

2-1. 問題解決の手順

③ 設計

学習の目標

- ☐ 設計の流れと必要な要素を知る。
- ☐ 設定した課題に向けた設計を行う。

考えてみよう

材料と加工の技術を利用して、できるだけ小さなスペースで、本や小物を整理するには、どのようにしたらよいだろうか。

■ 基本的な構想における問題解決の工夫

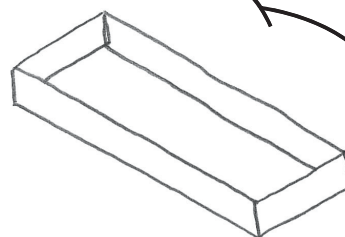
材料と加工の技術を利用して，どのように問題を解決するか全体の構想を立てます。構想を立てる際には，問題を解決するために設定した課題に基づいて，その課題を解決するための工夫をスケッチにして，よりよい構想にしていきます。



25図 構想を立てる例

ものを収納するのに困っている。
小物置きがあったら便利に片づけ
られるかもしれない。

【利便性を追求する工夫】



材料費を少なくして作る
ことはできないかな？

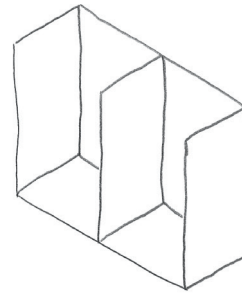
【経済性を追求する工夫】

じょうぶにするには、少なく
ともどのくらい板の厚さが必
要かな？

【安全性を追求する工夫】



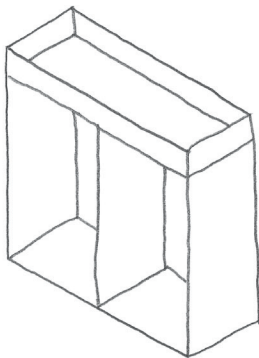
教科書を整理しないとね。
こういう本立てがあったら
いいな。



【利便性を追求する工夫】

机の上に二つは
置けないな。

**【機能性を追求
する工夫】**



これなら一つのもので
教科書が整理できて、
小物も置ける！
製作費を少なくして、
すぐに作れそうだ！

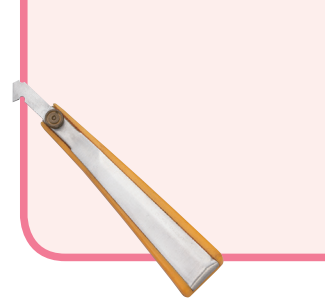


48-4



豆知識

製作品を決めるときは、思い浮かべるだけでなく、文字や数字、図などを使って表現したり、他の人からのアドバイスを受けてりしてまとめていくことが大切である。



■ 製品の機能

製品には，使用する用途や目的に応じて，機能が備わっています。製品の機能には，使いやすさや大きさ，置く場所に応じた利用，便利さ，美しさなどの要素があります。自分で製作をするときは，製品の機能を参考にしてみましょう。

26図 機能検討の例

置く場所に応じて

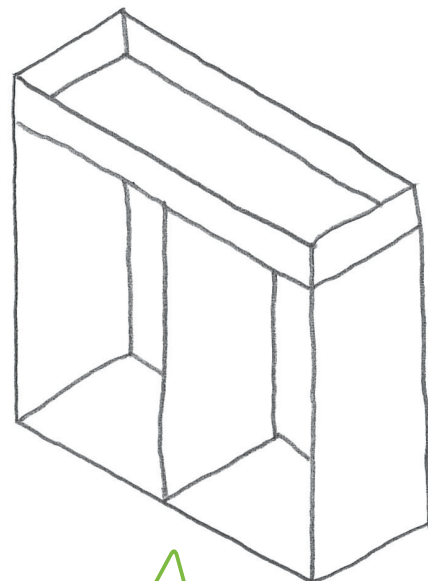
- 置く場所に応じた形や寸法を考える。
- 移動しやすいようにする。

便利さ

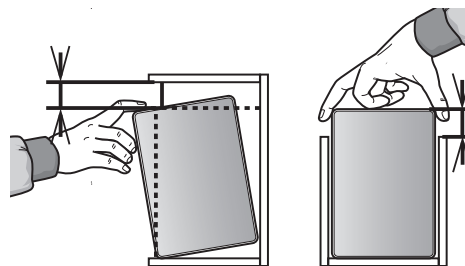
- 便利に利用できる。
- 他の用途にも使える。

美しさ (意ししょう性)

- 全体のバランスのよさ。
- 機能性に富んだ美しい形。
- 安全に配慮した形。

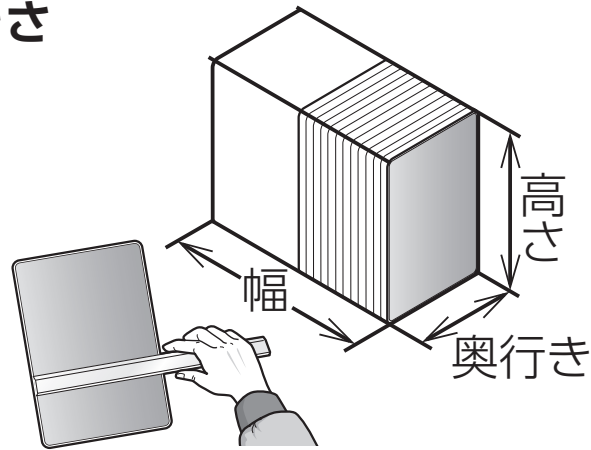


使いやすさ



- 収納するものが出し入れしやすい形や寸法を考える。
- 他の人でも使いやすい形を考える。

大きさ



- 収納するものの大きさを測る。
- 収納するものの大きさと数量から全体の大きさを決める。

どのような機能をもたせると、課題を解決することができるかな？

複数の機能をもった製品も考えられるね。



課題?

身の回りの製品には、さまざまな機能を取り入れた製品が多く存在します。本棚や家庭の整理棚を観察して、その製品と自分の作る製作品について、機能や構造、利用方法、環境性、安全性について観察してみよう。そして、このような製品のよいところを取り入れて、よりよい製作品になるように考えてみましょう。



このブックエンドは、多くの本を挟んでも安定しているだけでなく、^{いがた}鋳型で製作することで、曲線美など独創的な形状にしている。



豆知識

家具類は、①収納性（何をどれだけ入れるか）、②操作性（出し入れなどの使いやすさ）、③運搬性（持ち運びなどの移動）、④耐久性、⑤安全性などを考えて製作されている。

2-2. 計測・制御による問題解決

実習例 1 立入禁止エリアを判断する 金属回収ロボット



》問題の発見と課題の設定

①問題の発見

技術室に金属片が落ちていて危険だと思ったので、ロボットを使って分別・回収できるようにしたい。

《社会とのつながり》

- ・金属片を自動で回収してくれる【安全性，経済性】

②課題の設定

作品が置いてあるなど，立入禁止エリアを判断して入らないようなロボット掃除機を制作することにした。

》設計・制作

①全体の構想

- 障害物や立入禁止エリアをセンサで検知して，それらがあったら下がって回転して進むようにする。
- 使用するロボットを決め，動き方を考える。

【入力】・接触センサ：障害物との接触

・光センサ：床の明るさの変化

【処理】・マイコンボード：障害物や床の判定

【出力】・モータ（二つ）：前進，後退

②詳細設計とまとめ

- 使用するプログラミング言語の機能を確認した。
- システムの構成と，部品の取り付け位置を検討した。
- 処理の流れをUMLにまとめた。
- 設計したことを制作工程表にまとめた。

③制作

設計をもとに制作する。各部品や全体の動きを見ながらプログラムを修正した。

》評価・改善

①評価

制作したロボットを動かして，問題を解決できたかを評価した。

②改善

次のような課題が見つかった。

- ・ 同じ場所を回り続けてしまうことがある。
- ・ カーテンなど，接触センサが反応しない障害物も避けられるようにしたい。

スマートフォン



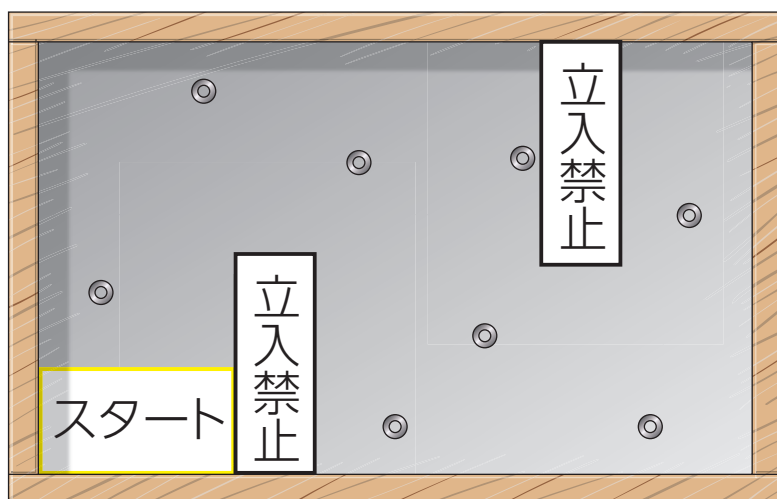
違うプログラミング
言語で表すとど
うなるかな？

```
void setup(){  
  aControl.init();  
  aControl.runwait();  
  
  while(!(aControl.getContactSwatchSt  
    {  
      aControl.controlMotor(1-2-3  
      aControl.controlMotor(2,1-3)
```



●課題や条件の確認

問題の場面を想定して，ここでは壁に囲まれたコースの中に散らばっている金属片（座金）を回収します。



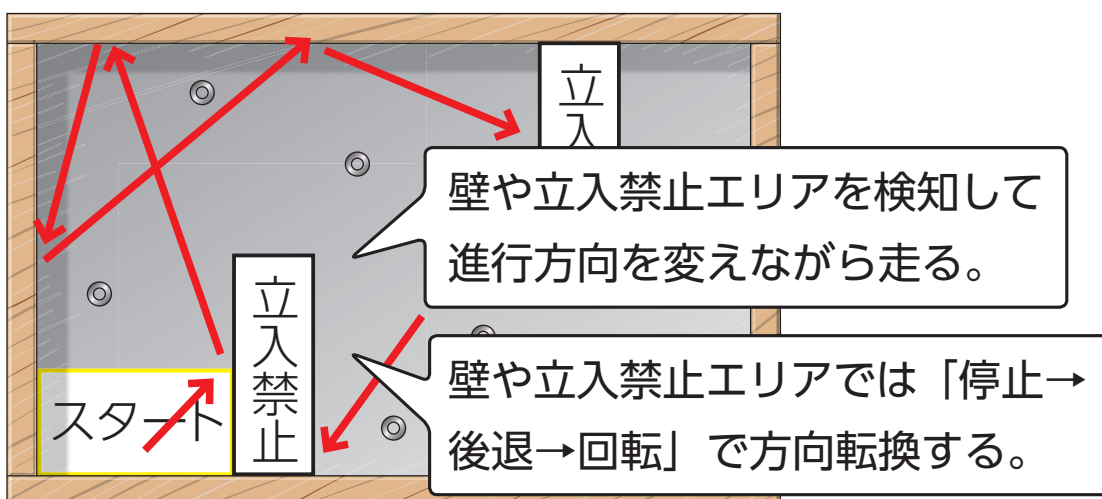
両面テープなどで固定



ロボットの底面や側面に磁石を取りつけて，コース上の座金を取れるようにします。

すべての座金を拾うには，どのようにロボットが動くといいかな？

●ロボットの動きの検討



動かし方によって効率やきれいさが変わってくるね。



プログラミング言語によって動かし方が異なるため、使用するプログラム言語の機能を確認します。

```

モータ( 右▼ , 後退▼ , 速度(0-200) 200 )
モータ( 左▼ , 後退▼ , 速度(0-200) 200 )
1 秒待つ
モータ( 右▼ , 前進▼ , 速度(0-200) 100 )
モータ( 左▼ , 後退▼ , 速度(0-200) 100 )
1 秒待つ
モータ( 右▼ , 前進▼ , 速度(0-200) 200 )
モータ( 左▼ , 前進▼ , 速度(0-200) 200 )
    
```

後退→左回転→前進する
プログラムの例

●ロボットと部品の確認

使用するロボットによって機能が異なるため、問題に応じて必要な動作ができるものを選択・制作します。



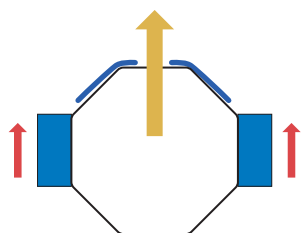
センサ

センサの動作を調べ、どのセンサを使えばよいか選択します。

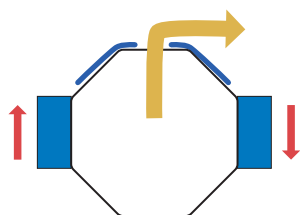
ここでは、障害物に接触センサを、立入禁止エリアに光センサを利用することにした。

アクチュエータなど

何秒動かすと何cm動くのか、何度回転するのかなど、制御する機器の性能を調べます。



[前進] 左右のモータを「前進」させる



[右回転] 左モータを「前進」、右モータを「後退」させる

二つのモータで動かす場合は、左右のモータの回転方向で動きを変える。



豆知識

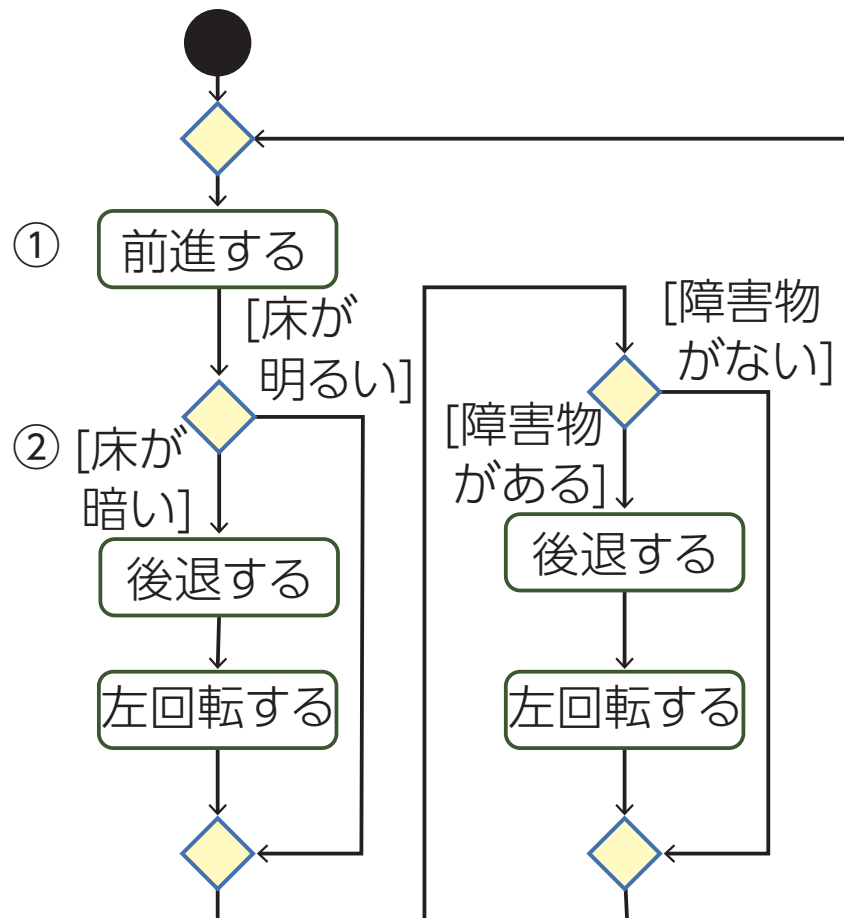
プログラムの中で、同じプログラムを何回も使うことがある。このようなとき、その部分に名前をつけて独立させ、呼び出して利用する。これをサブルーチンといい、プログラムを作成するときの基本的な技法の一つである。

8表 障害物や立入禁止エリアを見分ける金属回収ロボットの制作工程表

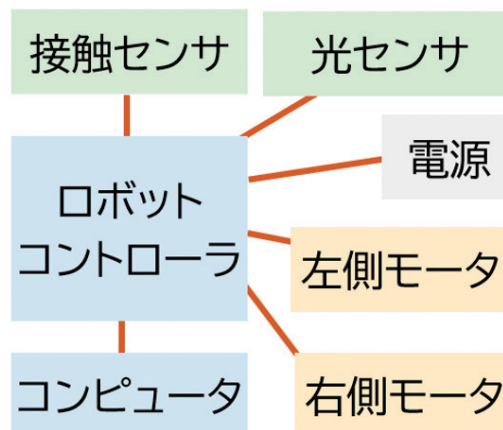
発見した問題	技術室に金属片が落ちていると危険なので、ロボットを使って分別・回収できるようにしたい。		
設定した課題	障害物や立入禁止エリアを避けて部屋に落ちている金属片を回収するロボットを制作する。		
課題を解決するために必要な仕様 (全体の構想)	入力	処理	出力
	<u>接触センサ</u> ・ 障害物との距離 光センサ ・ 床の明るさ	<u>ロボットコントローラ</u> ・ 障害物の判定 ・ 床の明るさの判定	<u>モータ (二つ)</u> ・ 方向転換

システム構成
と取り付け位
置およびアク
ティビティ図
(詳細の設計)

アクティビティ図



システム構成



253-5

86図 プログラムの例 QR



センサやアクチュエータの取り付け位置



＜接触センサ＞ 前向きに取り付ける

＜光センサ＞ 下向きに取り付ける

＜モータ＞ 左右に2個取り付ける



豆知識

プログラムの誤りを「バグ (bug)」という。コンピュータの登場前から使われていた言葉だが、初期のコンピュータで、実際に小さな虫（これをバグと呼ぶ）が機器に入り込むことで接続不良を起こしたことがあった。

