ゆめほたる環境科学技術塾

micro:bitプログラミング

関数

ゆめほたる環境科学技術クラブ

はじめに

今回は、演習で信号機プログラムをつくります。

先ほどつくった「LED信号機」をマイクロビットとつないでください。

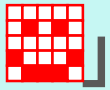
- LED信号機のグランドーマイクロビットの「GND」
- LED信号機の緑ーマイクロビットの「1」
- LED信号機の赤ーマイクロビットの「2」

「高度なブロック」＞「入出力端子」＞「デジタルで出力する」でLED信号機を点灯・消灯できます。

- 緑を点灯：端子「P1」 値「1」、緑を消灯：端子「P1」 値「0」
- 赤を点灯：端子「P2」 値「1」、赤を消灯：端子「P2」 値「0」

はじめに（つづき）

「LED信号機」をつくっていない人は、

- 「緑を点灯」のかわりに「えがおアイコン  」を表示
- 「赤を点灯」のかわりに「ねがおアイコン  」を表示
- 「点めつ」のかわりに「ハートアイコン  」を点めつ

させてください。

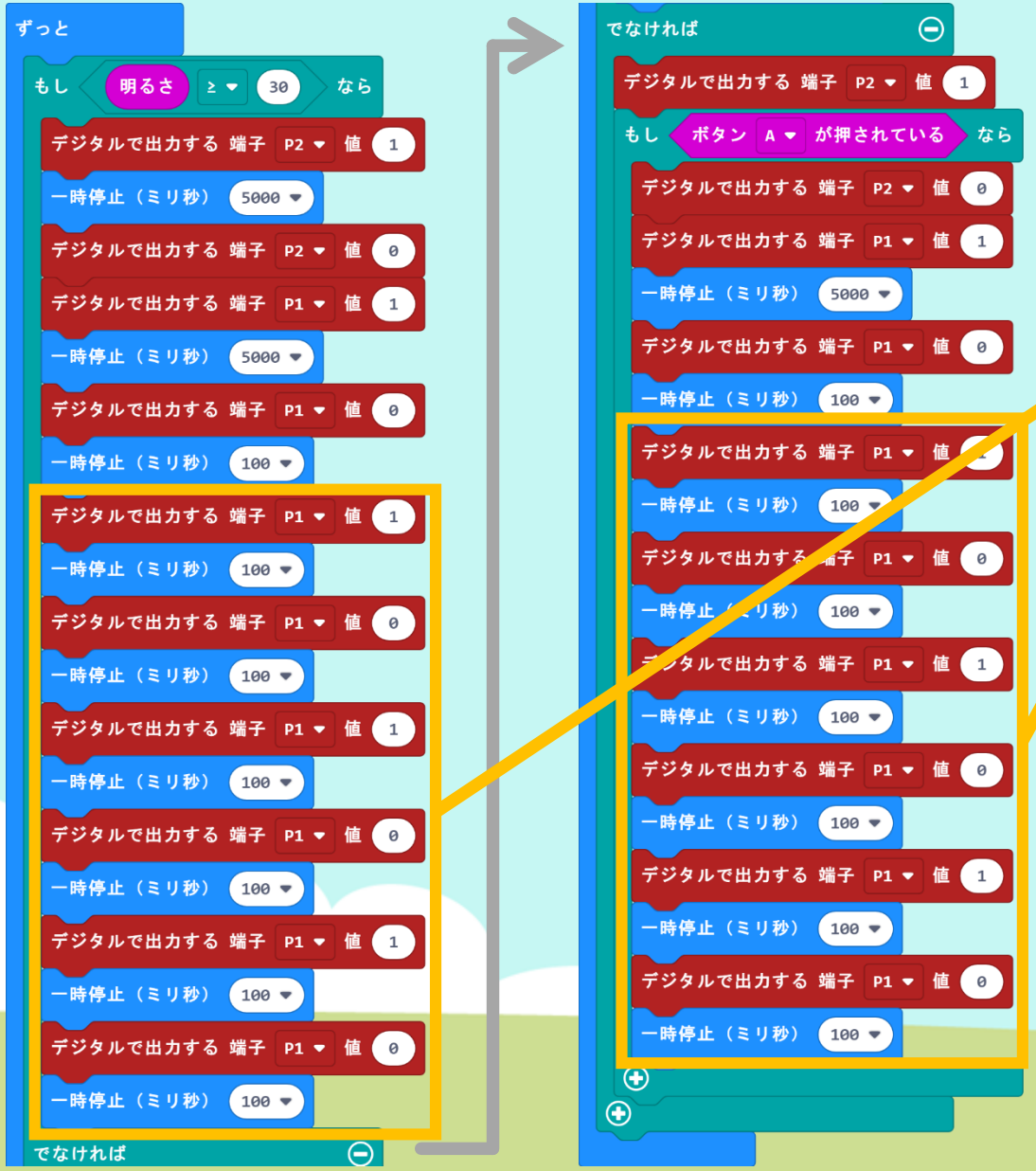
夜間押しボタン式信号をつくろう

昼は赤と青が交互に、夜はボタンをおしたときだけ青になる信号をつくります。

- 昼（明るさが30以上）なら、「赤LED（ねがお）を5秒表示」→「緑LED（えがお）を5秒表示」→「点めつ」をくりかえし。
- 夜なら「赤LED（ねがお）を表示」。
- 夜にAボタンがおされたら、「緑LED（えがお）を5秒表示」→「点めつ」のあと、「赤LED（ねがお）を表示」にもどる。
- 「点めつ」は「緑LED（ハート）を0.1秒表示」→「緑LED（ハート）を0.1秒消す」を3回くりかえし

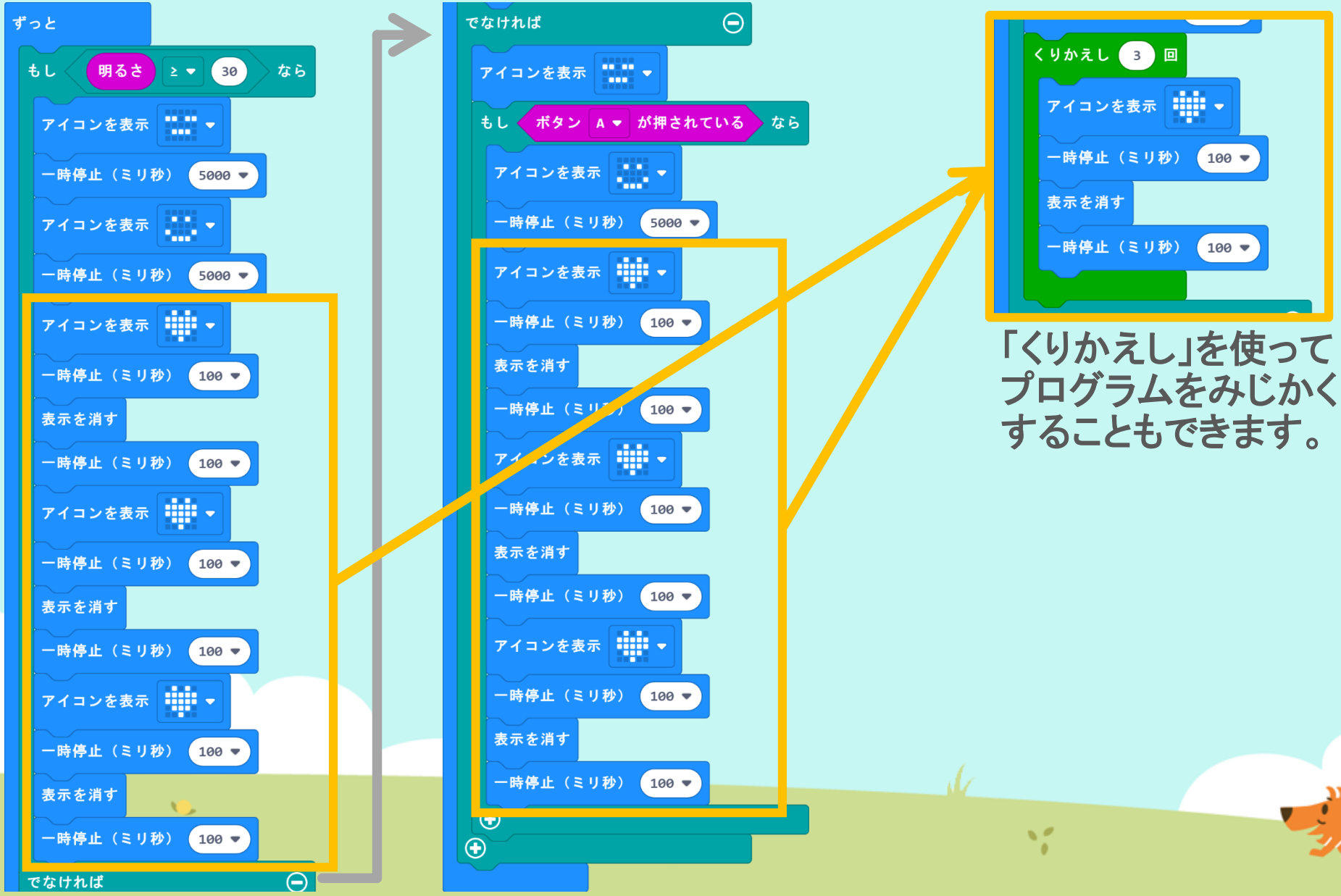
マイクロビットに書きこんでください

作成例～LED信号機の場合



「くりかえし」を使って
プログラムをみじかく
することもできます。

作成例～アイコンの場合



「くりかえし」を使って
プログラムをみじかく
することもできます。

改良プログラム（「関数」をつかう）

「点めつ」の時間がみじかすぎるので、「0.1秒」を適当な時間に変更してください。

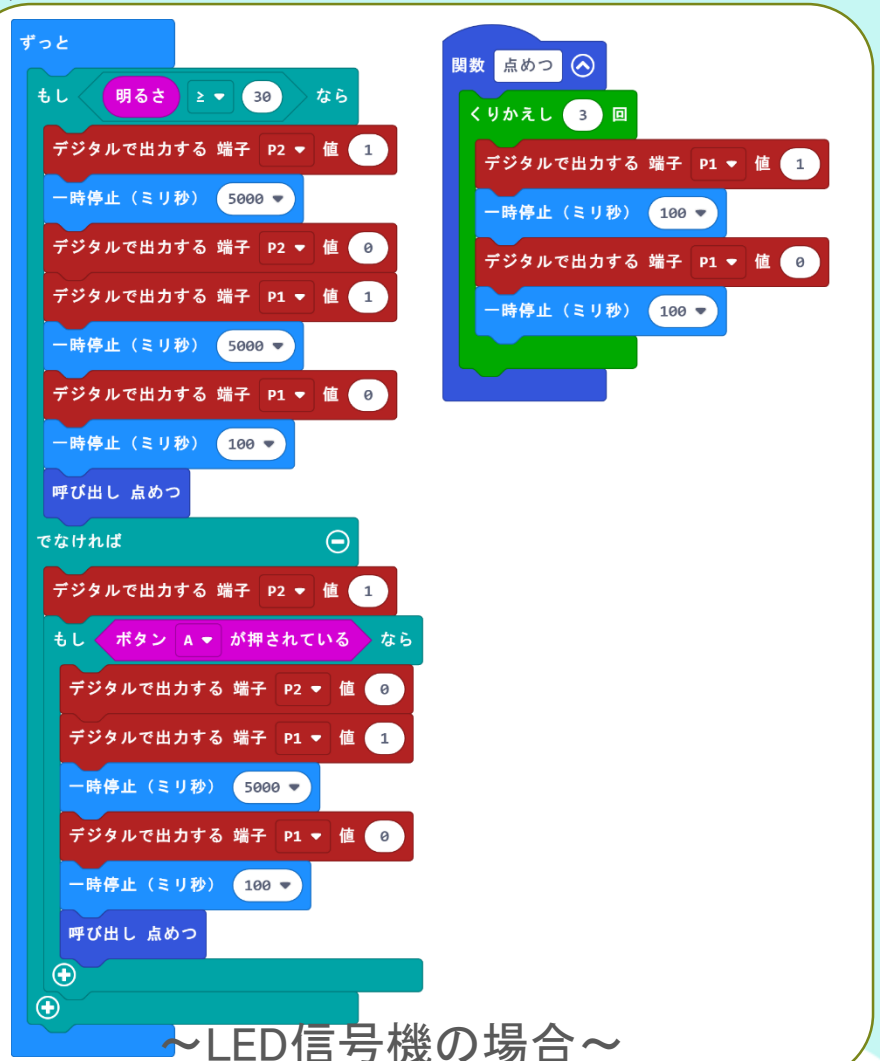
変更したのは何カ所でしたか？「くりかえし」を使わない場合で6カ所、使った場合でも2カ所変更したはずです。

このように、いろいろなところで同じ処理をする場合、「**関数**」を使うと1カ所にまとめることができます。

- 「高度なブロック」＞「関数」の「関数を作成する」をクリックし、「doSomething」と書いてあるところを「点めつ」と書きなおして「完了」をクリックします。
- 「関数 点めつ」ブロックがあらわれるので、その中に点めつの処理をつくります。
- 点めつさせたい場所に、「高度なブロック」＞「関数」の「呼び出し 点めつ」を置きます。

マイクロビットに書きこんでください

改良プログラム（関数）－ 作成例



～LED信号機の場合～

～アイコンの場合～

長かったプログラムがこんなにスッキリしました！

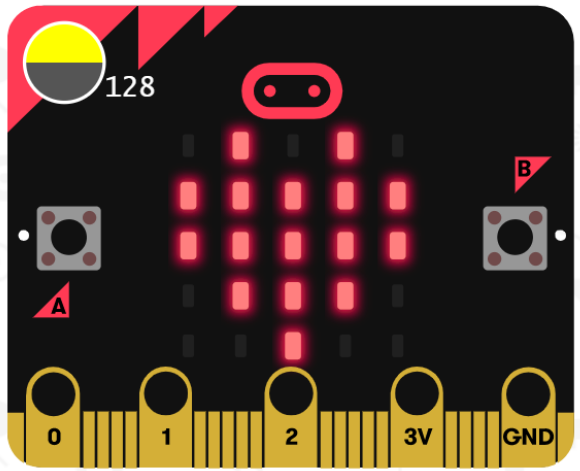
まとめ

ここまでやってきたことを組み合わせると、いろいろな複雑なプログラムもつくることができます。

ただ、みなさんの中には「ここで習ってきたのは子供向けのかんたんなプログラムで、おとながやっているプログラムはもっとむずかしいはず」と思っている人もいるかもしれません。

先ほどつくったプログラムの画面の上の方に「ブロック」「JavaScript」と書いてあります。この「JavaScript」をクリックしてみてください。

まとめ



エクスプローラー

検索...

- 基本
- Input
- 音楽
- LED
- 無線
- ループ
- 論理
- 変数
- 計算
- 高度なブロック

```
function 点めつ () {
    for (let index = 0; index < 3; index++) {
        basic.showIcon(IconNames.Heart)
        basic.pause(100)
        basic.clearScreen()
        basic.pause(500)
    }
}
basic.forever(function () {
    if (input.lightLevel() >= 30) {
        basic.showIcon(IconNames.Asleep)
        basic.pause(5000)
        basic.showIcon(IconNames.Happy)
        basic.pause(5000)
        点めつ()
    } else {
        basic.showIcon(IconNames.Asleep)
        if (input.buttonIsPressed(Button.A)) {
            basic.showIcon(IconNames.Happy)
            basic.pause(5000)
            点めつ()
        }
    }
})
```

ダウンロード

signal

まとめ

実はみなさんは、このようなコードでプログラムをつくっていました。

MakeCodeエディタが、このコードを分かりやすいようにブロックのかたちで表示してくれていただけです。

つまり、みなさんがつくったプログラムも、おとなの人がつくっているプログラムも、実は同じようなものです。

まとめ

高校の「情報Ⅰ（プログラミング）」での学習内容はこちらです。

（イ）

アルゴリズムと
プログラミング

(1) 外部装置との接続
計測・制御，センサ，アクチュエータ，計測・制御プログラム
(2) 基本的プログラム
アルゴリズム，プログラム，フローチャート，順次・分岐・反復，変数
(3) 応用的プログラム
配列，乱数，関数，WebAPI
(4) アルゴリズムの比較
探索アルゴリズムの比較，ソートアルゴリズムの比較

【出典】文部科学省 高等学校情報科「情報Ⅰ」教員研修用教材
（https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1416756.htm）

まとめ

- 順次：書いた順番に作業すること。
- 分岐：条件にあっているかどうかで作業内容を変えること（もし～なら）。
- 反復：ある作業を何度もくりかえすこと。
- アルゴリズム：「順次」「分岐」「反復」をくみあわせてつくった作業手順。
- プログラム：コンピュータに作業を行わせるための手順を書いたもの。
- 変数：数値や文字列の値などを入れておく「ハコ」のようなもの。
- 関数：いくつかの作業をまとめたもの。

どれも、みなさんがすでに学習したものです。

まとめ

けっこう「すごい」ことをやってきたことがわかると思います。

（イ）

アルゴリズムと
プログラミング

（1）外部装置との接続
計測・制御，センサ，アクチュエータ，計測・制御プログラム
（2）基本的プログラム
アルゴリズム，プログラム，フローチャート，順次・分岐・反復，変数
（3）応用的プログラム
配列，乱数，関数，WebAPI
（4）アルゴリズムの比較
探索アルゴリズムの比較，ソートアルゴリズムの比較

ゆめほたる環境科学技術塾

micro:bitプログラミング

おわり

ゆめほたる環境科学技術クラブ