

講師用

ロボット博士養成講座

ロボティクスプロフェッサーコース

不思議アイテムⅢ-1②

(第3回/第4回テキスト)

必ず、生徒に授業日と自分の名前を記入
させるようご指導をお願いいたします。

だい かい じゅぎょう び
第3回授業日 2024年 月 日

だい かい じゅぎょう び
第4回授業日 2024年 月 日

な まえ
名前



ロボット博士養成講座
ロボティクスプロフェッサーコース

2024年5月授業分

ロボット博士養成講座

ロボティクスプロフェッサーコース

不思議アイテムⅢ-1②

第3回

赤外線が発信源を探そう

講師用

目 次

- 0. 赤外線が発信源を探そう
 - 0.0. 「赤外線が発信源を探そう」でやること
 - 0.1. 必要なもの

 - 1. 赤外線 LED と赤外線受光素子の性能
 - 1.0. 赤外線 LED と赤外線受光素子の性能を知る

 - 2. 赤外線が発信源を探す
 - 2.0. 赤外線センサー回路
 - 2.1. 赤外線センサーの動作確認
 - 2.2. 通信パターンを見てみよう
 - 2.3. 赤外線の飛距離と範囲の確認

 - 3. 赤外線センサーロボットの自動化
 - 3.0. 赤外線から逃げるロボットをイメージする
 - 3.1. 各手順のフローチャートとプログラム

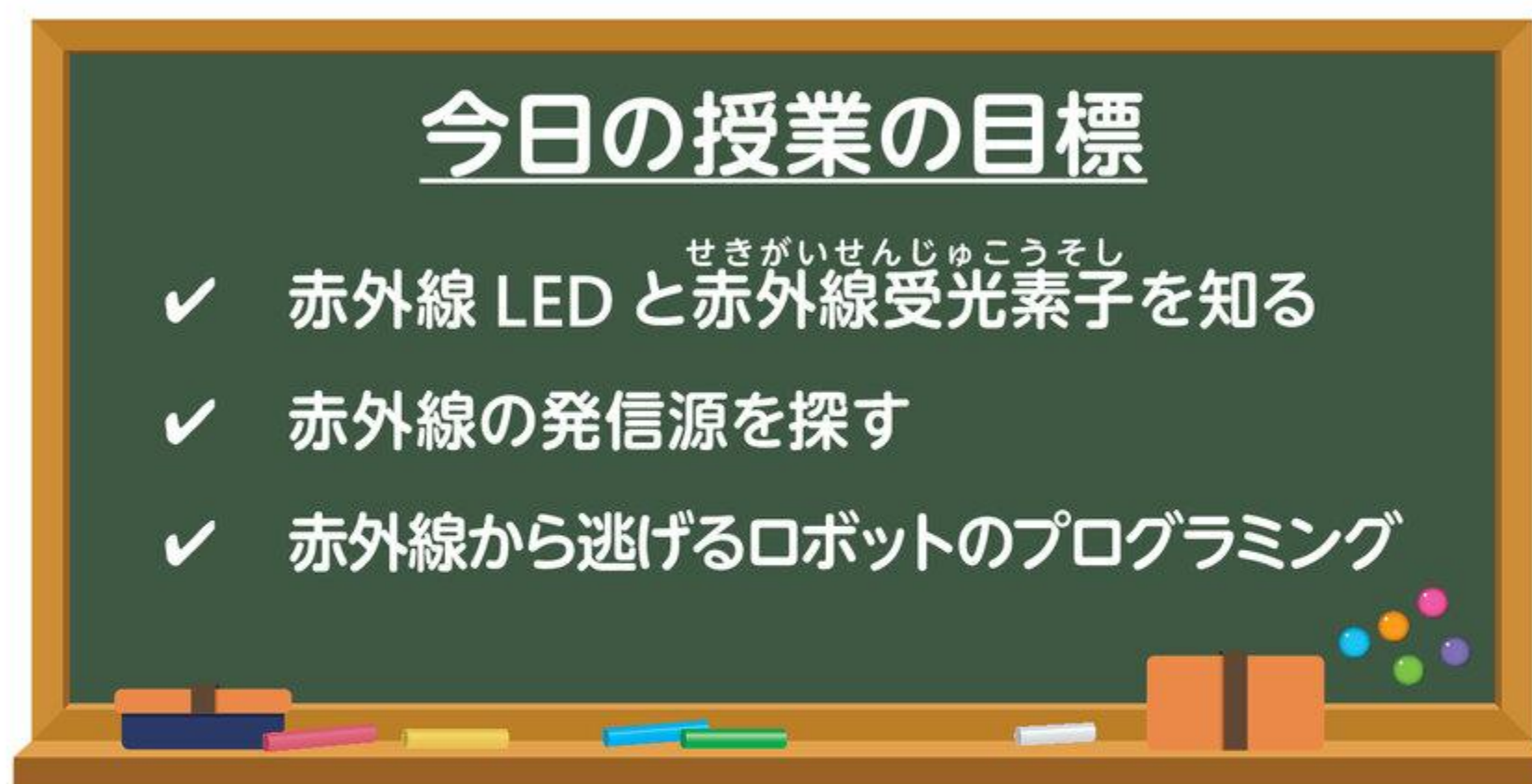
 - 4. まとめ
- 授業開始にあたって
 - 授業のはじめは、着席させ、大きな声であいさつしてから始めます。

 - 今回の目標をパネルで用意するか、黒板に予め書いておきます。
(授業の目標を明確化することは大変重要なことですので、生徒によく理解させます)

目安時間は授業時間 120 分のうち、休憩 10 分程度取ることを想定しています。
生徒の進捗状況により、休憩時間などを調整して授業を行ってください。

0. 赤外線が発信源を探そう（目安5分）

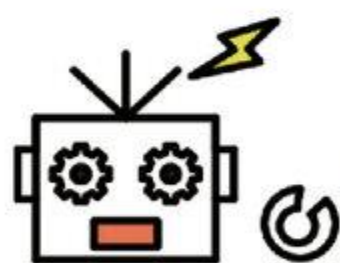
0.0. 「赤外線が発信源を探そう」でやること



前回までに赤外線の性質を理解し、赤外線を活用したコミュニケーション方法を体験しました。

今回は赤外線の強さや飛距離など、使用している赤外線LEDや赤外線受光素子の性質の理解を深めましょう。ロボットに使用している部品の性質を知ることが、ロボット自体の能力を発揮するうえでも大切です。部品の性質を知り、それに合わせたプログラミングをしていきましょう。

まずは、赤外線がだいたいどの方向から飛んできているのか特定し、それができたら赤外線を受けたら反応して赤外線が当たる範囲からロボットが逃げていくように動かします。前回やったロボットバトルにこの機能を加えたら、赤外線の砲撃を受けたら自動的に逃げるロボットにできるかもしれませんね。



部品の性質を知るといっても己を知ることと同じジャー。
知ればもっと自由に動けるようにナルゾ。

0.1. 必要なもの

前回つくったロボットと以下のパーツを準備しておきましょう。

USB ケーブル	1	緑色 LED	1	赤外線リモコン	1
					

図0-0 必要なもの

授業に入る前に赤外線センサーロボットの配線を確認しましょう。

モーターはそれぞれ以下のように接続し、スピーカーはロボプロシールドの **A4** 端子に接続しましょう。また、今回はコントローラーを使用しないのでリボンケーブルは外しておきましょう。

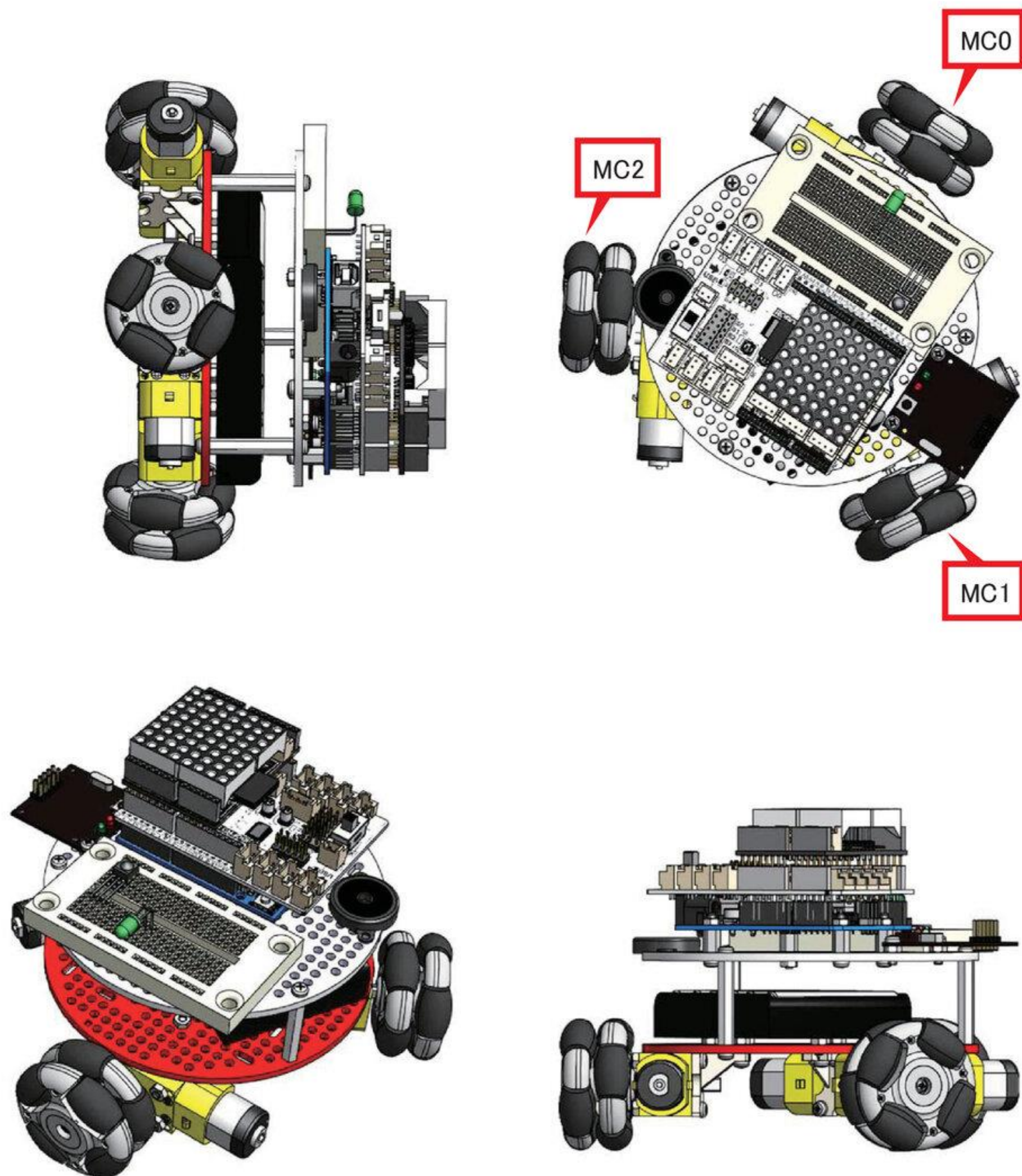


図0-1 赤外線センサーロボット

1. 赤外線 LED と赤外線受光素子の性能 (目安 10分)

1.0. 赤外線 LED と赤外線受光素子の性能を知る

前回は、相手から飛んできた赤外線が自分の赤外線受光素子に当たるとライフが減るというゲームをしました。前回までは、相手から飛んできた赤外線を赤外線受光素子が受光することは確認できましたが、「どの方向から飛んできているのか?」、「どのくらいの距離まで飛べるのか?」はあまりわからなかったと思います。

今回は、授業で使っている赤外線LEDと赤外線受光素子それぞれの性能をもう少し詳しく調べていきましょう。

1) 赤外線受光素子

赤外線受光素子はその名の通り、赤外線を受信するための電子部品です。

ロボプロでは赤外線通信で文字などをやりとりするとき、リモコンや赤外線LEDは赤外線受光素子に対し、1回の通信で68個のデータを送信しています。

赤外線受光素子がこれをしっかり受信することができれば、どのような文字・内容が送信されてきたか読み取れるわけです。

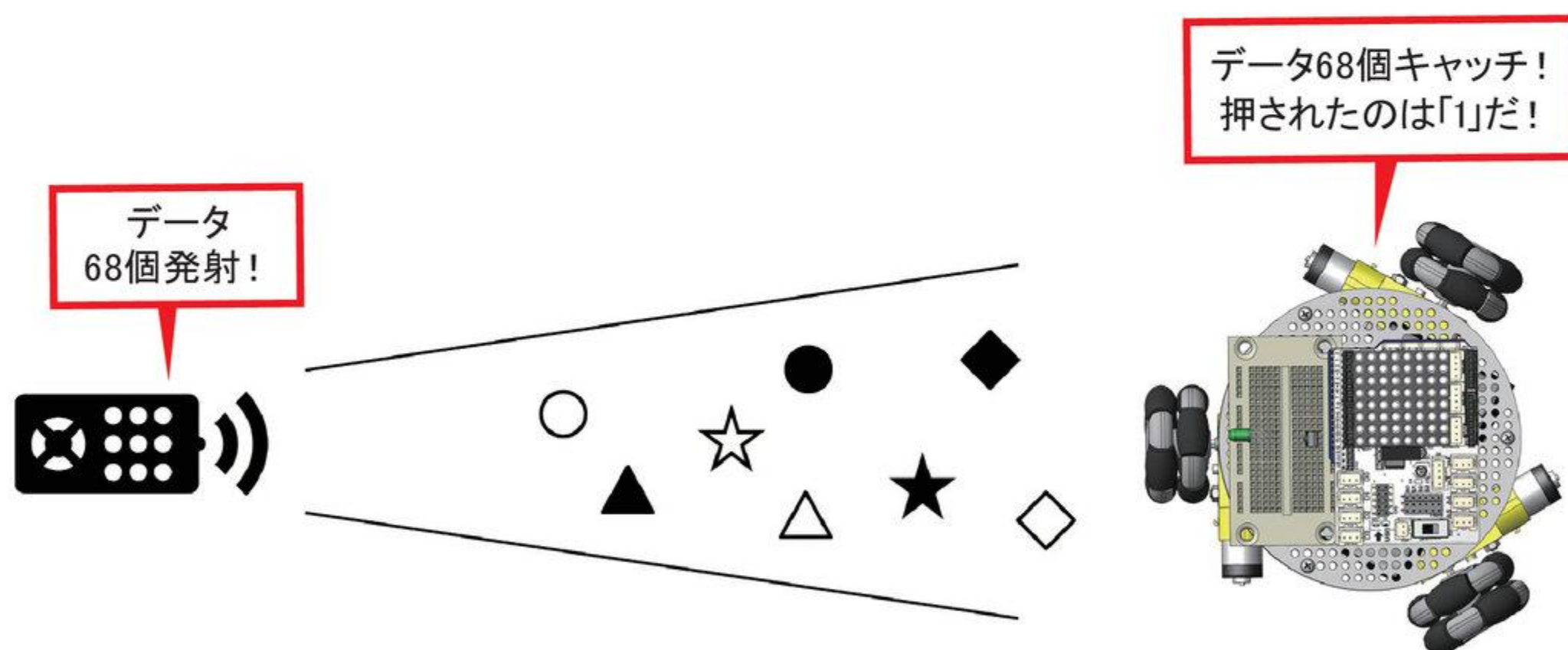


図1-0 赤外線受光素子が信号受信に成功

以前にも説明したとおり、赤外線通信は「指向性が高い」、つまりねらった方向にしか赤外線が飛んでいかないという性質があります。

家庭でも、テレビやエアコンなどのリモコンはしっかり機械に向けていないと操作できないのではないのでしょうか。

ロボプロで使用している赤外線LEDの「仕様書 (データシート)」を見てみると、大まかに正面から左右10度ずつ程度にしか信号が届かないことがわかります。

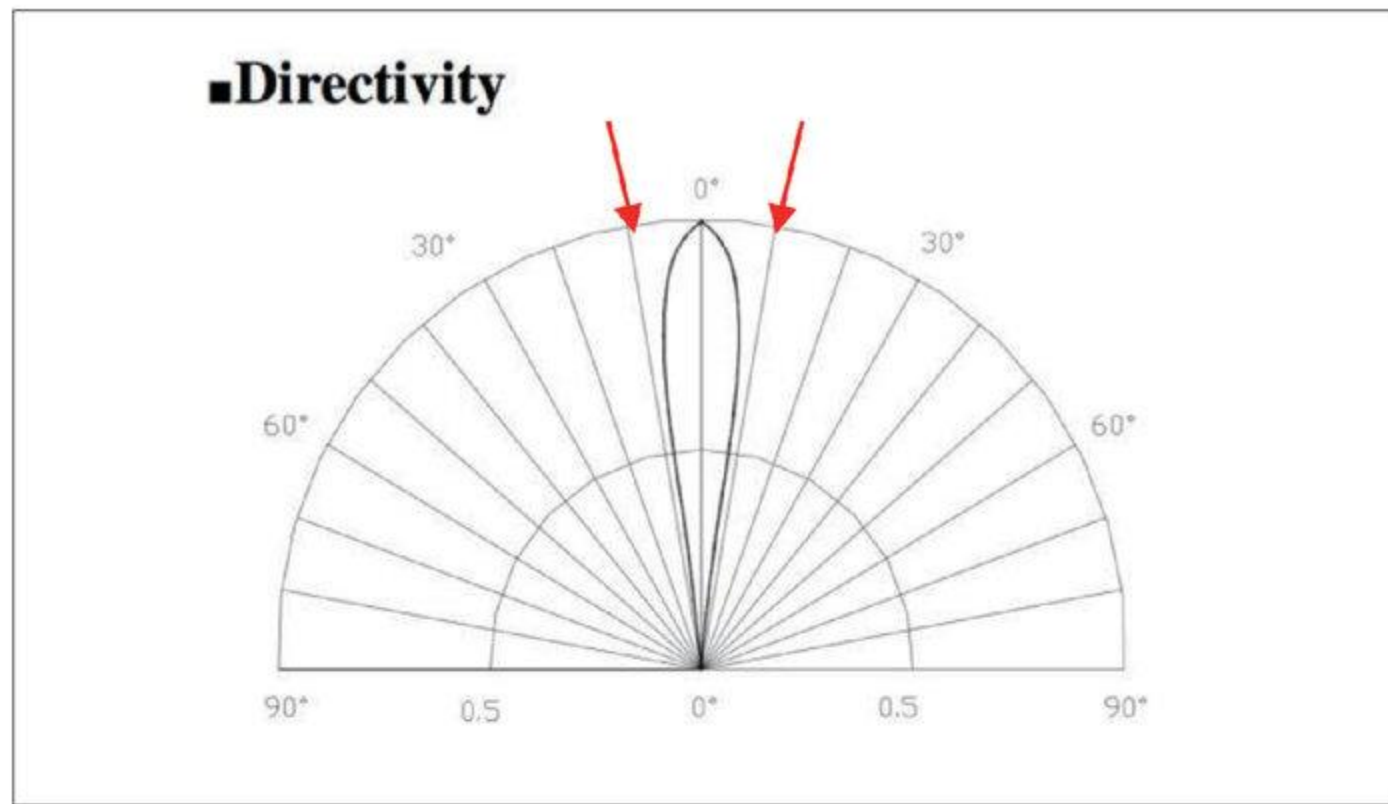


図1-1 赤外線LEDの信号範囲

範囲から外れてしまうと、信号が一部しか届かず解析ができないことがあります。リモコンであれば「何らかのボタンが押されていることはわかるが、どのボタンが押されているかまではわからない」という状態になってしまうわけですね。

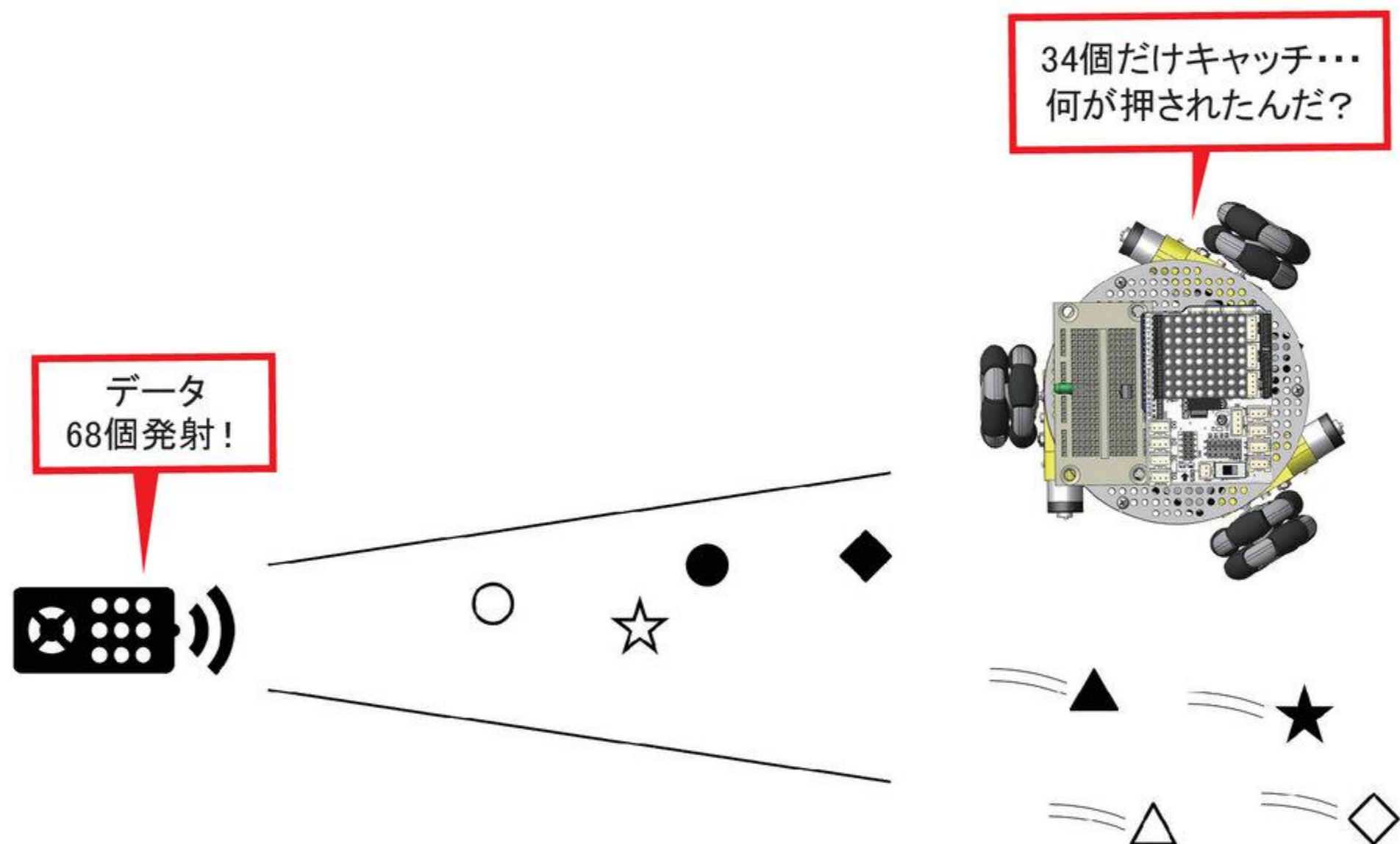


図1-2 赤外線受光素子が信号受信に失敗

前回の授業の最後に登場したプログラム[IRtankFire]では、正面方向から砲撃を受けたときだけ「被弾」として、体力を減少させていました。実は「どれくらい角度がずれたら砲撃が効かなくなるか」も、このデータ数をもとに決めています。

2) プログラム

□プログラム「IRtankFire」より抜粋

```
int hit_rate = 60;

(中略)

if (irrecv.decode(&results)){
    if(results.rawlen >= hit_rate){

(中略)

    }
}
```

命令「results.rawlen」

実行結果：データをいくつ受信したかチェックする

使い方：if(results.rawlen >= 60){ //もし68個中60個以上のデータを受

信していたら

results.rawlen を使うと、発射された68個のデータのうち、いくつ受信できたかを検知します。つまり、**図1-0**の状況であれば results.rawlen の値が68に、**図1-2**の状況であれば results.rawlen の値が34になります。

「IRtankFire」では hit_rate の初期値が60なので、68個中60個以上のデータを受信できたときだけ「被弾」としています。

もし hit_rate の数字を小さくしたら、データがあまり受信できていないときでも「被弾」扱いになりますね。つまり、正面から大きくずれた敵からも砲撃を受けてしまうことになります。逆に hit_rate を68とすれば、68個すべてのデータを受信できた時でない「被弾」にならないので、かなり狙いすまして砲撃する必要があります。

ちなみに、その上の if (irrecv.decode(&results)) という条件式は、赤外線を受信したときをさします。つまり、まず赤外線を受信したかどうかを判定してから、results.rawlen の値をチェックしているのです。

2. 赤外線発信源を探す (目安 50 分)

2.0. 赤外線センサー回路

電子回路も前回の「赤外線センサーロボット」から変更はありません。赤外線LEDの抵抗は100Ωを使用します。赤外線LEDは9番ピン、赤外線受光素子はアナログの5番ピンに接続しています (シーカーもビーコンも同じです)。

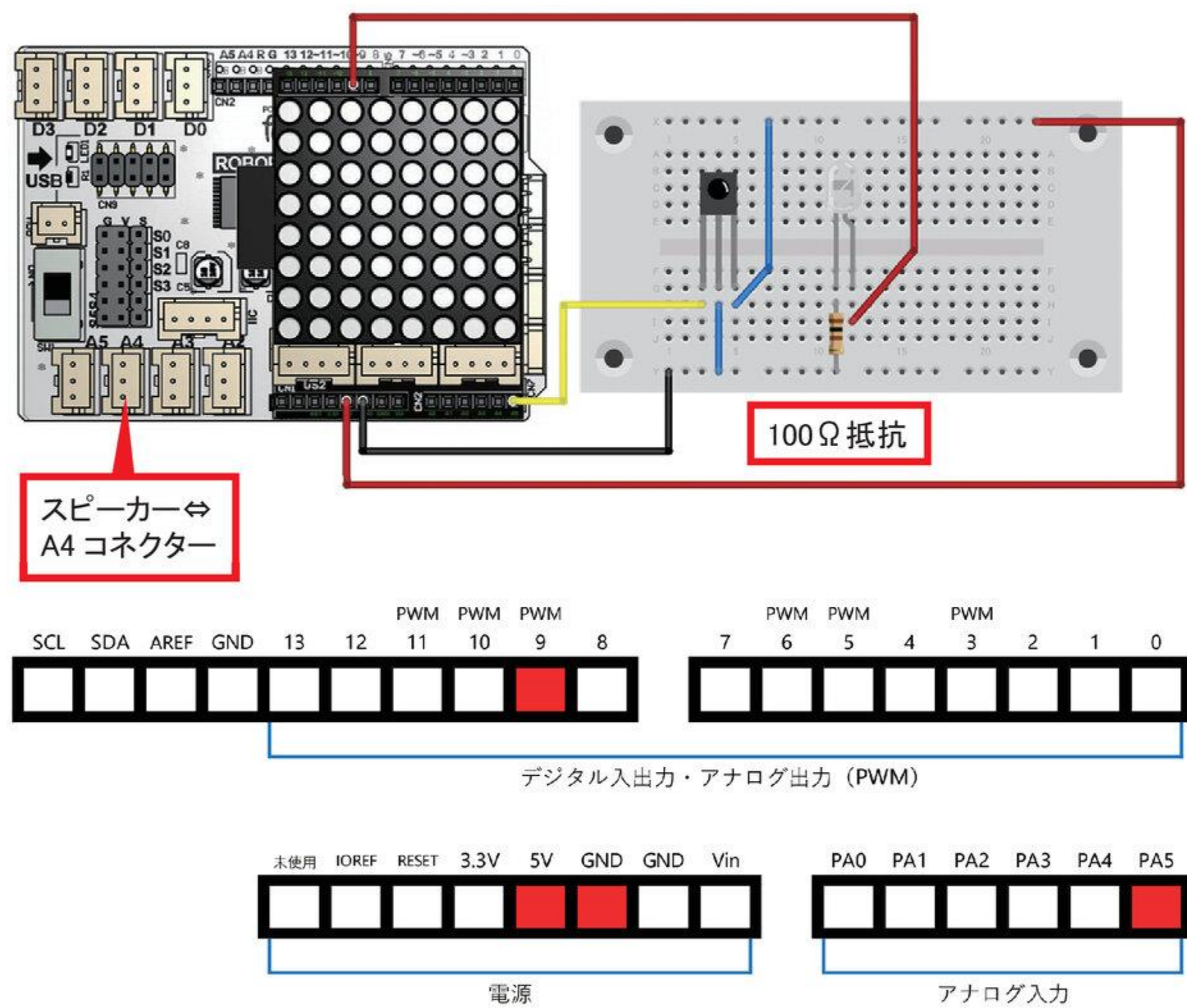


図2-0 赤外線センサーの回路およびマイコンボードの入出力詳細図

2.1. 赤外線センサーの動作確認

1) せきがいせんじゅこうそし 赤外線受光素子の確認

念のためにせきがいせんじゅこうそし赤外線受光素子の動作確認をしておきましょう。赤外線リモコンを準備してください。

以下のプログラムを実行して、せきがいせんじゅこうそし赤外線受光素子に向けて赤外線リモコンの0～9の数字ボタンを押すと押した数字がマトリクスLEDに表示されます。せきがいせんじゅこうそし赤外線受光素子の真正面にリモコンを向けてください。

∞プログラムの書き込み

RoboticsProfessorCourse3 > infraRed3 > IRJackRCV

2) 赤外線通信テスト

今度は、赤外線送信を確認しましょう。ここでは、ペアのひとりが送信プログラム「IRsendTest」を実行し、もうひとりは受信プログラム「IRJackRCV」を実行して通信をします。送信側になる人は、以下のプログラムを実行してください（送信、受信を交互に行ってください）。ビーコン・シーカーとも同じ配線です。

∞プログラムの書き込み

RoboticsProfessorCourse3 > infraRed3 > IRsendTest



図2-1 赤外線通信テスト

実行結果：ビーコンは赤外線リモコンのボタンを押したときと同じ信号を0～9の順番に0.5秒間隔で送信する。かんかくシーカーのマトリクスLEDにはビーコンから送信された数字が表示される。

うまくいかない場合は、以下の点を確認してみましょう。

POINT

- ① 配線が間違っていないか 図2-0を見て点検しましょう。
- ② LEDの極性が逆になっていないか 図2-0を見て点検しましょう。
- ③ ささっている抵抗が100Ω（茶・黒・茶・金）か点検しましょう。
- ④ ジャンパー線が断線しているものは処分して、交換しましょう。

結果はどうでしたか？ ビーコンとシーカーのマトリクスLEDで同じ数字は表示されましたか？

もし、赤外線LED、赤外線受光素子には問題がないのに、うまくシンクロしない場合は、別のペアとの距離が近すぎて、違う場所からの信号を受け取っている可能性があります。そのときは、距離を離して行ないましょう。

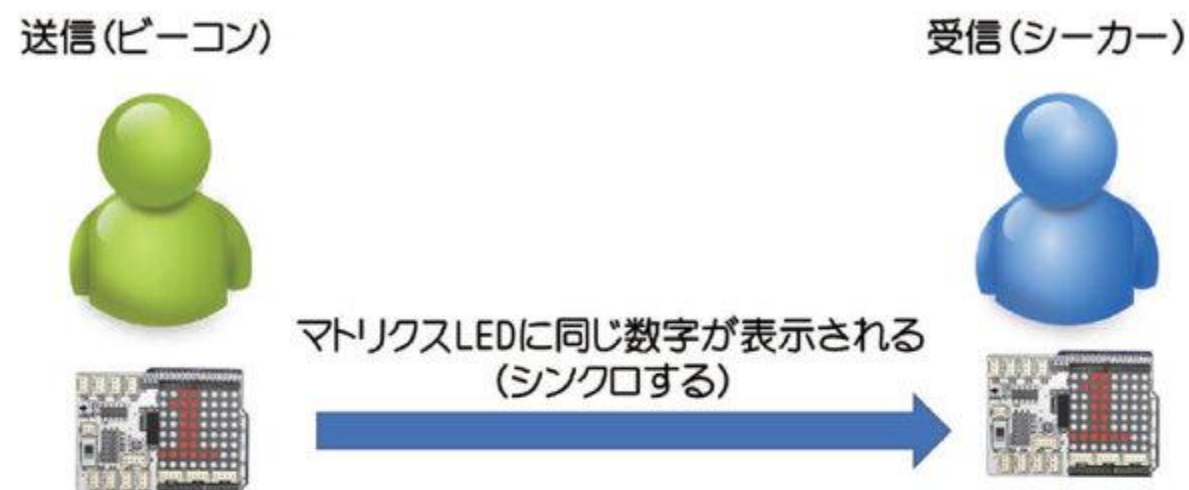


図2-2 赤外線通信テスト結果

うまく送受信ができたなら、役割を交代してテストを行ってください。

今回の授業では、こんな感じで、赤外線の信号を出す「ビーコン」の役割と、赤外線を検知するシーカーの役割を交互に行ってください。

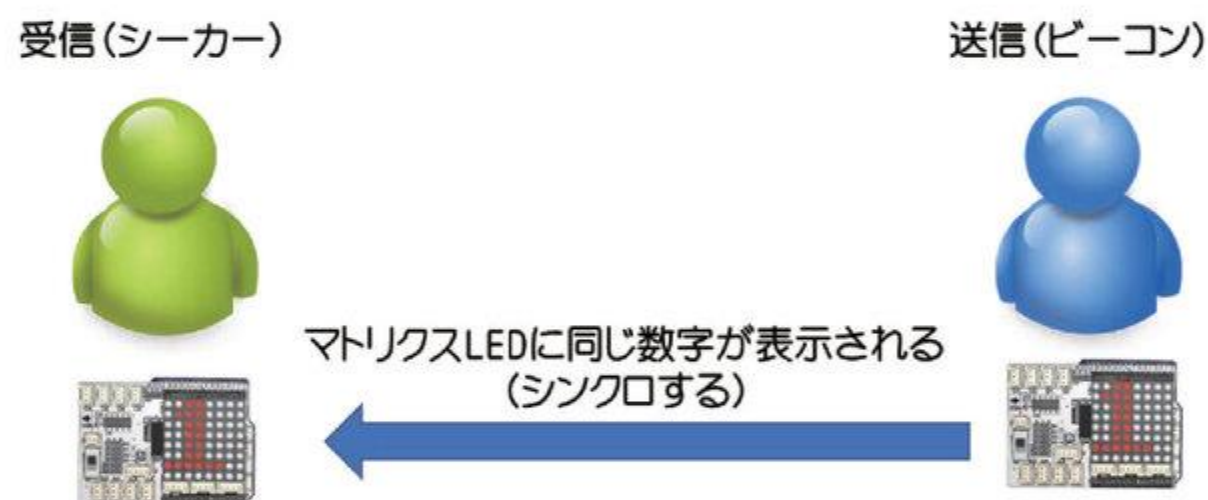


図2-3 役割を交代



豆知識

ビーコンとシーカーとは

「beacon (ビーコン)」とは、標識、灯台、狼煙、かがり火、無線標識、航空標識などの意味をもつ英単語です。また、光や電波などを発する装置はビーコンといわれ、その光を見た人や信号を受信した電子機器などが現在地を知るのに使われます。

「seeker (シーカー)」とは、自動検知システムを意味することがありますが、今回の実験では、赤外線を探す役割としています。

2.2. 通信パターンを見てみよう

さて、前回までの授業でわかったように、赤外線を目で見えることはできません。信号の状況を観察したくても、赤外線を見ることができ^{きざい}るカメラなどの機材で確認するほかに、良い方法はないものでしょうか。

やってみよう！

赤外線信号のパターンを可視化^{かしか}してみよう。LEDを赤外線LEDから可視光線を出す緑色LEDに差しかえて信号のパターンを見てみよう。アノードとカソードの向きには気をつけて差しかえてね。

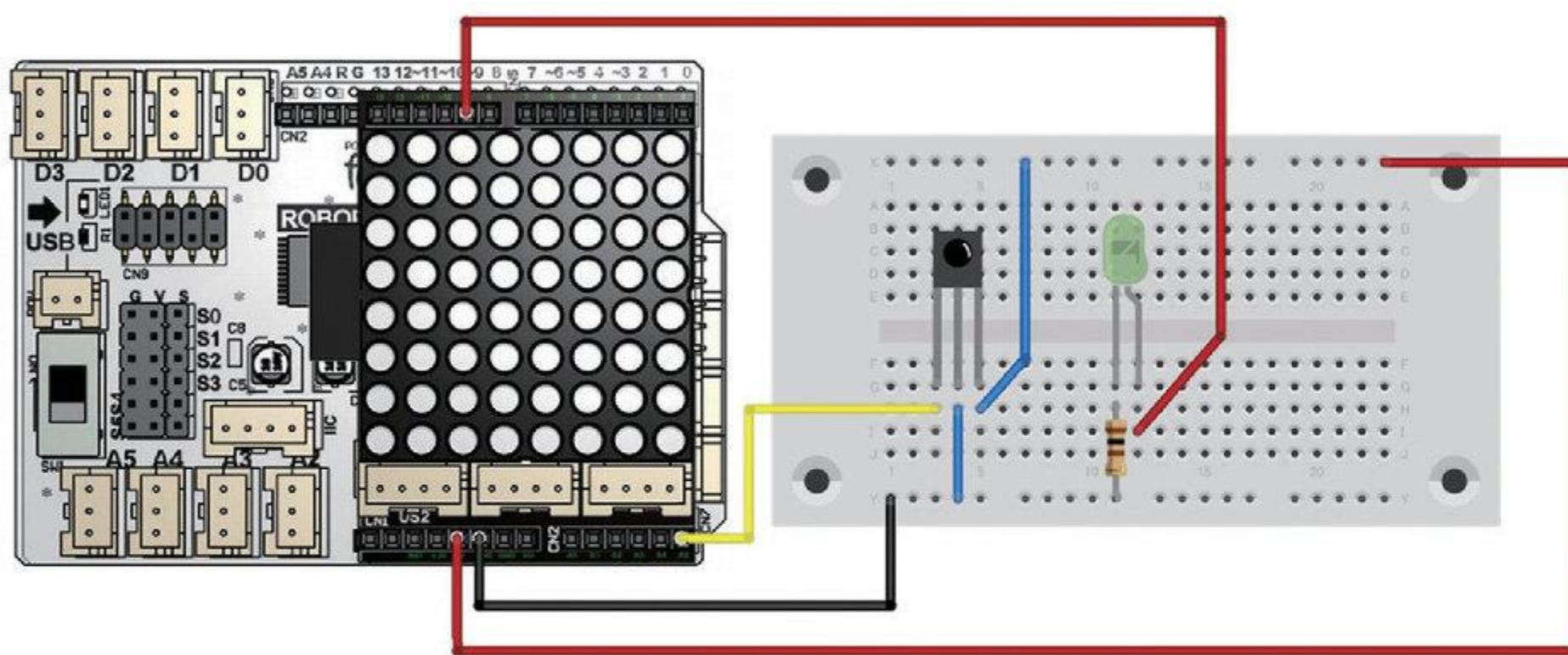


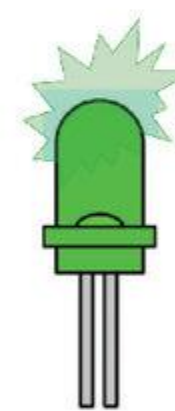
図2-4 緑色LEDに交換

実行結果：赤外線LEDに合わせ、緑色LEDも明滅する。

一文字のデータを送るだけで、ものすごい回数の点滅をしていることがよくわかりますね。

目で見ただけでは非常にわかりにくいですが、送信しているデータの内容によって点滅のしかたも変化しています。もし、「この点滅ならこのデータだな」と見ただけで判別できる人がいれば、緑色LEDでも通信を行うことができます。

実行を終えたら、再び赤外線LEDに差しかえておきましょう。



2.3. 赤外線ひきよりの飛距離はんいと範囲の確認

やってみよう！

赤外線LEDの飛距離ひきよりと範囲はんいを実験で確認しよう。
 さきほどと同じようにビーコンとシーカーの役割を交互こうごに行ってね。
 電源には電池ボックスを使って行おう。

1) 飛距離ひきよりを測定そくていする

以下の図2-5のように真正面ましようめんを向けた状態で、どのくらいの（何cmくらいの）距離きょりまでしっかりと信号が届くでしょうか？以下のプログラムを実行して、実験しましょう。

プログラムの書き込み

【ビーコン】 RoboticsProfessorCourse3 > infraRed3 > IRsendTest

【シーカー】 RoboticsProfessorCourse3 > infraRed3 > IRJackRCV

実行結果：通信が途切れはじめると「×」が表示される。

飛距離ひきよりはどのくらいになるか、測定して記録しておきましょう。

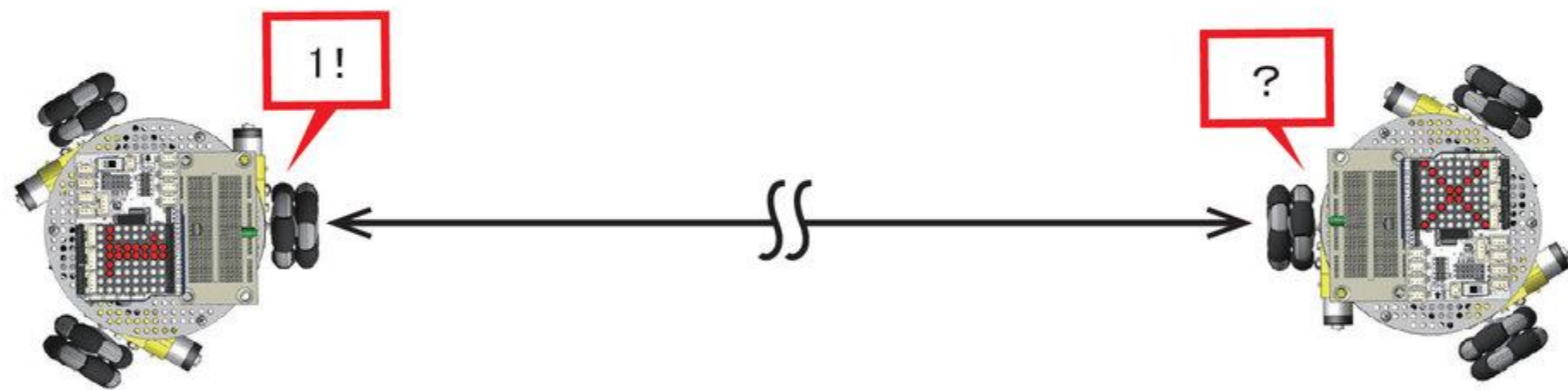


図2-5 赤外線LEDの飛距離ひきより

測定距離そくていきょり  約300 cm

講

測定距離は環境によって誤差が出ます。
 上記は参考数値です。

2) ^{そうじゅしんはんい}送受信範囲 (横方向にずらす) ^{そくてい}を測定する ①

次に、横にずらしてみ、^{はんい}範囲を測定してみましょう。
 以下の図のように 50cm くらい離して、^{ましようめん}真正面の位置から ^{じよじよ}徐々に横にずらしてみましょう。
 だいたい何cmくらいのずれまで送受信ができるか、記録しておきましょう。

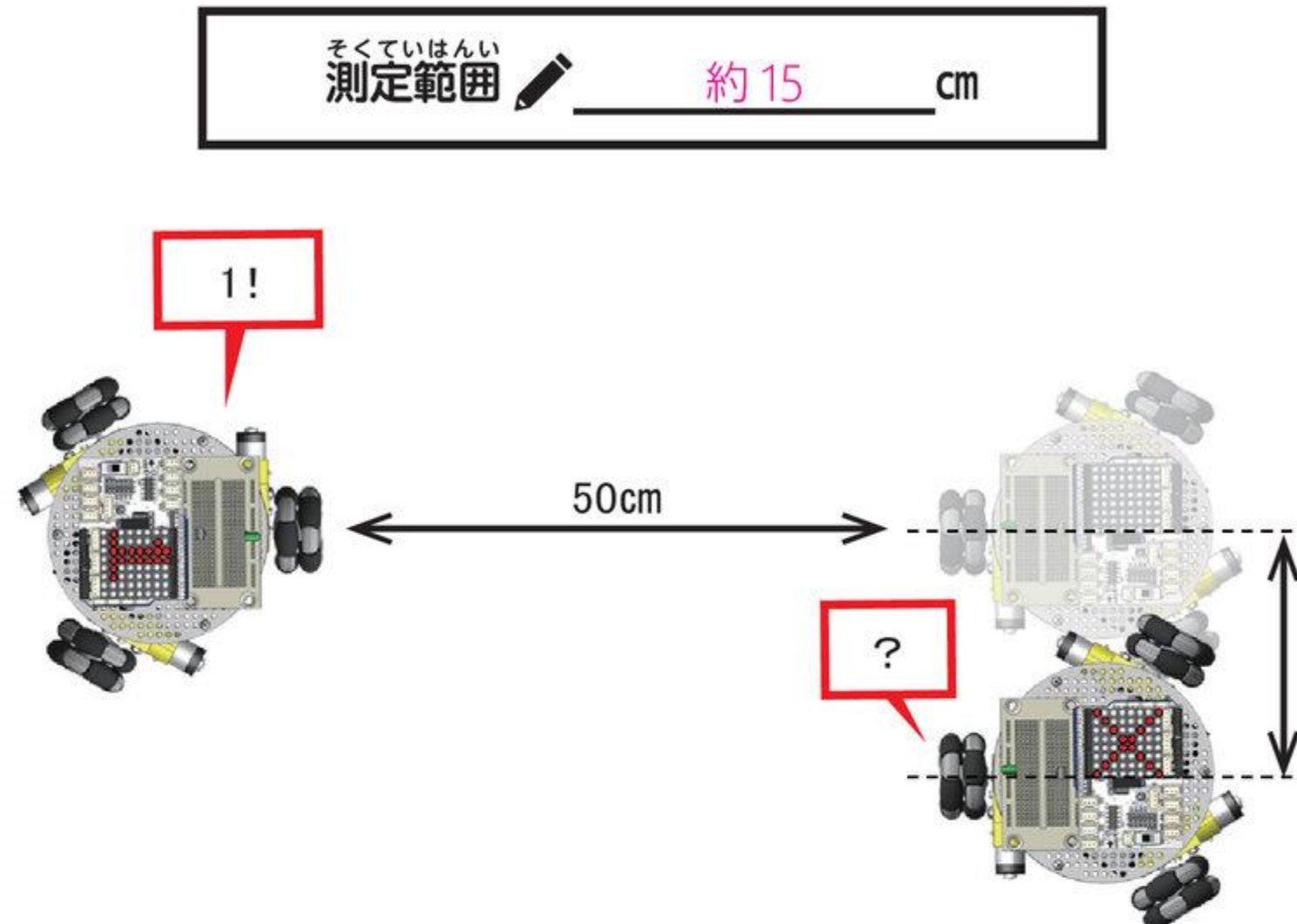
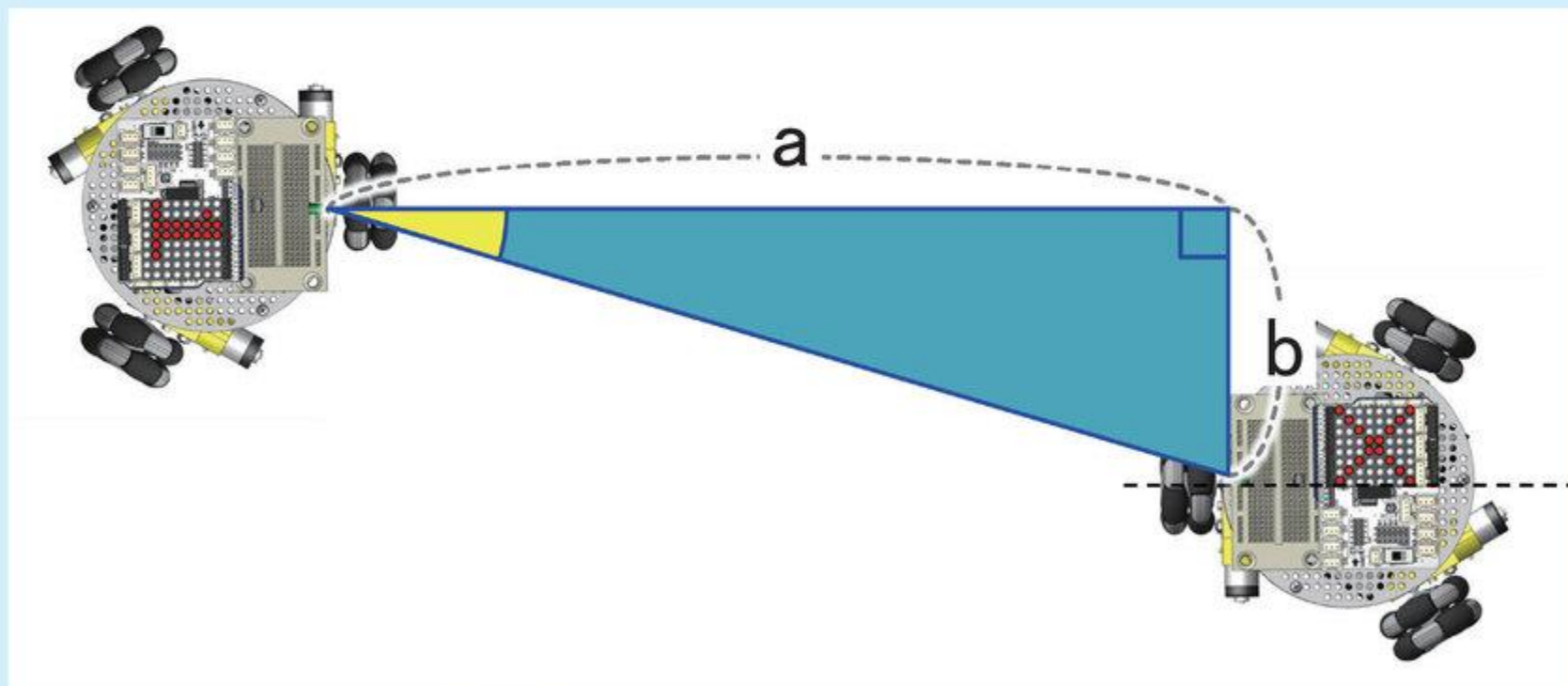


図2-6 ^{そうじゅしんかのうはんい}送受信可能範囲の測定

コラム 検知可能角度を求める

上の方法だと、「何cmずれたら検知できなくなるか」はわかりますが、「何度ずれたら検知できなくなるか」がわかりません。ただ数学には「直角三角形の2辺の長さのバランスから角度を求められる」という考え方が存在します。



上の図のa、bにあたる部分の距離をはかり、 $b \div a$ を計算しましょう。この値から、正面からのずれの角度 (図の黄色の角) を求めることができます。

今回は下の表を見るだけで大まかに角度がわかるようにしておきました。ちなみに、高校数学では「正接 (タンジェント)」という名で学ぶ計算方法です。名前だけでも覚えておくと役立つかもしれませんね。

b÷aの値	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
角度[度]	5.7度	11.3度	16.7度	21.8度	26.6度	31.0度	35.0度	38.7度	42.0度	45.0度

3) 送受信範囲 (角度) を測定する

次に、ビーコンを中心において、その周囲を、シーカーをビーコンに向けて移動させながら、どのくらいの角度まで信号が届いているのか、だいたいの角度を測定しましょう。ビーコンとシーカーの前方が向き合った角度を0度として考えます。受信できたポイントを図に書き込んでもいいです（受信がうまくいっていない場合マトリクスLEDに「×」が表示されます）。

実験結果記入  : ビーコンの前方で約±45度くらいの角度。狭い場所で乱反射すると同じ結果にはなりません。広い空間で行ってください。

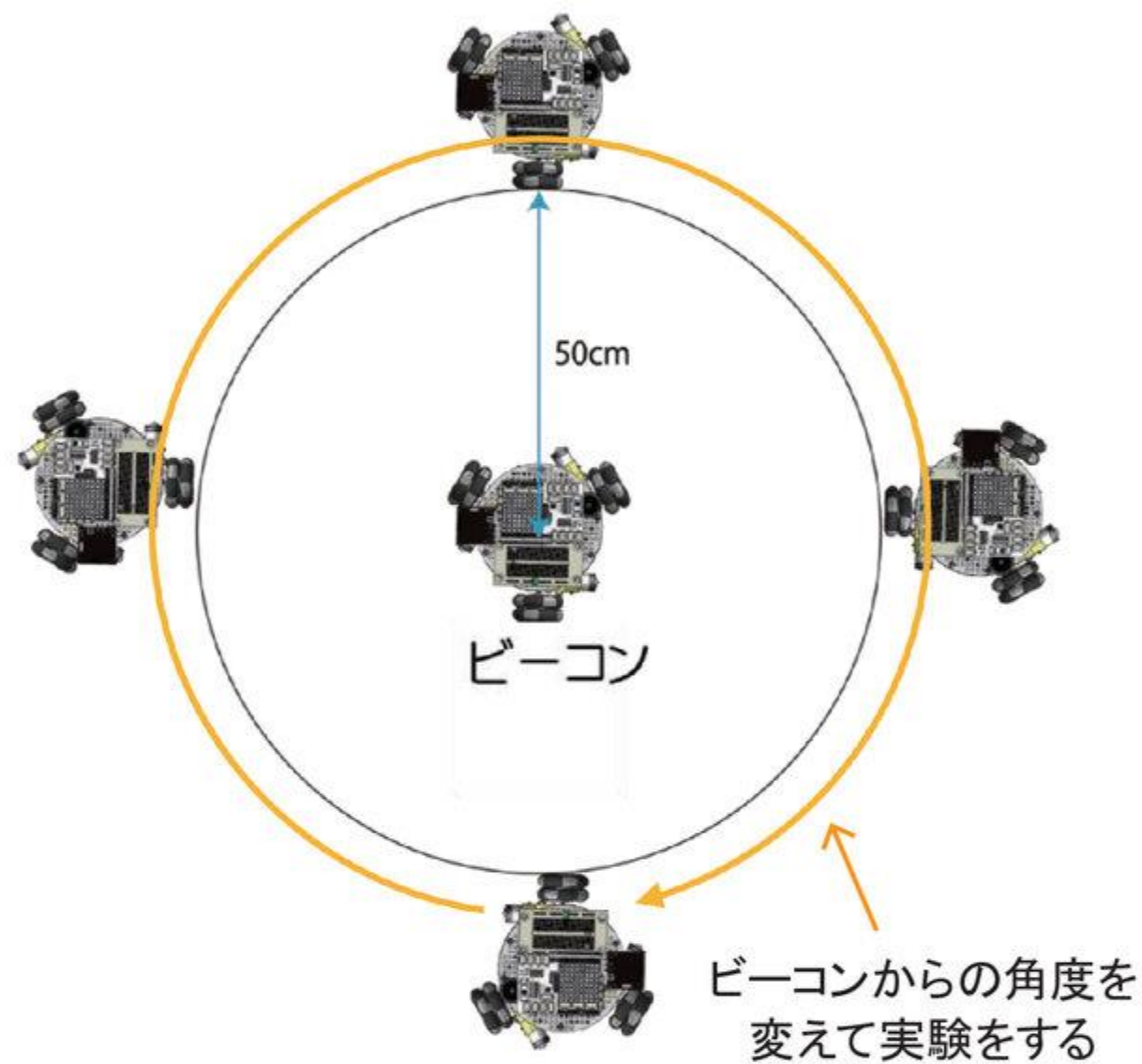


図2-7 角度の測定

3. 赤外線センサーロボットの自動化（目安 40 分）

3.0. 赤外線から逃げるロボットをイメージする

赤外線いの性質がわかってきたところで、その性質を活かせるロボットを考えていきます。ここでは、赤外線が当たったら、自動的に動くロボットにしましょう。まずは、動作のイメージとロボットへの命令の手順を確認しましょう。

1) 動作イメージ

まずは、動作イメージを確認しましょう。図のように最初はビーコンとシーカーを向かい合わせにします。ビーコン、シーカーともにプログラムを書き込むと、赤外線が当たる範囲はんいから逃げるようにシーカーが横に動きます。

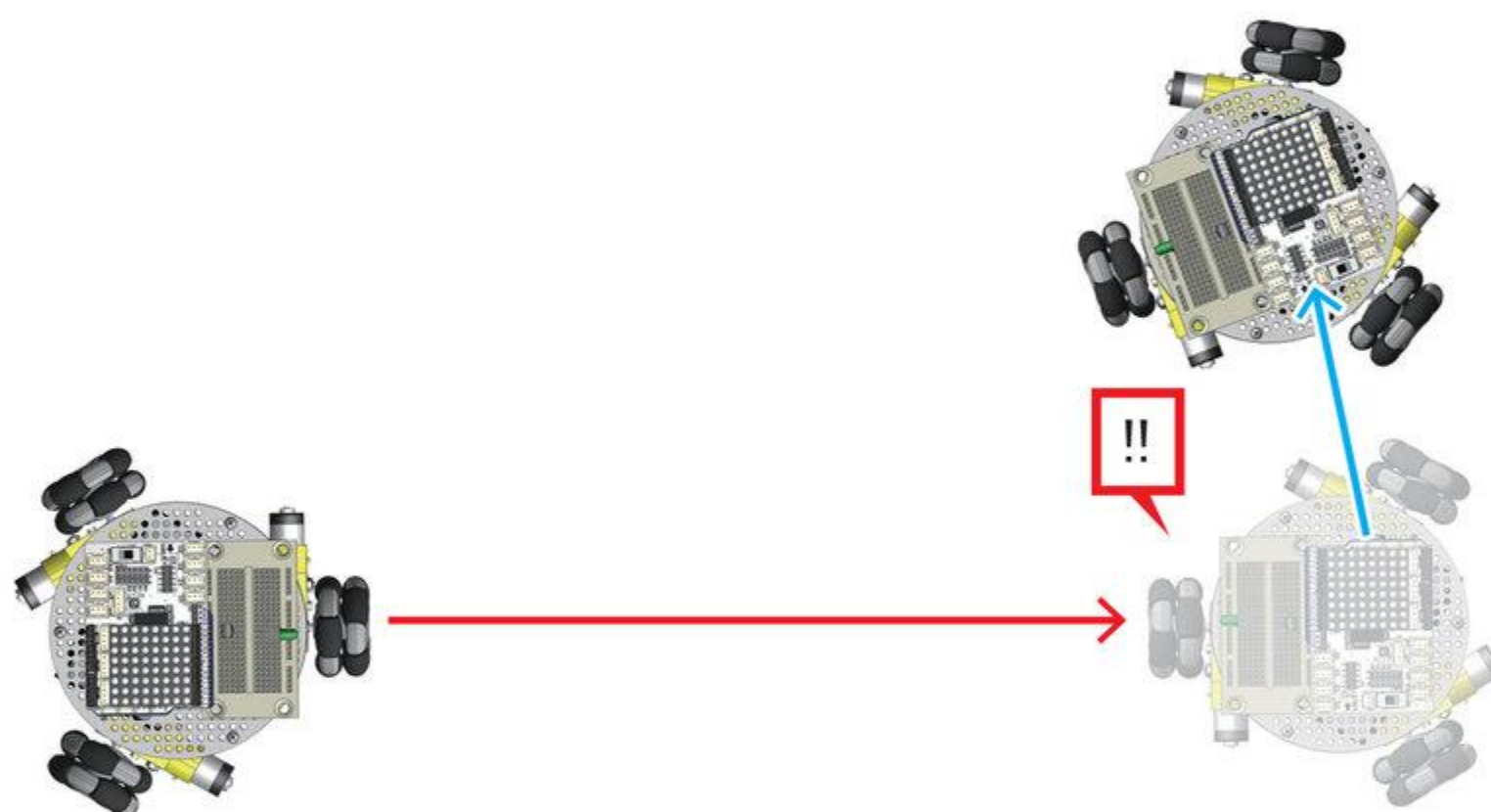


図 3-0 動作イメージ

2) シーカーの処理手順しよりてじゆん

シーカーの動作をつくる処理手順しよりてじゆんは大きく分けて3つになります。この手順を参考に次のページでフローチャートを完成させましょう。



POINT

- 手順 1) 赤外線受信：シーカーがビーコンからの赤外線信号を受信
- 手順 2) 赤外線判定：シーカーが、赤外線の強さを判定
- 手順 3) 回避運動かいひうんどう：シーカーが右へ旋回せんかいする

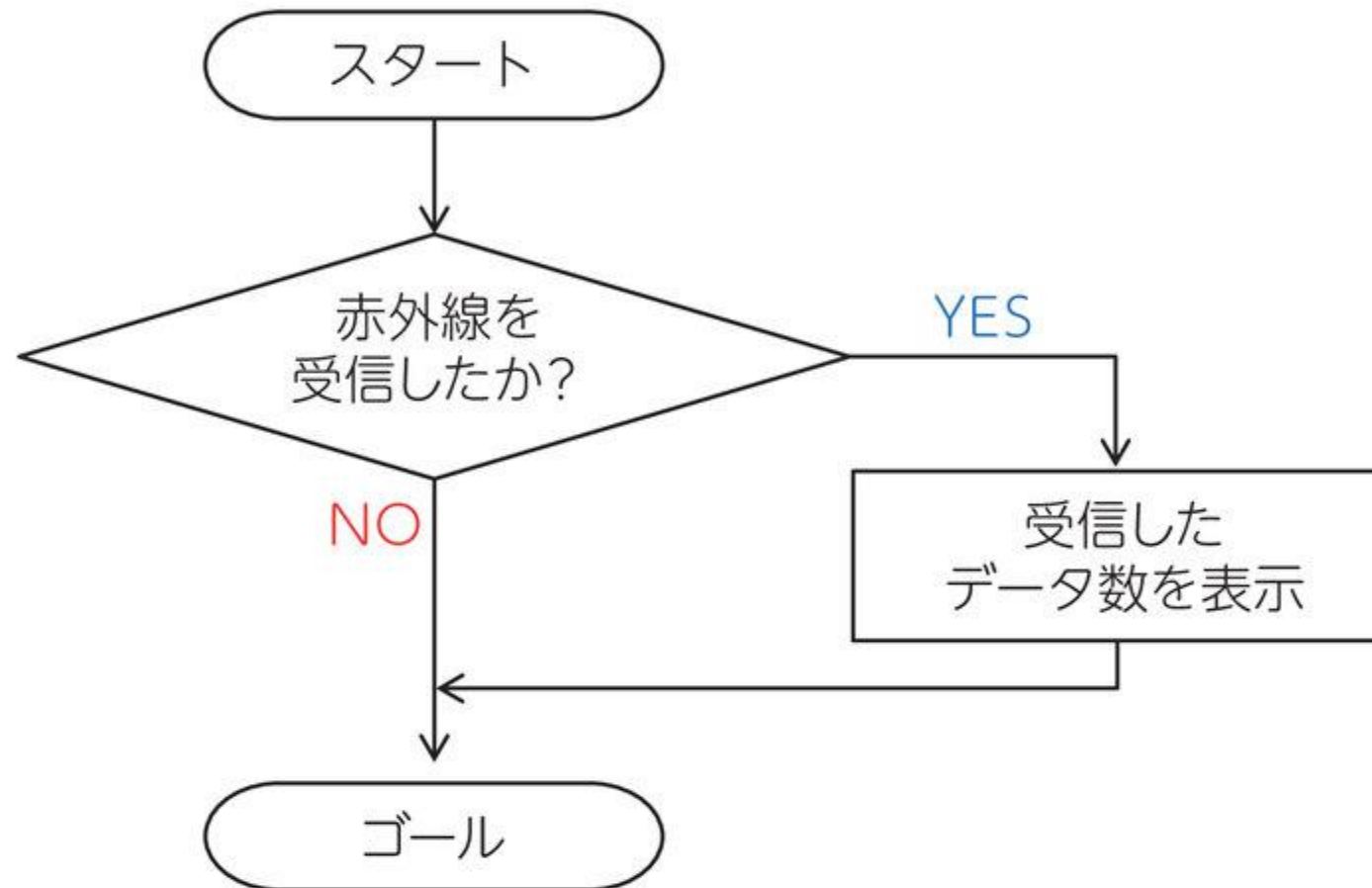
3.1. 各手順のフローチャートとプログラム

1) 赤外線受信：シーカーがビーコンからの赤外線信号を受信

赤外線から逃げるためには、赤外線を受信したことを検知するプログラムが必要です。前回のプログラム「IRtankFire」や今回の説明を参考に、まずはベースとして赤外線検知の部分を書いてみましょう。

やってみよう！

以下のフローチャート通りになるようプログラムを書いてみよう！



💡 ヒント

今回は「RPLib.h」のほかに、マトリクスLEDへ命令を出すのに「Sprite.h」と「Matrix.h」、赤外線通信を行うために「IRremote.h」と、計4つのライブラリが必要だね！

`void loop` 以外の部分にどのような設定が必要かは、これまでのプログラムを参考にしてみよう！

「赤外線を受信したか」を判定する条件式は `irrecv.decode(&results)`、受信したデータ数の検知は `results.rawlen` でした。よって、次のようなプログラムになった人が多いのではないのでしょうか。

```

void loop(){
  if(irrecv.decode(&results)){
    myMatrix.putd2(0, 0, results.rawlen);
  }
  delay(500);
}
  
```

このプログラムをシーカーに書き込んで、ビーコンやリモコンから赤外線を発信してみましょう。

シーカーに数字が表示されるものの、そのあと数字がまったく変化しない、という問題が発生します。

前のページの問題は、せきがいせんじゅこうそし赤外線受光素子^{せきがいせんじゅこうそし}がいったん赤外線を受信すると、`results.rawlen`がずっとその時の値のまま固定されてしまうために起こります。

よって、マトリクスLEDに`results.rawlen`の値を表示し終わったら、いったんこの値をリセットし検知を再開させる必要があるのです。

以下の一文を追加してみましょう。

```
void loop(){
  if(irrecv.decode(&results)){
    irrecv.resume();
    myMatrix.putd2(0, 0, results.rawlen);
  }
  delay(500);
}
```

`resume();` という関数を実行することで、赤外線を受信を再開させることができます。

講

上記のプログラムは以下のものです。

RoboticsProfessorCourse3 > infraRed3 > IRavoider1

以下のようなフローチャートが解答例ですね！

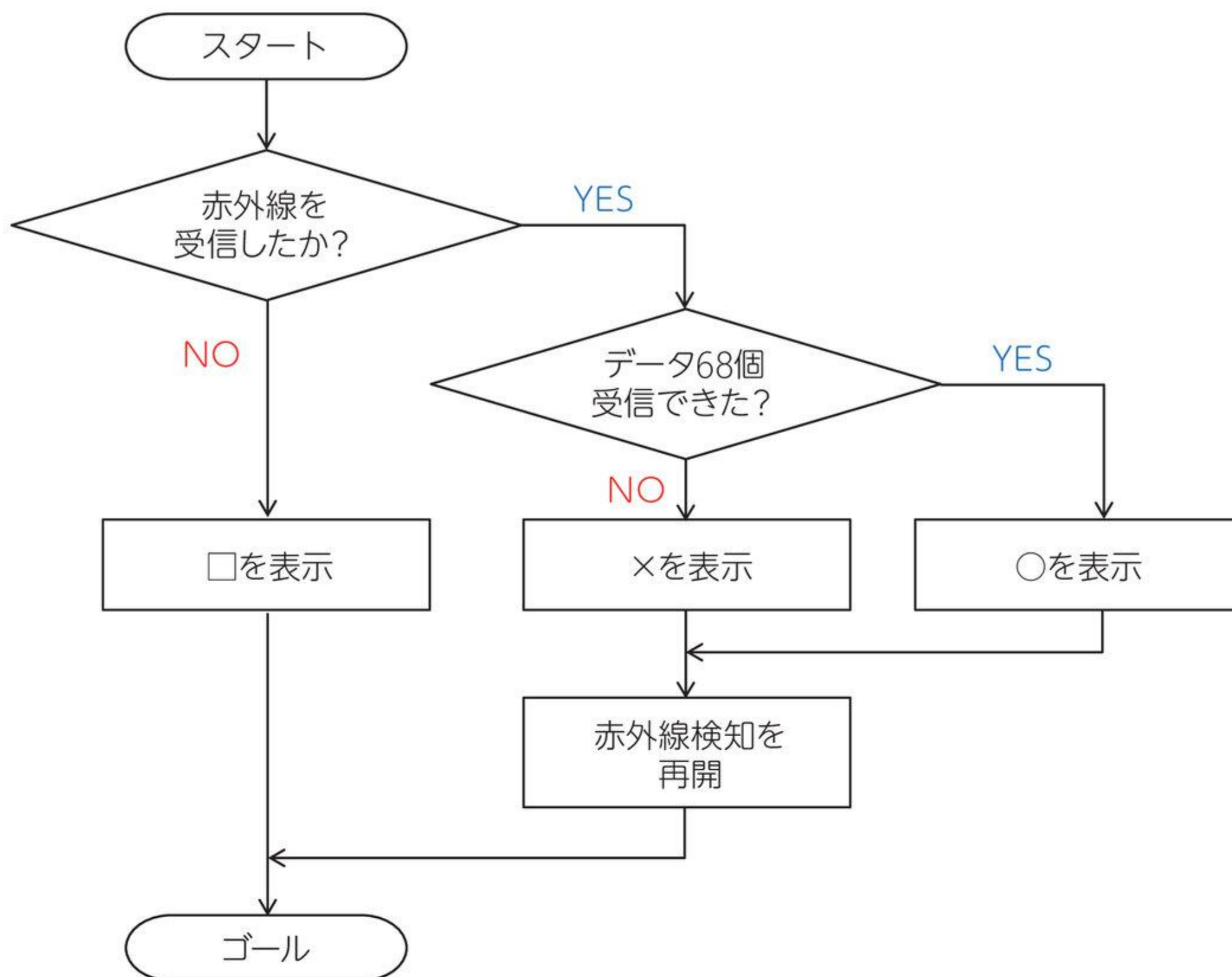


図3-1 赤外線判定をするためのフローチャート

今回も、赤外線を検知したあとは検知再開命令 `resume();` が必要です。

3) 回避運動：シーカーが右へ^{せんかい}旋回する

さて、いよいよ今日の最終ミッションです。とはいえ、今回つくってきたプログラムをベースにしていけば、さほど難しいプログラムではないはずですよ。

ステップアップ

プログラムを書きかえ、「赤外線から逃げるロボット」をつくろう！

モーターを回転させ、ロボット自身を移動させる動作が追加されます。
オムニホイールロボットを移動させる方法は、

- ① `mc0.rotate(100);`などを組み合わせ3つのモーターを回転させる。
- ② `omniBot.move(100, 0, 0);`などで前後左右回転の速度を指定する。
のいずれかになります。

講

- ①は `RPmotor mc0(MC0);`などのモーターの名付け処理（インスタンス生成）が、
- ②は `#include`でヘッダーファイル「`RPomniDirect.h`」を取り込む処理が、それぞれ追加が必要です。

解答例は以下のプログラムです（②の方法を取り入れています）

RoboticsProfessorCourse3 > infraRed3 > IRavoider3

チャレンジ課題

赤外線から逃げるロボットがつくれたら、さらにプログラムを書きかえてさまざまな動きをするロボットにしてみよう！

たとえば、赤外線を検知できるぎりぎりの距離^{きょり}を保ち続けるロボット、リモコンの押された数字に応じて進む向きが変化するロボット…。

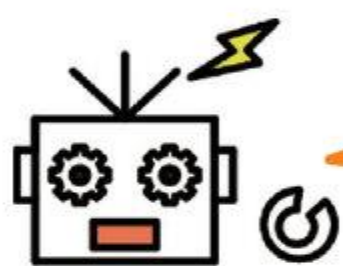
「IRtankFire」にこのプログラムを「移植」すれば、敵の砲撃をよけつつすかさず反撃する、なんて動作もつくれるかもしれないね！

4. まとめ（目安5分）

今回は、赤外線LEDと赤外線受光素子の性質を確認して、ロボットに組み込みました。また、赤外線LEDの指向性を利用して、「赤外線を回避する」ロボットをつくり実験をしました。

赤外線は目に見えない電磁波なので、こうした実験で距離や範囲を理解することで、その機能をうまくロボットに取り入れることができます。

次回は、「赤外線の発信源を探す」アルゴリズム（手順）について詳しく解説します。第6回で行う、サッカーロボット入門編まで、少しずつプログラムを学習していきましょう！



ではまた次回、新たなるロボットの力を学んでいきましょう！

《次回必要なもの》

次回は今回使ったロボットと以下のパーツを持ってきてください。

USB ケーブル	1	緑色 LED	1	姿勢検出シールド	1
					

図 4-0 次回必要なもの

講

- 今回の授業の目標を再確認します。
 - ・赤外線LEDと赤外線受光素子を知る
 - ・赤外線の発信源を探す
 - ・赤外線から逃げるロボットのプログラミング
- 今回の授業で学んだ感想や面白かったことなどを、生徒から聞いてみましょう。
- 次回のテーマは「赤外線を追従する1」であることを告知します。