

Mission3 プログラミングでマイコンを制御してみよう！

【1】必要な機器やアプリ

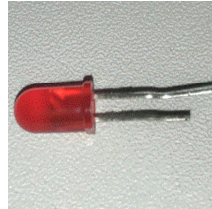
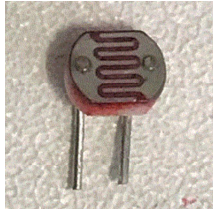
このパートでは、次のものが全て必要です。よく確かめて準備しよう。

○Google Chrome ブラウザ ※Safari等のブラウザではうまくいかないかもしれません。

○なでしこボード

○光導電セル(CdSセル)

○赤色LED

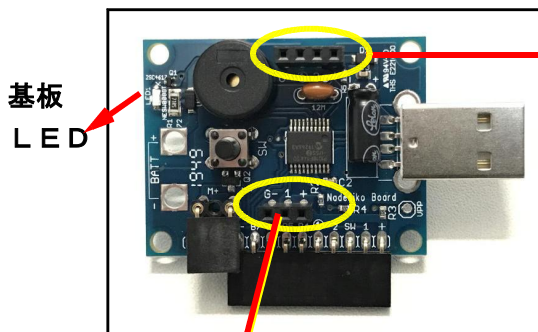


※LEDは部品の向きがあります。電流は長い足(アノード)から、短い足(カソード)に向けて流れます。

【2】なでしこボードの仕組みと準備

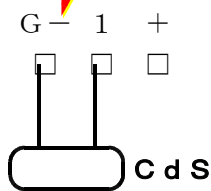
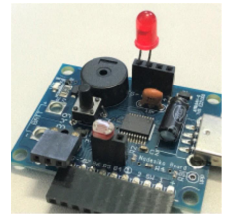
なでしこボードはPICマイコンの仲間です。センサの値を測る、LEDを光らせる、ブザーを鳴らす、といったことができます。

手順1 次の写真を見ながら、入力端子1に光導電セル(CdS)を、出力端子1に赤色LEDを取り付けてみよう。
(※この赤色LEDは、抵抗内蔵のものを使ってください)



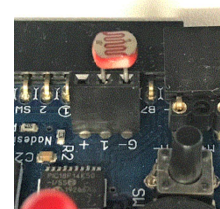
出力端子1

赤色LEDの長い足を「①」に、短い足を「G-」に差しこもう。部品の向きに気を付けて！

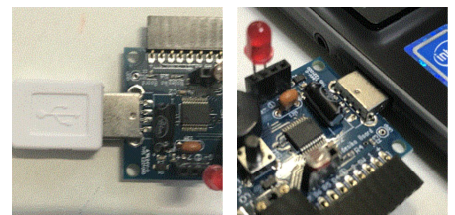


入力端子1

CdSの足を「G-」と「1」に差しこもう。部品の向きはありません。



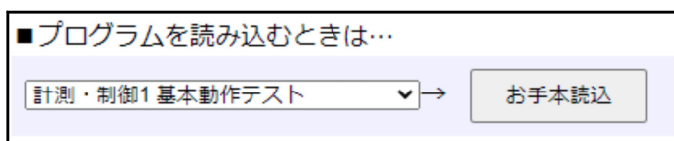
手順2 なでしこボードをコンピュータのUSBポートに接続しよう。「ピポッ♪」と音が鳴ったら準備OKだよ。



【3】使い方を覚えよう！

最初に、なでしこパッドでなでしこボードを使うやり方を覚えよう。お手本を開いて準備しよう。

手順1 お手本から「計測・制御1 基本動作テスト」を選ぼう



手順2 なでしこパッドとなでしこボードを接続しよう

なでしこパッドでは、WebHIDという技術を利用してなでしこボードと通信しています。そのため最初の1度だけ、なでしこボードを使うことをコンピュータ側へ伝える必要があります。

お手本にある、最初の2行で、ボードを使う準備をします。

ボード接続。
もし、ボード未接続ならば、終わる。

ボードをコンピュータへ接続してから初めて実行すると、
次のように「どの機器と通信したいか」を指定するように要求
されます。そこで「なでしこボード(Nadeshiko board)」を選んで
で **【接続】ボタン** をクリックします。

その後、プログラムは実行されますが、1回目は正しく動作しない
ことがあります（実行されずに終了することもあります）。



手順3 プログラムを実行しよう

接続がうまくいったら、[実行] ボタンを押すと、プログラムの4行目以降が処理されます。
プログラムを何度か実行してみよう。

- ・基板上の白いLEDが2回チカチカする（Lチカっていうやつだね！）
- ・その後、基板上のブザーが「ピポピポ♪」と鳴る
- ・その後、画面上に数値が4回表示される。基板に取り付けたCdSを手で隠して暗くすると、画面上の数値が変化するよ！

何度も実行して、動作を確かめてみよう！

手順4 使い終わったときは…

なでしこボードを使い終わったときは、ボードをゆっくりとコンピュータから抜き取ろう。
その時、金属部分を触らないように気を付けてね。

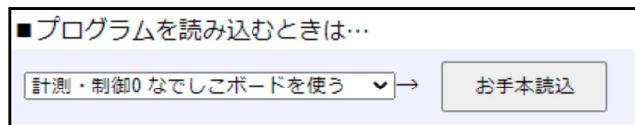
年 組 番 名前 ()

【4】なでしこボードを使うときは、ひな形をつかってみよう！

では、ここからはプログラムを自分で作ってみよう。

プログラムを作るとき、最初の2行はいつも同じになるから、あらかじめひな形を用意してあります。

なでしこパッドを開いたら、画面下方の「**■お手本を選ぶ**」というセレクトボックスから「**計測・制御0 なでしこボードを使う**」を選んで「**お手本読込**」ボタンをクリックしよう。



すると、次のようにプログラムのひな形が表示されるので、5行目からプログラムを入力していこう。

```
1 ボード接続。
2 もし、ボード未接続ならば、終わる。
3
4 // ↓ここからプログラムを入力しましょう。
5
6
```

【5】とりあえずLチカ！

最初はLEDをチカチカ光らせてみよう！ こういうプログラムを、プログラミングの世界では「Lチカ（エルチカ）」と呼んでいるよ。

課題1 基板のLEDをチカチカさせよう！

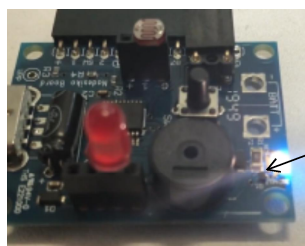
まず、基板にある白いLEDを光らせてみよう。

- ・点灯させる命令は **LEDオン**
- ・消灯させる命令は **LEDオフ**
- ・n秒待つ命令は **n秒待つ**

を使います。これを組み合わせて、LEDをチカチカさせてみよう。

ひな形の5行目から、次のプログラムを入力して、実行しよう！

```
4回
LEDオン。
1秒待つ。
LEDオフ。
1秒待つ。
ここまで。
```



基板のLEDが
Lチカしたかな？

課題2 赤色LEDをLチカさせよう！

次は、出力端子1に取り付けた赤色LEDをチカチカさせよう。

- ・このLEDを点灯させる命令は **出力1オン**
- ・このLEDを消灯させる命令は **出力1オフ**

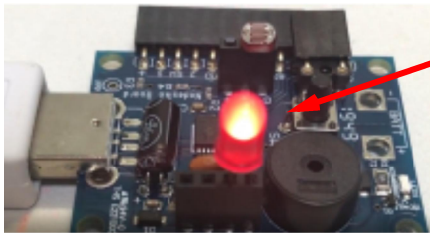
を使います。これを組み合わせて、LEDをチカチカさせてみよう。

ひな形の5行目から、次のプログラムを入力して、実行しよう！

```

4回
 出力1オン。
  1秒待つ。
 出力1オフ。
  1秒待つ。
ここまで。

```



赤色LEDがチカしたかな？

課題3 2つのLEDを交互に点灯させてみよう！

では、次の順に処理するプログラムを考えて作ってみよう。

- ・ 白色LED…●消灯，赤色LED…●消灯 で1秒待つ
- ・ 白色LED…○点灯，赤色LED…●消灯 で1秒待つ
- ・ 白色LED…○点灯，赤色LED…○点灯 で1秒待つ
- ・ 白色LED…●消灯，赤色LED…○点灯 で1秒待つ
- ・ これを4回繰り返す。

★このプログラムを自分で考えて作ってみよう。先生にチェックしてもらおう。

【6】ブザーを鳴らそう！

課題1 とりあえずピーッ♪

ボード上にあるブザーを鳴らしてみよう。とりあえず，ピーッ♪と1回だけ鳴らしてみよう。

```

発音。

```

実行したら，ボードからピーッ♪と音が鳴ったかな？

課題2 音の長さを変えてみよう！

次に，音の長さを変えてみよう。

```

0.5秒発音。

```

こんなふうに，秒数を指定するとその長さでブザーが鳴るよ（但し最長で2秒まで）。単に**発音**と書いたときは，**0.5秒発音**と同じ意味になるよ。

課題3 音の高さを変えてみよう！

音の高さも変えられるよ。音の高さは，0～22の範囲で指定できるよ。

```

11を発音。 // シの音
14を0.2秒発音。 // レの音

```

- 0 2 4 5 7 9 11 12 14 16 17 19 21 22
- ド レ ミ ファ ソ ラ シ ド レ ミ ファ ソ ラ シ

数値を変えて，色んな音を出してみよう。

★「お手本」から，「計測・制御1.1」や「計測・制御1.2」を開いて，音を鳴らしてみよう！

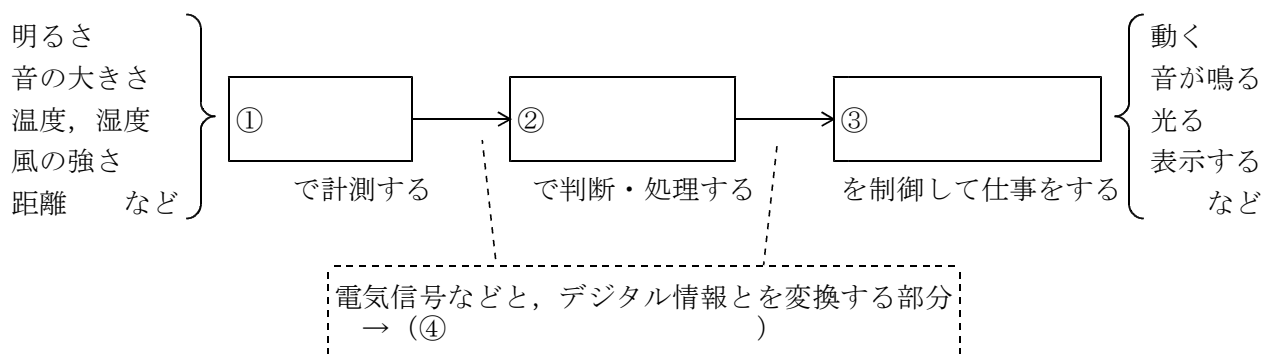
★かんたんな電子メロディをつくってみよう！

年 組 番 名前 ()

【7】 センサで明るさを測ってみよう

課題1 計測・制御システムの仕組みを知ろう

周囲の状況を計測して、その結果を利用して機器を制御する仕組みを**計測・制御システム**といいます。



課題2 CdSで明るさを測ろう

センサとは、周囲の明るさや温度などを測る部品や装置のことです。なでしこボードでは、センサが測った明るさや温度などを、数値で扱うことができます。

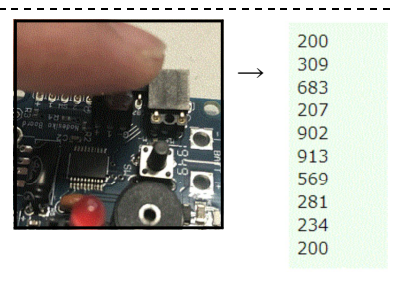
CdS(光導電セル)は、周囲の明るさによって抵抗値が変化する部品です。これを利用して、周囲の明るさを測ることができます。なでしこボードは**明るいとき最小値0**、**暗いとき最大値1023**で測定値を返します。

ボードの入力端子1に**CdS**を差しこんで、**センサ1測定**という命令を使うと、CdSの測定値が**センサ1**という変数に格納されます。

ひな形の5行目から、次のプログラムを入力して、実行しよう！

```
10回  
  センサ1測定。 ← 「センサ1」という変数に測定値が入る  
  センサ1を表示。  
  1秒待つ。  
ここまで。
```

プログラムを実行すると、画面に測定値（センサ1という変数の値）が表示されます。自分の手でCdSをおおって暗くすると、どんな値になるか確かめよう。



課題3 センサライトをつくってみよう！

もし～ならば～違えば～ここまで を使って、センサライトの動作を再現してみよう。

- ・周囲が暗いときは、自動的にLEDオン（点灯）する。
- ・周囲が明るいときは、自動的にLEDオフ（消灯）する。
- ・これを繰り返す

市販のセンサライトと全く同じ動作をつくるのは難しいから、基本的な動作だけ再現してみよう。

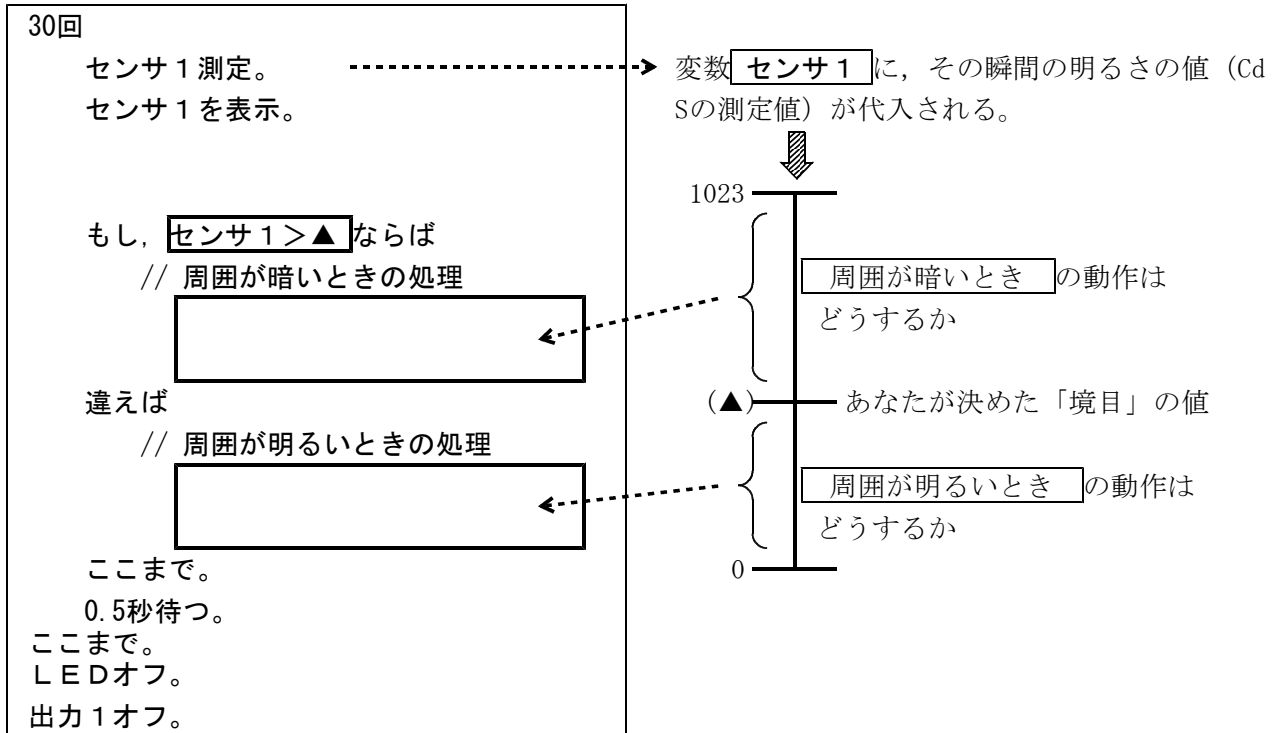
手順1 お手本から「計測・制御2 もし~ならば~違えば」を選ぼう

■プログラムを読み込むときは…

計測・制御2 もし~ならば~違えば

お手本読み込

手順2 分岐処理をつくって、実行してみよう！



★ヒント！

「周囲が暗いとき」の条件式は センサ1 > 600ならば のように不等号を使って書けるよ。

★覚えよう！

コンピュータを利用して自動で動作している機器は、次のような仕組みで構成されています。

入力： 周囲の明るさや温度などを (①) で計測する。

判断： ①で計測した値を利用して、コンピュータ(プログラム)で判断・処理する。

出力： その結果に従って、モータやLEDなどの (②) を制御する。

このように、センサからの入力 → コンピュータでの判断・処理 → アクチュエータへの出力という流れで構成される仕組みを、(③) と呼びます。

答え ①：センサ ②：アクチュエータ ③：計測・制御システム

課題4 <チャレンジ!> 冷蔵庫の動作を再現してみよう!

課題3のプログラムを改良・応用して、次のような冷蔵庫の動作を再現してみよう。自分で考えて、プログラムを作ってみよう!

①【技能】

冷蔵庫は、扉を開けると中が明るくなりますよね。それと同じように、明るいときは「LEDオン」で、暗いときは「LEDオフ」に改良してみよう。



②【思考】

扉が開いているときに、音が鳴っていれば、閉め忘れを防ぐことができます。明るいときは「LEDオン」で「音が鳴る」ように改良してみよう。



年 組 番 名前 (_____)

【8】3通りに場合分けしてみよう！

分岐（もし～ならば～違えばもし～違えば）を使うと、より細かく場合分けすることができます。

★条件式のかき方を理解しよう！

- もし暗かったら とか もし暗いならば とかいても、なでしこは理解できない。
 ↓
 「明るいとき」と「暗い時」の境目の値を(例えば800と)決めて、
 もし、センサ1 > 800ならば と数式でかけば、なでしこが理解できるのだ！

★次のかき方を理解しよう！

- もし、(条件式1)ならば
 (条件式1にあてはまるときの処理)
 違えばもし、(条件式2)ならば
 (条件式1にあてはまらないが、条件式2にあてはまるときの処理)
 違えば
 (条件式1と条件式2の、どちらにもあてはまらないときの処理)
 ここまで。

課題1 3つの場合に分ける分岐処理をつくってみよう

30回
 センサ1測定。 -----> 変数 **センサ1** に、明るさの値が入る
 センサ1を表示。

もし、 **センサ1 > ▲** ならば
 // とても暗いときの処理

違えばもし、 **センサ1 > ●** ならば
 // やや暗いときの処理

違えば
 // 明るいときの処理

ここまで。
 0.5秒待つ。
 ここまで。
 LEDオフ。
 出力1オフ。

1023

とても暗いとき はどうするか

(▲) あなたが決めた「境目」の値

やや暗いとき はどうするか

(●) あなたが決めた「境目」の値

明るいとき はどうするか

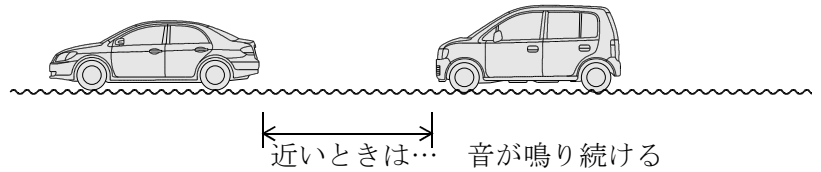
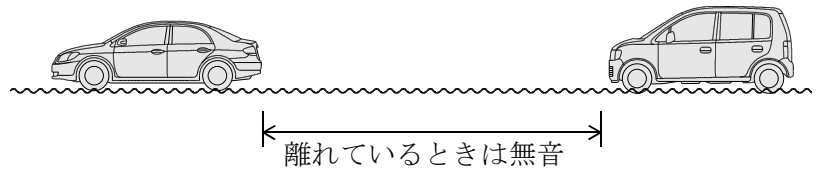
0

年 組 番 名前 ()

【9】衝突防止アラームを作ってみよう！

なでしこボードは別の種類のセンサも使えますが、CdSが一番単純で分かりやすいので、これを応用して、自動車の衝突防止アラームを作ってみよう。

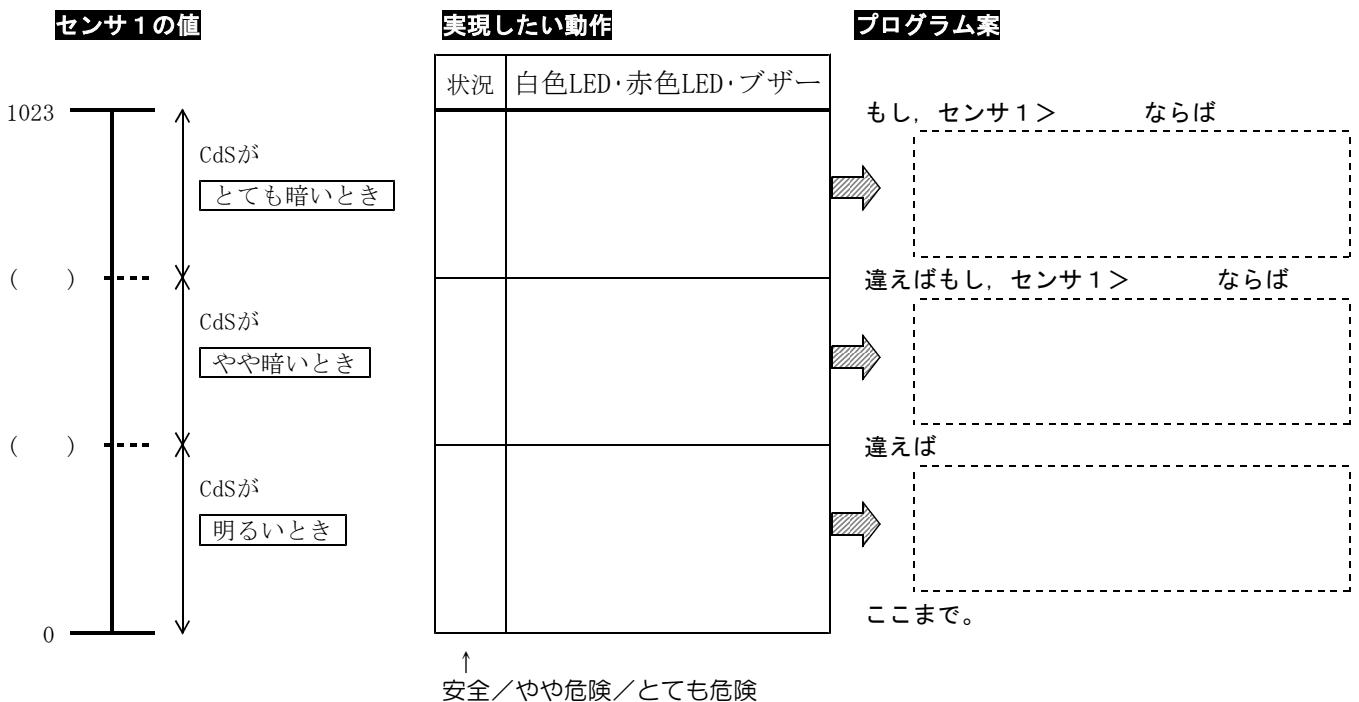
2台の車が離れているかどうかは、実際には距離センサなどで測りますが、今回はCdSで代用しましょう。



課題1 動作を場合分けして考えてみよう

このようなプログラムを考えるときは、動作を**場合分け**して構想を具体化します。次の図を使って、あなたが実現したい動作を考えてみよう。

<動作を3通りで考える場合>



課題2 プログラムをつくって、試してみよう！

大まかな構想が決まったら、次はそれぞれの場合の処理を考えてみよう。

- ・「音が鳴る」ようにしたいときは … ブザーを鳴らす命令 **aをn秒発音** を使ってみよう。
- ・「LEDが光る」ようにしたいときは … **LEDオン**や**LEDオフ**、**出力1オン**や**出力1オフ** を使ってみよう。
- ・ **安全なとき** … 何もしないから、何も書かなくていい！…とっていませんか？
ブザーも鳴らないし、LEDも光らないなら、どんな命令を入れたらいいかな？
n秒待つを上手に使ってみよう。

とにかく、何度も実行して、何度も直してみよう！

このプログラムは、全員が同じプログラムになるわけではなく、一人一人が異なるプログラムを完成させると思います。だからこそ、自分で何度もプログラムを書いて、何度も実行してみよう。エラーが出たら、どこが間違っているかを見つけて、直してみよう。

他の人にプログラムを見てもらおうと、よりよい動作に近づけるよ！ あきらめずに、粘り強く頑張ろう！

年 組 番 名前 ()
