

秋季プログラミング教室

令和5年（2023年）10月1日／山口市大殿地域交流センター

- 主催・共催／講師
 - － 山口-UKコンピュータープログラミング協会
 - 会長 弘中 富士彦 先生
 - － 山口大学教育学部
 - 情報教育コース 野村厚志
- 協賛：公益財団法人 マツダ財団

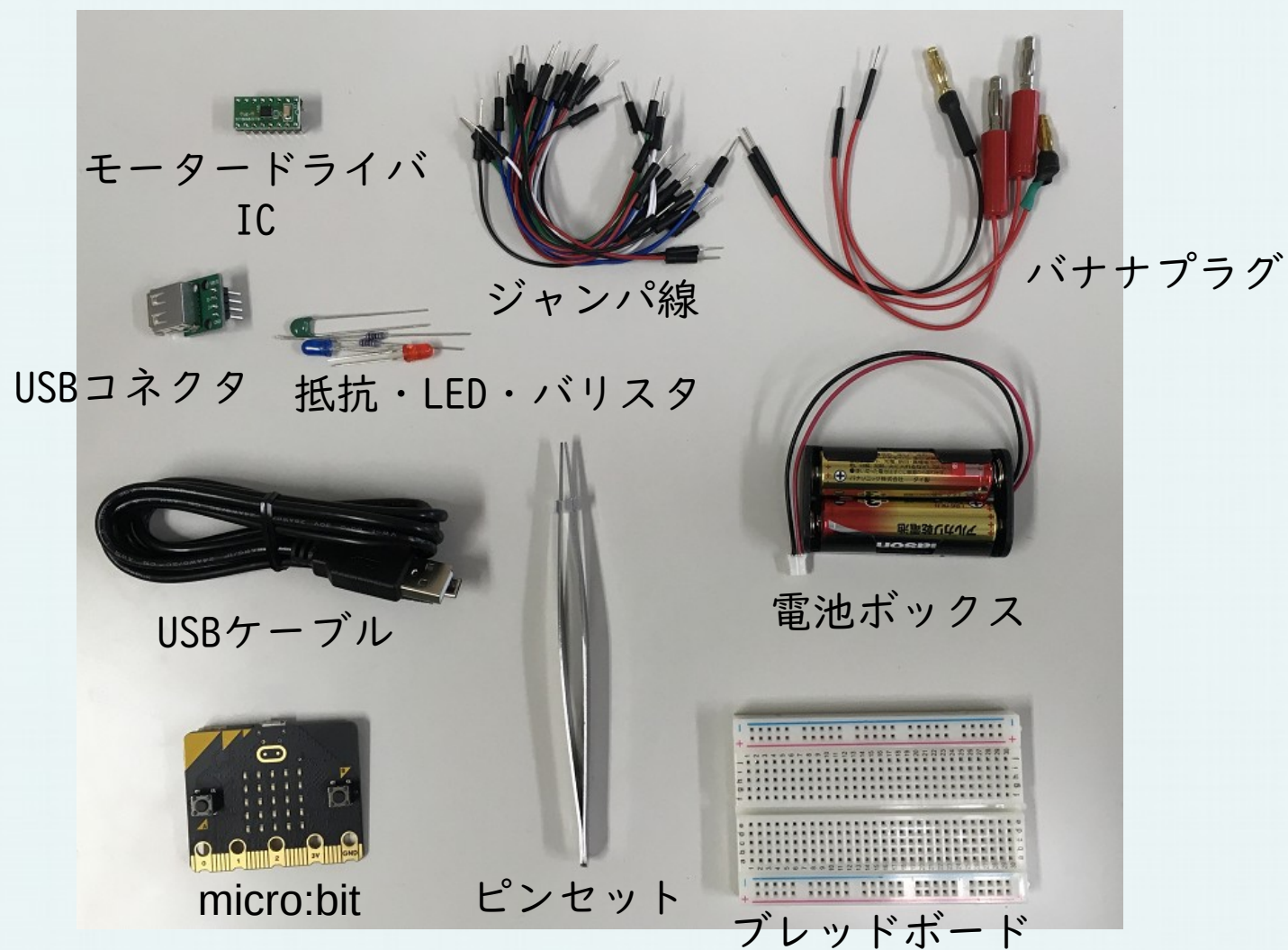


本日の予定

- 開会行事
- 演習 1 (13:40-14:00)
makecodeによるプログラミング
ブレッドボードの使い方
- 演習 2 (14:00-15:00)
リレーをmicro:bitで制御する
- 演習 3 (15:10-16:00)
モーターをmicro:bitで制御する
- 閉会行事 (16:00-16:30)
I2Cインターフェースデモ



配付物

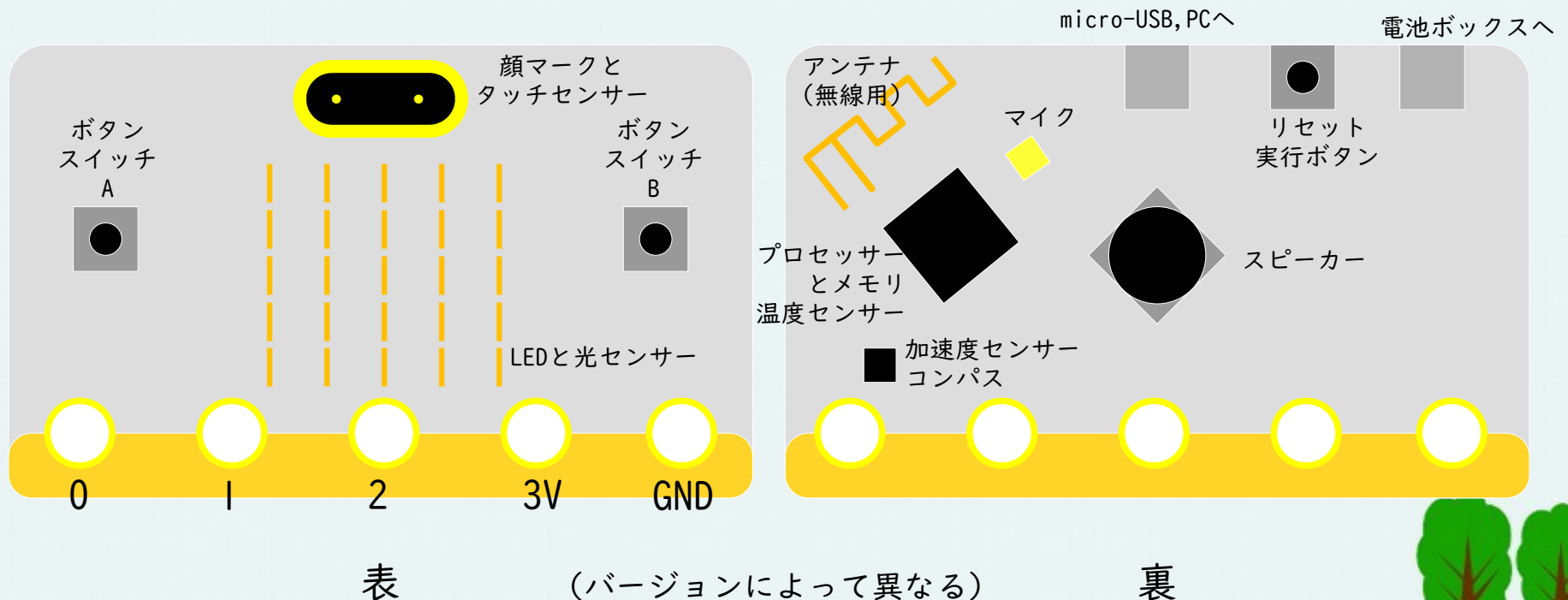




演習 1. makecodeによるプログラミング

演習1-1. 「micro:bit」の装置確認

- 「micro:bit」は小さなコンピュータ
- 公式website → <https://microbit.org/ja/>
 - 「はじめよう」>「ユーザーガイド」>「概要」に写真と解説
 - 装置の詳細→ <https://tech.microbit.org/> > 「Hardware」



演習1-2. プログラム作成の準備

インターネット版, Windows・Mac・Chromebook 用

1. Microsoft社の「makecode」のサイトを利用

- <https://makecode.microbit.org/>
- micro:bitのサイト>プログラムしよう

2. 「新しいプロジェクト」をクリック

- プロジェクト名「**test**」と入力
- 「作成」をクリック

プロジェクトを作成する 

プロジェクトに名前をつけてください。

[> コードのオプション](#) 作成 

演習1-3. プログラムの作成と実行

• 画面の様子

ここから説明を表示することもできる。

ブロック

?

実行結果が再現される

(無線を使うと2台めが出てくる)

基本

入力

音楽

LED

無線

ループ

論理

変数

計算

拡張機能

高度なブロック

最初だけ

アイコンを表示

- この灰色領域内でブロックを組み合わせてプログラムを作成する
- 左側のメニュー「基本」～「高度なブロック」をクリックして表示されるブロックを選ぶと、ここに配置される。
- ドラッグ&ドロップで組み合わせていく。
- 不要となったブロックは左側のメニュー領域にドラッグ&ドロップ

ダウンロード

test

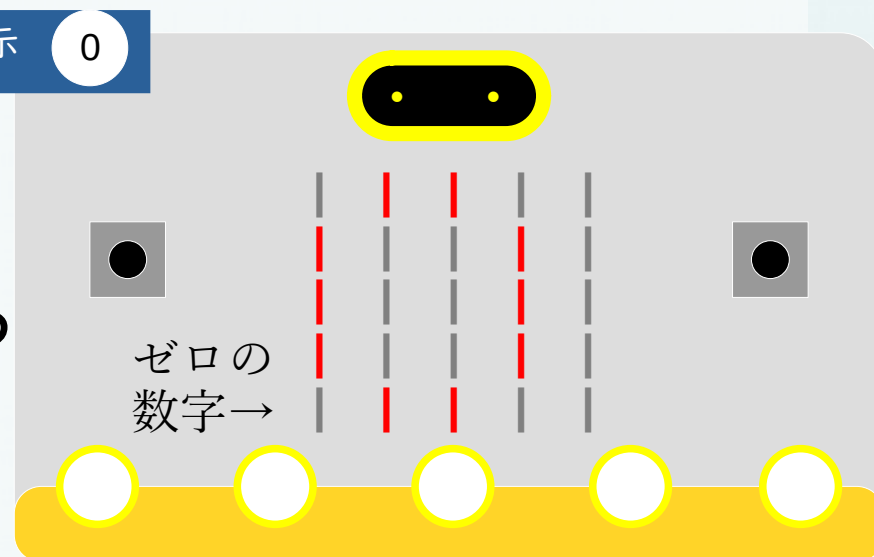
演習1-4. プログラミングの基礎

- 順次：上の方のブロックから実行すること
- 分岐（場合分け）：条件に応じて実行するブロックを変えること。「〇〇なら□□を行う」
- 反復（繰り返し）：ブロックを何度も実行すること。10回繰り返すとか、ある条件が当てはまる間、繰り返すとか。



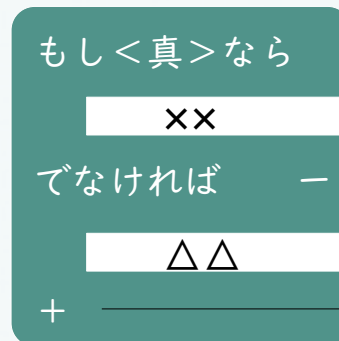
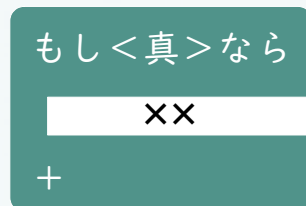
演習1-5. 順次（上から順に実行）

- 各ブロックは、コンピュータで計算・実行できる「機能」と関係づけられている。
- 「基本」メニューのブロックの「機能」の例：
 - LEDに「数を表示」
 - LEDの「表示を消す」
- ブロックを上下に連結することで機能を順番に組み合わせることができる。

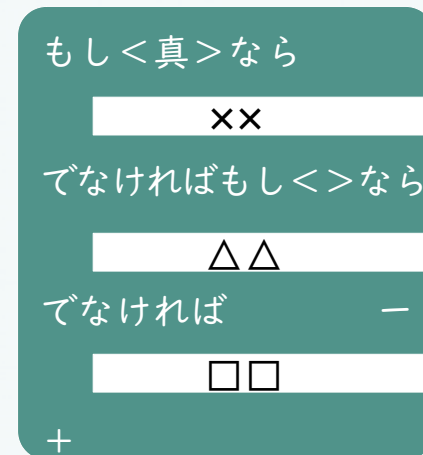


演習1-6. 分岐（場合分け）

- 「もし〇〇なら××を行う」
- 「もし〇〇なら××を行い、
でなければ△△を行う」
- ブロック：「論理」→「条件判断」



→
+をクリックすると
場合分けのブロック
が増える



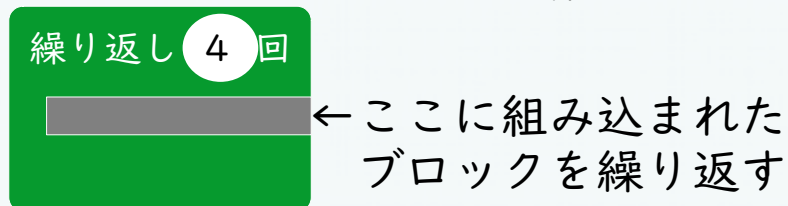
補足：分岐の「もし〇〇なら」の「〇〇」で場合分けのための条件を指定します。条件の指定方法は後で説明します。



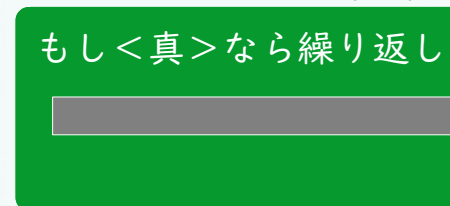
演習1-7. 反復（繰り返し）

- 機能を何度も（複数回）繰り返すこと。
- 「反復の回数」を指定する場合と、
「反復の条件」を指定する場合とがある。

↓ここで回数を指定



↓ここで条件を指定



補足1：反復の「条件」の指定方法は、次で説明します。

補足2：「ずっと」のブロックも「ずっと繰り返す」という反復になります。

演習1-8. 分岐や反復における条件の指定方法

- 記号で結ばれた式の左辺と右辺の2つの数や変数の値を比べて条件が満たされているか調べる方法

記号	意味	例
$=$	左辺と右辺が等しい	$i = 2$
$\neq$	左辺と右辺が等しくない	$i \neq 2$
$<$	左辺が右辺より小さい	$i < 2$
$\leq$	左辺が右辺以下	$i \leq 2$
$>$	左辺が右辺より大きい	$i > 2$
$\geq$	左辺が右辺以上	$i \geq 2$



演習1-9. 条件を組み合わせる指定方法 (かつ・または・～ではない)

- 2つの条件を「かつ」や「または」の言葉（論理語）で組み合わせ、全体として条件が満たされているか調べる方法
- 「～ではない」は1つの条件を否定

論理語 (演算名)	意味	例
かつ (論理積)	2つの条件が 両方とも満た される場合	$i=2$ かつ $j=3$
または (論理和)	2つの条件の うち、少なく とも1つは満 たされる場合	$i=2$ または $j=3$
ではない (否定)	ある条件が満 たされない場 合	$i=2$ ではない (つまり i は2以 外の場合)



演習1-10. 順次のプログラム例

- 「基本」のメニューから
「数を表示○」
「一時停止（ミリ秒）」
- 実行すると、LEDに数字の
1, 2, 3が順に1秒ごとに
表示される。

最初だけ

数を表示 1

一時停止（ミリ秒）1000

数を表示 2

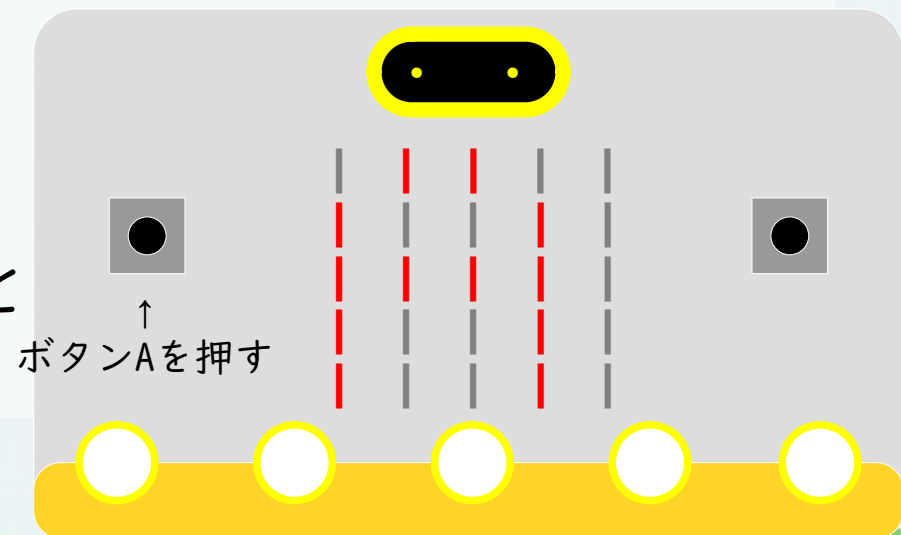
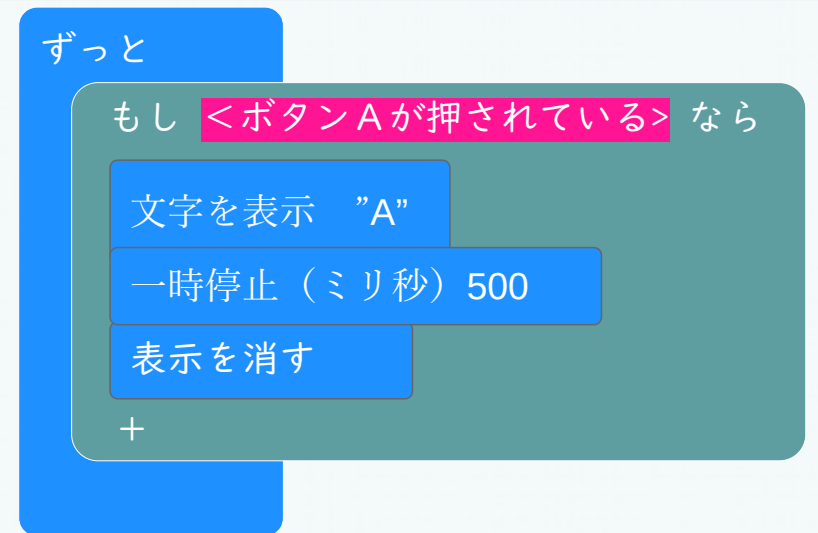
一時停止（ミリ秒）1000

数を表示 3



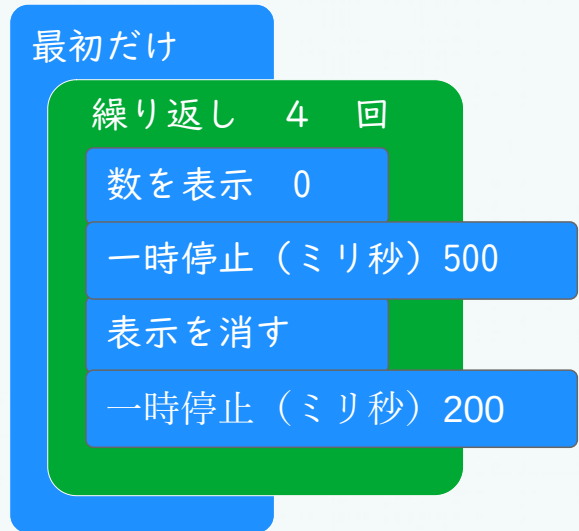
演習1-11. 分岐のプログラム例

- 「基本」のメニューから
 - 「文字を表示」
 - 「一時停止XX」
 - 「表示を消す」
- 「論理」
 - 「もし<真>なら」
- 「入力」
 - 「ボタンAが押されている」
(これを<真>のところにを入れる)
- 実行した後に、ボタンAを押すとLEDで「A」の形が表示



演習1-12. 反復のプログラム例

- 「基本」のメニューから
「数を表示 0」
「一時停止（ミリ秒）」
「表示を消す」
- 「ループ」
「くりかえし 4 回」
- 実行すると数字の「0」が4回点滅する。



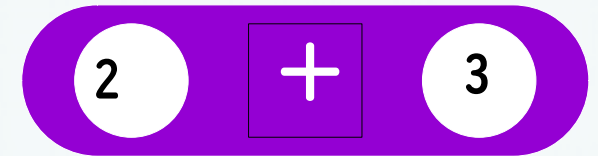
演習1-13. 定数と変数, 変数への代入

- 定数：
 - 具体的な数：1, 2, 3
 - 具体的な文字列：“Hello”
- 変数：
 - 計算に使う数や計算結果を覚えておくところ
 - 変数には自分で名前をつける：
回数, 個数, n, x, y など



演習1-14. 計算・算術演算

- 定数や変数を用いて計算
 - 四則演算： $+$ $-$ $\times$ $\div$
 - 平方根
 - 四捨五入、切り捨て・切り上げ
 - 乱数
- プログラムを実行することにより、コンピュータの内部状態は変化していく。
 - プログラムは、コンピュータの内部状態を変化させること（記憶装置に対する操作）によって、目的の処理を行うものとも考えることもできる。



演習1-15. 変数, 演算と分岐を用いたプログラム例

- 変数「n」に1~9の整数乱数を代入
- 分岐：
 - もし<nが偶数>なら、LEDに「o」を表示
 - そうでなければ、LEDに「x」を表示

• プログラム例

最初だけ

変数nを1から9までの乱数にする

もし<nを2で割ったあまり=0>なら

文字列を表示 "o"

でなければ

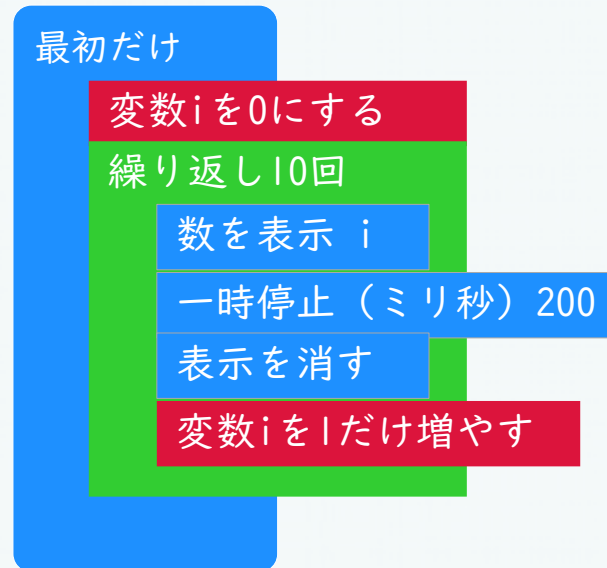
文字列を表示 "x"

+

演習1-16. 変数と反復を用いた プログラム例

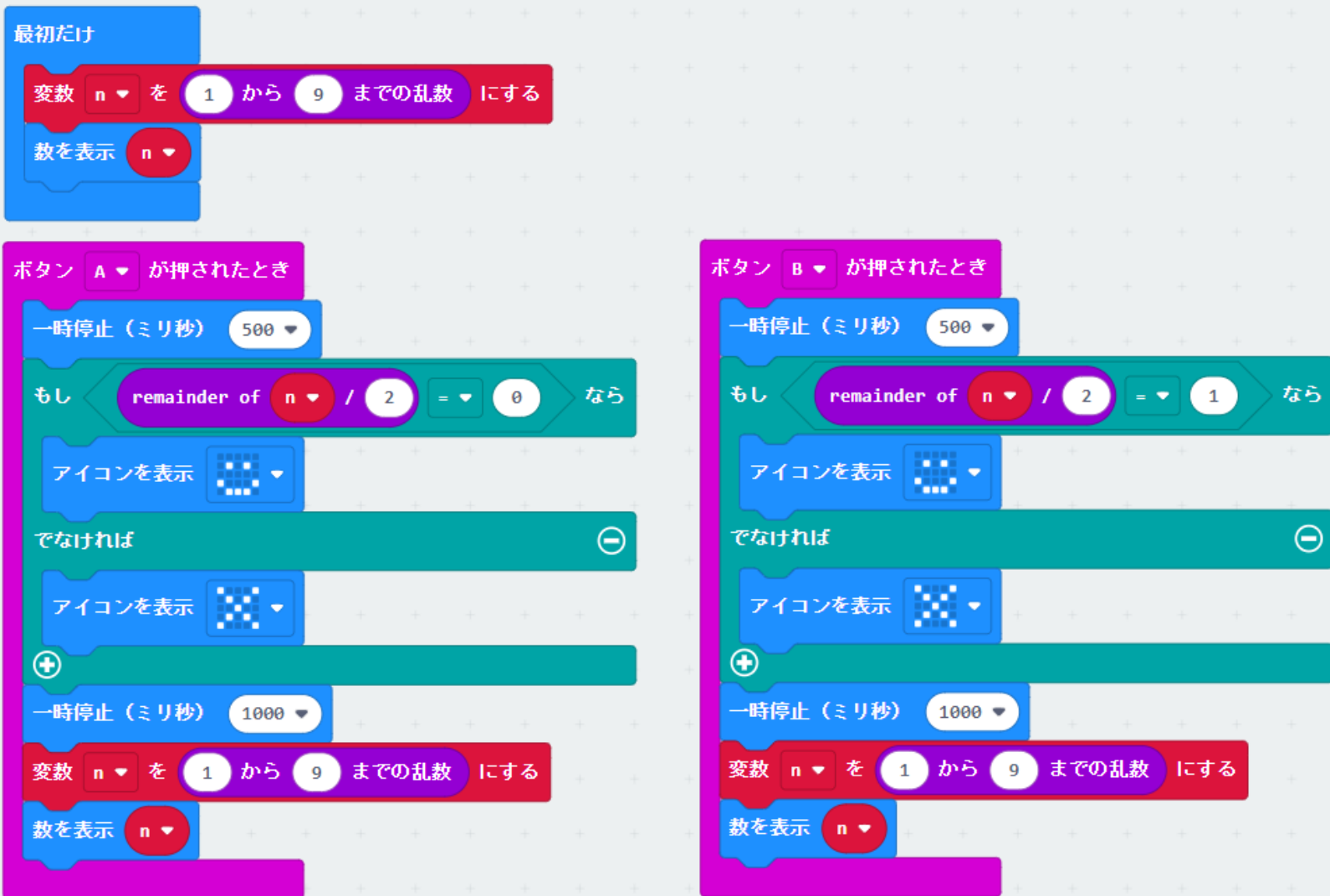
- 0, 1, 2, ..., 9と数をLEDに表示
- 変数*i*に0を代入
- 10回反復する
 - 変数*i*が覚えている数を表示
 - 変数*i*を1増やす

- プログラム例



演習1-17. 分岐のプログラム例

表示された数の奇数・偶数を答える



演習1-18. 作成したプログラムの micro:bitへのインストールと実行

- 方法1：PCとmicro:bitを「デバイスを接続する」により結びつける。
- 方法2：「ダウンロード」をクリックすると
 - makecodeのサイトから手元のコンピュータへプログラムをダウンロード・保存
 - 保存場所は「ダウンロード」フォルダ



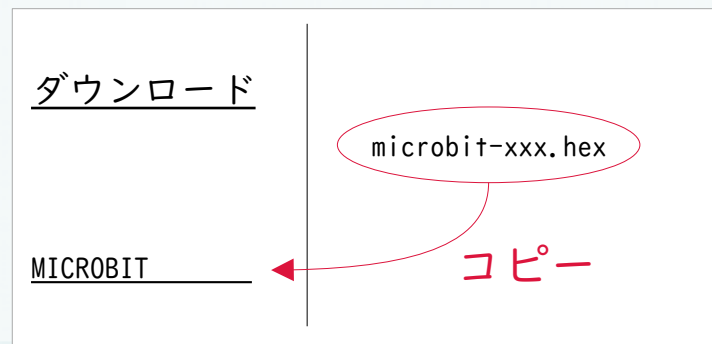
ダウンロード完了... ×

ヘルプ

完了

演習1-19. 作成したプログラムの micro:bitへのインストールと実行

- フォルダ「ダウンロード」に格納されているプログラム「microbit-xxx.hex」をドラッグ&ドロップでmicro:bitにコピーすると、そのプログラムがインストール・実行される。
- プログラムの最初から実行し直すにはmicro:bitの裏面にある「リセット実行ボタン」を押す。

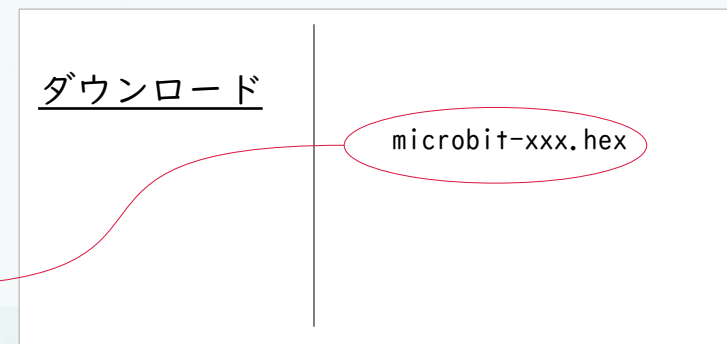


パソコン内部の補助記憶装置



演習1-20. 保存プログラムの読み込み・修正

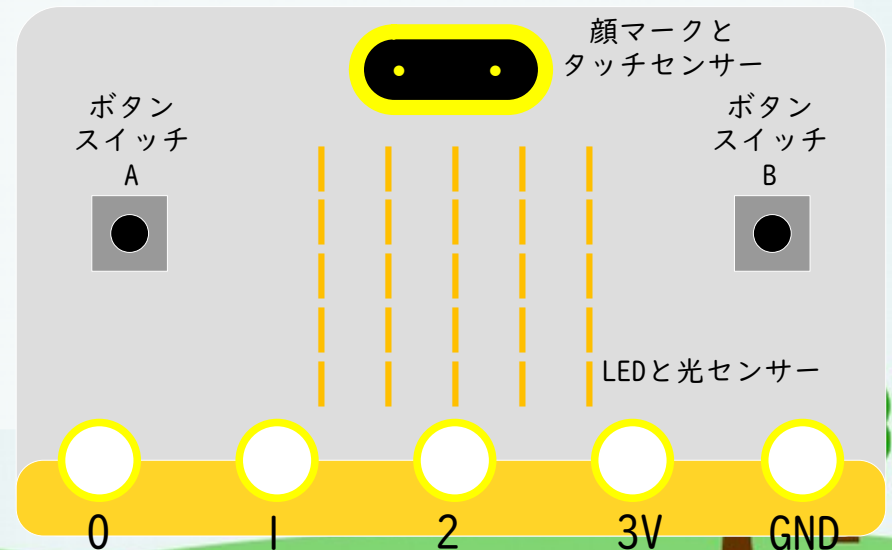
- コンピュータに保存されているプログラムを **makecode** サイトにドラッグ&ドロップすると、再度プログラムの修正が可能となる。
- **makecode/micro:bit** の最初のページで、右上の「↑読み込む」から、手元のコンピュータに保存されているファイルを読み込んでもよい。



パソコン内部の補助記憶装置

演習1-21. micro:bitのI/Oポート

- micro:bitの下部の「0」, 「1」, 「2」が外部装置との接続端子 (I/Oポート)
- 「3V」は「GND」に対して+3Vの電位
3Vの電池があり、電池のマイナス端子が「GND」でプラス端子が「3V」と思ってよい。
- その他：3番～は小さな端子
 - Makecodeの？>ハードウェア>入出力端子「pins」をクリックすると詳細な説明



演習 I-22. makecodeでのI/Oポートの利用例

- 0番端子からデジタルの「1」を出力
- 1番端子からアナログの「50」を出力（但しPWM変調）

ブロック ?

基本
入力
音楽
LED
無線
ループ
論理
変数
計算
拡張機能
◎入出力端子

最初だけ

デジタルで出力する端子P0 値1
アナログで出力する端子P1 値50

※補足：
・ デジタルの0, 1はL, Hと表すことがある
・ アナログは0～1023
・ 「出力する」では、出力端子が、0に対応する0Vや1に対応する3Vとなる。

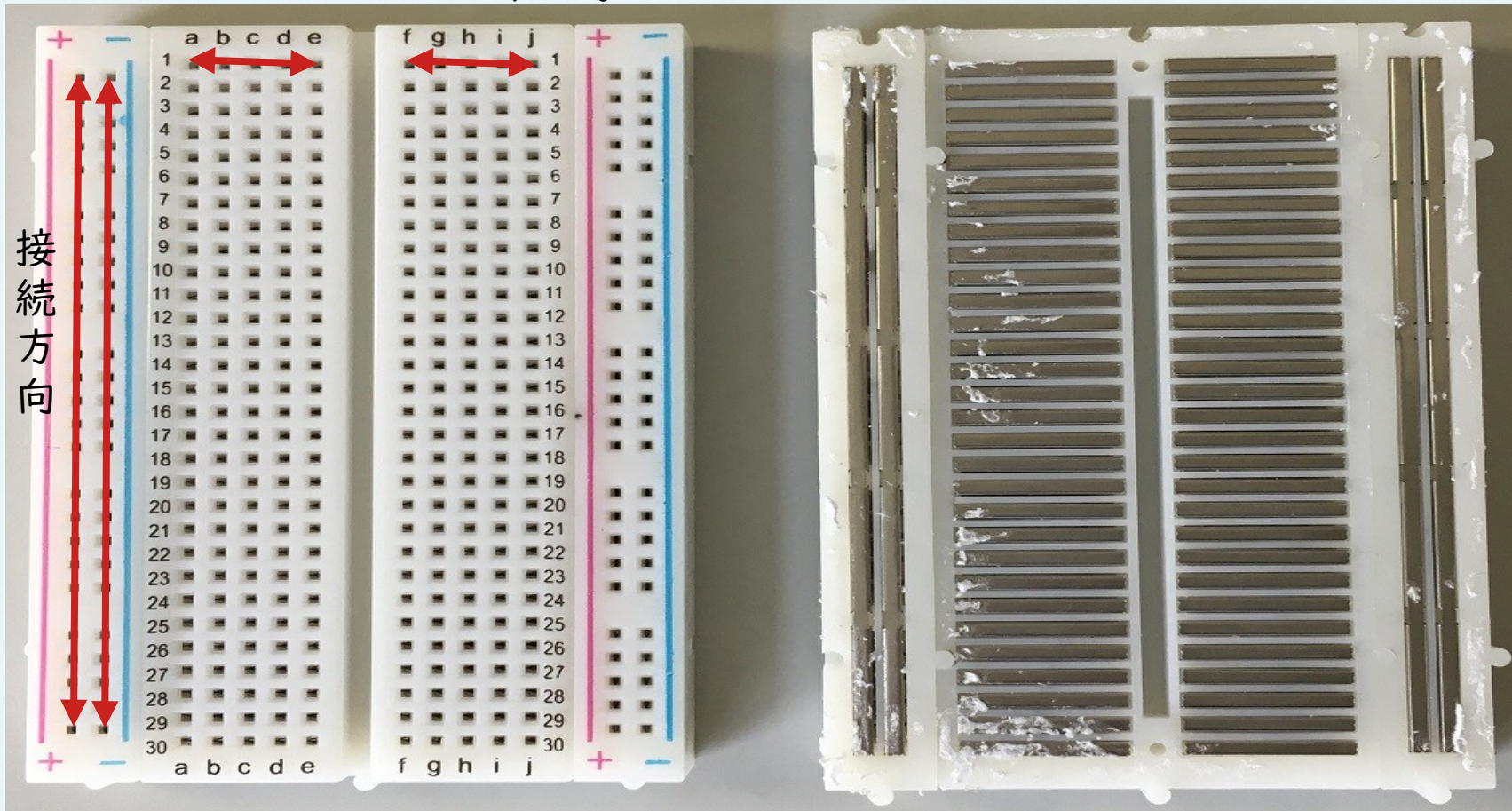
ダウンロード test

演習1-23. ブレッドボードの使い方

<https://akizukidenshi.com/catalog/g/gP-12366/>

接続方向

(各行1~30: a~e, f~jが接続)

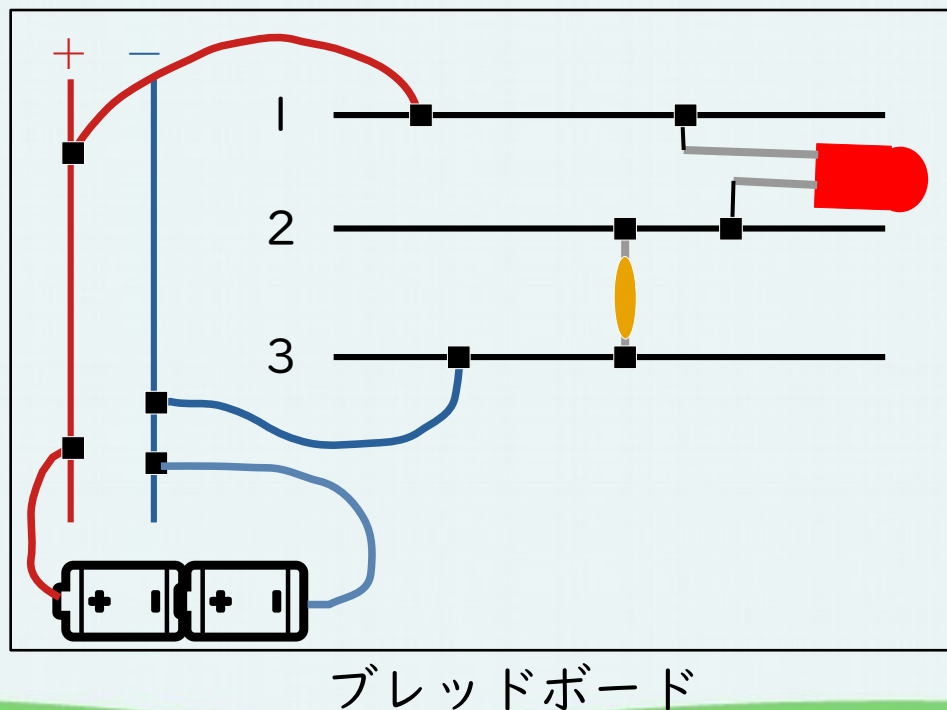
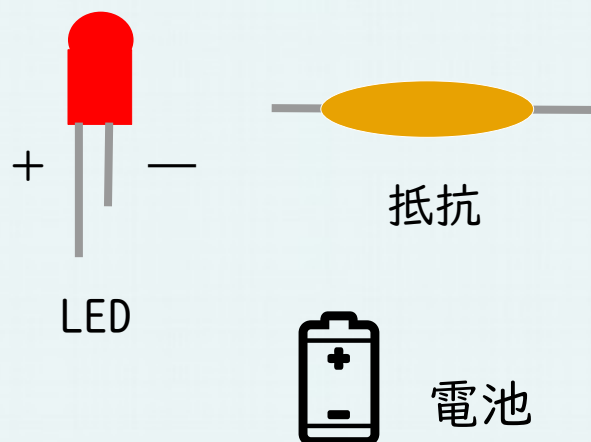


表

裏

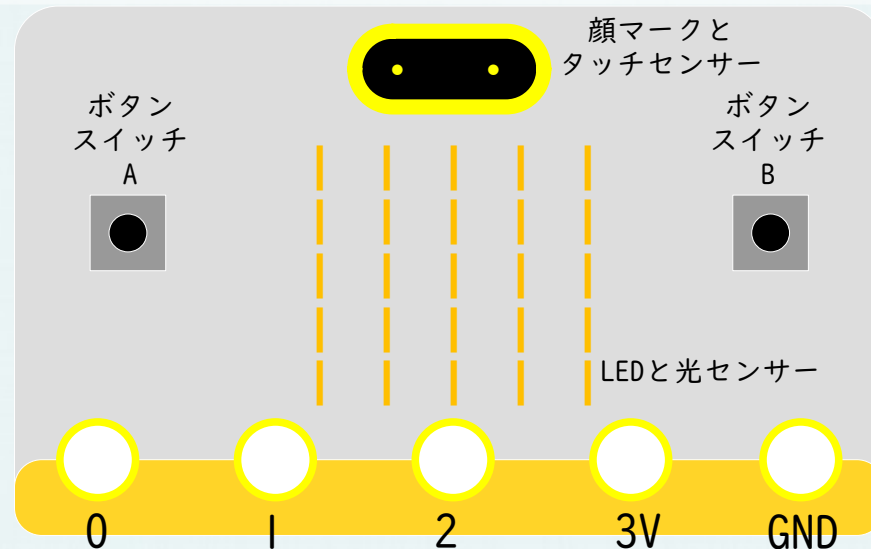
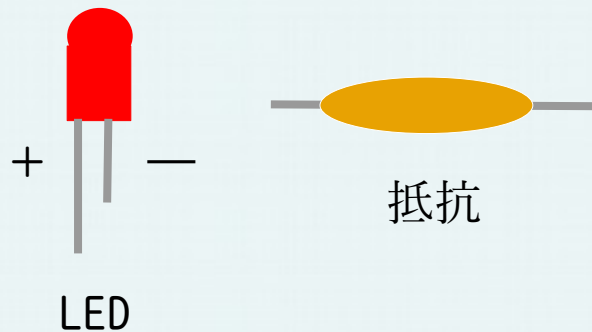
演習1-24. 確認・電池でLEDを点灯

- LEDと抵抗を直列接続
 - LEDは極性あり／抵抗は極性なし
- 例：電池の⊕→LED→抵抗→電池の⊖



演習1-25. I/OポートにLEDを接続

- P0にLEDを接続して点灯させる
- LEDと抵抗を直列接続
 - LEDは極性あり／抵抗は極性なし
- 例：P0→LED→抵抗→GND



演習 I-26. LED を点灯・消灯させる micro:bit のプログラム

- プログラム例：

ずっと

デジタルで出力する端子 P0 値 1

一時停止（ミリ秒）500

デジタルで出力する端子 P0 値 0

一時停止（ミリ秒）500



The background features a light blue sky with a large yellow sun in the top right corner and several white, fluffy clouds. The bottom of the slide shows a green landscape with rolling hills, stylized green trees with brown trunks, and small pink flowers scattered across the grass.

演習2.

リレーをmicro:bitで制御する

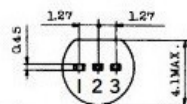
micro:bitで小型リレーを動かしてみよう

2SC1815 TRANSISTOR (NPN)

FEATURES

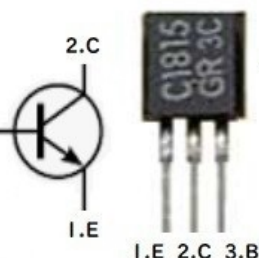
- Power dissipation

MAXIMUM RATINGS (T_a=25°C unless otherwise noted)



ボトムビュー

1. E エミッタ
2. C コレクタ
3. B ベース



正面からみて「エクボ」で覚えよう!

Symbol	Parameter	Value	Unit
V _{ceo}	Collector-Base Voltage	80	V
V _{ceo}	Collector-Emitter Voltage	50	V
V _{ebo}	Emitter-Base Voltage	5	V
I _c	Collector Current -Continuous	150	mA
P _c	Collector Power Dissipation	400	mW

注 リレーやLEDの実負荷は、30V・100mA以下で使用してください。左記のスペックは、絶対(最大)定格です。リレーやソレノイド負荷の場合は、保護回路を必ず付けて下さい。

micro:bit カードエッジ
コネクター端子



*注 A1はA0・A2他も使用可

FET回路は必要

リレーを5V用に交換すれば
Allmay2でも同じ回路で作動する

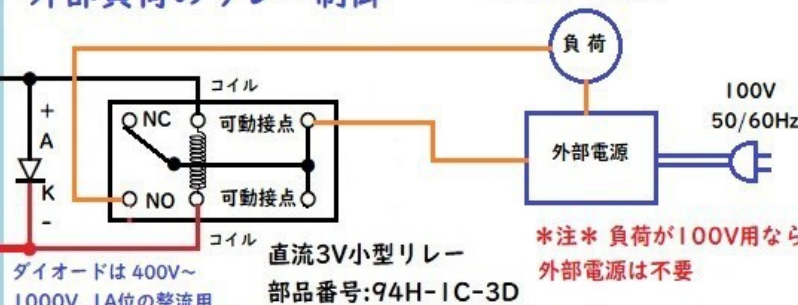
サージ電圧	保護ダイオード
選定リスト	IN4004
	400V 1A
	IN4006
	600V 1A
	IN4007
	1000V 1A

注
動作が不安定な時は
1KΩの抵抗値を調整する

上記回路を演習時は、以下の回路で配線する

外部負荷のリレー制御

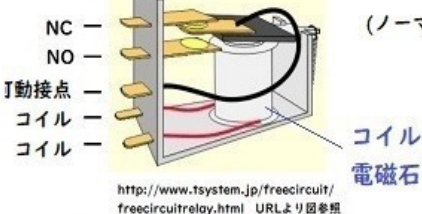
外灯等の負荷



注 負荷が100V用なら
外部電源は不要

注 リレー制御の場合は
負荷は交流(AC)でも、直流(DC)でも良い

リレー内部のイラスト

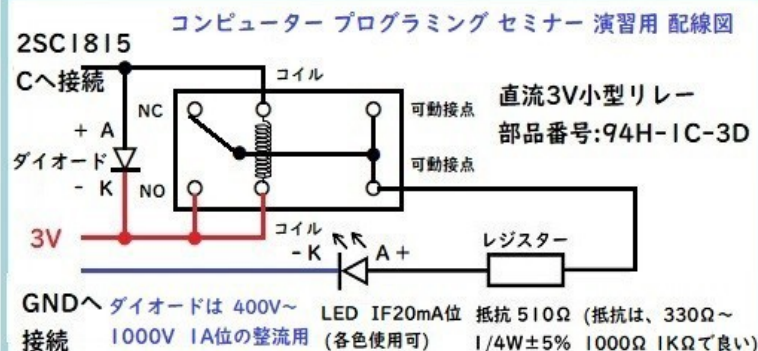


リレー(継電器)とは、コイルに電流を流すことにより、電磁石で内部の接点を動かし電流をON・OFFしたり切り替える電気部品です。

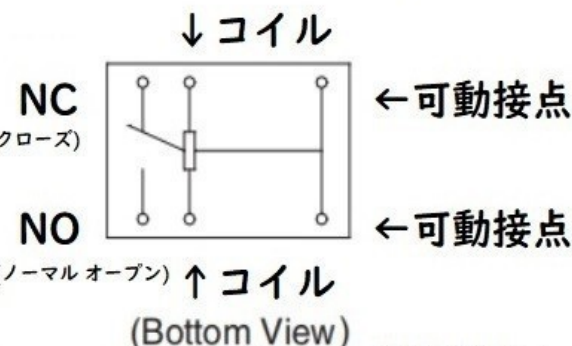
LED



A 長い



GNDへダイオードは400V~
接続 1000V 1A位の整流用
LED IF20mA位 抵抗 510Ω (抵抗は、330Ω~
(各色使用可) 1/4W±5% 1000Ω 1KΩで良い)



3V小型リレー
接点容量: 3A 946H-1C-3D
コイル抵抗 45Ω 電流 66.7mA

リレー 部品入手先 秋月電子通商
<https://akizukidenshi.com/catalog/g/gP-07651/>
<https://akizukidenshi.com/catalog/> ← HP
秋月通商は電気電子部品の技術データが無料で参照可能

micro:bitでリレーを動かす時、トランジスターやFETが必要な理由
micro:bitのI/Oポートの出力端子の駆動能力は、5/1000 A ±5mA
しかありません。従って、リレーやソレノイド・モーター等の大電流
をドライブする場合は、電流増幅用のデバイスを使用します。

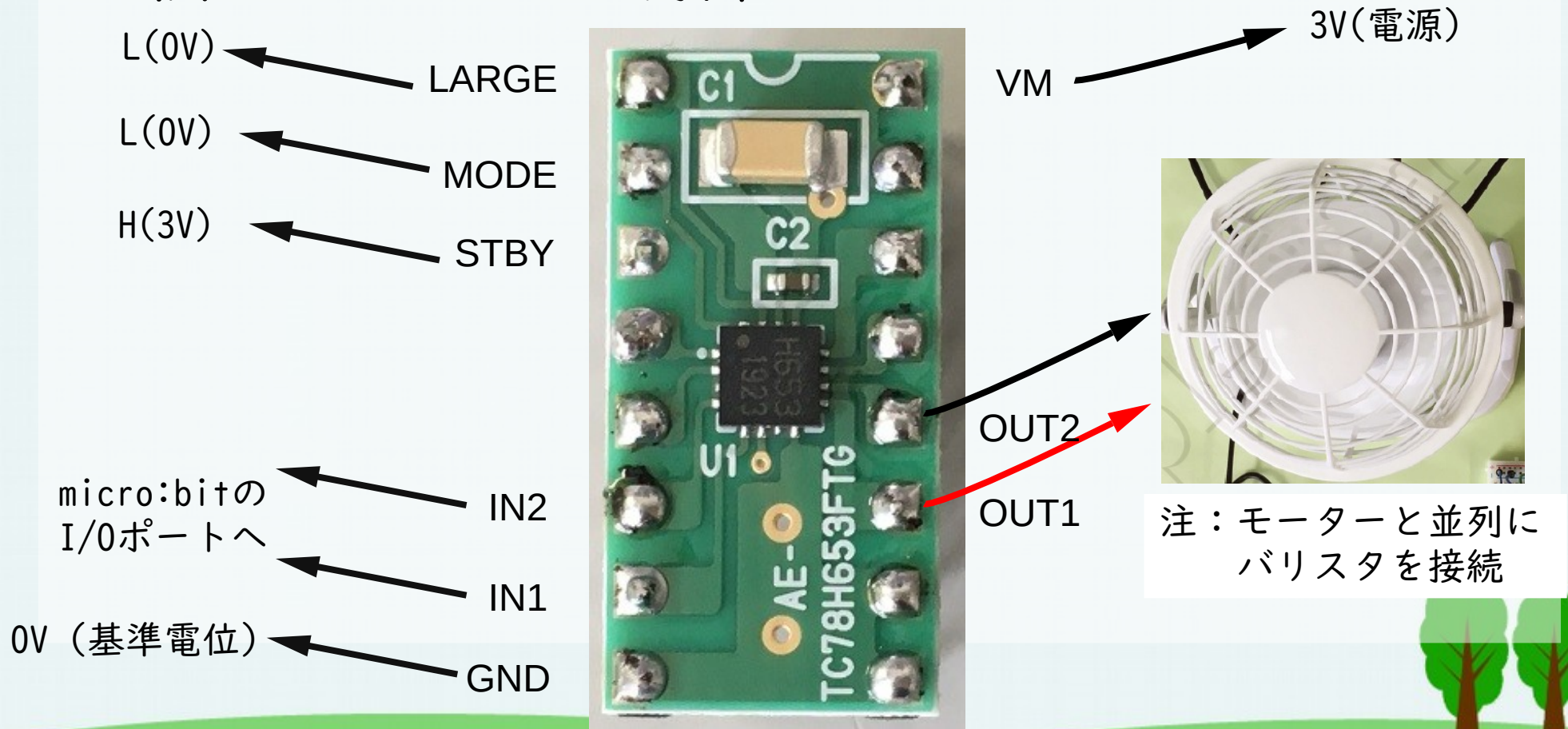


演習3.

モーターをmicro:bitで制御する

演習3-1. モーターの回転を micro:bitでコントロール

- 東芝：TC78H653／秋月電子：AE-TC78H653FTG
- 2個のモーターを制御できる



演習3-2. モーター制御用ICの使い方

https://akizukidenshi.com/download/ds/toshiba/TC78H653FTG_datasheet_ja_20190129.pdf

- 例：IN入力モード・スモールモードの場合
 - MODE:L(0), LARGE:L(0), STBY:H(1)
 - VM:電源 (+3V), GND:0V
 - IN1:H(1), IN2:L(0)
=> OUT1:H(1), OUT2:L(1)
 - IN1:L(0), IN2:H(1)
=> OUT1:L(0), OUT2:H(1)

回転方向が
逆になる





付録.

付録. 無線通信を用いた 数や文字の送受信

- micro:bitには無線通信の機能がある。
- 2つのmicro:bitの間で、数や文字を送信・受信することができる。
- 無線通信を行うには、準備が必要
- 設定や送信・受信のためのブロックがある。



付録. 無線通信を行うための準備

- 準備：無線のグループを設定
 - グループ番号：0～255の整数の中から1つを割り振る
 - 通信するmicro:bitには同じグループ番号を設定
 - 同じグループ間で通信できる
- 「無線」のメニューの
 - 「Group」の「無線のグループを設定 1」
 - これを「最初だけ」のブロックに入れる
(グループ設定は最初に1回行えばよい)

最初だけ

無線のグループを設定 1



付録. 無線により数や文字列を送り合う

- 「無線」のメニューの
 - 「Send」
 - 「無線で数値を送信 0」
 - 「無線で文字列を送信 “ ”」
 - 「Receive」
 - 「無線で受信した時 `receivedNumber`」
 - 「無線で受信した時 `receivedString`」



付録. 無線通信を用いたプログラム例

- ボタンAを押すと、整数乱数を送受信して表示

ブロック

基本
入力
音楽
LED
無線
ループ
論理
変数
計算
拡張機能
高度なブロック

最初だけ

無線のグループを設定 |

ボタンAが押されたとき

変数nを0から9までの乱数にする

無線で数値を送信 n

数を表示 n

無線で受信した時 receivedNumber

数を表示 receivedNumebr

ダウンロード test

(無線を使うと2台めが出てくる)

質問があったら

- 山口-UK コンピューター プログラミング協会
会長 弘中 富士彦
– yamaguchi.uk.c.p.s@gmail.com
- 山口大学教育学部 野村厚志
– anomura@yamaguchi-u.ac.jp

