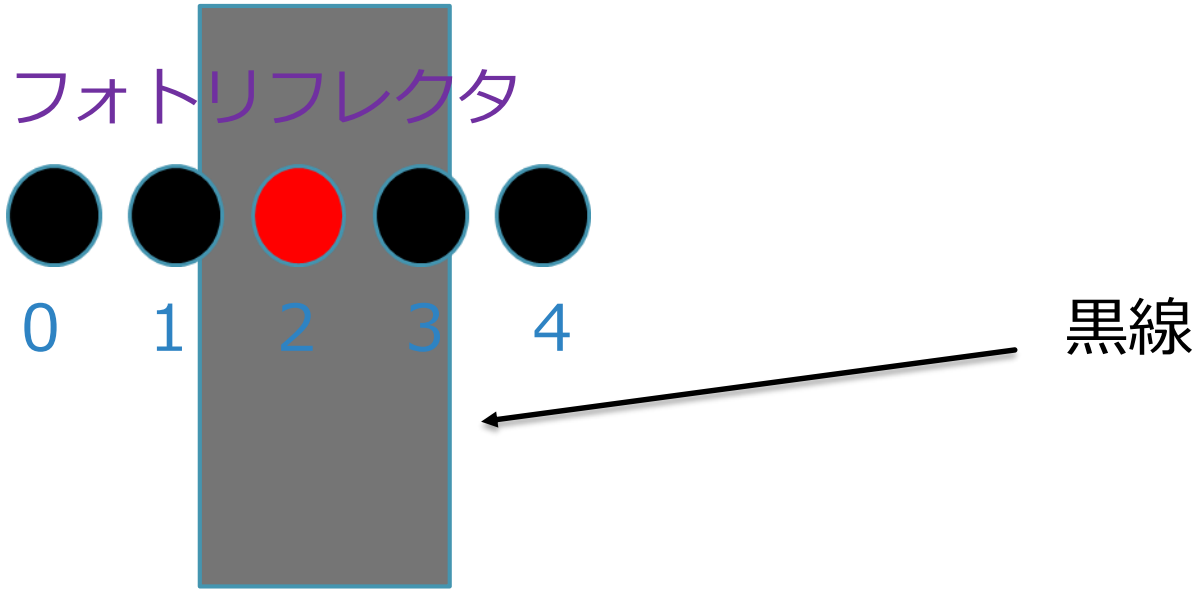
アルゴリズムの説明

- ①：黒線上の最も中央に位置するフォトリフレクタ（以下「CFR」と呼ぶ。）を調べることにより、車体が黒線上のどの位置にあるかを調べる。
- （個々のフォトリフレクタの電圧値を比較）



アルゴリズムの説明

- ①：黒線上の最も中央に位置するフォトリフレクタ（以下「CFR」と呼ぶ。）を調べることにより、車体が黒線上のどの位置にあるかを調べる。

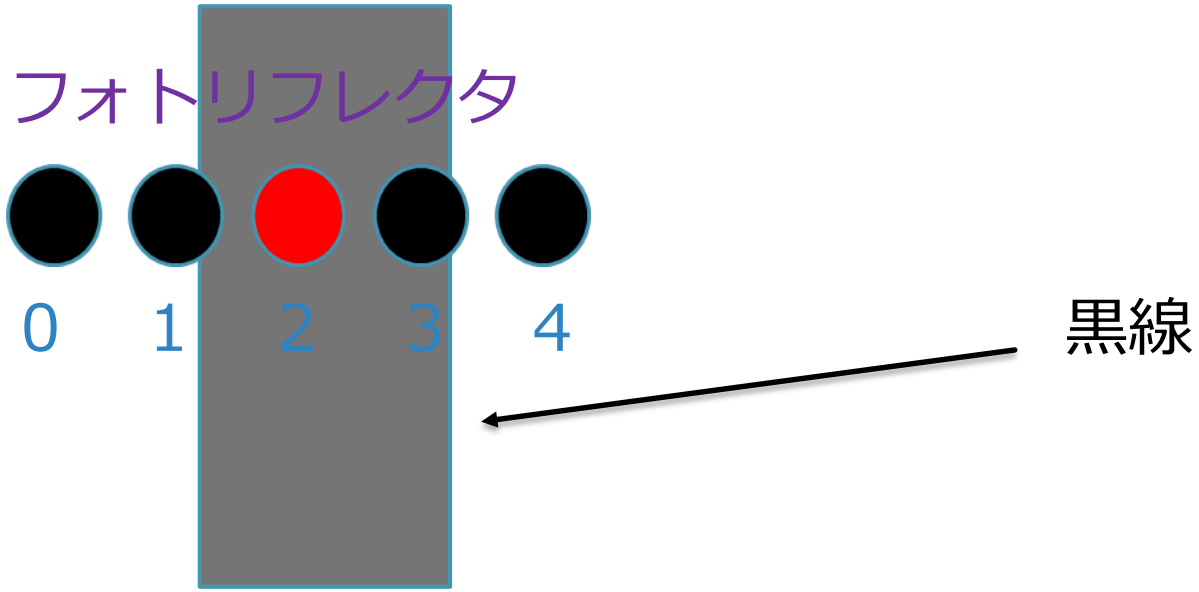


□例

番号２のフォトリフレクタが最も黒線上にある場合

アルゴリズムの説明

- ②：①で調べたフォトリフレクタ（CFR）が黒線上の少し右側に位置するか、中央に位置するか、少し左側に位置するかを調べることにより、車体の位置情報を高精度に調べ、その位置によってさらに細かく処理を指示することを実現している。

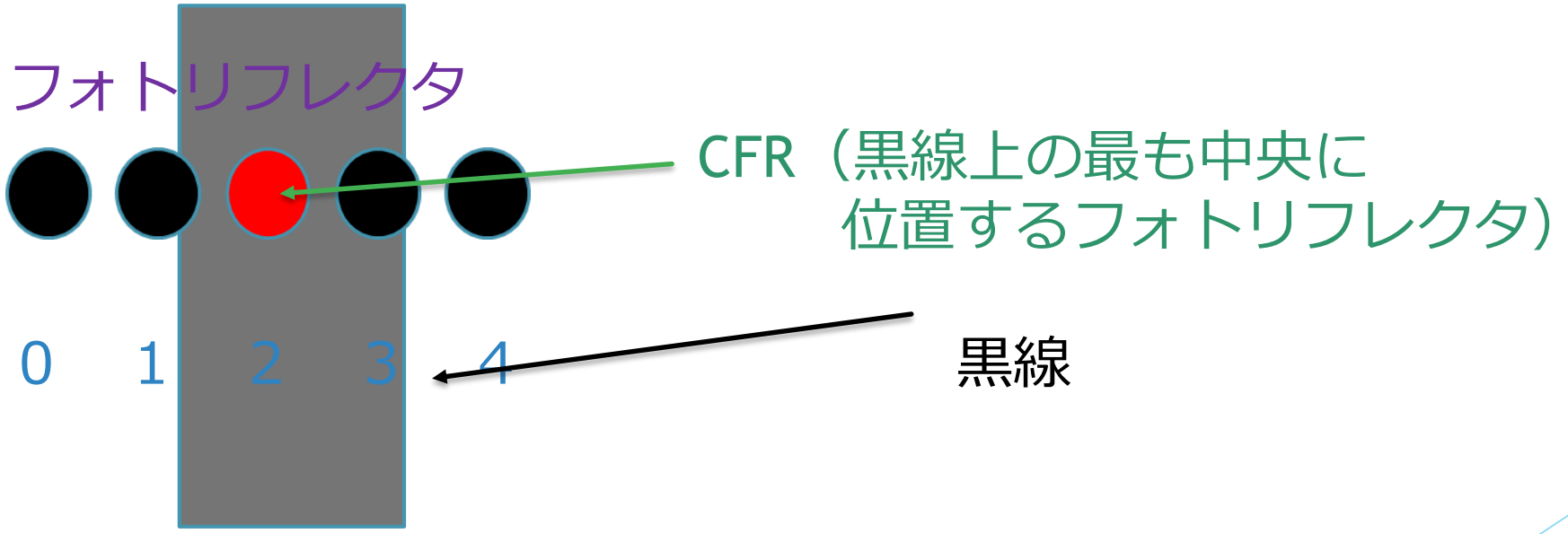


□例

番号２のフォトリフレクタが最も黒線上にある場合

アルゴリズムの説明

- 下図の例で説明すると、CFRの両端のフォトリフレクタの電圧値を比較することにより、CFRが黒線上の少し右側に位置するか、中央に位置するか、少し左側に位置するかを調べている。

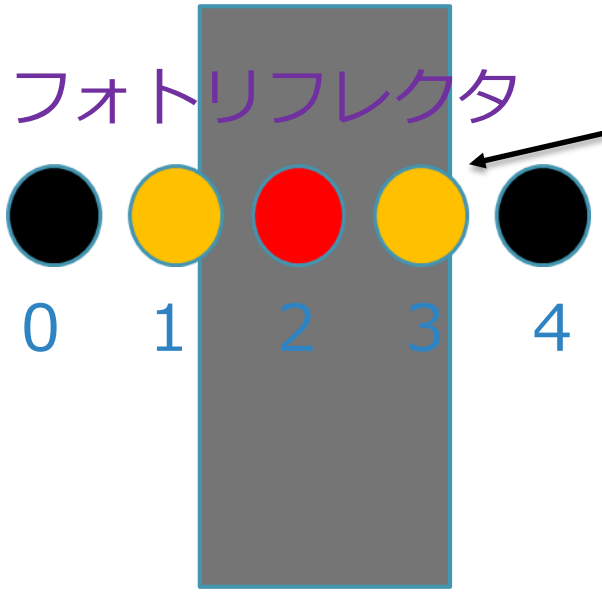


□例

番号２のフォトリフレクタが最も黒線上にある場合

アルゴリズムの説明

- 下図の例で説明すると、CFRの両端のフォトリフレクタの電圧値を比較することにより、CFRが黒線上の少し右側に位置するか、中央に位置するか、少し左側に位置するかを調べている。
-



番号３のフォトリフレクタの方が、
番号１のフォトリフレクタよりも
電圧値が高い

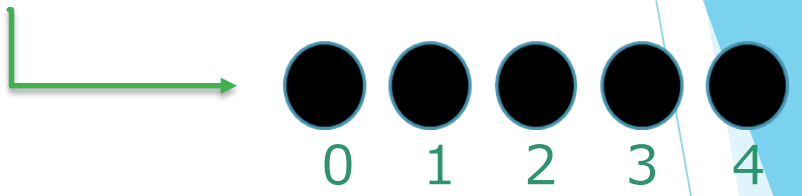
- ⇒中央を走行しているが、
- コースの少し左側を走行している

□例

番号２のフォトリフレクタが最も黒線上にある場合

アルゴリズムの説明

- 黒線の最も中央に位置するフォトリフレクタの番号における、処理の種類



- ・番号０：２パターン
- （右に完全にコースアウト or 軽度のコースアウト）
- ・番号１：３パターン（少し右寄り or 中央 or 少し左寄り）
- ・番号２：３パターン（少し右寄り or 中央 or 少し左寄り）
- ・番号３：３パターン（少し右寄り or 中央 or 少し左寄り）
- ・番号４：２パターン
- （左に完全にコースアウト or 軽度のコースアウト）

- □計１３パターンの処理

工夫した点

- ・最も端に車体が出てしまった時、その場でコースを修正するように左右のモータを逆回転させることで、カーブを速く走行できるようにした点。
- ・電圧値が最大のフォトリフレクタに対応するLEDを点灯させることで、車体が今どの位置にあるかを目視により確認できるようにした点。
- ・switch文を用い、さらにcase中でif文を使用することで、
  黒線上の車体の位置を細かく場合分けすることができ、
  それによって細かく処理を命令することができたので、
  安定した走行を実現できた点。
- ・プログラムの構造を深さ２の木構造にすることにより、loop関数の効率を上げた点。

プログラムの構造

黒線上の最も中央に位置する
フォトリフレクタ

CFRの探査

- switch文

- if文

番号０
- 重度のコースアウト（右）
- 軽度のコースアウト（右）

番号1
- コースのやや右側
- コースの右側
- コースの少し右側

番号２
- コースの中央（右寄り）
- コースの中央（真ん中）
- コースの中央（左寄り）

番号3
- コースの少し左側
- コースの左側
- コースのやや左側

番号4
- 軽度のコースアウト（左）
- 重度のコースアウト（左）

予測所要時間（２周分）

- 約３８秒