

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
センサーロボット①

第2回

フローチャートとプログラムをかこう

講師用

目 次

0. フローチャートとプログラムをかこう

0.0. 「フローチャートとプログラムをかこう」でやること

0.1. 必要なもの

1. フローチャート

1.0. フローチャートとは

1.1. フローチャートをかく

2. いろいろなセンサーロボット

2.0. つかまらないロボット

2.1. 色探しロボット

3. 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。

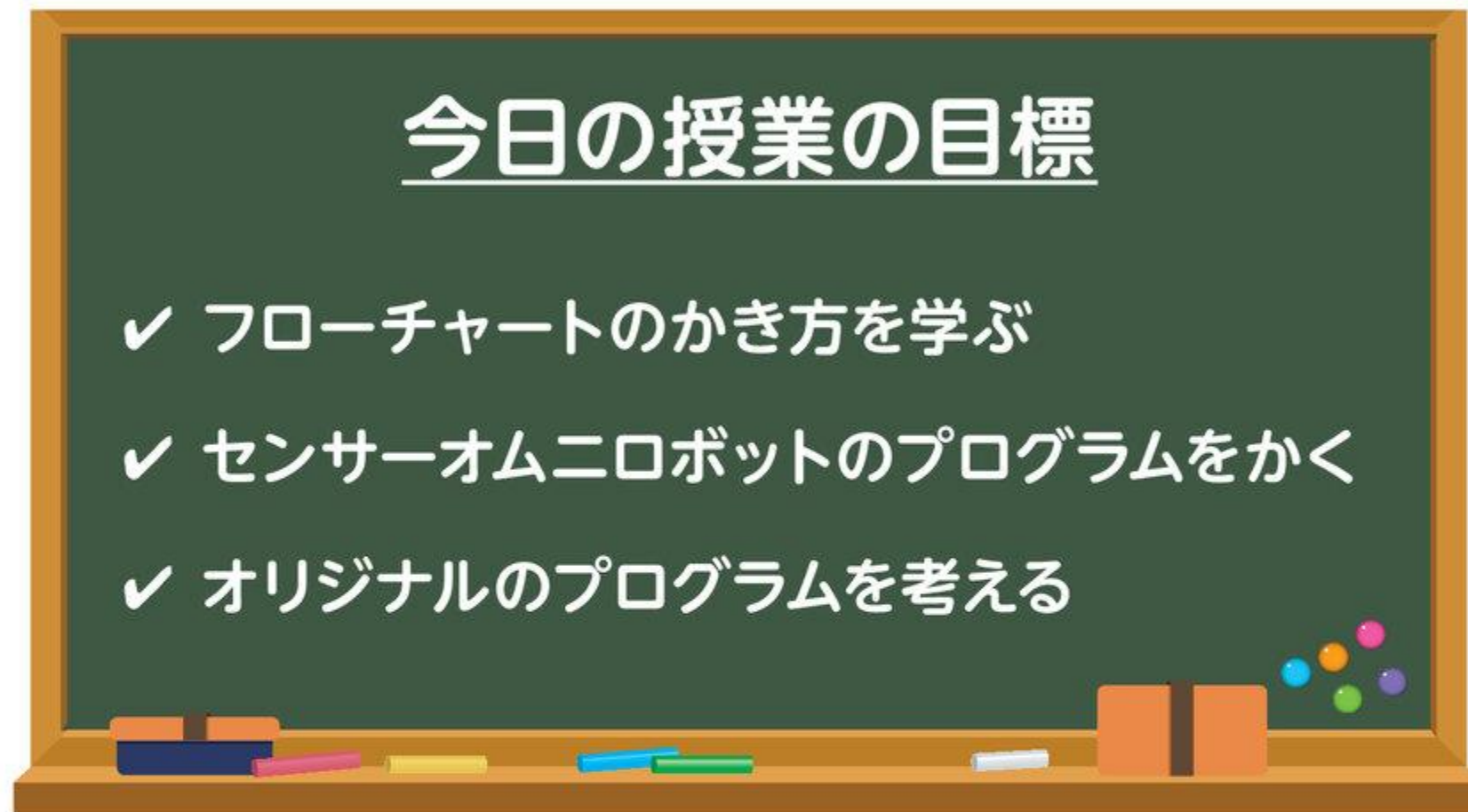
○ 今回の目標をパネルで用意するか、黒板に予め書いておきます。

(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)

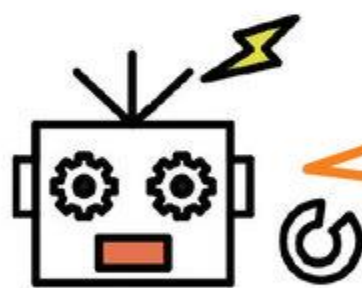
目安時間は授業時間 120 分のうち、休憩 10 分程度を取ることを想定しています。
生徒の進捗状況により、休憩時間などを調整して授業を行ってください。

0. フローチャートとプログラムをかこう (目安5分)

0.0. 「フローチャートとプログラムをかこう」でやること



今回は、前回製作したセンサーオムニロボットにさまざまな動きをさせるプログラムをかいていきます。かなり複雑なプログラムになるものもあるので、ここで一度、^{しよ}処理の手順を流れ図にした「フローチャート」を勉強しましょう。難しいことはありません。たとえば、アドベンチャーゲームをつくるような感覚でしょう。そして、フローチャートをもとにプログラムをかいて、ロボットを動かしましょう。



これができれば、どんなプログラムにも通用するゾウ

講

複数のセンサーを使って情報を読み取ったり、さらにセンサー情報をもとにモーターやスピーカーなどで出力をする場合、プログラムの処理手順がどんどん複雑になります。複雑な処理をプログラムする際には、「どの順番でどんな処理をするのか」をフローチャートでわかりやすく整理し、視覚化することが重要になります。

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。



図 0-0 必要なもの

今回使用するロボットは、前回製作したセンサーオムニロボットです。

オムニホイールロボットに超音波距離センサー 2 個とカラーセンサーを搭載した複合ロボットです。マイコンボードにはロボプロシールドとマトリクス LED シールドを接続し、7 セグメント LED にセンサー情報などを表示させます。

なお、7 セグメント LED の向きには注意が必要です。図 0-1 を参照し、チェックしましょう。

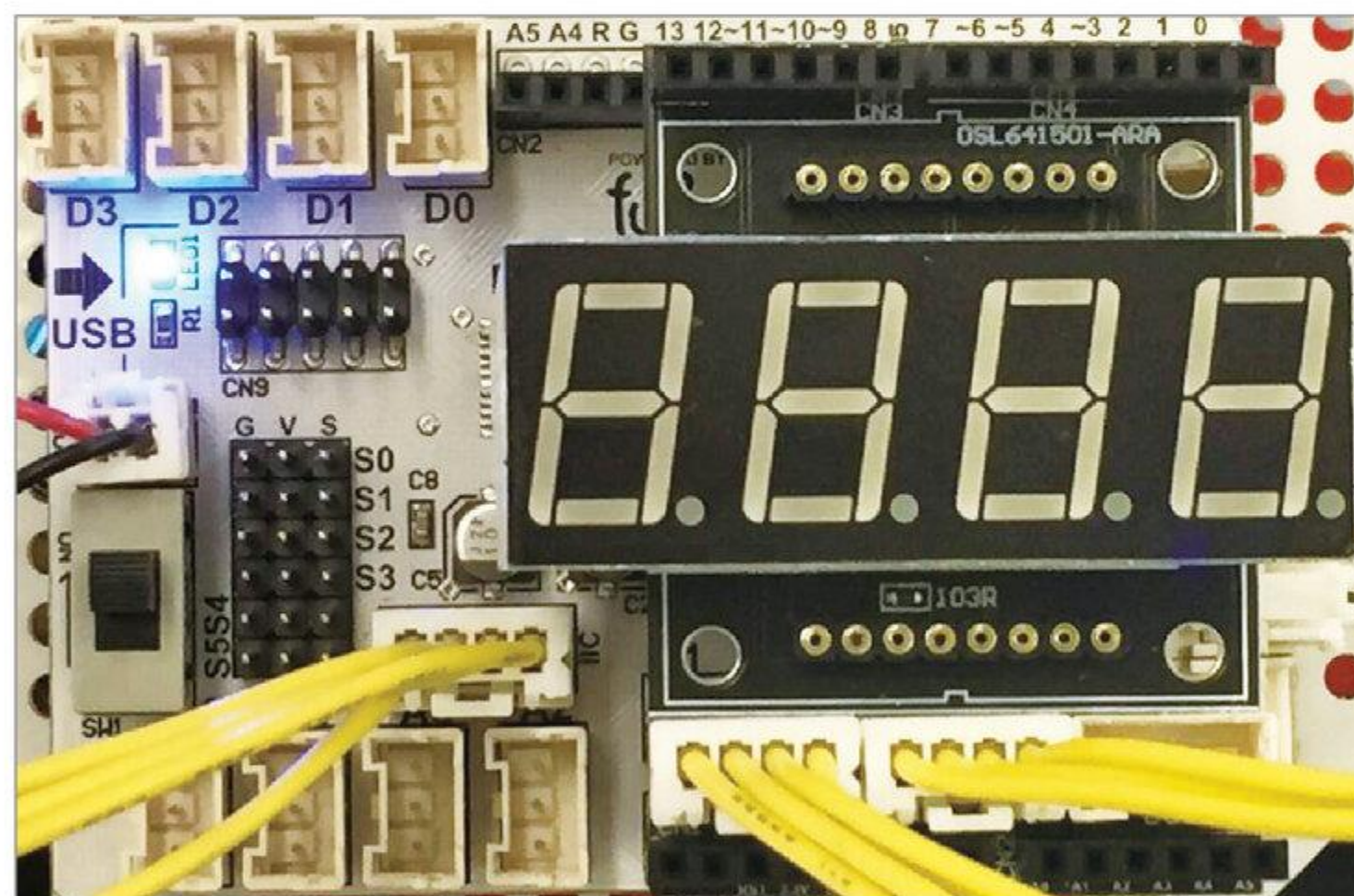


図 0-1 7 セグメント LED の接続

1. フローチャート (目安 15分)

1.0. フローチャートとは

さて、目的に沿っていきなりプログラムをかきはじめることもできますが、複雑な処理になると途中でわからなくなってしまうことがあります。また、たとえプログラムが完成したとしても、しばらく時間が経過してプログラムを修正したいときに、構成を忘れてしまって大変な思いをすることがあります。

そんなときは、これまでも何度か登場した「フローチャート」を使いましょう。

フローチャートを作成しておけば、直感的にプログラムの構成がわかりますし、慣れないうちはとくに、フローチャートを作成してからプログラムをかいた方が効率も良いことがあります。下記のようにフローチャートをつくれれば、どんな動きをするロボットかわかりやすくなります。

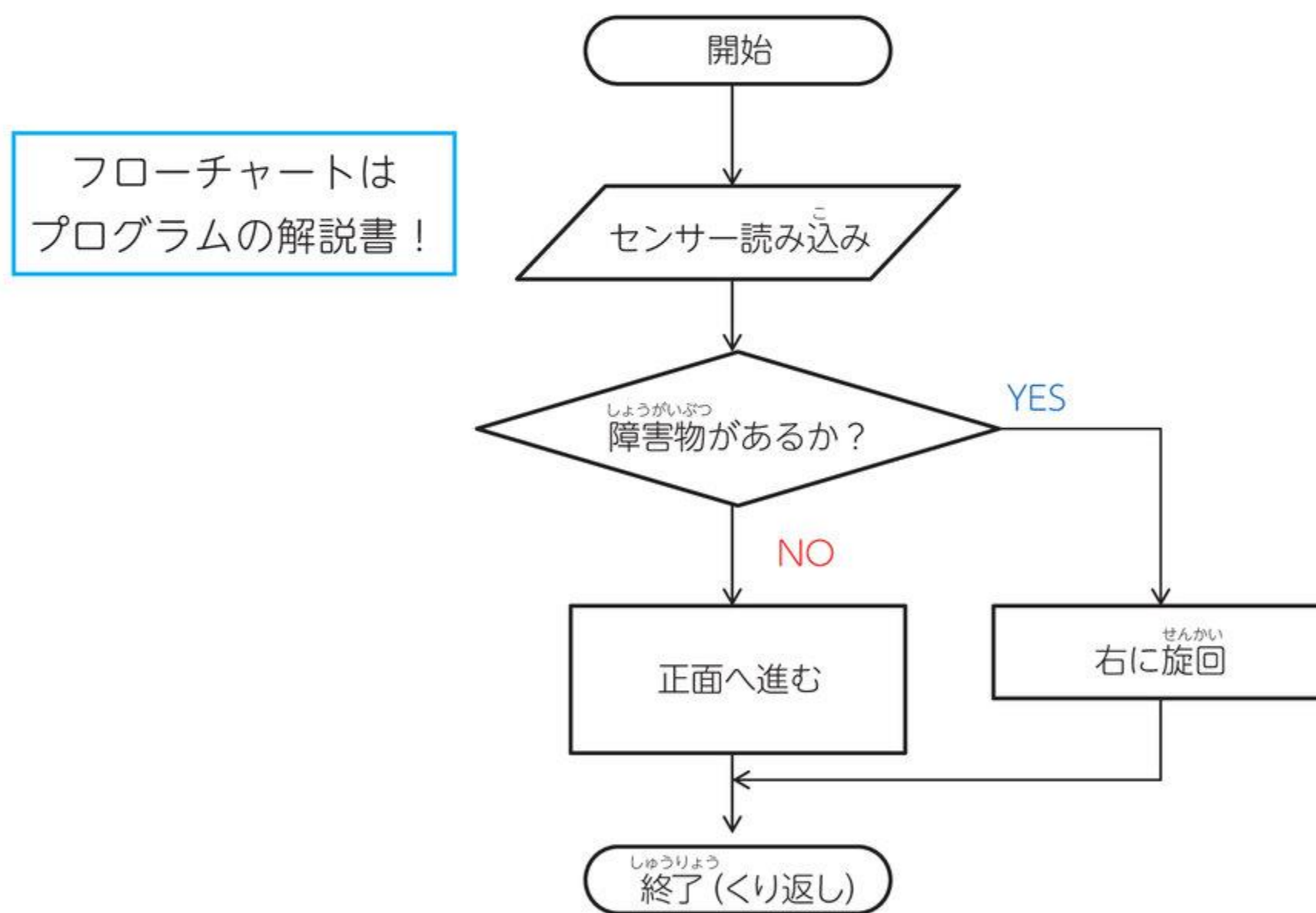


図 1-0 フローチャート例

では、**図 1-0** の例はどんな動きのロボットでしょうか、文章にしてみましょう。

講 センサーが障害物があるかを判断し、障害物がない場合は正面に進む。障害物がある場合は右に旋回する。

講

ロボットのアルゴリズム (処理手順) には、フローチャートを活用します。さまざまなセンサーを使用するケースでは、プログラムをかく前に処理手順の整理をすることが大切です。ここが今回のポイントです。

1.1. フローチャートをかく

まずは、どんなフローチャートの部品があるか見ていきましょう。

表 1-0 フローチャートの部品



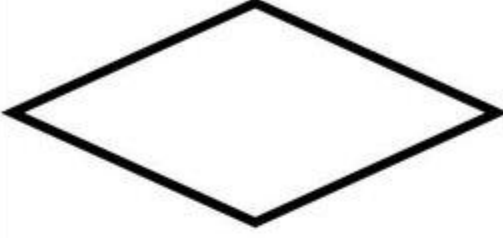
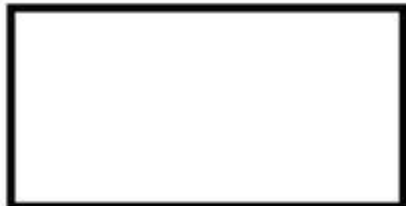




			
端子 (開始・終了)	入力データ	判断分岐	処理

図 1-0 のフローチャートも、表 1-0 の 4 種類の部品からできていますね。それでは、この部品を使って、フローチャートをつくる練習をしてみましょう。


フローチャートは端子記号  で始まり、 などを縦に順番に並べていきます。

途中で判断分岐  が入ったりしたら、そこから分岐の数だけ線を出して、その先にそれぞれの処理などを縦に並べていきます。

各記号を線で結んでいき、最後に端子記号  で終わります。

やってみよう！

次の処理をフローチャートにしてみよう。P.6 にある方眼紙を使ってフローチャートを、P.5 のヒントを手がかりにつくりあげてみよう。

処理：距離センサー  (前面センサー) が 30 センチ以内に障害物を検知したら停止する。それ以外の場合は前進する。

講

フローチャートをいきなりかく前に、ヒントを読解し、処理を整理させてください。P.7 に解答例があります。

POINT

ヒントを手がかりに次のページの方眼紙に、フローチャートをつくっていこう。

ヒント①

この処理を表に整理すると、次のような要素があるよ。

入力データは US1 の超音波距離センサー
判断分岐は 30cm以内かどうか
処理は分岐して YES なら停止する NO なら前進する

ヒント②

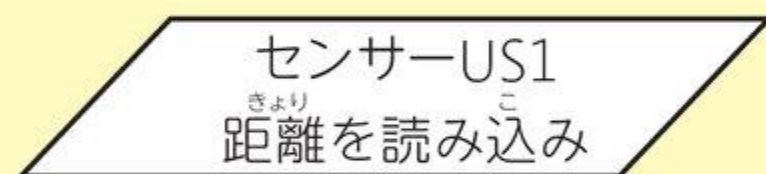
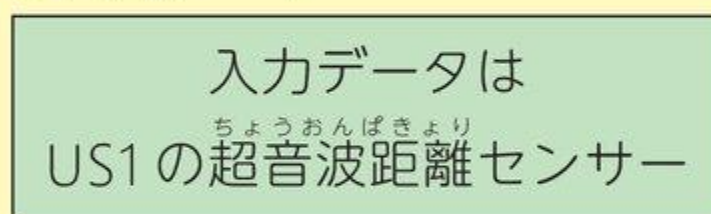
フローチャートの始まりと終わりは、次のようにしよう。



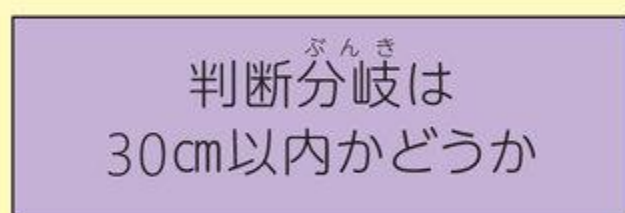
ヒント③

表で整理した要素を部品にしよう。

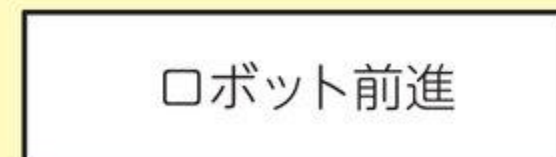
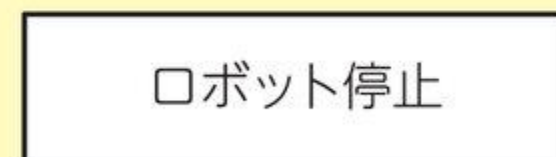
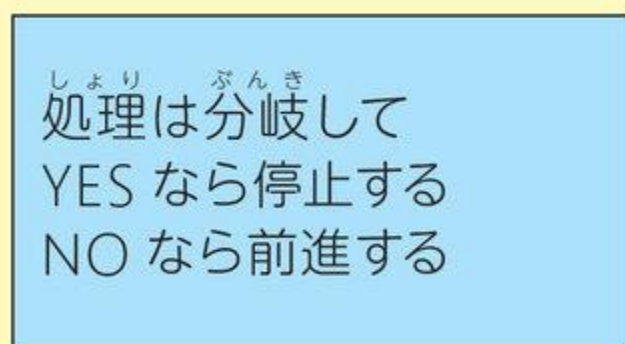
●入力データ



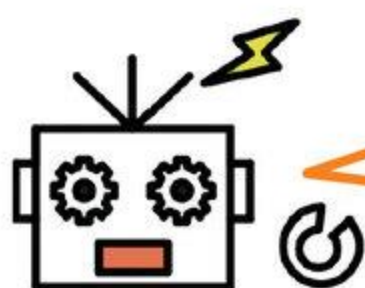
●条件分岐



●処理結果

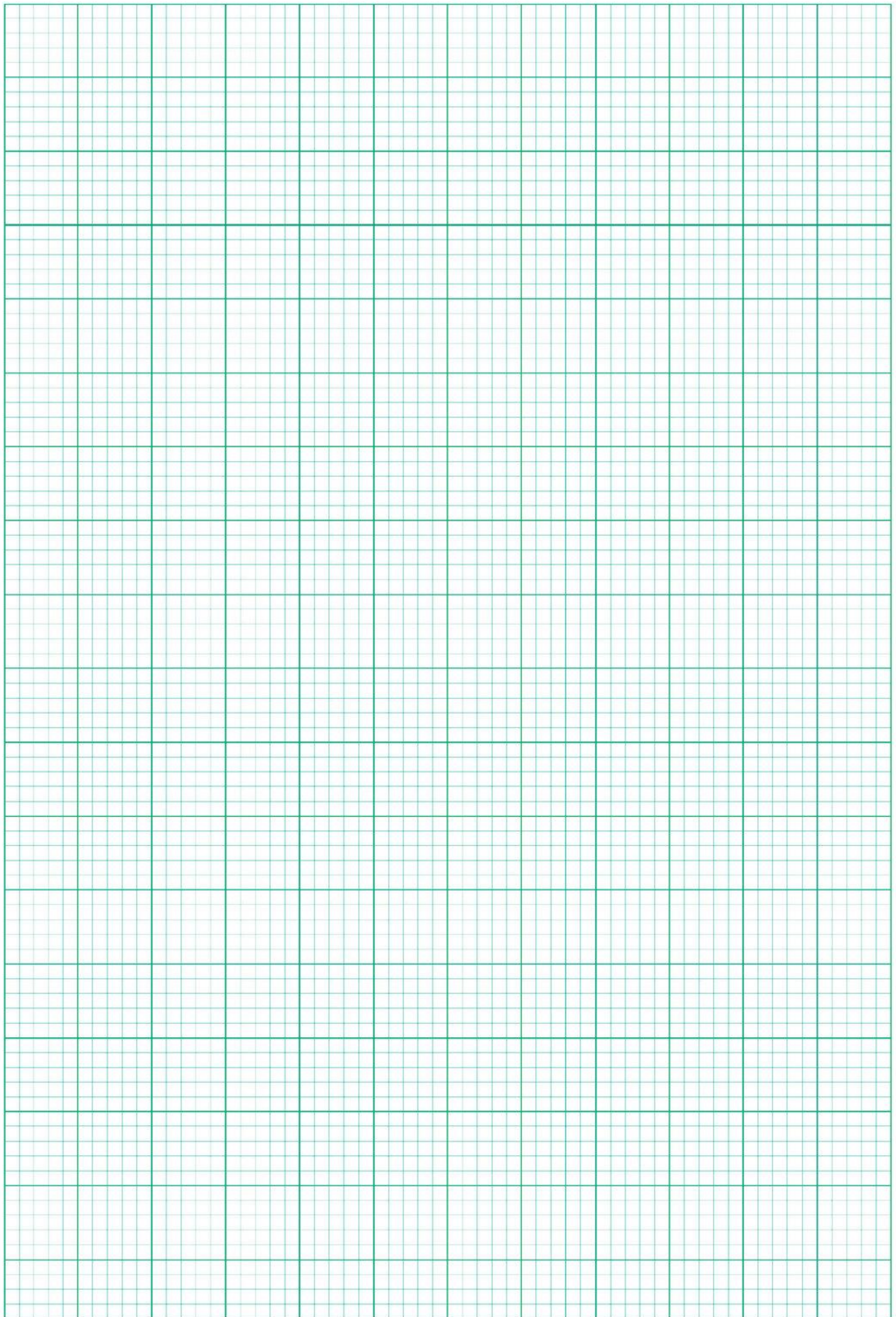


このように、処理手順を表に整理してからフローチャートの部品にしていくと整理しやすくなるよ。それでは、フローチャートを完成させよう。



文章にして、表にして、フローチャートにする。最後にプログラムをかく。メンドウに感じるかもしれないけど、大切なんだよ。

手書きでフローチャートをつくろう



ヒント①～③を手がかりにすると以下のようにかくことができます。

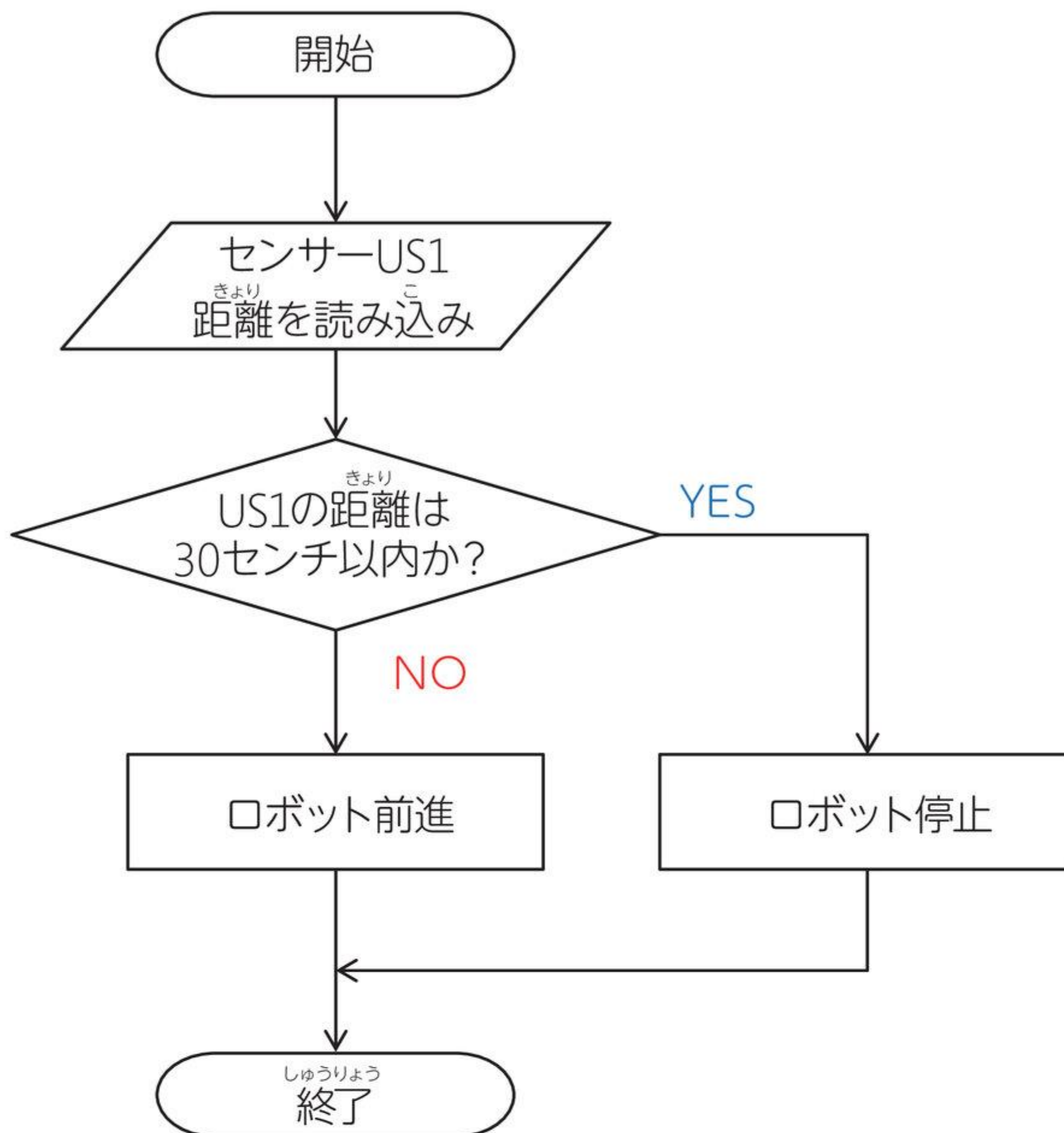


図 1-1 「やってみよう！」の解答例フローチャート

フローチャート解答例は 30cm以内に障害物があるかどうかを判断分岐としています。

講

プログラムが複雑になればなるほど、ロボットの動作やセンサーの反応（処理）をフローチャートにすることが必要です。

フローチャートはプログラムの設計図なので、誰が見てもわかりやすい内容にすることが大切です。

判断分岐は「YES」と「NO」で結果を分岐させます。

2. いろいろなセンサーロボット (目安 80 分)

2.0. つかまらないロボット

さて、フローチャートのつくり方はわかってきましたか？ 今度は実際にセンサーオムニロボットを使って、いくつか動作をさせてみましょう。

まず最初に、どの方向から人が近づいてきても、「つかまらないロボット」をつくってみましょう。使える超音波距離センサーは2個ですが、プログラムを工夫して、全方向360度を見られるようにしましょう。さらに、近づいた距離によって、危機感が高まって、逃げる速度が変わるなどといったオプション機能も追加してみましょう。

わかった人は、この先の説明は飛ばして、フローチャートをつくってプログラムをかく準備をしていてください。方眼紙は巻末のページにあるので確認してください。

イメージがつかない人は、以下の手順を読んで考えていきましょう。

手順1) 動作をイメージする

まずは、しっかりと動作のイメージをしてみましょう。図2-0のような動作をするロボットを目指します。以下のように機能別に二段階にわけて考えてみます。

①サーチモード：図のように回転しながらセンサーで周りを見張っている状況です。

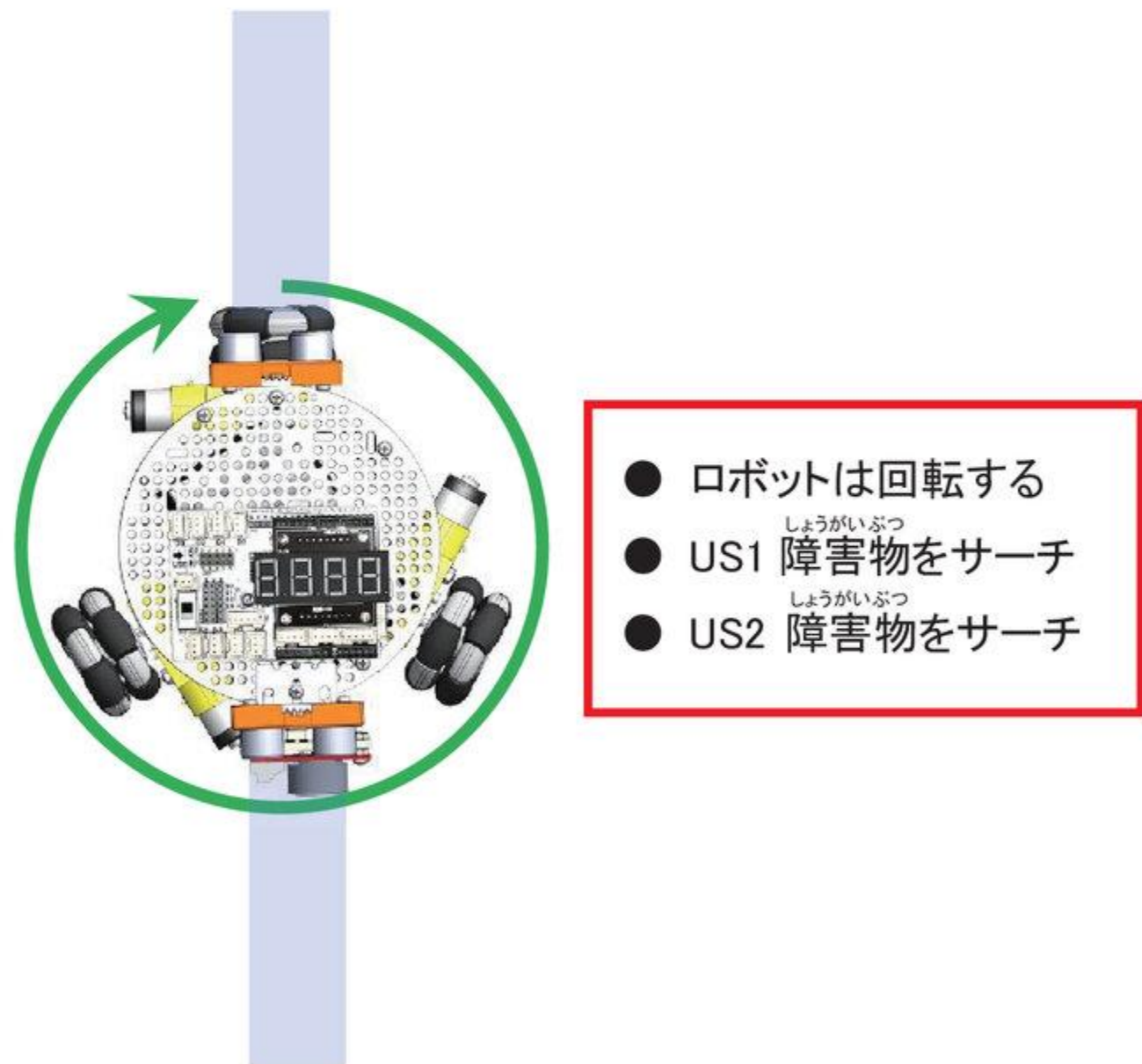
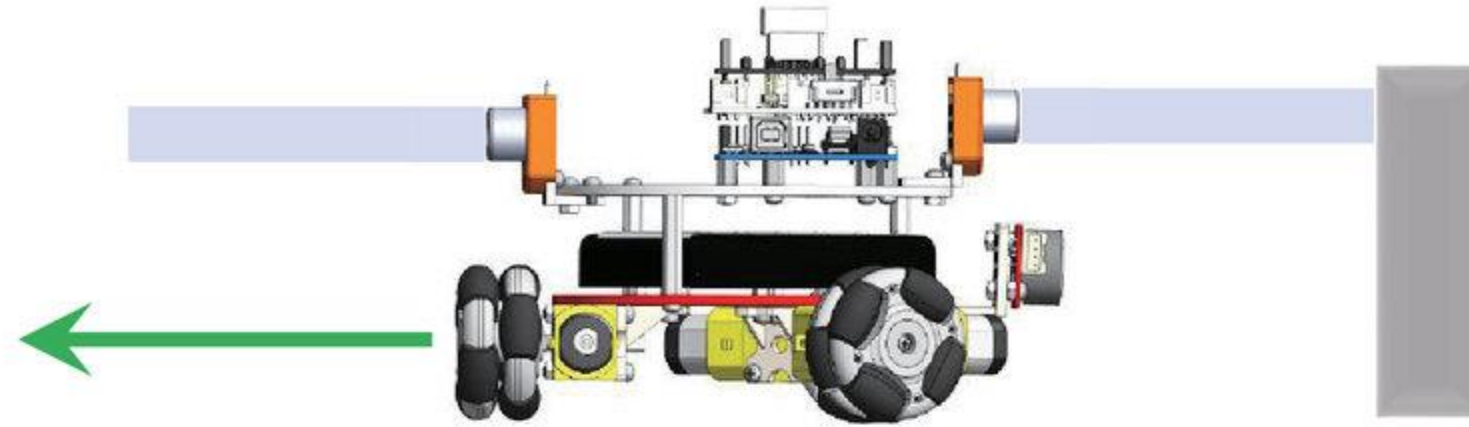


図2-0 つかまらないロボットのイメージ①

②逃げるモード：いざ、障害物を発見したら逆方向に逃げていく、という機能を加えます。



- 超音波距離センサーが、指定された距離内に障害物を検知すると、障害物のある方と逆の方向に反応がなくなるまで逃げていく

図 2-1 つかまらないロボットのイメージ②

手順 2) 表にまとめフローチャートをかく

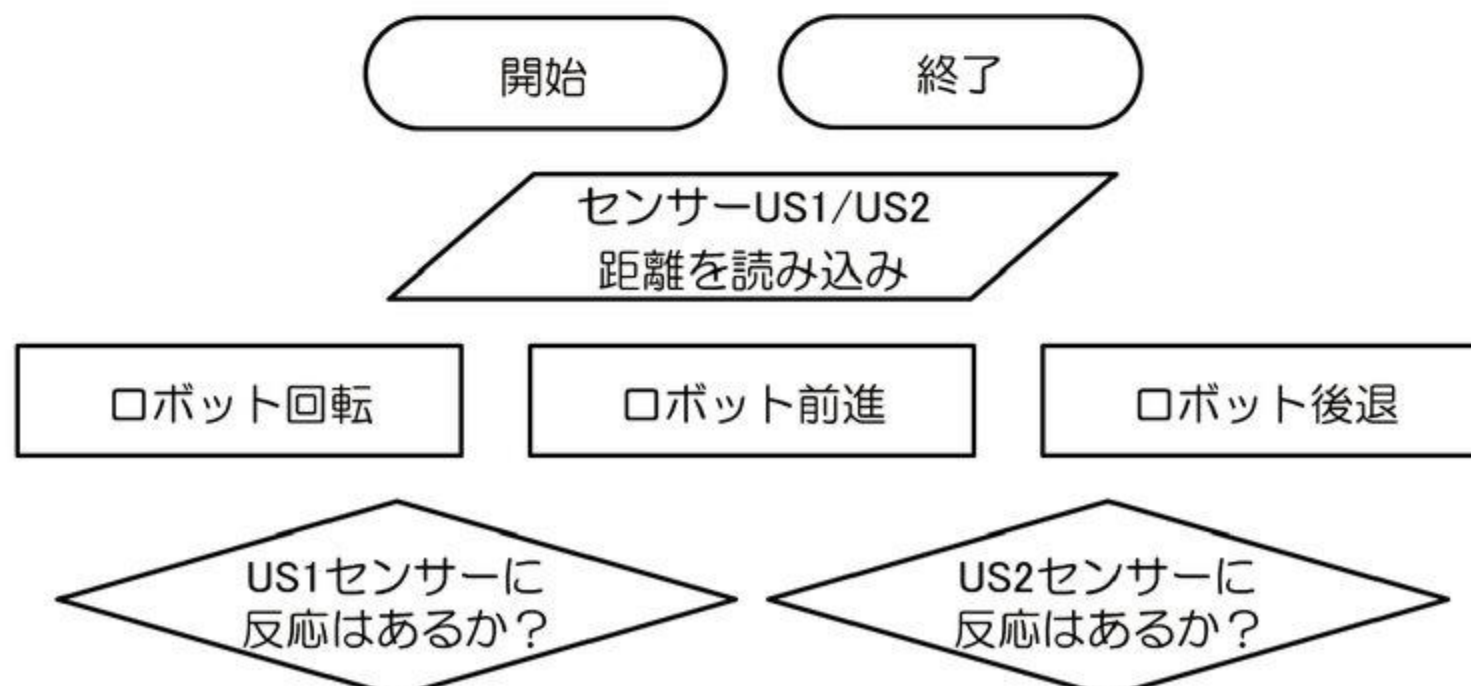
このロボットの処理の流れを具体的にかいてみると次のように整理できます。

表 2-0 つかまらないロボットの処理

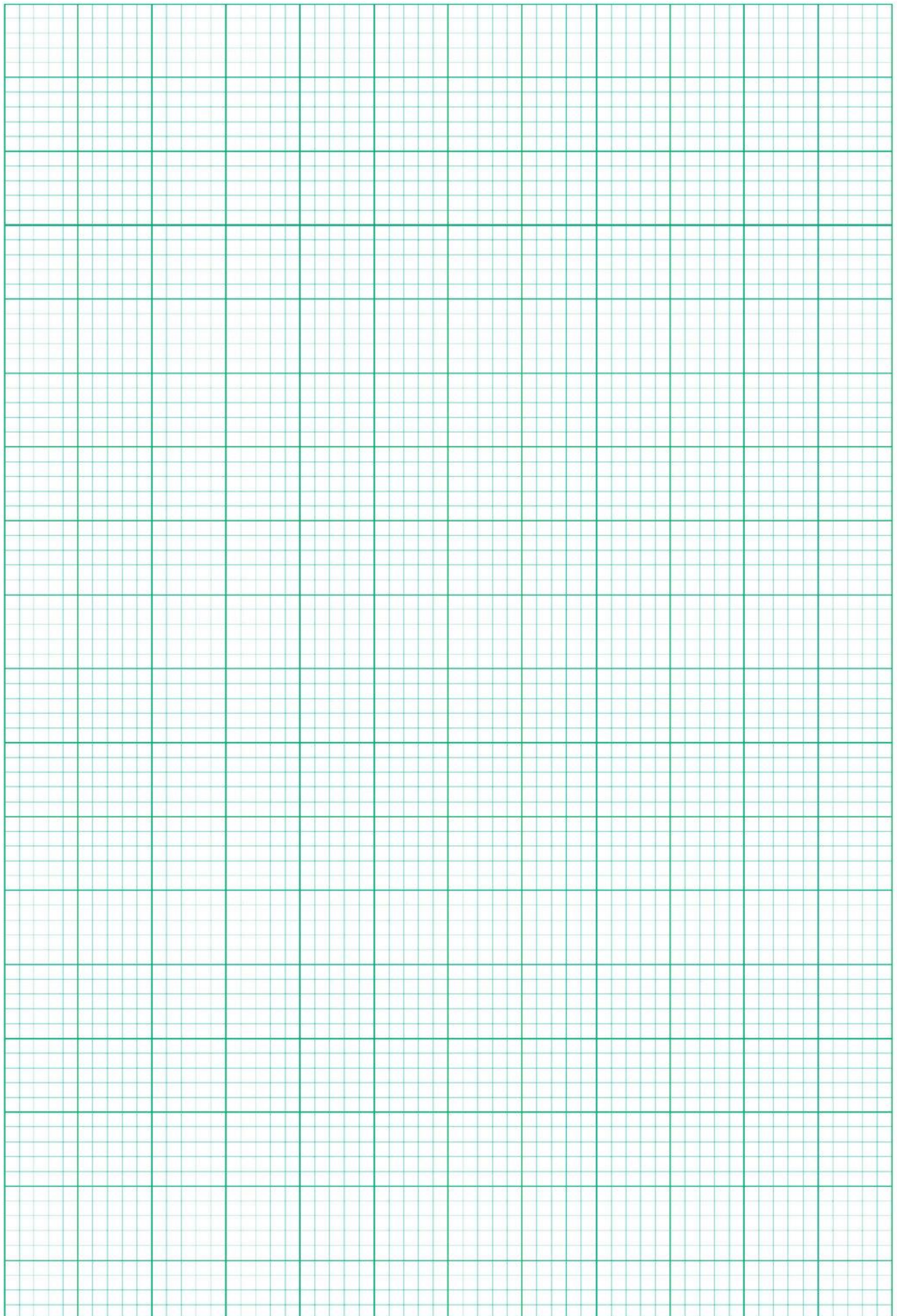
入力データはセンサー US1 とセンサー US2 の距離情報入力
判断分岐は近くに障害物があるかどうか
YES ならば 反対方向に逃げるモード NO ならば サーチモード
逃げるモード 障害物を検知すると逆方向に逃げる
サーチモード その場で回転しながら周りの状況を収集する

ステップアップ

上で整理した流れを参考に、「つかまらないロボット」のフローチャートをかいてみよう！
下のパーツを使ってみてね。同じパーツを複数回使っても OK だよ！



手書きでフローチャートをつくろう



フローチャートはうまくかけましたか？ 巻末に解答例も載せておくので確認してみましょう。このフローチャートをもとにしたのが以下のプログラムです。

∞ プログラムの書き込み

RoboticsProfessorCourse2 > CombRobot2 > OmniRunAway2

今度は、このプログラムにいろいろな機能を追加してみましょう。

チャレンジ課題

プログラム「OmniRunAway2」をかきかえ、以下の①～③のオプション機能を追加していこう！

オプション① ダッシュモード

処理：「OmniRunAway2」に加え、逃げるモードのときの移動が障害物に近いときほど速くなるようにする。なお、逃げるモードに移行する検知距離は5cm～30cmとする。

💡 ヒント

逃げるモードのときの移動速度を、-30や30といった定数ではなく変数 `d1` や `d2` を用いた数式にしてみよう！
`d1` に代入される値が小さければ小さいほど、式の結果は大きくなるようにすればいいんだね。どんな数式ならいいかな？

オプション② 横方向に逃げる

処理：オプション①の機能に加えて、もし `US1`、`US2` どちらのセンサーも障害物を検知したら横方向に3秒間移動するようにする。

💡 ヒント

論理演算子をたくさん使うね！ if文をかいていく順番も大事だよ！

オプション③ 逃げるモード+ α

処理：オプション①・②の機能に加えて、逃げるモードを3秒間に延長するかわりに、逃げるモード時にも進行方向をチェックするようにする。

💡 ヒント

`delay(3000);` とするだけだと、3秒間逃げるモードを続けるだけになってしまうね！ センサーをチェックし、何もなければ短時間だけ逃げる、という動きを、計3秒になるまでくり返すようにできるかな？

講

解答例プログラムはそれぞれ以下です。

RoboticsProfessorCourse > CombRobot1 > OmniRunAway3

RoboticsProfessorCourse > CombRobot1 > OmniRunAway4

RoboticsProfessorCourse > CombRobot1 > OmniRunAway5

講

たとえば、センサーが障害物を検知して距離によって速度をかえながら3秒間逃げるプログラムは普通にかくと以下のようになります。

```
omniBot.move(0, -500/d1, 0);  
delay(3000);
```

しかし、この場合、逃げている間にセンサーが新たな障害物を検知できません。そこで、センサーを使いながら最大3秒間逃げるという処理にしています。

以下が完成したプログラムです。プログラムを確認して実行してください。

∞ プログラムの書き込み

RoboticsProfessorCourse2 > CombRobot2 > OmniRunAway5

講

「OmniRunAway2～5」と4段階に積み上げています。それぞれのプログラムをフローチャートと照らし合わせて学習してください。結果は「OmniRunAway5」の完成プログラムを実行して、ロボットを観察させてください。

なお、電池残量などによっては、うまく動作しないこともあるためご注意ください。

2.1. 色探しロボット

続いては、部屋の中をウロウロしながら、目当ての好きな色紙の前で動きを停止するロボットをつくりましょう。

講

授業の進行状況により、チャレンジ課題に取り組ませてください。

1) 動作をイメージする

処理を大きく2つに分けて考えてみると、以下のようになります。

1. 部屋をウロウロする（障害物を避けて進む）

2. 色を認識して、その前で止まる

2つとも難しくありませんよね。

「部屋をウロウロする」は、今回のテキストの一番はじめのフローチャートの処理でできますね。

ただし、カラーセンサーがロボットの後ろ側についているので、超音波距離センサーも後ろ側のUS2を使い、今まで後退だった方向が前進方向になります。

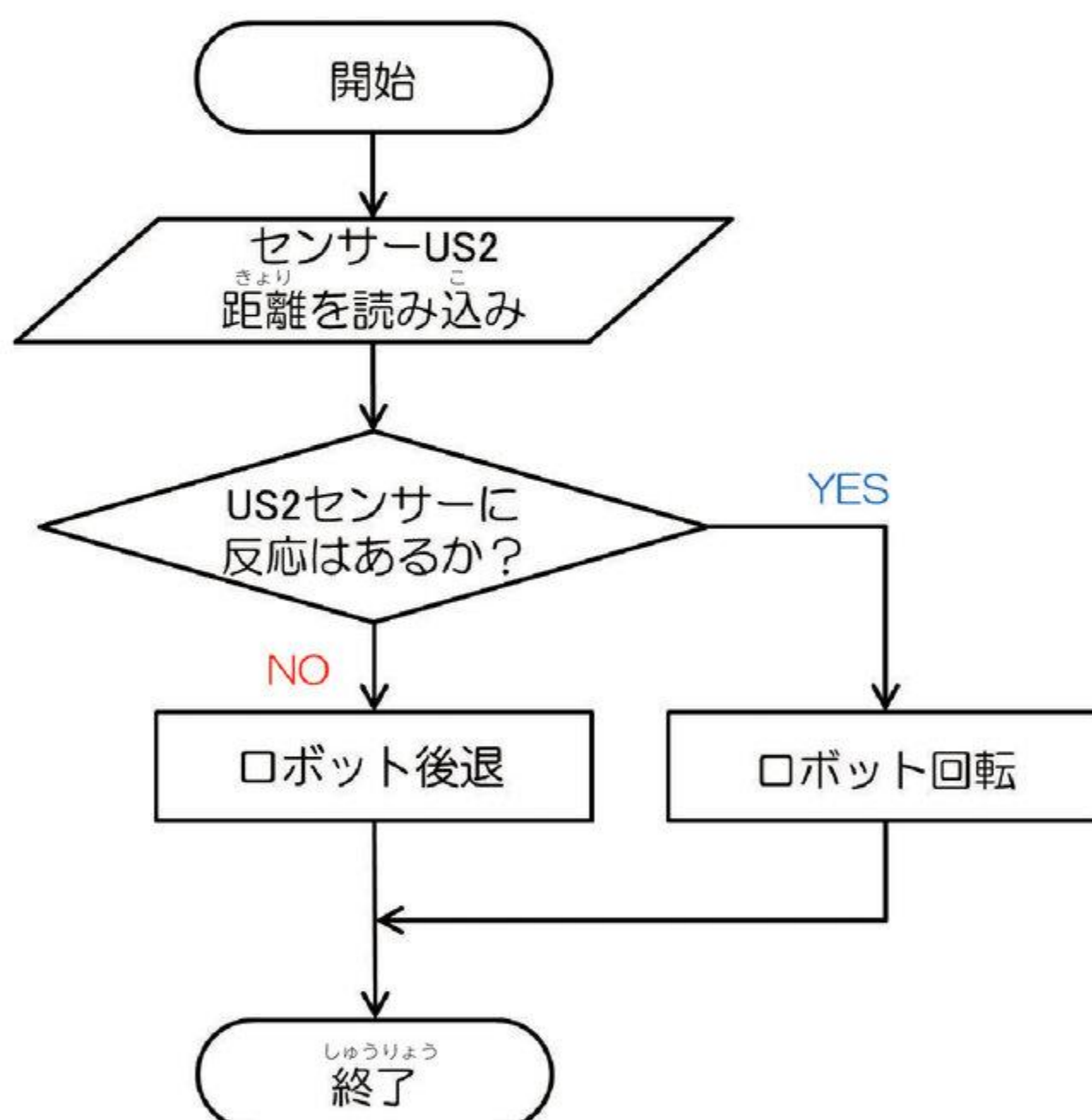


図 2-2 障害物を避けて進むフローチャート

講

カラーセンサーは、同じフロアの中でも明るさによって結果に差が出ます。明るい場所、暗い場所、数箇所を実験をして、一番良い結果が出る場所を探すようご指導ください。

また、電池残量が低下すると検出精度が低下します。センサーを複数使用するロボットは電池を消費しますので、センサーの精度が低下した際には交換させてください。

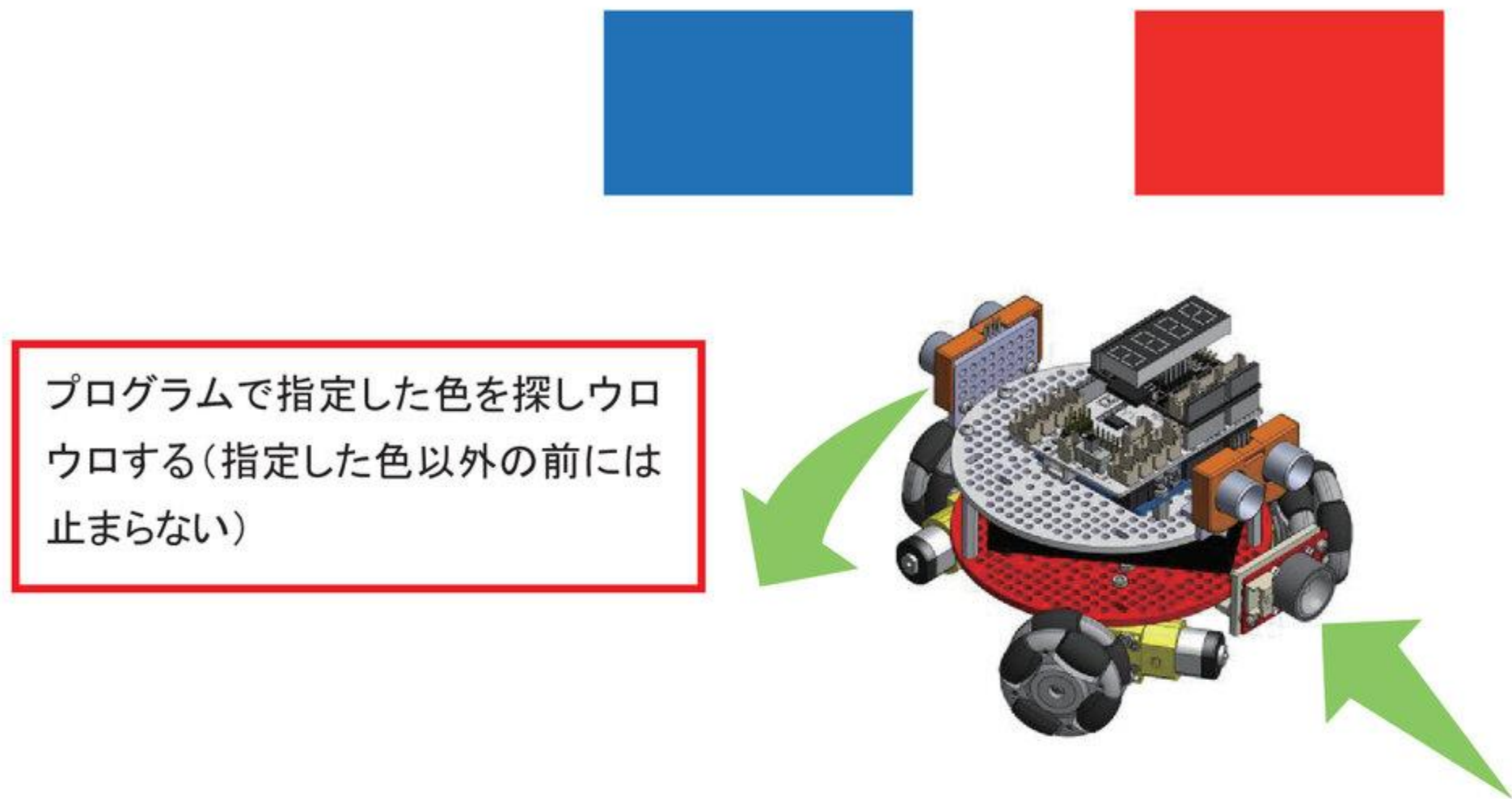


図 2-3 色探しロボットイメージ (ウロウロ)

「色を認識して、その前で止まる」も、センサーがかわるだけで処理は1.と似ています。

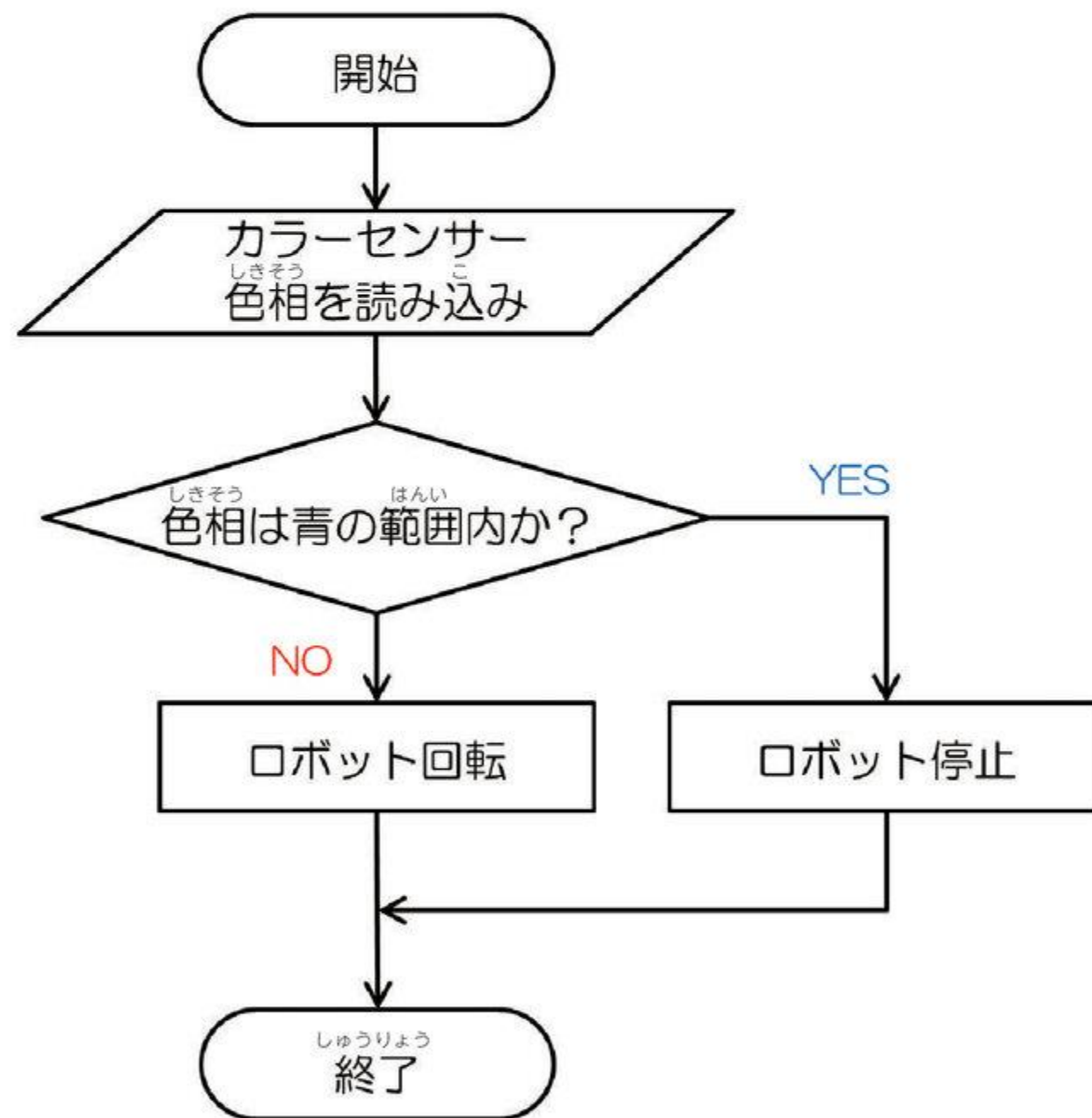


図 2-4 「色探しロボット」のフローチャート

たとえば、青色を探すのであれば、
色相の範囲条件を240度前後に
値を設定する

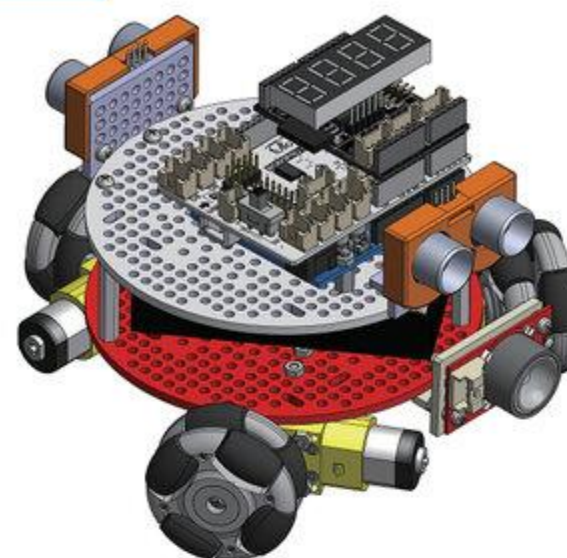


図 2-5 決めた色の前で止まる

上記のように青色の場合は色相の範囲条件として 240 度前後になるはずですが。

2) フローチャートをかく

全体の処理としては以下のようになっています。これを巻末ページのフローチャートにかき起こしてみましょう。

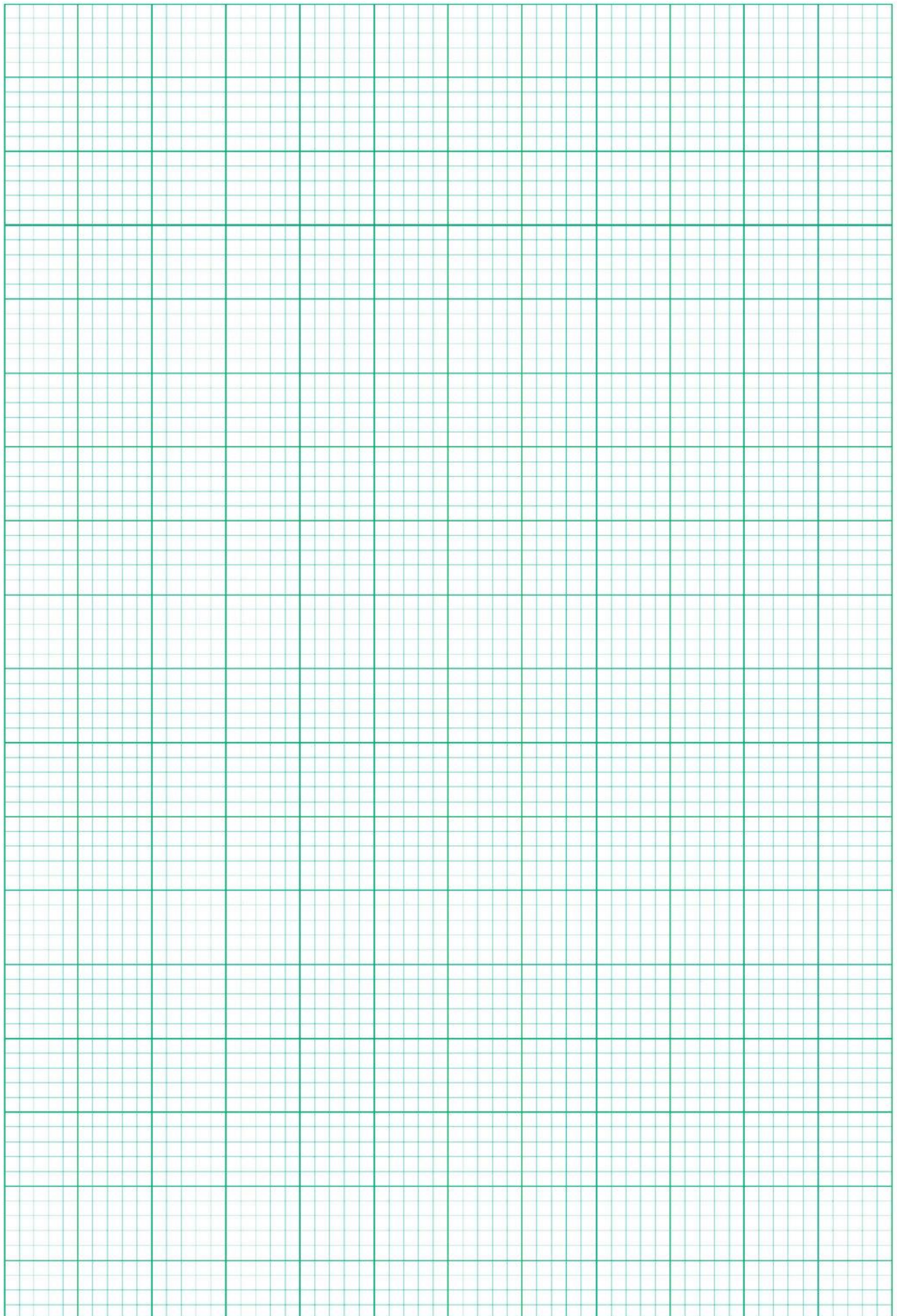
表 2-1 色探しロボットの処理

入力データは US2 の距離情報入力	
判断分岐は近くに障害物があるか	
YES ならば カラー検知モードへ移行 NO ならば 直進し続ける	
カラー検知モード 障害物まで5センチに近づいてから色を検知する 判断分岐：決められた色か判断する	
YES ならばストップ NO ならば障害物回避モード その場で旋回し、障害物を避ける	<p>講</p> <p>フローチャートをかかせてから、次ページの解答例に沿って、処理手順を説明してください。</p>

ステップアップ

図 2-2 と図 2-4 のフローチャートを合体させて、色探しロボットのフローチャートの完全版をかいてみよう！

手書きでフローチャートをつくろう



フローチャート解答例は次のとおりです。

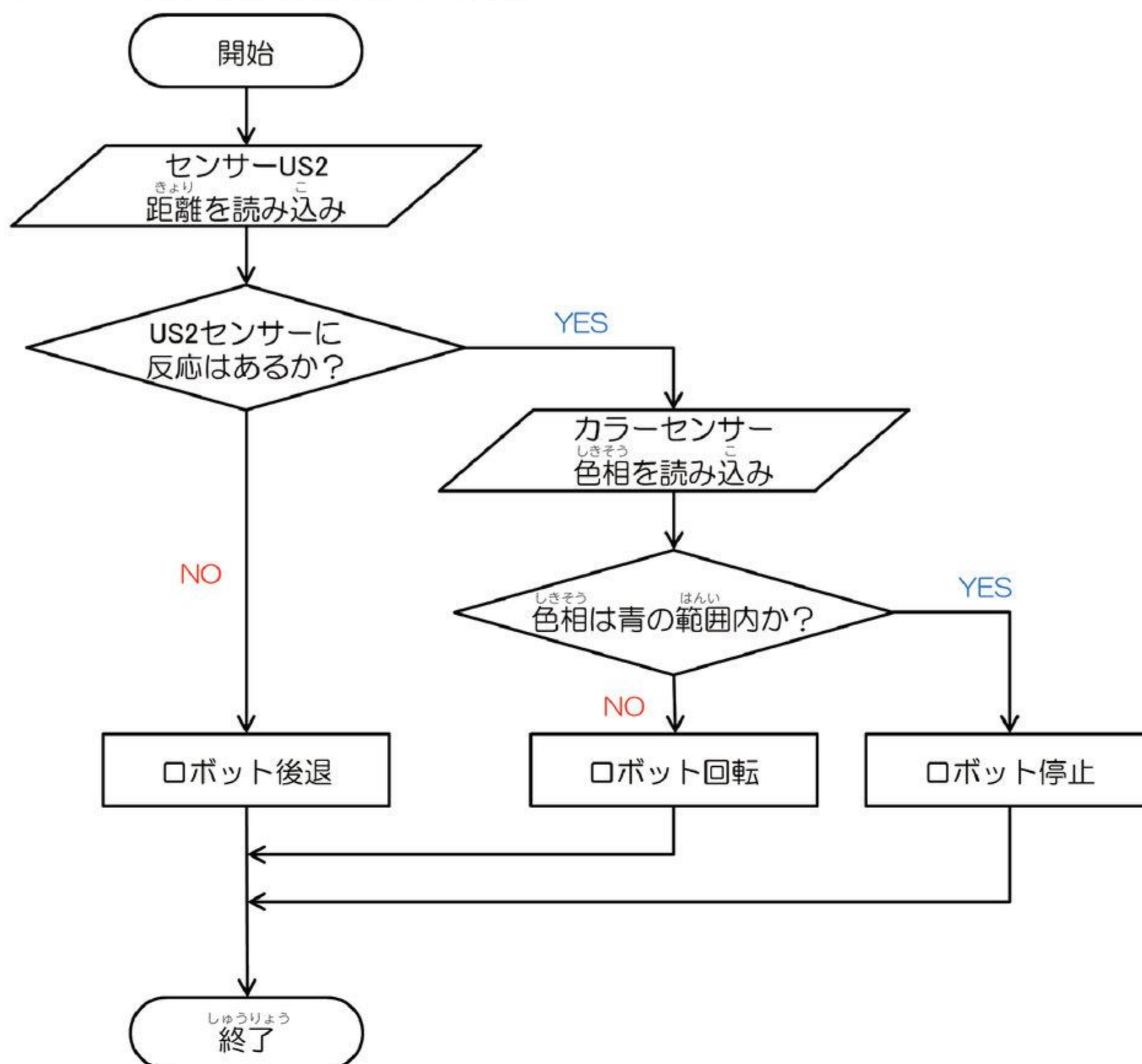


図 2-6 フローチャート解答例

チャレンジ課題

自信がある人は、色探しロボットのプログラムを自分でつくってみよう！

`void loop()` 以外の部分は第1回で使用したプログラム「ColorMotorTest」をベースに使えるよ！ `void loop()` の中は、第1回と第2回で使用したプログラムをうまく使ったり、完成したフローチャートを参考に if 文をかいたりしてみよう！

以下のプログラムを実行してください。

プログラムの書き込み

RoboticsProfessorCourse2 > CombRobot2 > ColorSeeker

ColorSeeker のプログラムの調整方法

講

教室の明るさなどにより、カラーセンサーで色が検知しにくい場合は、プログラム内の以下の部分で検知する色相を変更してください。

```
lc.setDec(0, h); // H 値を7セグメント LED で表示する
if(h >= 200) // 青色を探す 見つければ LED を光らせて終了する
```


3. まとめ（目安5分）

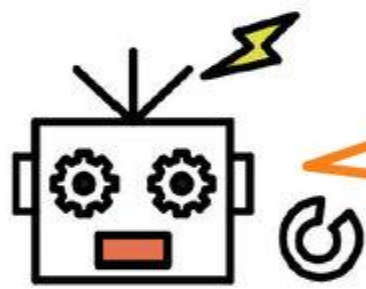
これで、プログラムがわかってきたでしょうか？
 今回のプログラムができるようになっていれば、相当レベルが上がっています！
 次回はさらに複雑なことをできるようにしましょう！

<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。

ラジオペンチ	1	ドライバー	1	USB ケーブル	1	スピーカー	1
							

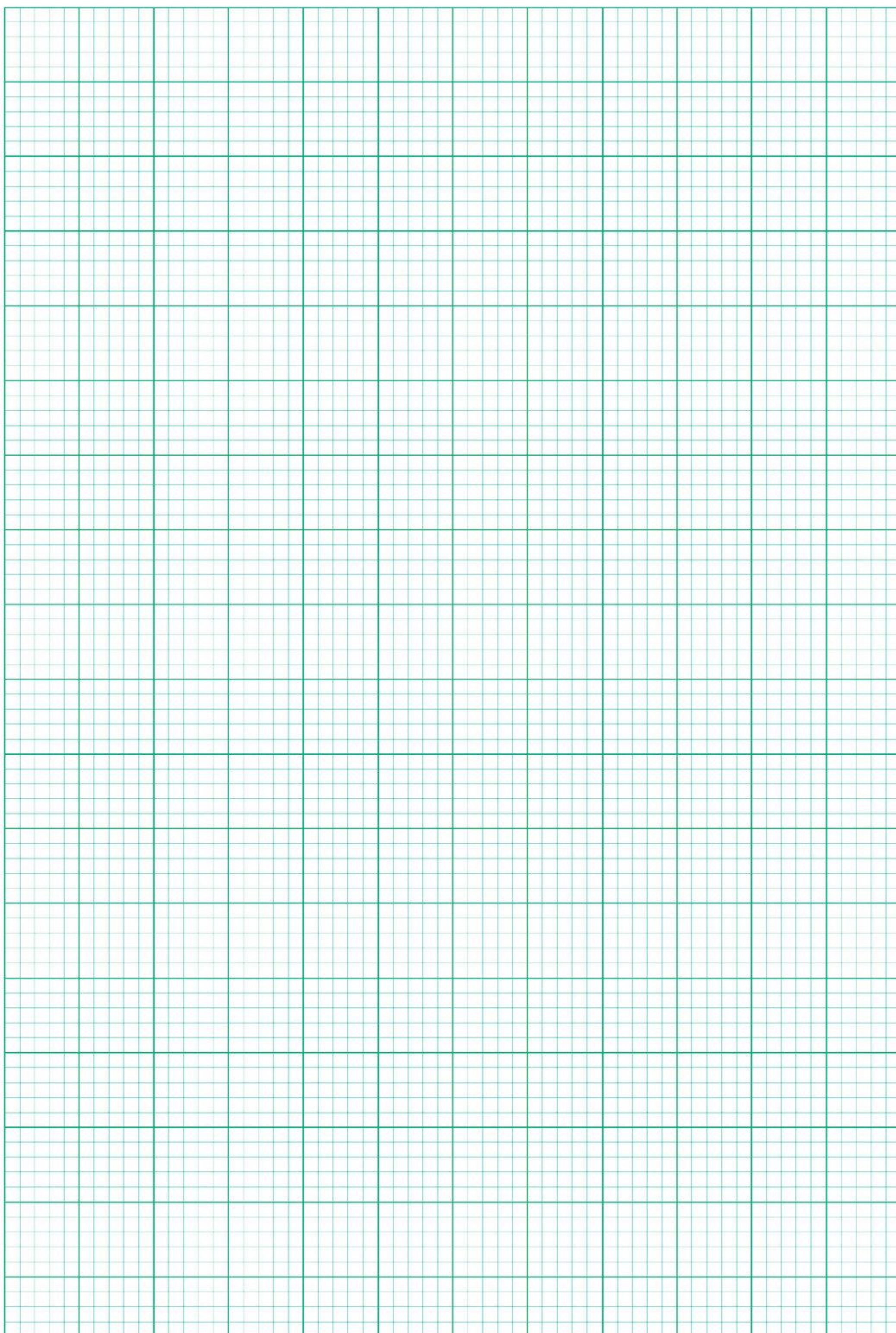
図 3-0 次回必要なもの



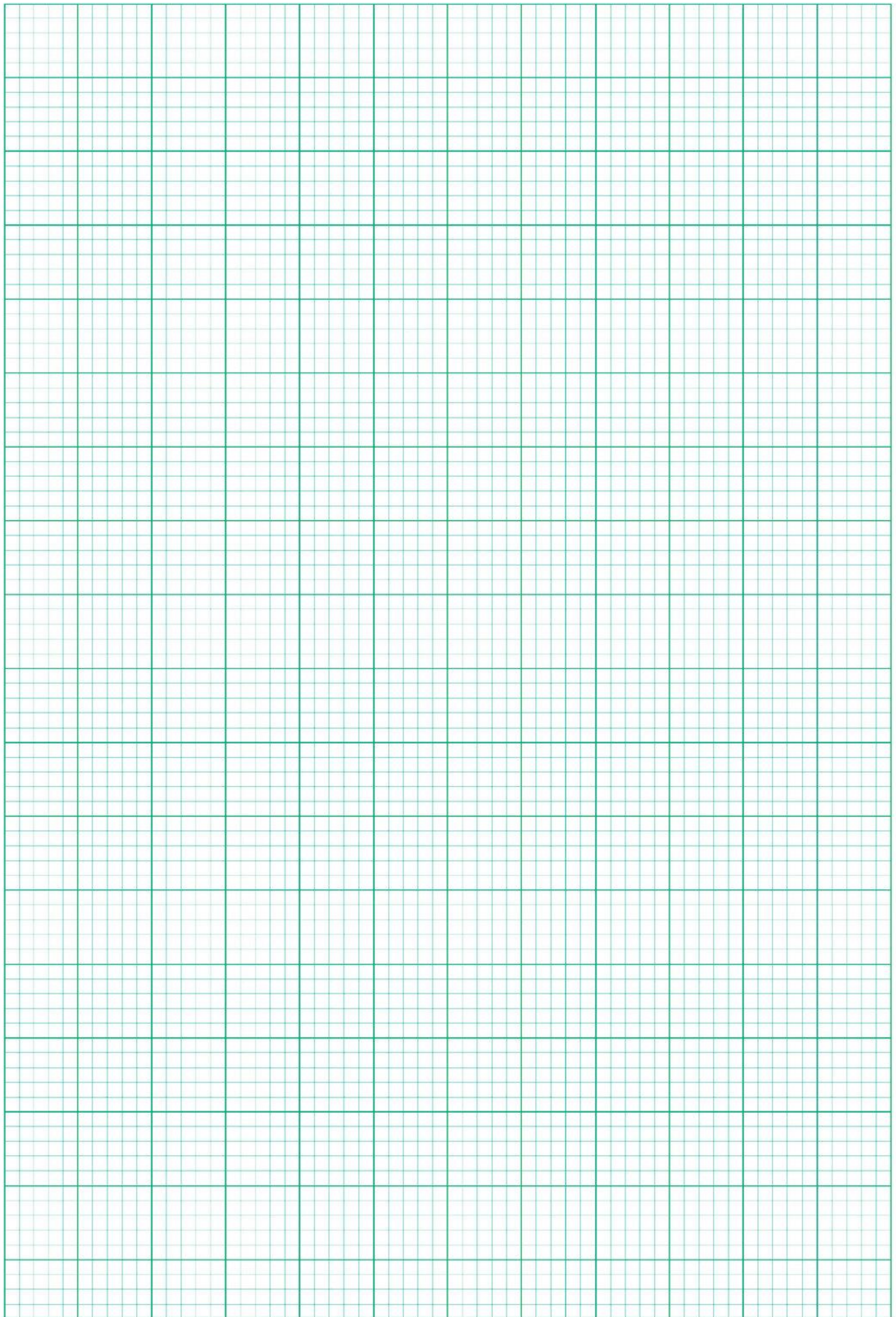
次回は、もっとフクザツなフローにチャレンジするヨ。

講

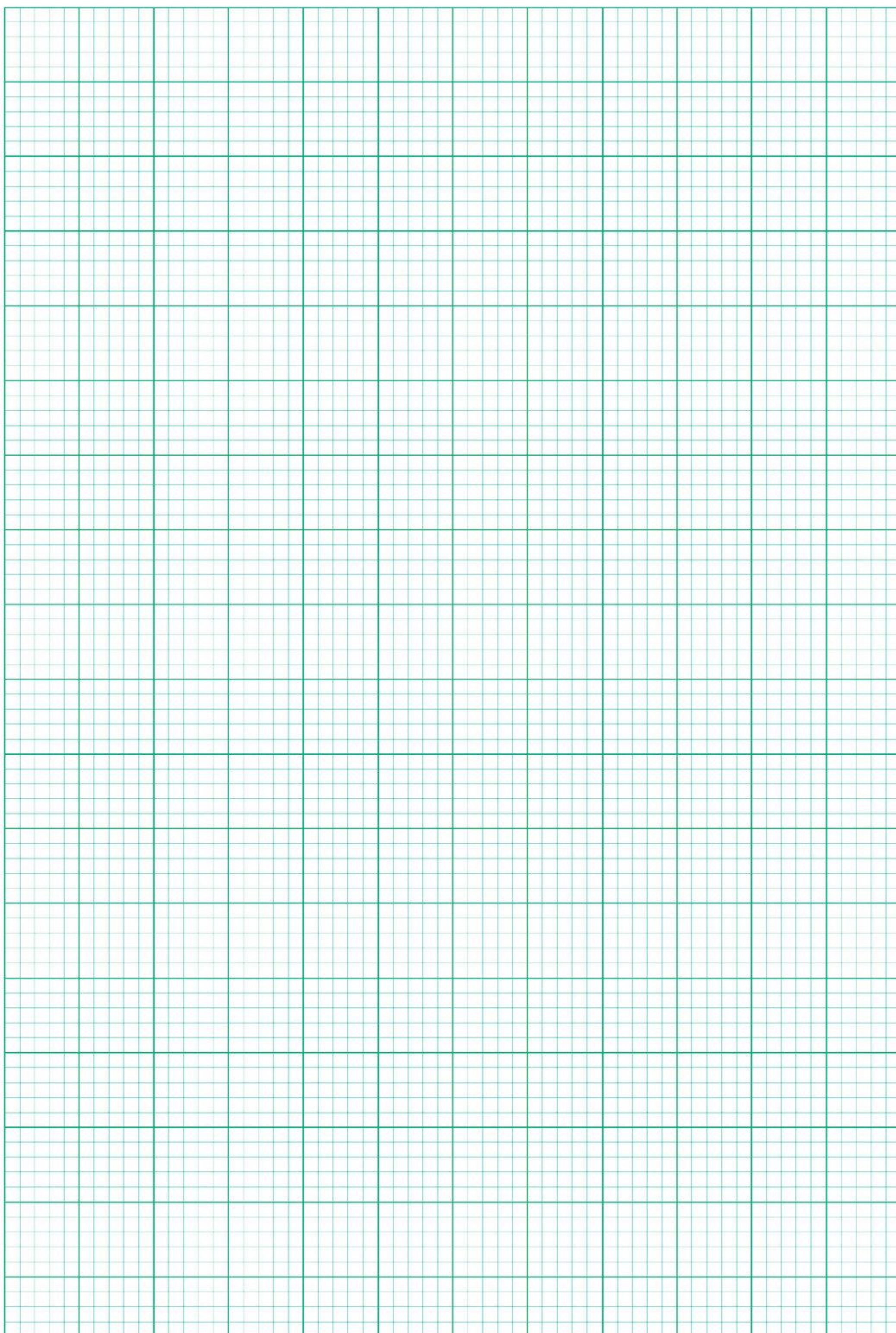
- 以下の理解度を確認します。
 - ・フローチャートのかき方を学ぶ
 - ・センサーオムニロボットのプログラムをかく
 - ・オリジナルのプログラムを考える
- 次回のテーマは「センサーで工夫しよう」であることを告知します。



自由方眼紙①



自由方眼紙②



自由方眼紙③

つかまらないロボットのフローチャートの解答例

