きのくにICT教育

レゴ® マインドストーム® EV3版

中学校プログラミング教育
学習指導案
補助資料

和歌山県教育委員会

目　次

中学校１年生編

単元の構成

1時間目：プログラミングでコンピュータに指示してみよう(Hour of Code)

2時間目：センサでコンピュータを動かそう(micro:bit)

3時間目：計算機を作ろう（Studuino）

4時間目：計算クイズを作ろう（Studuino）

5時間目：分かりやすく・使いやすくしよう（Studuino）

１時間目：プログラミングでコンピュータに指示してみよう(Hour of Code)

①使用するソフトウェア

Hour of Code（アワー・オブ・コード）　　http://code.org/hoc

- コンピュータサイエンス※教育週間に行われるプログラミング体験活動
  - 全世界的なムーブメント
  - 幅広い対象向けの教材開発

※コンピュータ活用全般を扱う学問分野（プログラミングは構成要素の１つ）

- 教材の目的
  - コンピュータサイエンスに対する興味を持たせる
    注）プログラミング技術を学ぶものではない

- パズル型プログラミング
  - コードブロックの組み合わせで迷路などの課題を解く

②ソフトウェアの操作方法

②－１　ブラウザ（インターネットエクスプローラなど）を立ち上げる

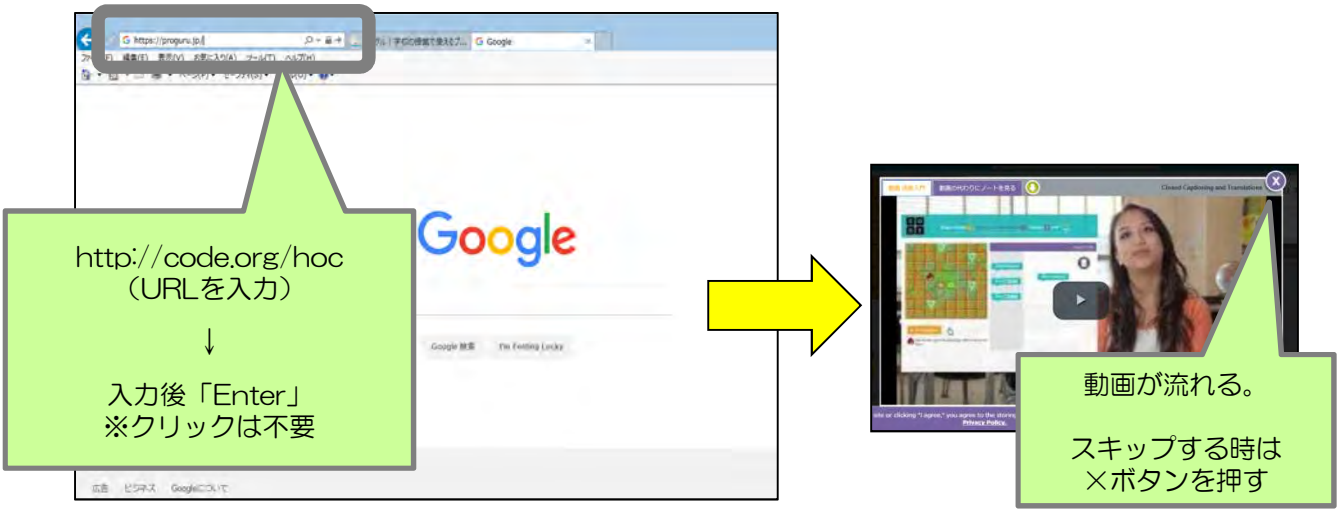
インターネット
エクスプローラ

グーグル
クローム

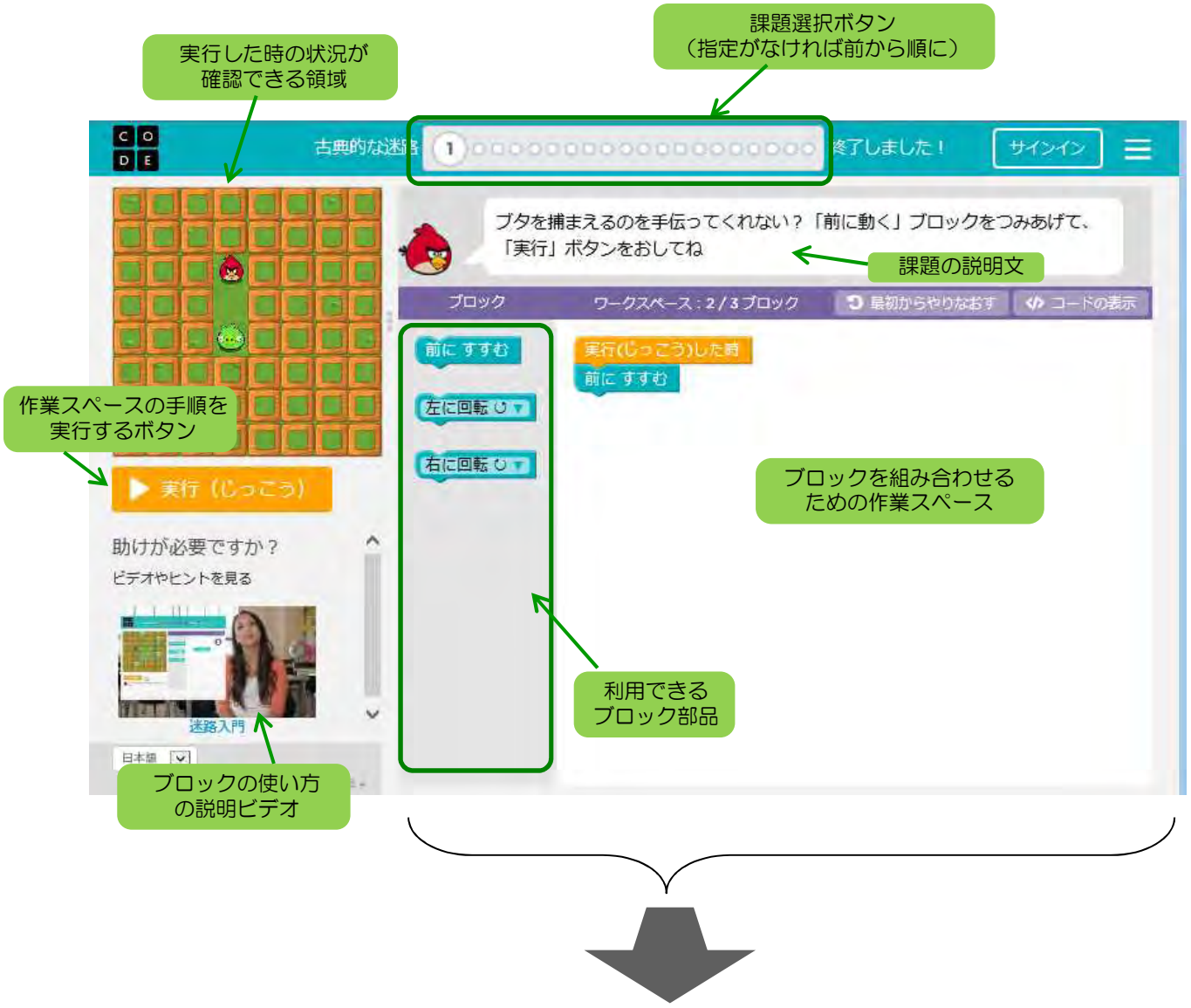

マイクロソフト
エッジ


サファリ
※アイパッド等

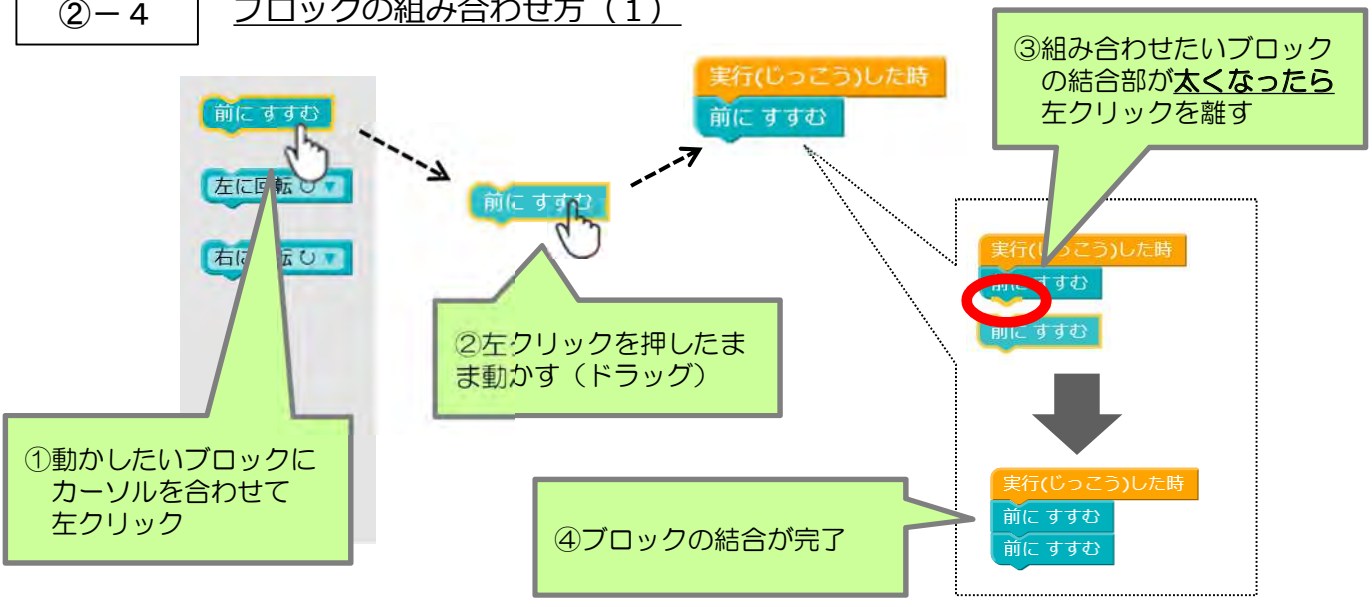
②－２　Hour of Codeのサイトへ移動　　※Googleの例



②－３　Hour of Codeの操作画面



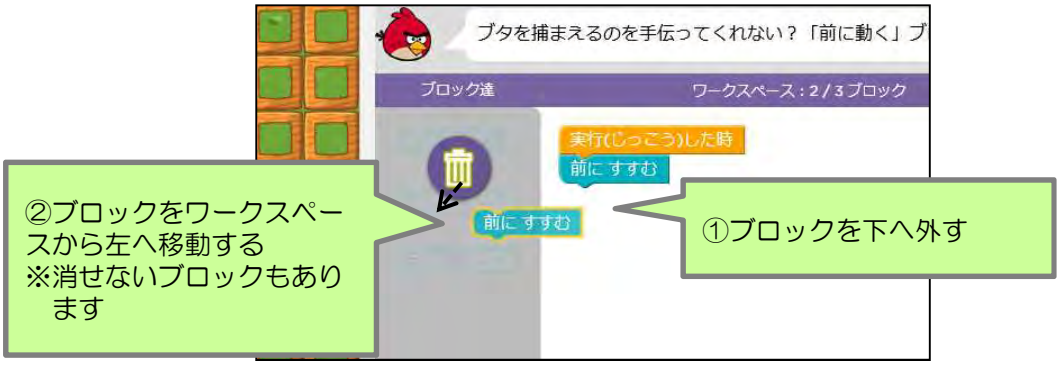
②－４　ブロックの組み合わせ方（１）



②－５　ブロックの組み合わせ方（２）

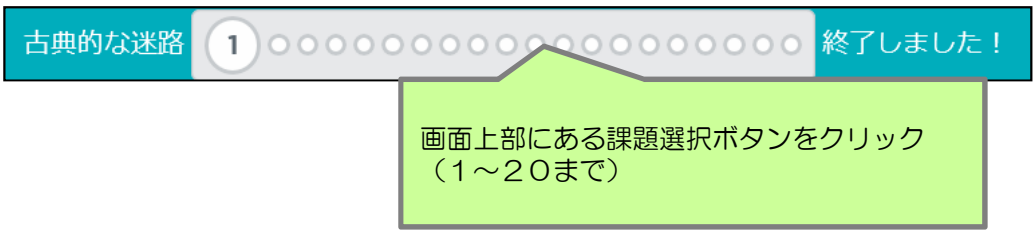
※「繰り返しブロック」「条件分岐ブロック」の詳細はP6参照

②－６　ブロックの消し方



②－７　プログラムの実行

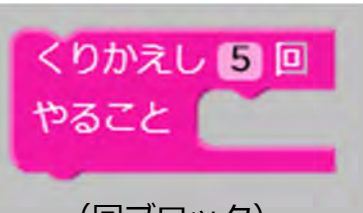
②－８　手動での課題（パズル）の変更



③大切なブロック

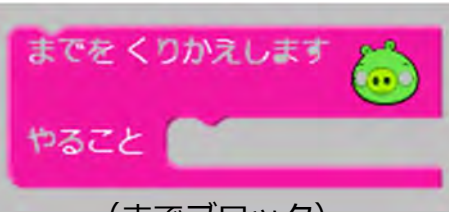
- プログラミング教育においては、「繰り返し」や「条件分岐」の概念を理解することが重要です。
- 授業においては、生徒が「繰り返し」「条件分岐」の両方のブロックに触れられるよう、状況を見ながら進行を工夫してください。

〈繰り返し〉



（回ブロック）

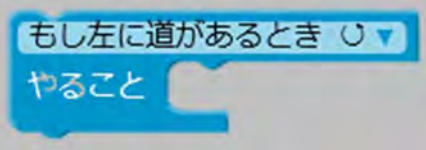
- 回数を指定して繰り返す
  - 内に含んだブロックを、**指定された回数だけ**実行する
- 課題6で登場



（までブロック）

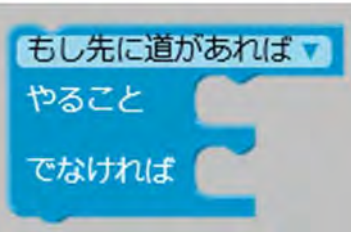
- ある状況になるまで繰り返す
  - 内に含んだブロックを、**キャラクタに会うまで何度も**実行する
- 課題10で登場

〈条件分岐〉



（もしブロック）

- 条件を指定して実行する
  - 内に含んだブロックを、**指定された条件が満たされる時だけ**実行する
- 課題14で登場



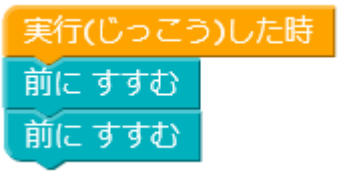
（もし／でなければブロック）

- 条件を指定して、満たされる時と、満たされない時とで、**場合分け**して実行する
- 課題18で登場

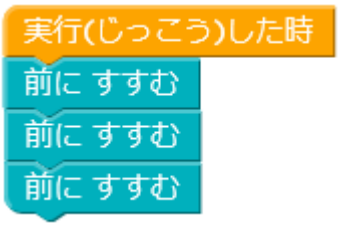
④プログラム例

- 正しいプログラムが一つとは限りません。確実に目的を達成するのであれば、それは正しいプログラムです。
- 大切なことは、その動きを自分の言葉で説明できることです。それができれば、工夫の余地にも気づきやすいでしょう。

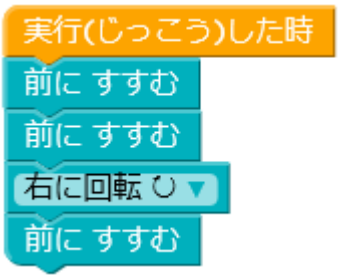
〈課題１〉



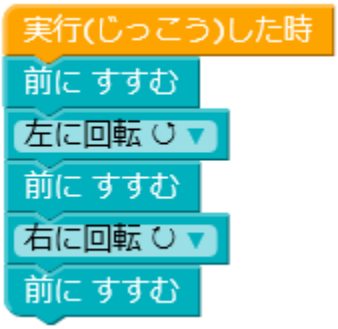
〈課題２〉



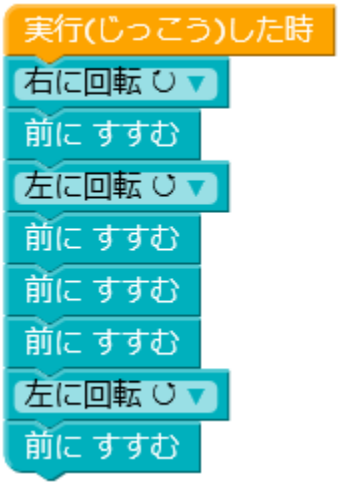
〈課題３〉



〈課題４〉



〈課題５〉



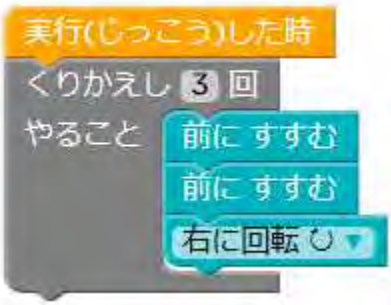
〈課題６〉

繰り返しの回数を
手入力で変更

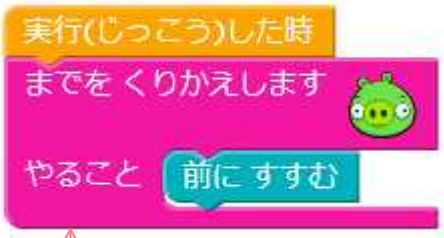
〈課題７〉

〈課題８〉

〈課題９〉

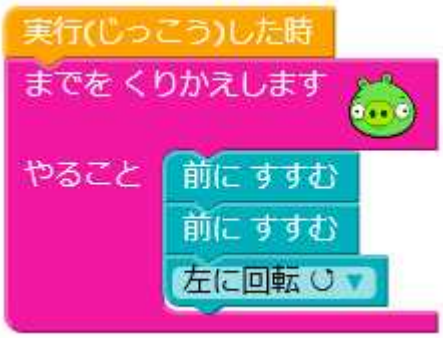


〈課題10〉



キャラクターに会うまで
プログラムを繰り返すブロック

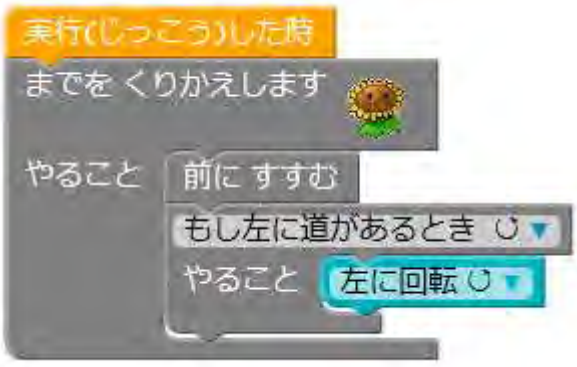
〈課題11〉



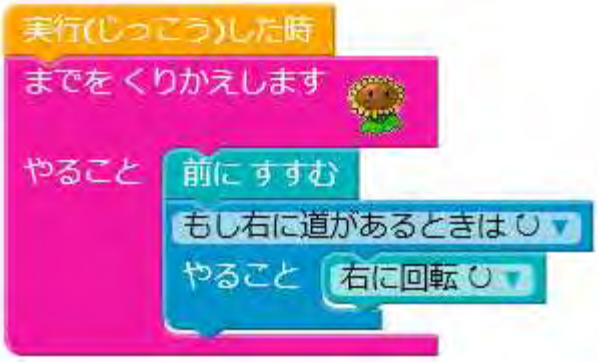
〈課題12〉

〈課題13〉

〈課題14〉

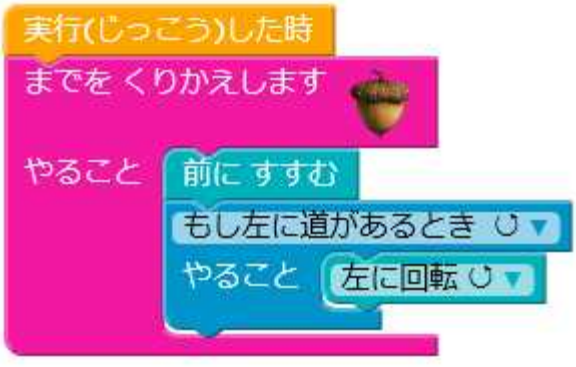


〈課題15〉



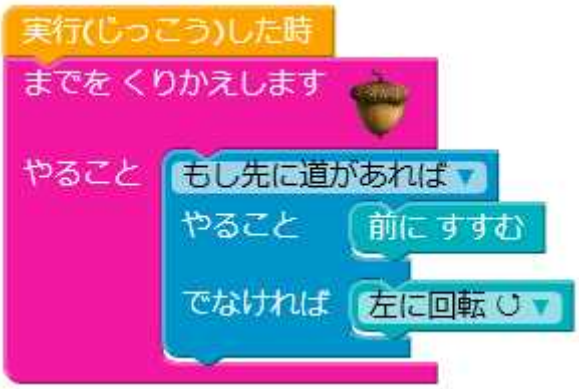
「もし左…」をプルダウン
メニューから
「もし右…」に変換

〈課題16〉

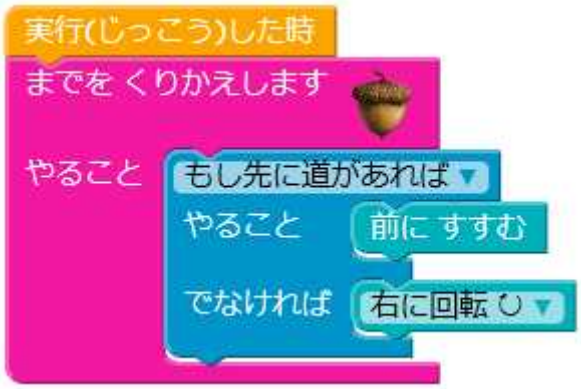


〈課題17〉

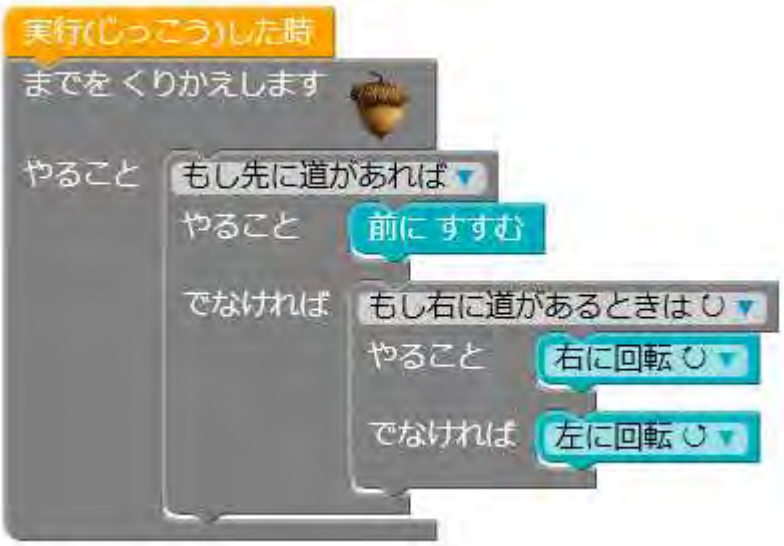
〈課題18〉



〈課題19〉



〈課題20〉

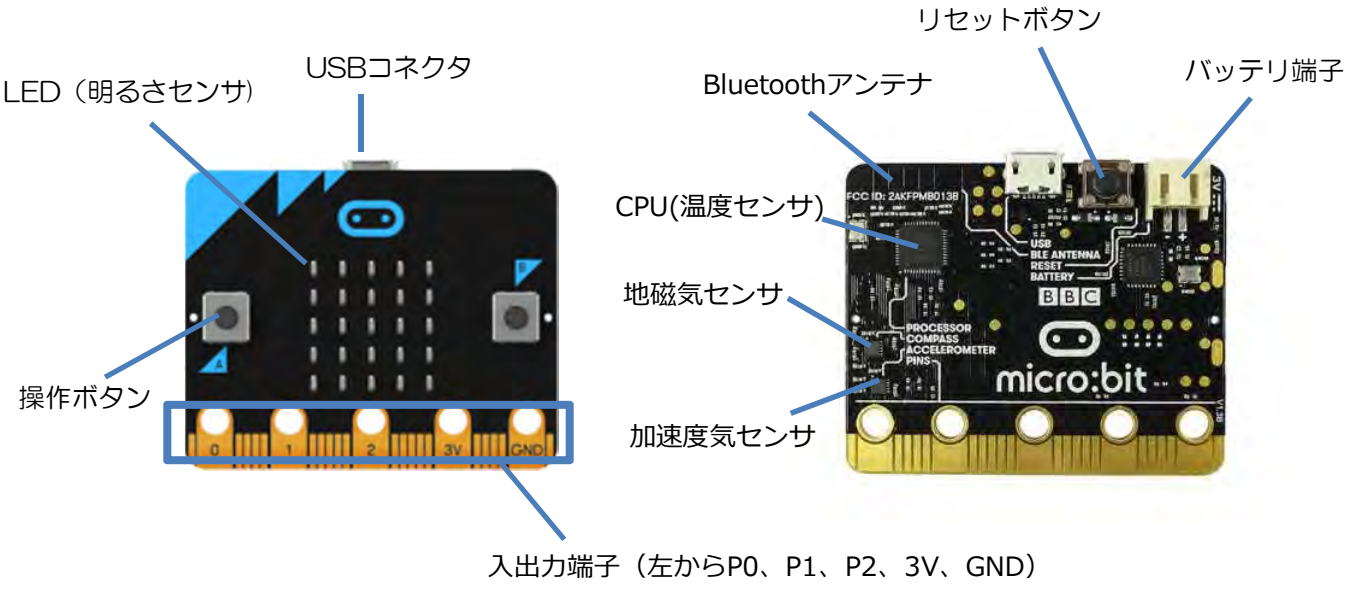


２時間目：センサでコンピュータを動かそう(micro:bit)

①使用する教材

micro:bit（マイクロビット）　　https://makecode.microbit.org/

- BBC（イギリス放送協会）が企画・開発した教育用シングルボードコンピュータ
  - 11, 12歳の小学生に無料配布
- 単体でできること
  - 簡易表示（LEDマトリクス）
  - センサ（ボタン，加速度（重力），明るさ，磁気）
  - 無線通信（Bluetooth Low Energy）
  - 汎用入出力端子（25ピンコネクタ）
  - モーターやスピーカーの出力，人のタッチ入力
- ブロックで作ったプログラムをダウンロードして実行



- パソコンとマイクロビットを接続するには、ＵＳＢケーブル（片方がＡタイプ、もう片方がMicro-Bタイプ）が必要です。　※Bluetoothで接続する場合は不要

②パソコン画面上（エディター）の操作方法

②－１　ブラウザ（インターネットエクスプローラなど）を立ち上げる

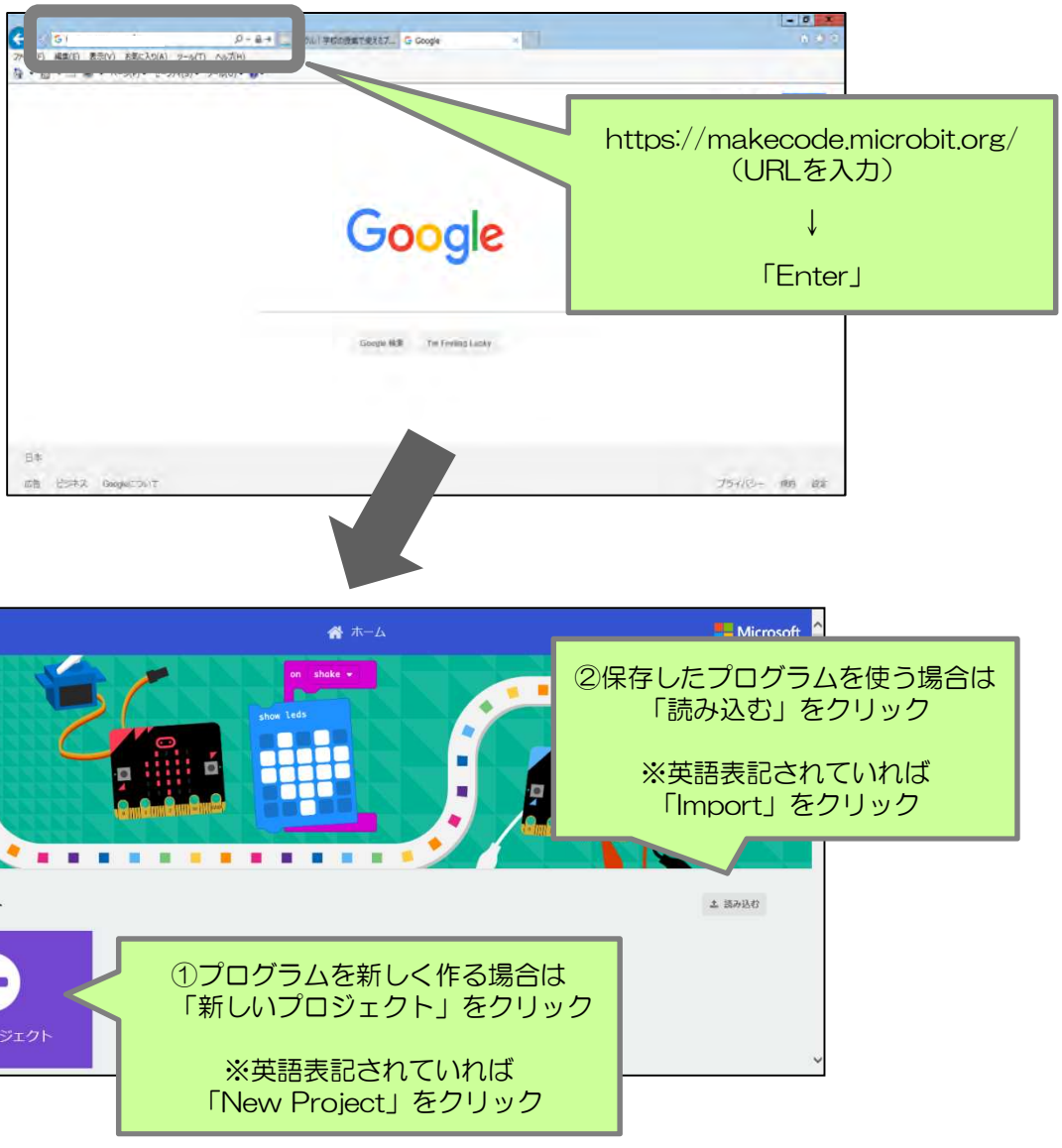
インターネット
エクスプローラ

グーグル
クローム

マイクロソフト
エッジ

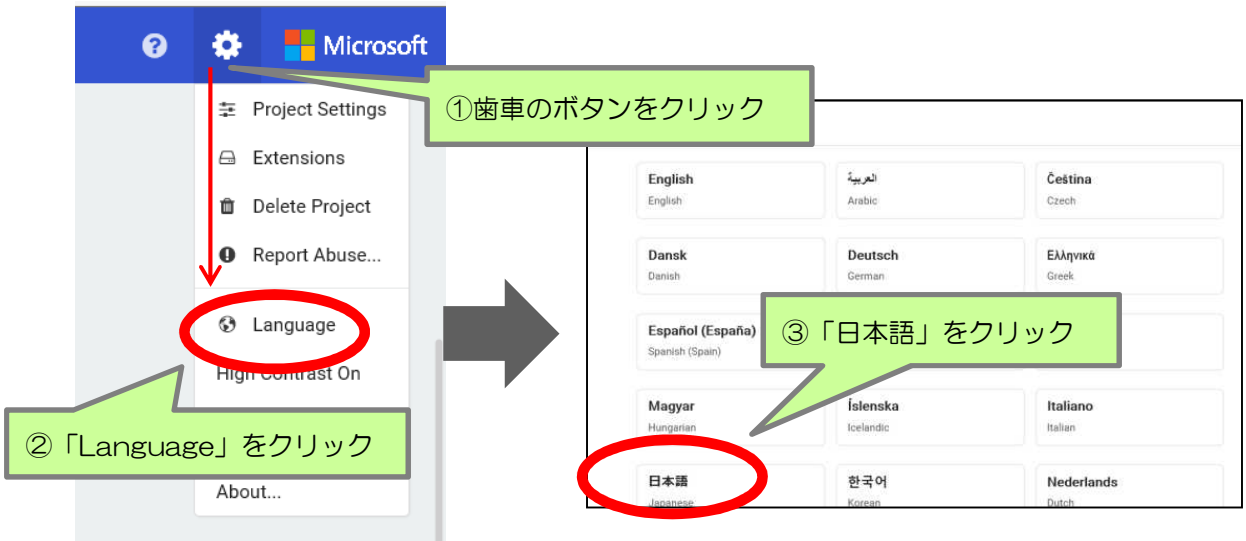
※アイパッドの場合は
マイクロビットのアプリの
インストールが必要

②－２　マイクロビットのサイトへ移動　　※Googleの例

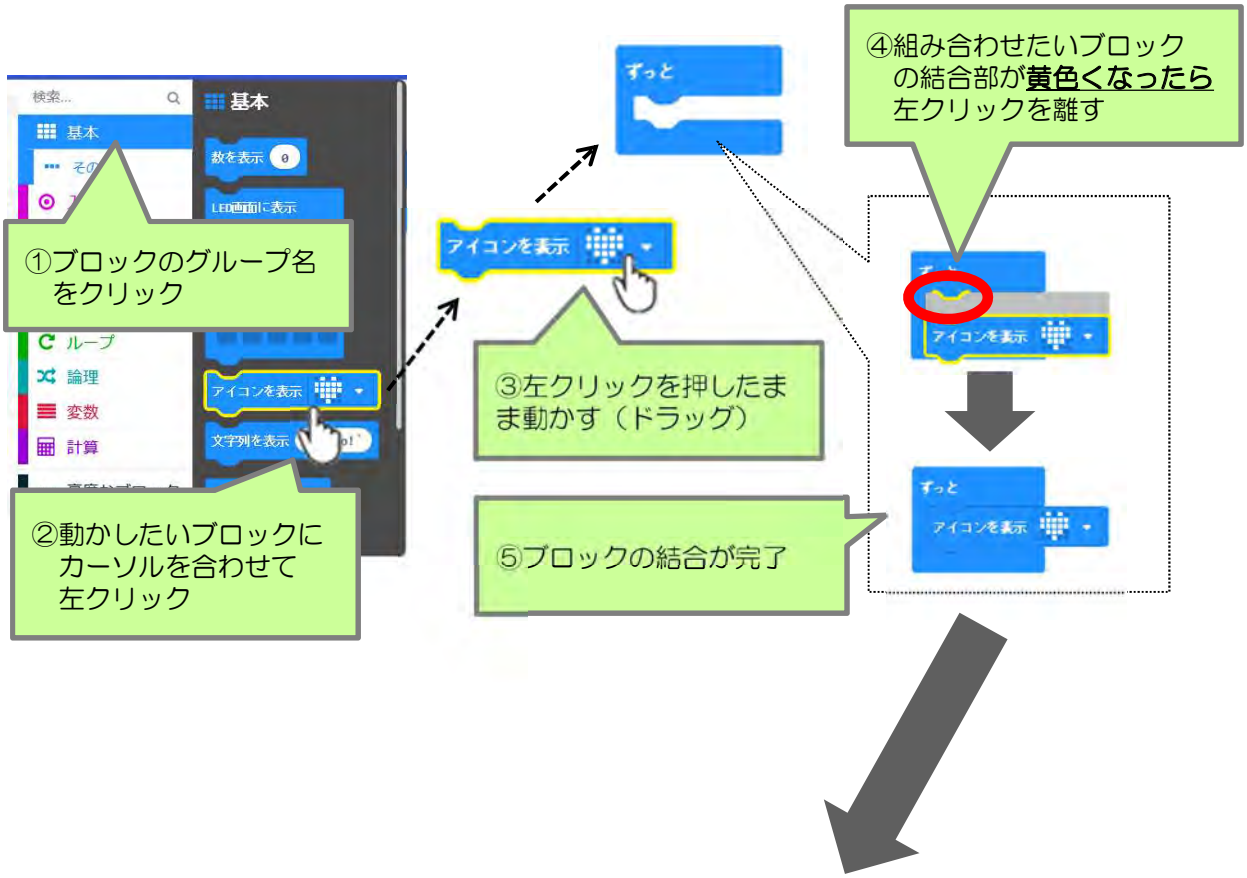


②－３　マイクロビットのエディターの操作画面

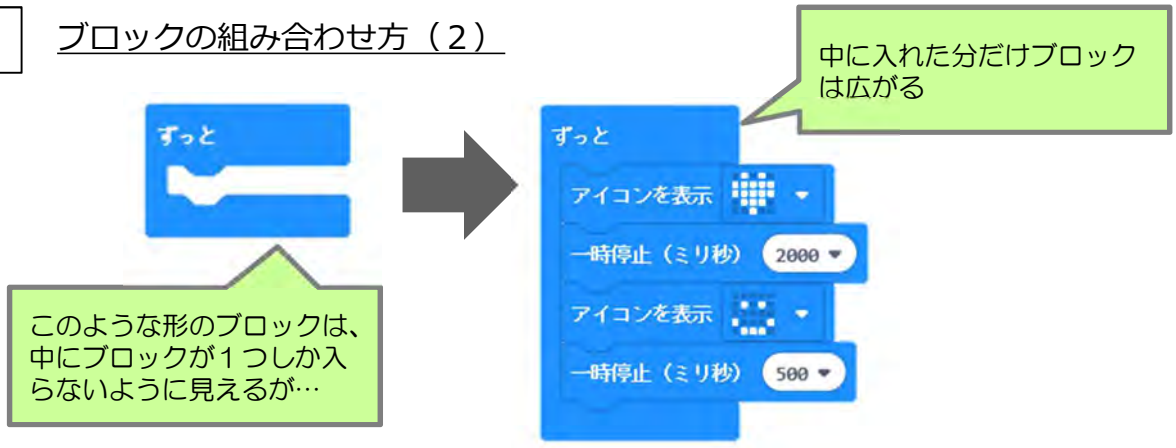
②－４　英語表記から日本語表記への変換方法



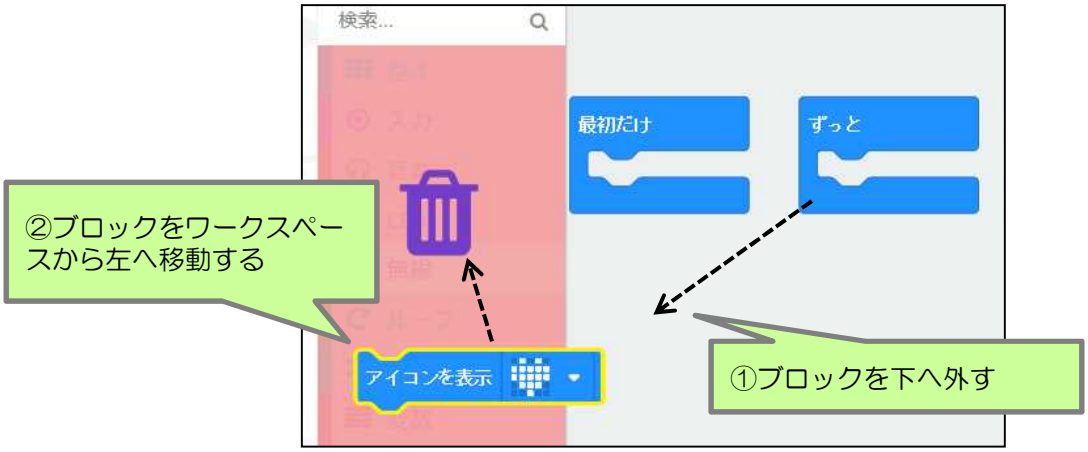
②－５　ブロックの組み合わせ方（１）



②－６　ブロックの組み合わせ方（２）

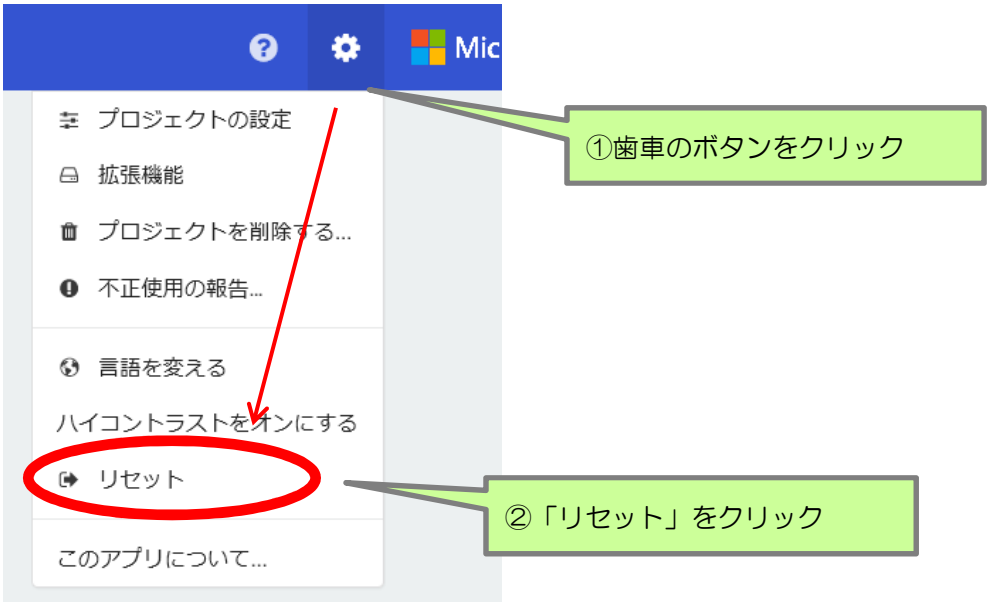


②－７　ブロックの消し方



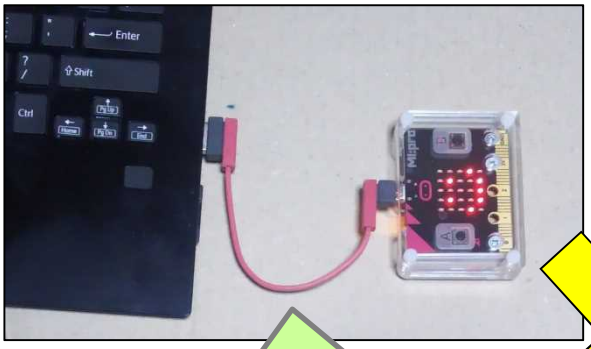
②－８　作業環境のリセット

- 同じパソコンで作ったプロジェクト（作業状況）は残っている
- これを削除するには、歯車メニューから「リセット」を実行

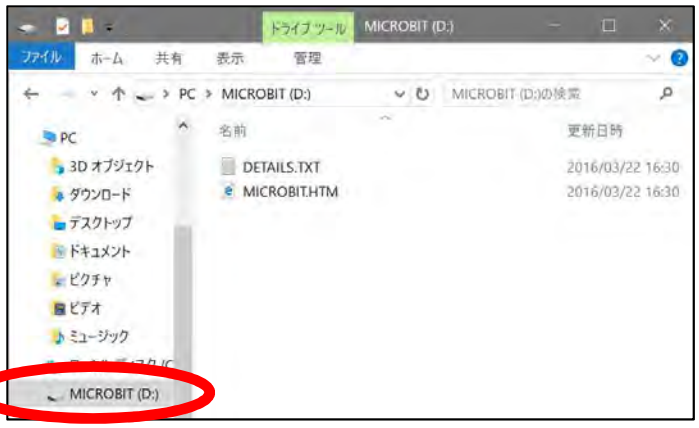


③マイクロビット（実機）でのプログラムの実行

③－１　マイクロビットの接続

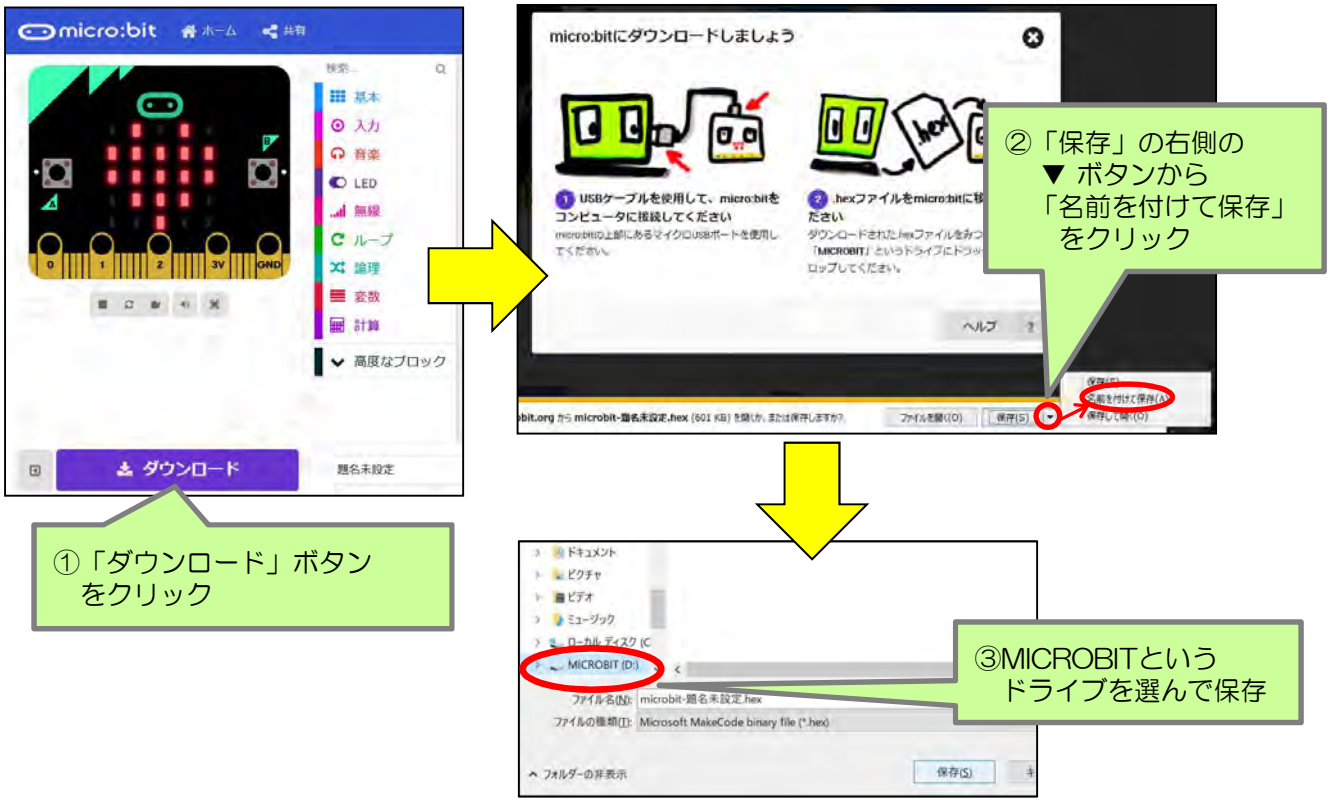


①ＵＳＢケーブルでパソコンと
　マイクロビットをつなぐ
※パソコンから電源も供給される



②「MICROBIT」という名前のドライブ（図ではＤ：）が作られる

③－２　プログラムのダウンロード

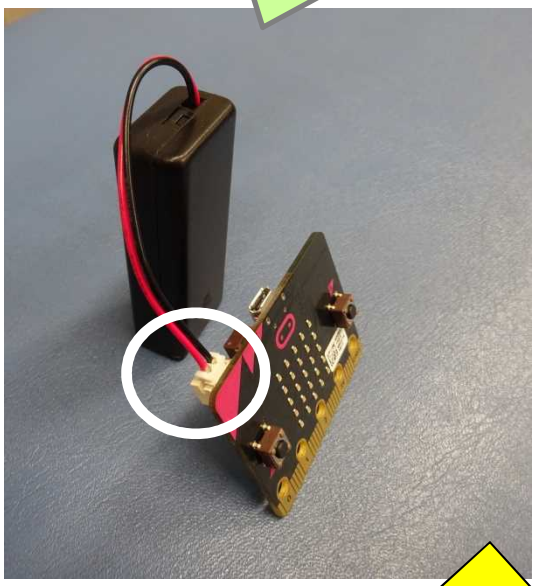


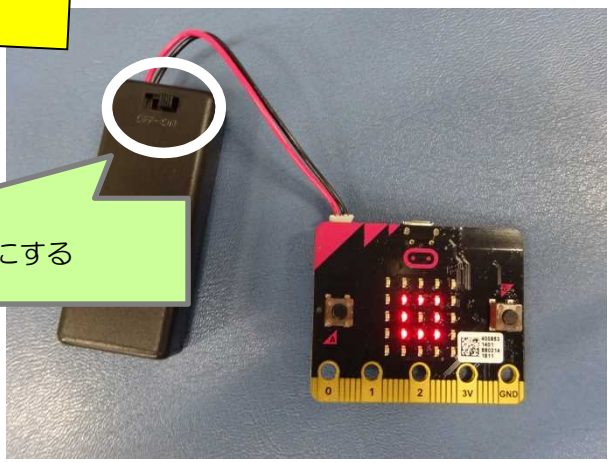
・プログラムの履歴を残したい場合は、
　一度パソコンに保存　→　ＭＩＣＲＯＢＩＴドライブへコピー

③－３　パソコンに接続しない使い方

●電池ボックスを利用すれば、パソコンに接続しなくてもプログラムを実行できる

①電池ボックスとマイクロビットを
接続

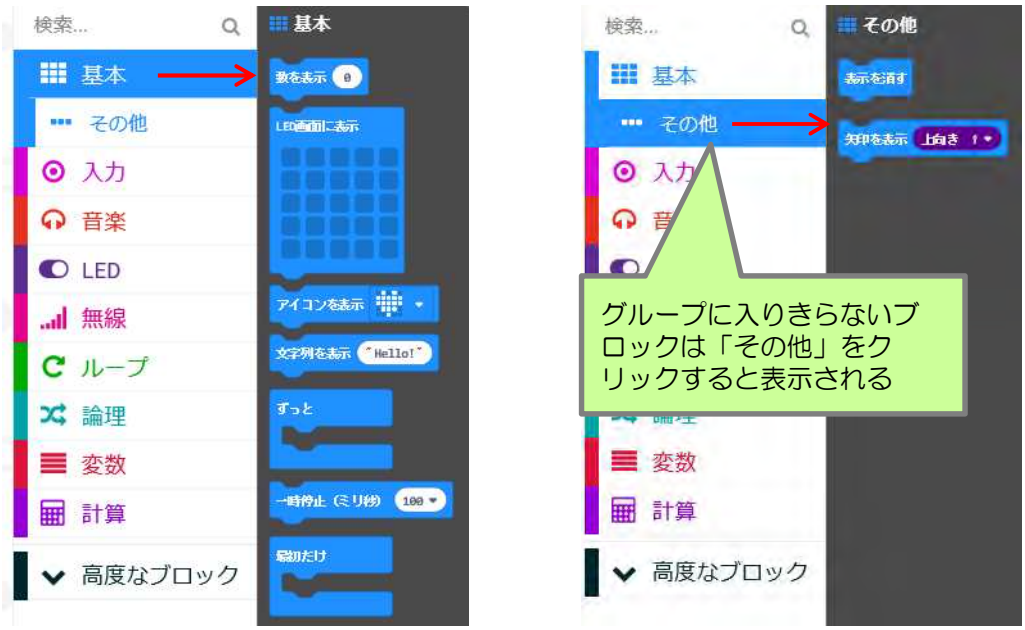




②電池ボックスの電源をONにする

④ブロックのグループ（主なグループ）

④－１　基本グループ



④－２　入力グループ（ボタン、センサ）

・ボタンの状況の取得
・各種センサの値の取得

④－３　ループグループ（繰り返し）

④－４　論理グループ（条件分岐）

⑤プログラミングでコンピュータを意図したとおりに動かす

⑤－１

ワークシートの課題：　２）何もしないときは「ハート」が表示、Ａボタンが押されたときだけ自分の名前がローマ字で表示されるようにしよう

〈プログラム例〉

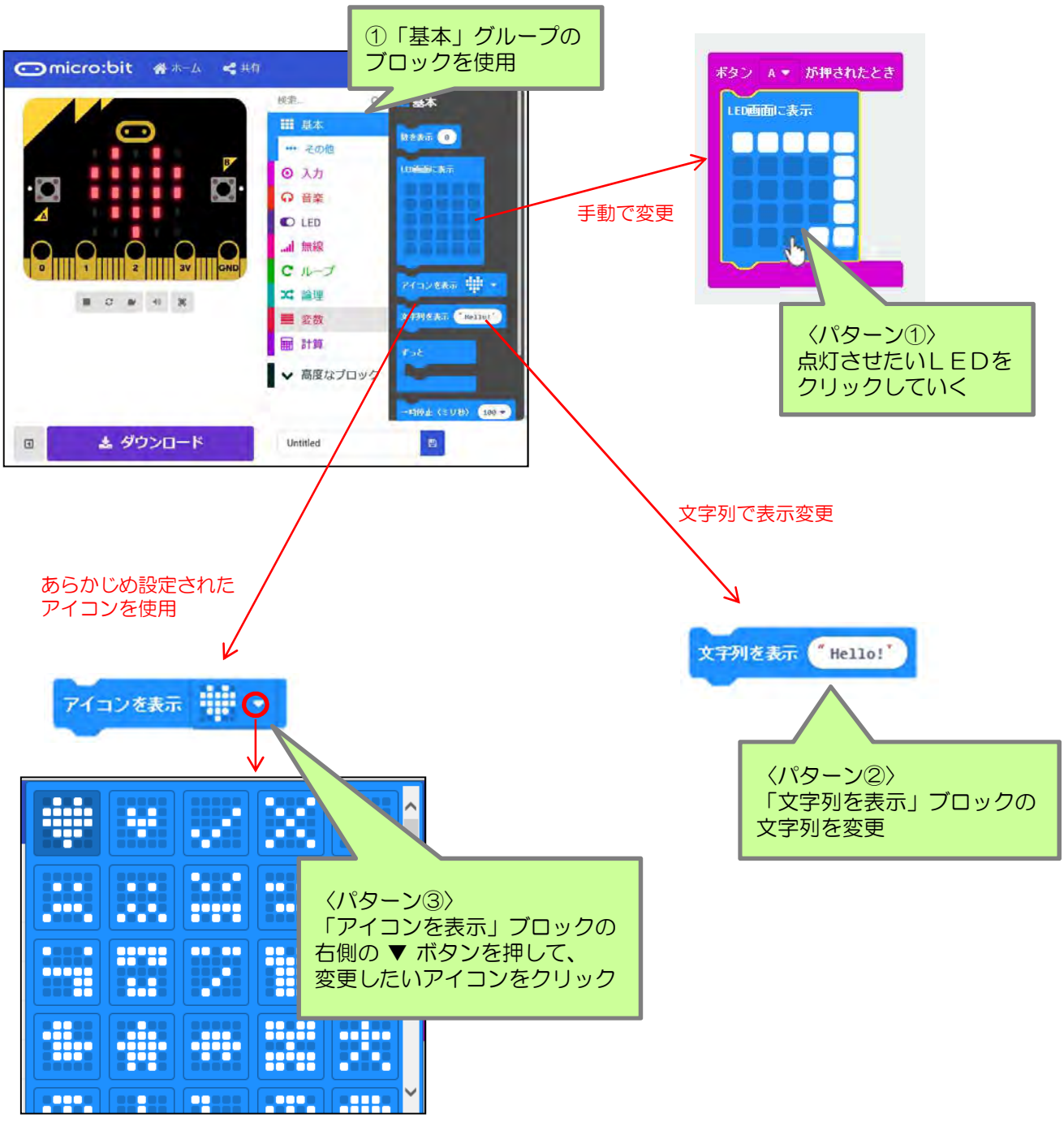
⑤－２

ワークシートの課題：　３）加速度センサーを用いて、ゆさぶられたときに、ＬＥＤ表示が変わるプログラム

〈プログラム例〉

ＬＥＤ表示の変え方は、次ページで説明

〈ＬＥＤ表示の変え方〉


①「基本」グループの
ブロックを使用
手動で変更
ボタン A が押されたとき
LED画面に表示
〈パターン①〉
点灯させたいＬＥＤを
クリックしていく
文字列で表示変更
あらかじめ設定された
アイコンを使用
文字列を表示 "Hello!"
〈パターン②〉
「文字列を表示」ブロックの
文字列を変更
アイコンを表示
〈パターン③〉
「アイコンを表示」ブロックの
右側の ▼ ボタンを押して、
変更したいアイコンをクリック

⑤－３　ワークシートの課題：　４）端子「P0」がタッチされたときにＬＥＤ表示が変わるプログラム

〈プログラム例〉

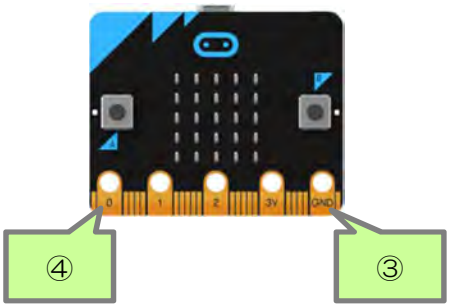
・エディター上で端子P0をクリックすると、「端子P0がタッチされたとき」（プログラム例右側）のプログラムが動く
・端子の電圧を0（0V の意味）から 1023（3.3V の意味）までの間の整数として返される　※画面では852

●実機でプログラムを実行する場合

①プログラムのダウンロード
②電池ボックスの接続
　→ ③－２　③－３　参照

③右手で「GND」端子をつまむ

④左手で「ＰＯ」端子をつまみ、左手を離す
　→　ＬＥＤ表示が変化することを確認



※マイクロビットの「端子Ｐ０がタッチされたとき」ブロックは、タッチする前に片方の手で「GND」端子も触っていないとプログラムは動作しない。
　また、左手で端子Ｐ０を触って、端子を離すまでの動きを「タッチ」としているので、タッチしたままだとＬＥＤ表示は変わらない。

※マイクロビットに保護ケースを装着して素手で触れない場合は、ミノムシリード線を使って素手と同じ要領でタッチする。

⑤－４　ワークシートの課題：　５）明るさの値の変化を調べる

〈プログラム例〉

●実機でプログラムを実行する場合

①プログラムのダウンロード
②電池ボックスの接続
} ③－２　③－３　参照

③いろいろな場所で明るさの値を調べる

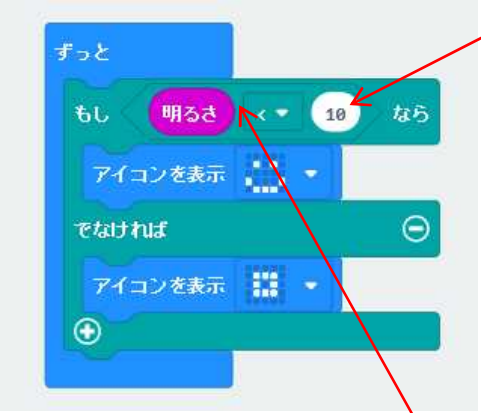
・値が大きいほど明るく、小さいほど暗い

・次の課題のために、机上で手をかざしたときの明るさも調べておく

⑤－５

ワークシートの課題：　６）何もしていないときは「小さい四角」が表示されていて、Micro:bitの上で手をかざしたときに「うれしい顔」が表示されるプログラム

〈プログラム例〉



条件分岐する明るさ（例では10）は、部屋の明るさなどによって異なるので、手をかざした時の明るさを先に調べておく

明るさが１０以下になったら（暗くなったら）うれしい顔を表示

明るさが１０以下ではない（明るくなったら）小さい四角を表示

「論理」グループの２つのブロックと、「入力」グループの「明るさ」ブロックを組み合わせる

〈　参　考　〉

このプログラムだと、うれしい顔が表示されてもすぐに小さい四角が表示されるため、手をかざすと、うれしい顔と小さい四角が交互に表示される

プログラムが繰り返されるたびに小さい四角を表示

明るさが１０以下になったら（暗くなったら）うれしい顔を表示

３時間目：計算機を作ろう（Studuino）

①使用するソフトウェア

Studuino（スタディーノ）

- 「ロボット」モードでは、アーテックロボを動かすためのプログラミングを行う（中学校３年生編で使用）
- 「キャラクター」モードはScratch（スクラッチ）1.4と同等で、パソコン画面上でのプログラミングを行う（中学校１年生編、２年生編で使用）
- 事前にダウンロード・インストールが必要

iPad

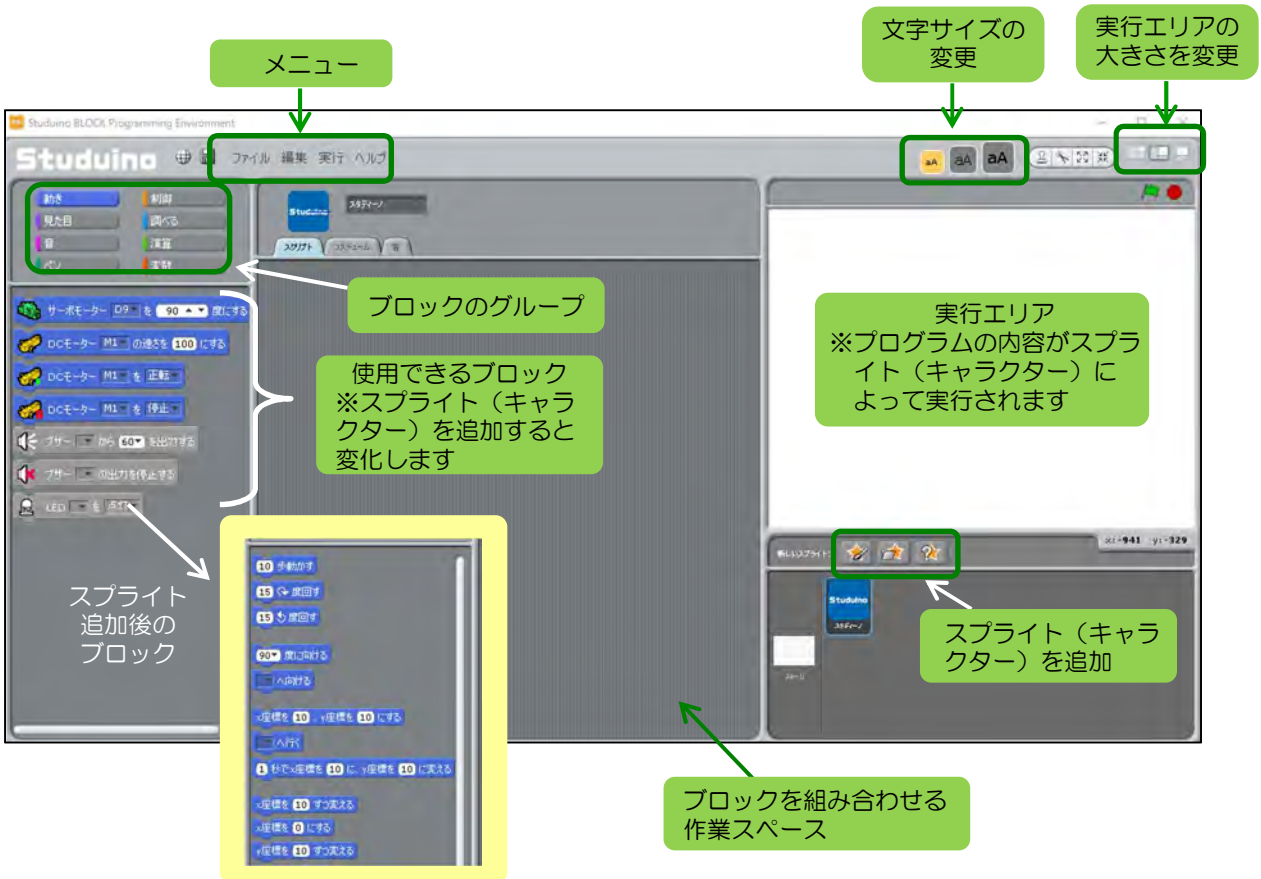
※iPadの場合はアプリケーション「Tickle（ティックル）App for Studuino」を使用
【アーテック社 取扱説明書掲載URL】 https://www.artec-kk.co.jp/studuino/ja/tickle.php
※ただし、「キャラクター」モードは無いため、他のソフトウェアでの対応が必要（「ロボット」モードのみ可）

②ソフトウェアの操作方法

②－１

Studuino（スタディーノ）の操作画面（キャラクターモード）

※iPadの場合はScratch 3.0（https://scratch.mit.edu/）などで対応



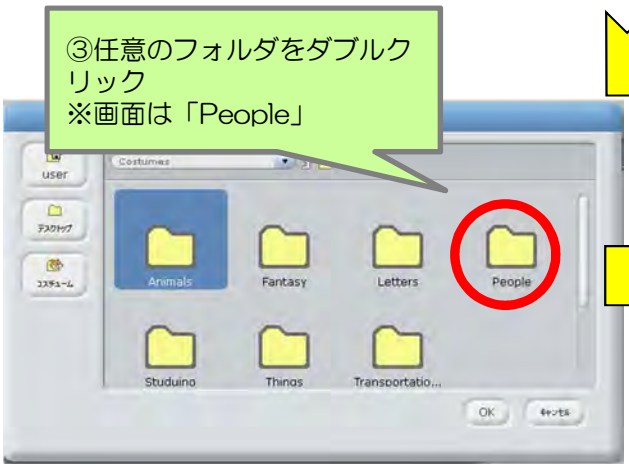
②－２

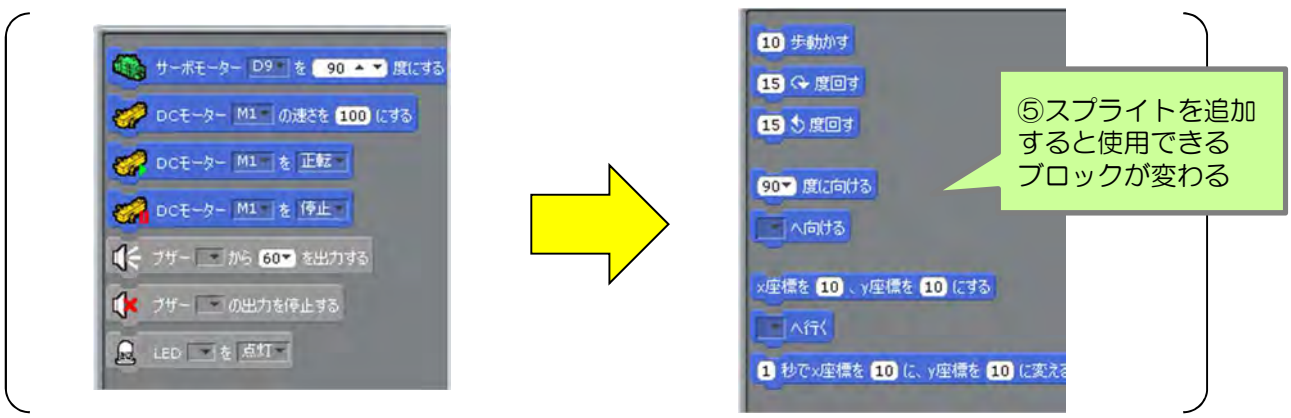
スプライトの追加方法

※スプライト＝命令を与えるキャラクター

例：roundman

●あらかじめ用意されているスプライトを使用する例





③演算子を用いたプログラム

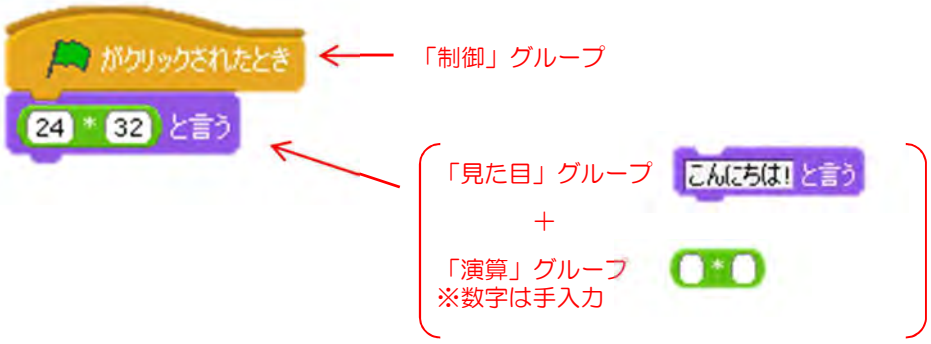
③－１　スプライトを追加　※スプライト＝命令を与えるキャラクター

②－２ 参照

例：roundman

③－２　プログラムの作成

〈プログラム例〉



〈実行エリア〉

④演算子に変数を組み合わせたプログラム

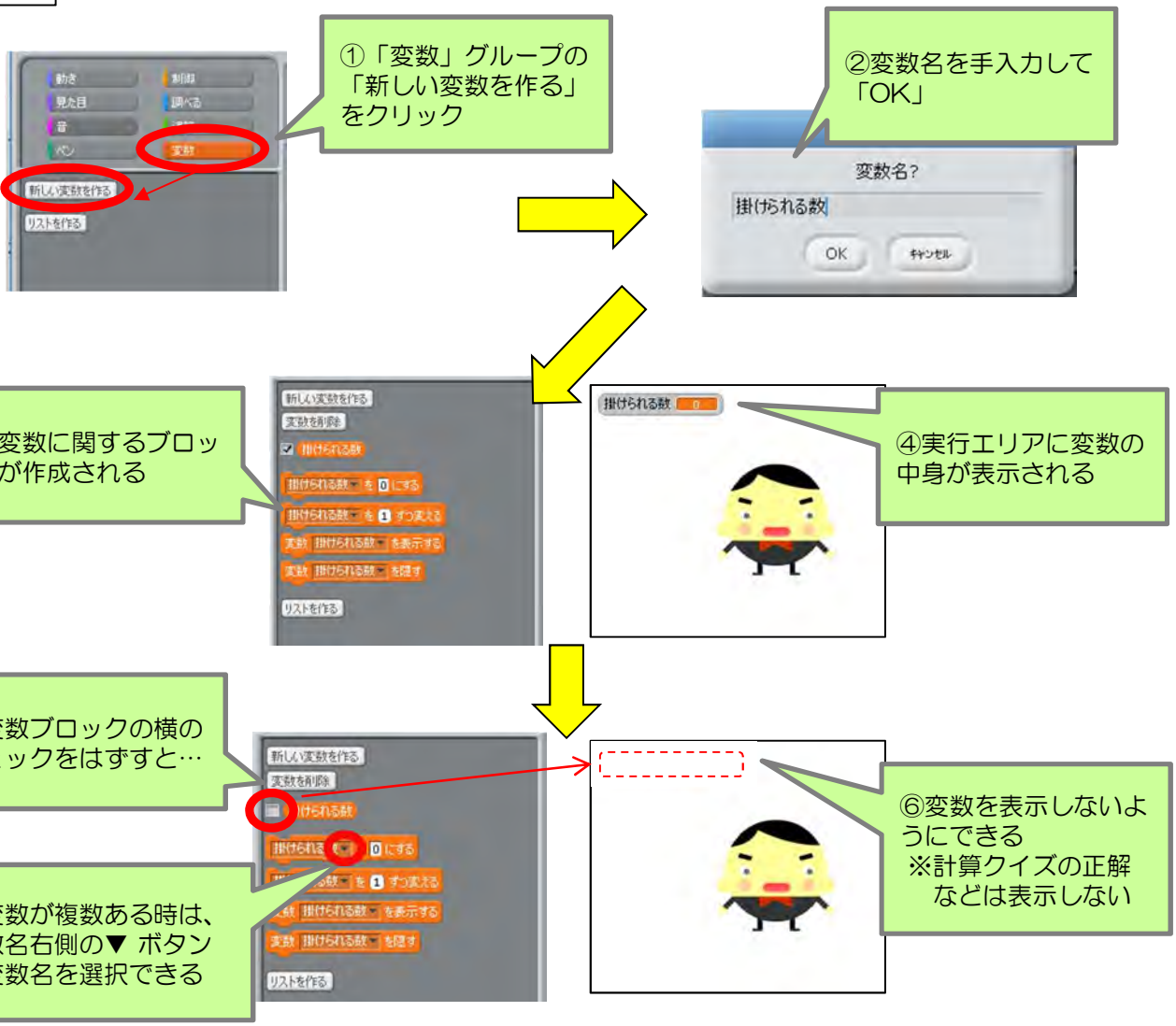
④－１　変数とは

- 「変数」とは、プログラムの実行にともなって変化していく情報（数値や文字）を格納しておく箱のことです。
- それぞれの箱に名前をつけることができます。

〈イメージ〉

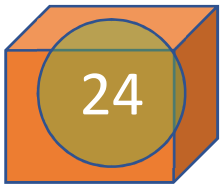
【変数】掛けられる数　　【変数】掛ける数

④－２　変数の作り方

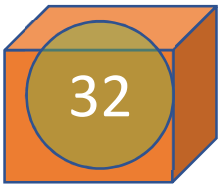


④－３　変数の有用性

・箱の中身を変えるだけで，関係する複数の動作をすべて変化させることができます

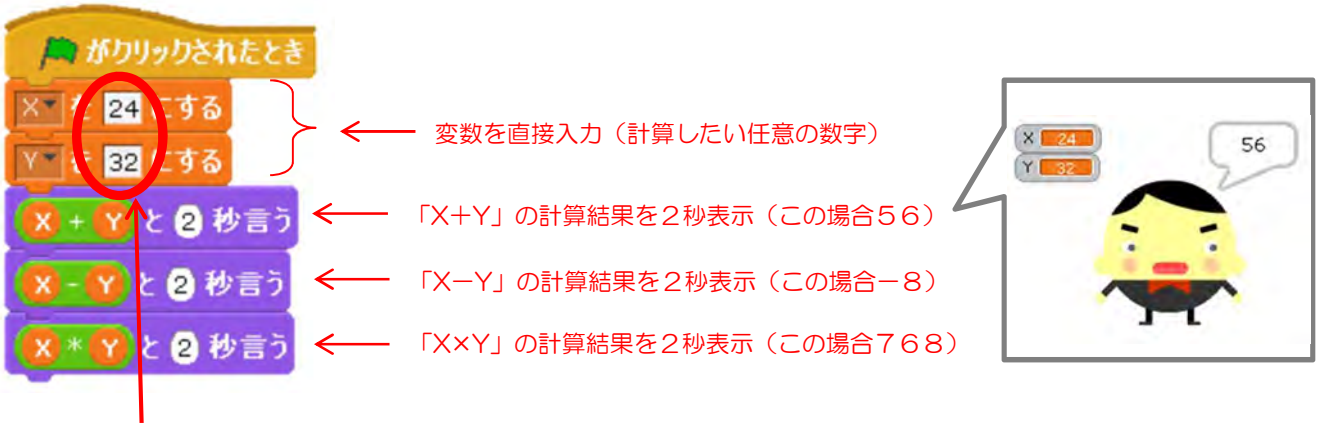


【変数】X



【変数】 Y

名前がXとYの変数

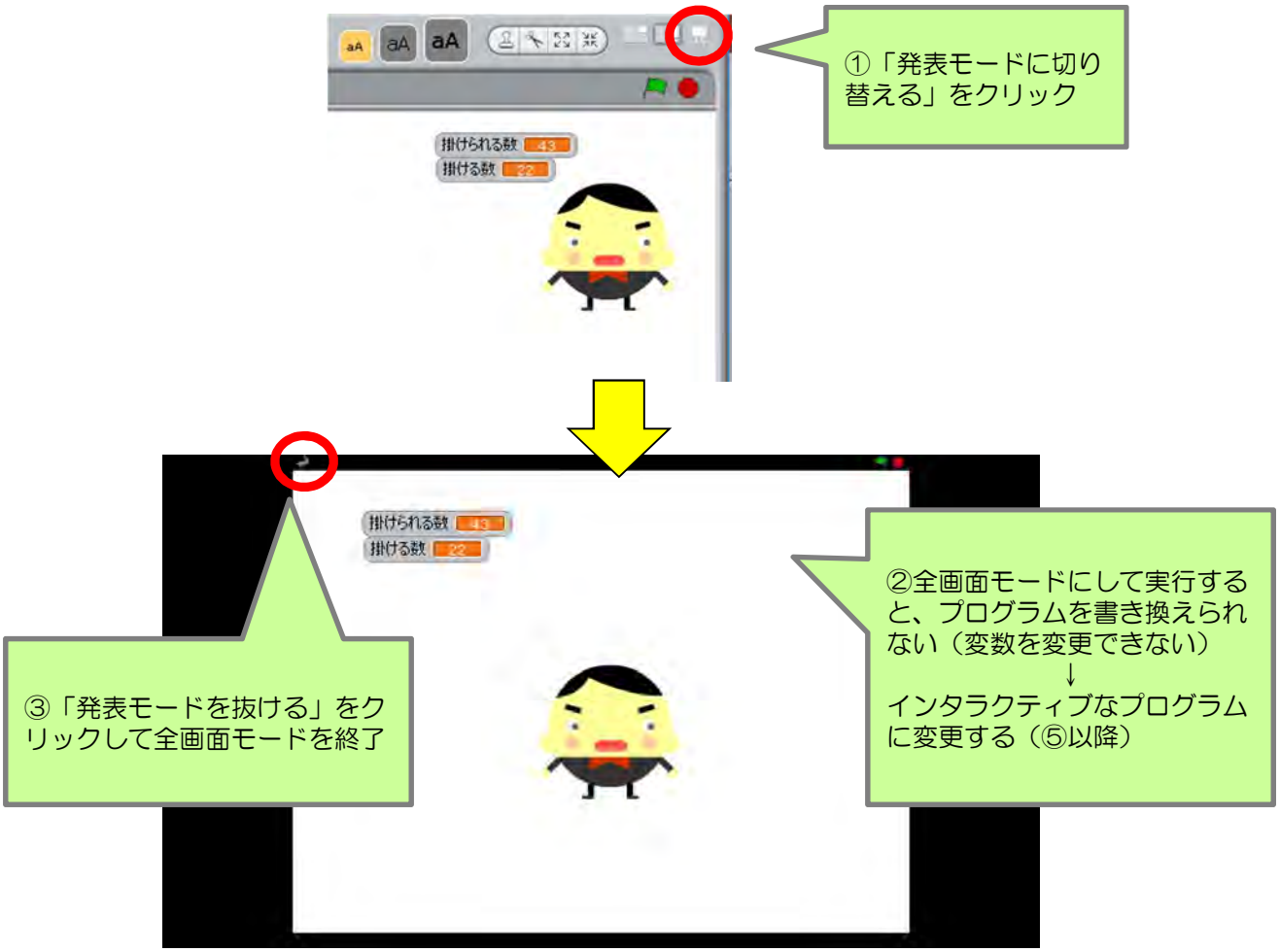


●変数（X、Y)の中身だけを入れ替えれば色々な計算ができる

④－４　変数を用いたプログラム

〈プログラム例〉

④－５　全画面表示


①「発表モードに切り替える」をクリック
掛けられる数 43
掛ける数 22
②全画面モードにして実行すると、プログラムを書き換えられない（変数を変更できない）
↓
インタラクティブなプログラムに変更する（⑤以降）
③「発表モードを抜ける」をクリックして全画面モードを終了

⑤インタラクション（相互作用）

⑤－１　概要（イメージ）

●実行時にユーザからの情報入力を受け付け，その情報によってプログラムの結果を変えるしくみ

①［　］と聞いて待つ
②［答え］

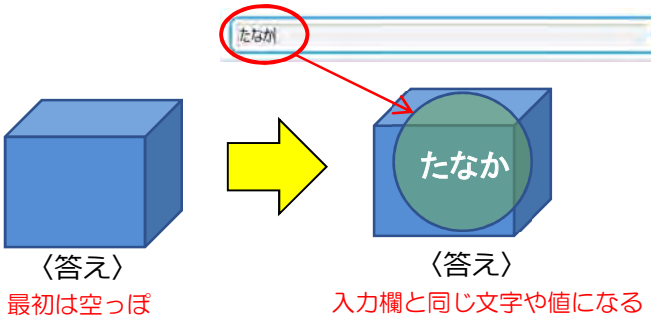
←「調べる」グループの、このブロックを使うと入力欄が表示され、好きな文章などが入力できる。

① ［　］と聞いて待つ

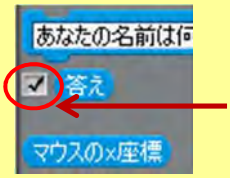
「・・・と聞いて待つ」ブロックは、**「・・・」に入力した文字列を吹き出しに表示**する。

② 答え

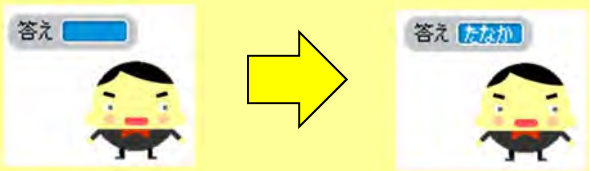
「答え」ブロックは、**入力欄に入力された文字や値が入る**、初めから用意されている変数のブロック。



●ソフトウェア上で「答え」ブロックの中身を見る方法



「調べる」グループのブロック一覧にある「答え」ブロック左のチェックを入れると、実行エリアに表示される



〈例〉

←「調べる」グループ
※四角内の値を吹き出しに表示

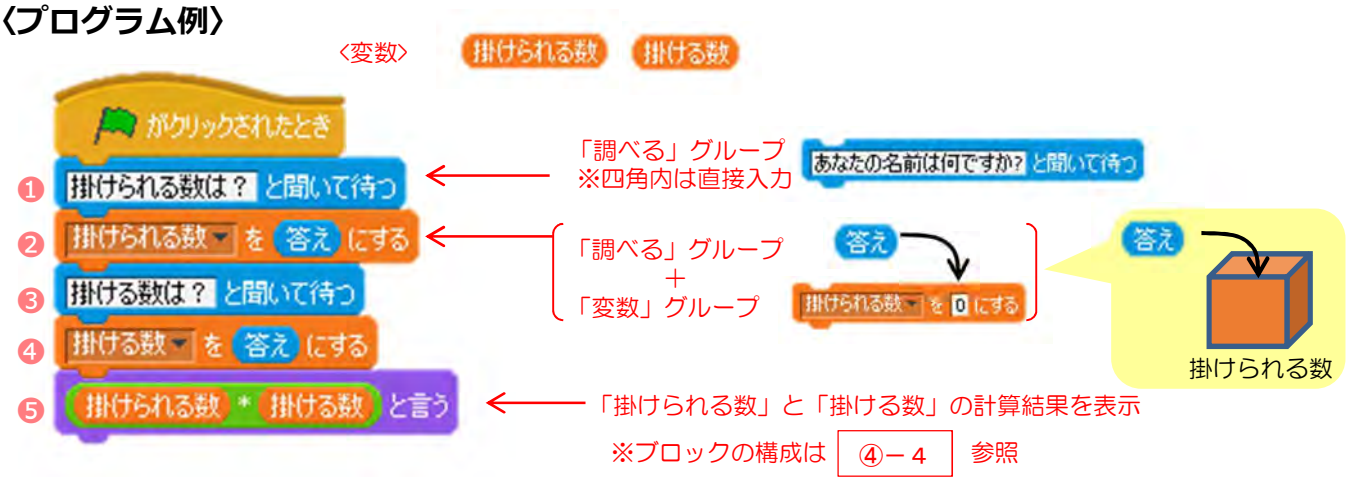
「見た目」グループ　こんにちは! と言う
＋
「演算」グループ　ハロー と ワールド をつなぐ
＋
「調べる」グループ　答え

「たなか」と入力すると…

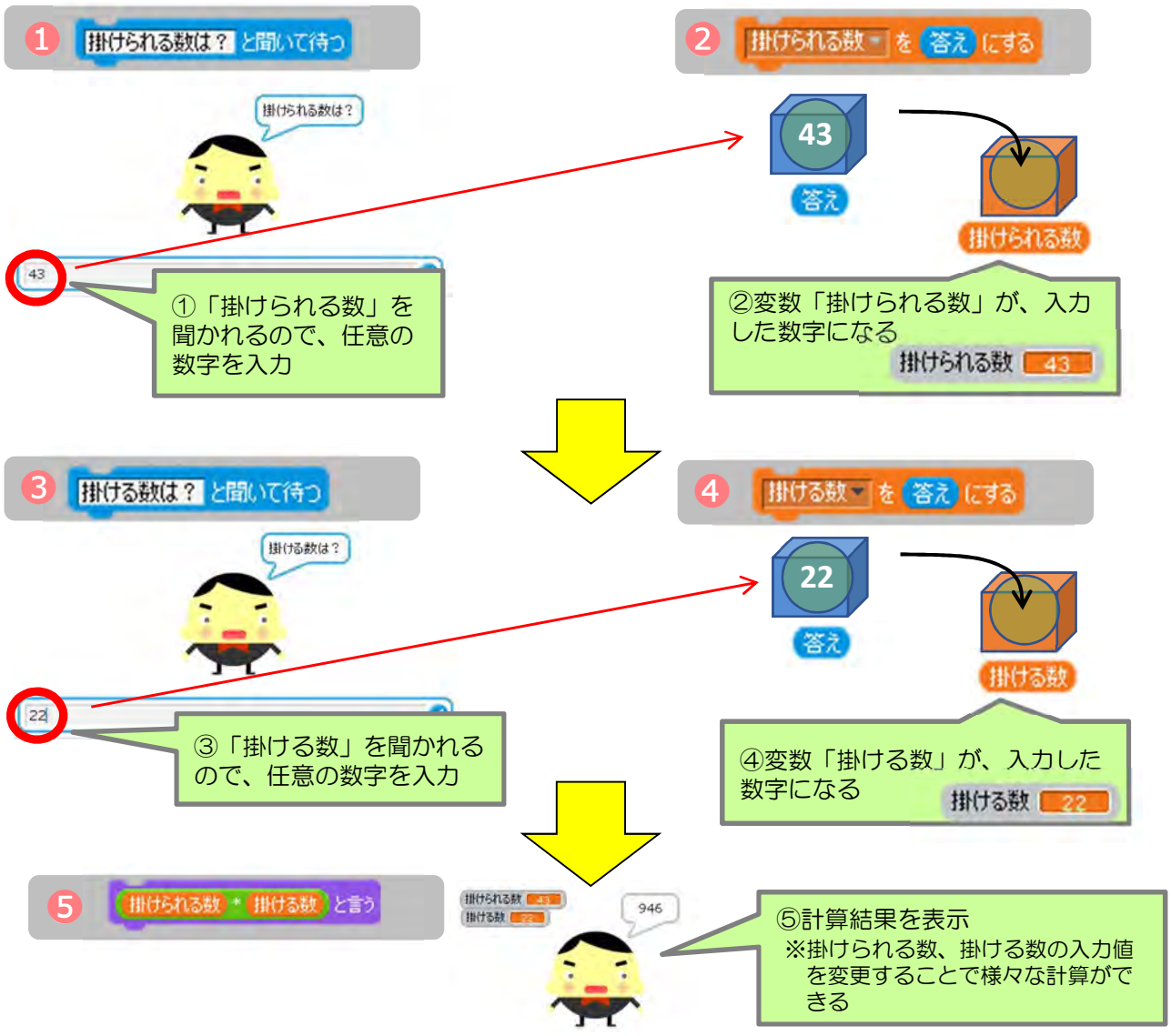
⑤－２　インタラクティブなプログラム

- ④－４ で作成したプログラムを変更していく

〈プログラム例〉

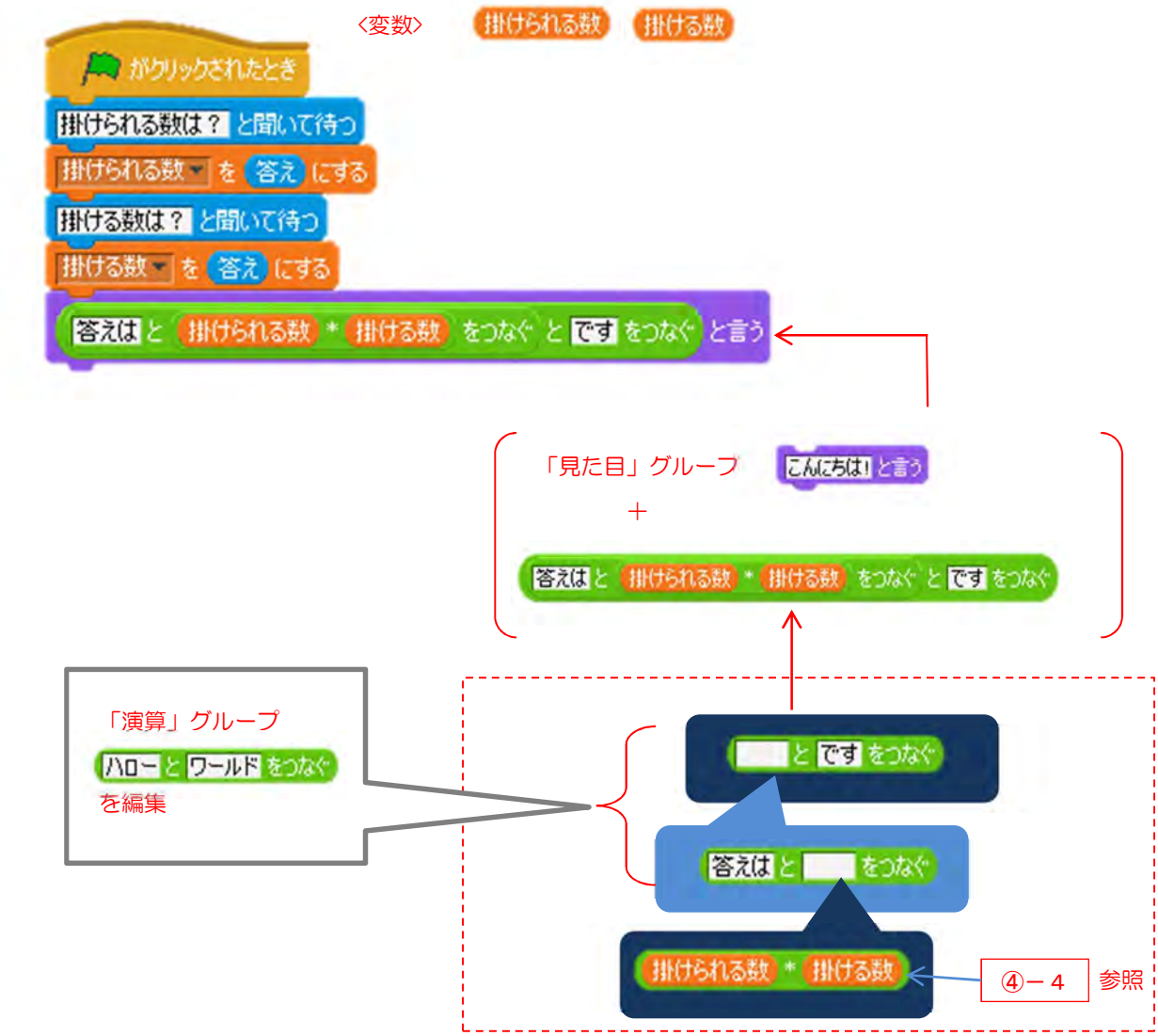


〈プログラムの実行〉



⑤－３　全画面表示　④－５ 参照

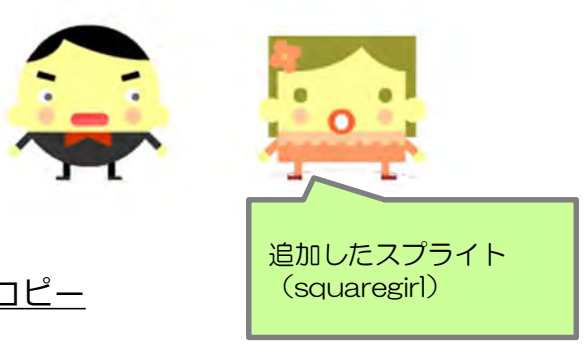
⑤－４　文字列連結ブロック（「答えは～です」と表示させる）



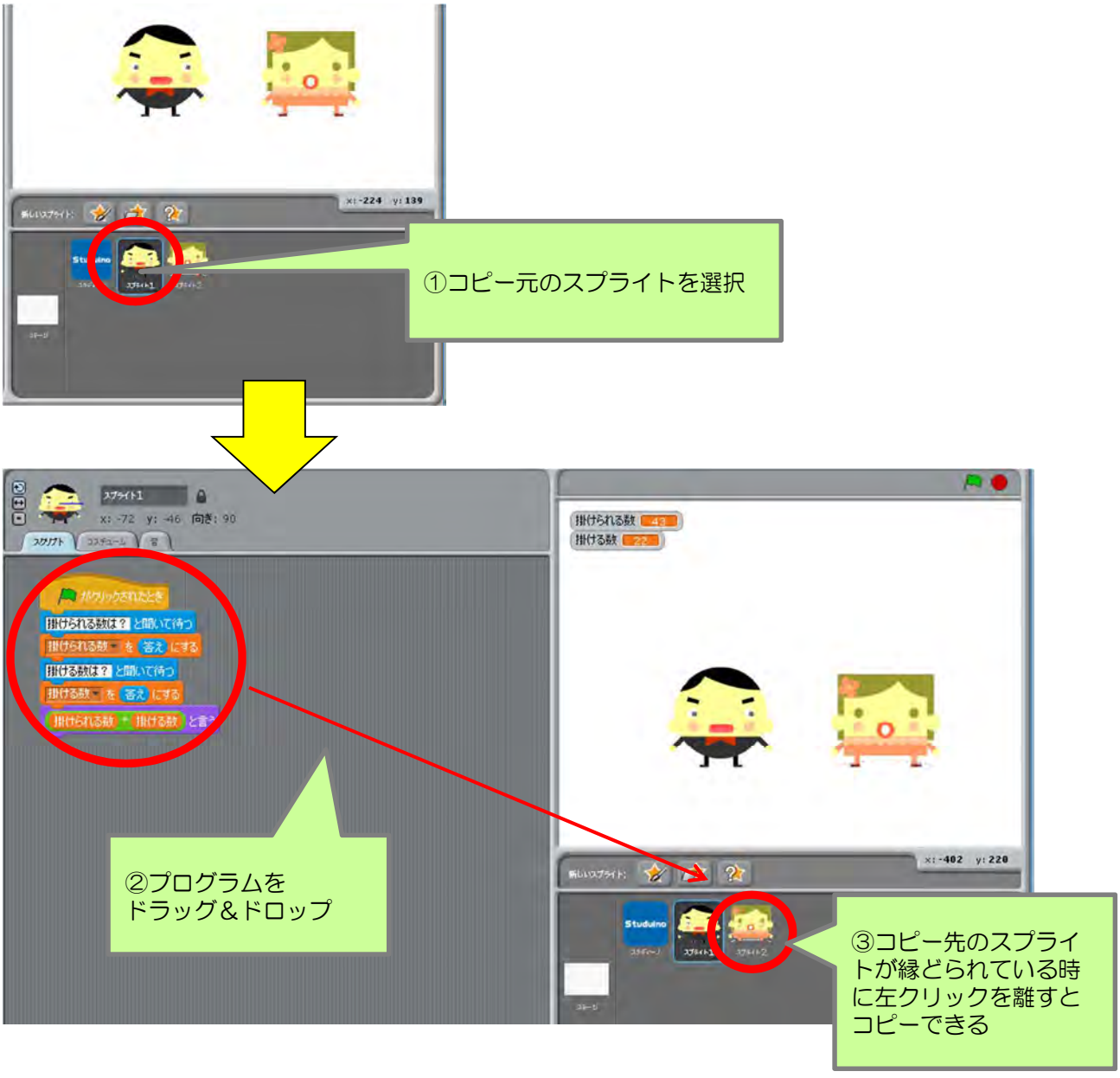
⑥他の計算（足し算や引き算）をするプログラムの作成

⑥－１　スプライトを追加　※スプライト＝命令を与えるキャラクター

②－２ 参照



⑥－２　プログラムをスプライト間でコピー



⑥－３　割り算のプログラム例

①「スプライト２」を選択

スプライト１からコピーされたプログラムがあるので、これを編集していく

②割り算に用いる変数「割られる数」「割る数」を作る

④－２ 参照

〈変数〉 割られる数 割る数

「制御」グループのブロックを変更
※プログラム開始の合図をスプライト１と別にしておく

直接入力により質問を変更

変数名右側の▼ボタンから変数を「割られる数」に変更

直接入力により質問を変更

変数名右側の▼ボタンから変数を「割る数」に変更

演算を変更（割り算は「／」で表す）

スプライト２をクリックするとプログラムがスタート

４時間目：計算クイズを作ろう（Studuino）

①乱数の使い方（指導過程１）

①－１

Studuino（スタディーノ）を起動し、スプライトを追加

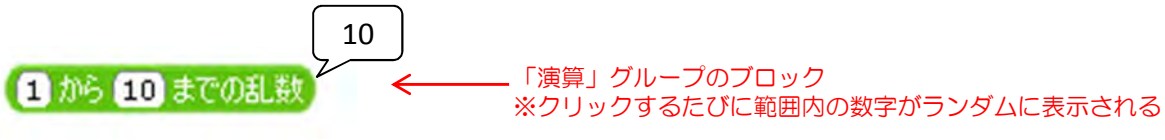
※スプライト＝命令を与えるキャラクター

中学校１年生編の３時間目 ②－２ 参照

例：roundman

①－２

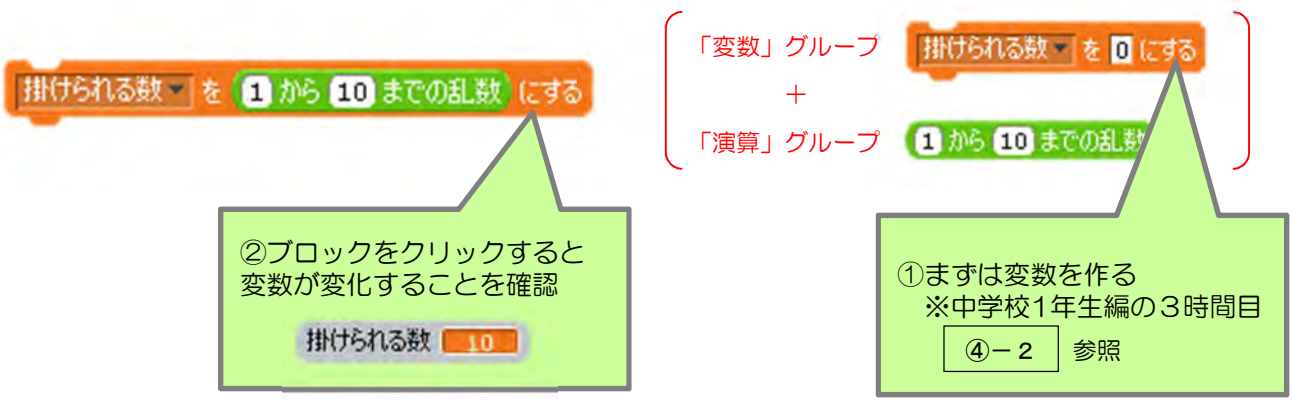
乱数の使い方



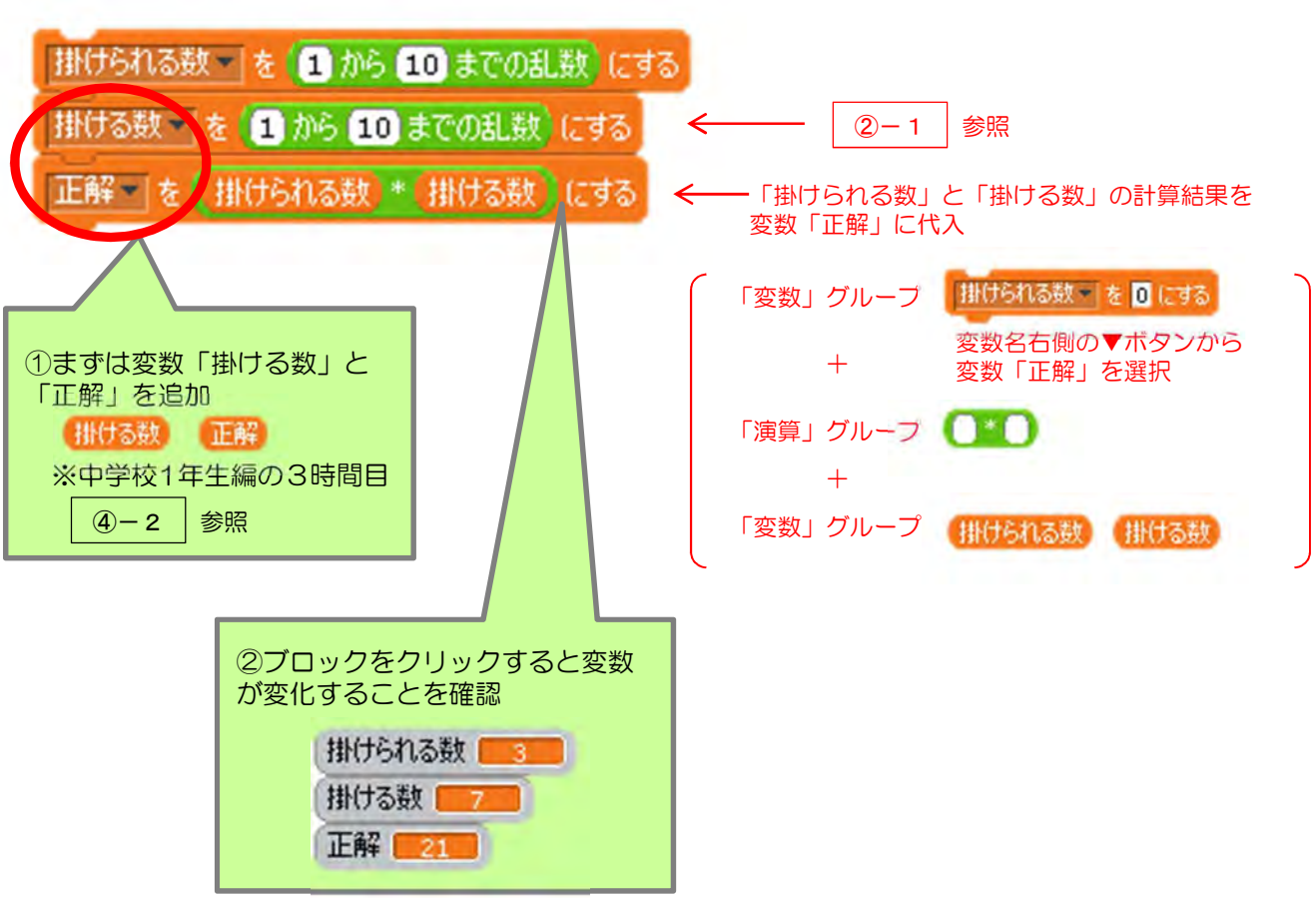
②乱数を用いた演算（指導過程２）

②－１

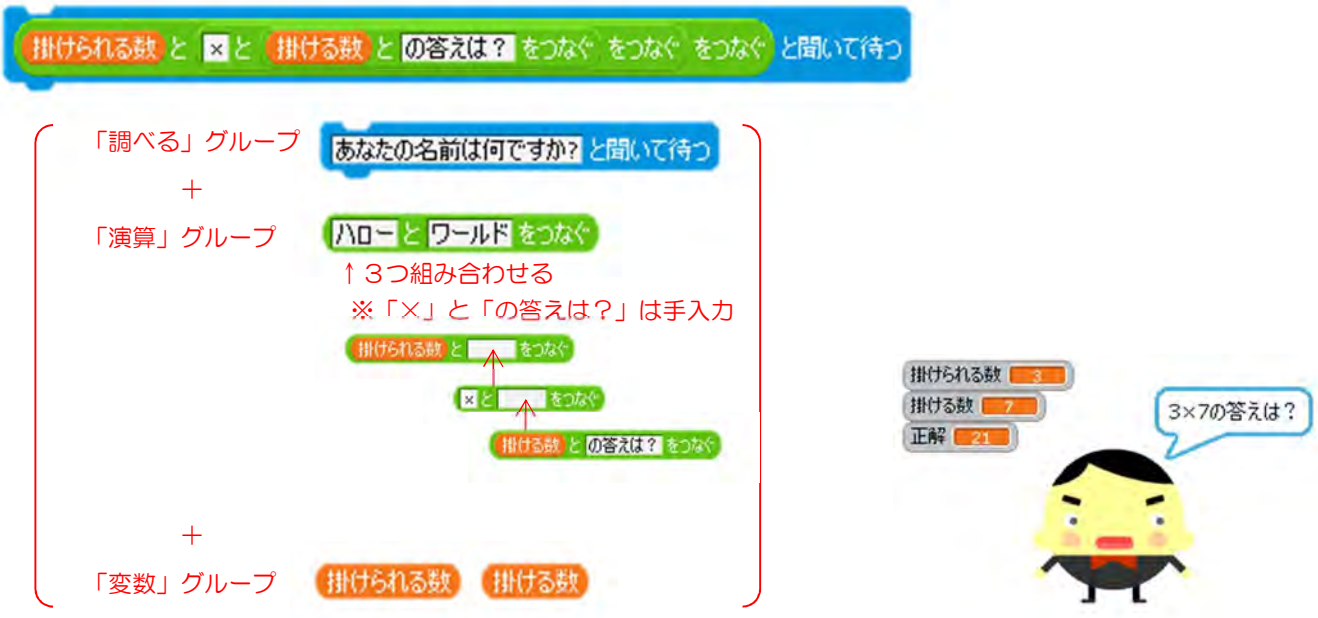
変数を乱数にする



②－２　乱数を用いた計算式



③キャラクターにクイズを出させる（指導過程３）

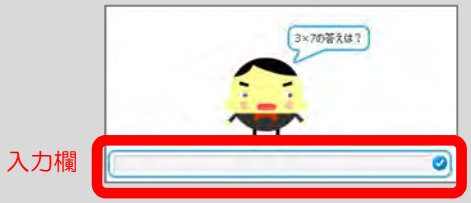


④正解と不正解の判別（指導過程４）

・「入力欄に入力された値」（答え）
が
「変数『正解』」と等しいかどうかで条件分岐
※③の場合は変数「正解」は２１ 正解 21

・等しければ「正解！」と言い、
等しくなければ「間違い！」と言う

答え 「答え」ブロックは、入力欄に入力された文字や値が入る、初めから用意されている変数のブロックです。



「制御」グループ

＋

「演算」グループ

＋

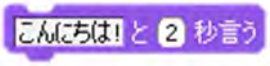
「調べる」グループ 答え

＋

「変数」グループ 正解

＋

「見た目」グループ

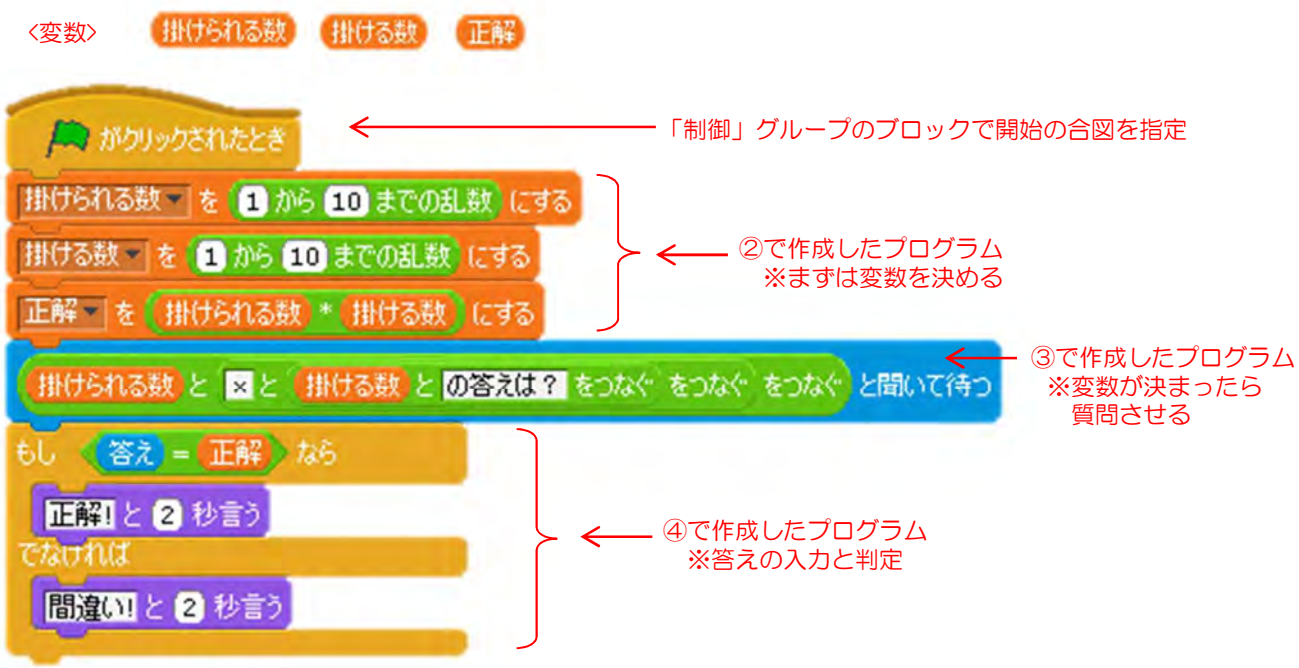


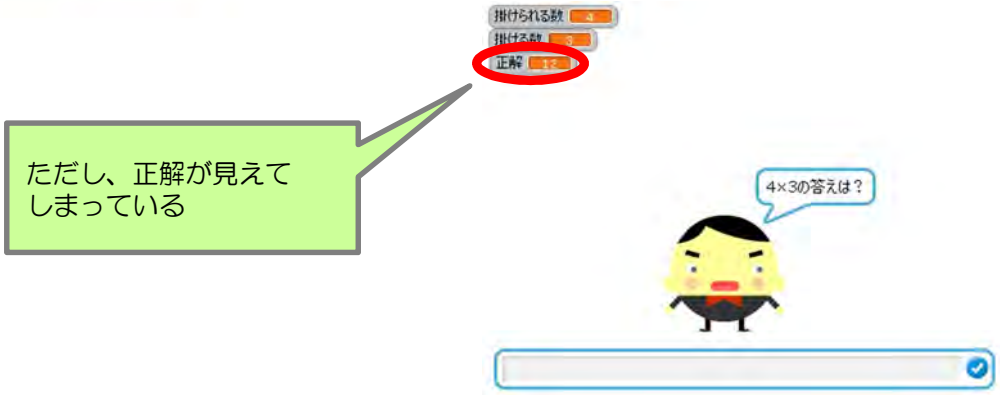
※手入力で変更

⑤計算クイズを作る（指導過程５）

⑤－１　１問だけの計算クイズ

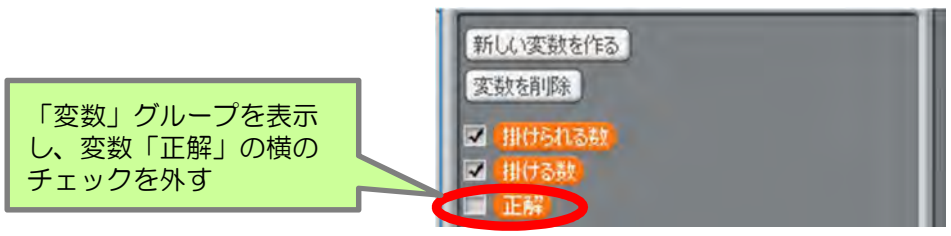
●②③④で作成したプログラムを結合する





⑤－２　クイズの改良

●正解を隠す



⑤－３　５問出題し、最後に正解数を表示させるようなプログラム

●⑤－１、⑤－２で作成したプログラムを改良する

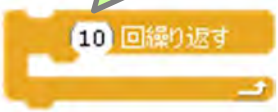
●プログラム改良の手順

・授業中のヒントとして、生徒に示しても良い（習熟度に合わせて適宜修正）

【はじめ】
・⑤－１、⑤－２で作成したプログラムをベースにする

【STEP1】
・5問繰り返すプログラムを作る

10 → 5に変更



「制御」グループのブロックを追加

【STEP２】
・「正解数」という変数を作る

「変数」グループ

【STEP３】
・「正解数」は、はじめはゼロ

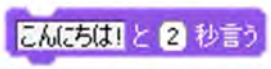
・「変数」グループのブロックを追加
※変数名右側の▼ボタンから変数を「正解数」に変更

【STEP４】
・正解するごとに、「正解数」を増やす

・「変数」グループのブロックを追加
※変数名右側の▼ボタンから変数を「正解数」に変更

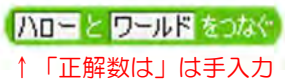
【STEP５】
・５問終わったら正解数を言う

「見た目」グループ



＋

「演算」グループ



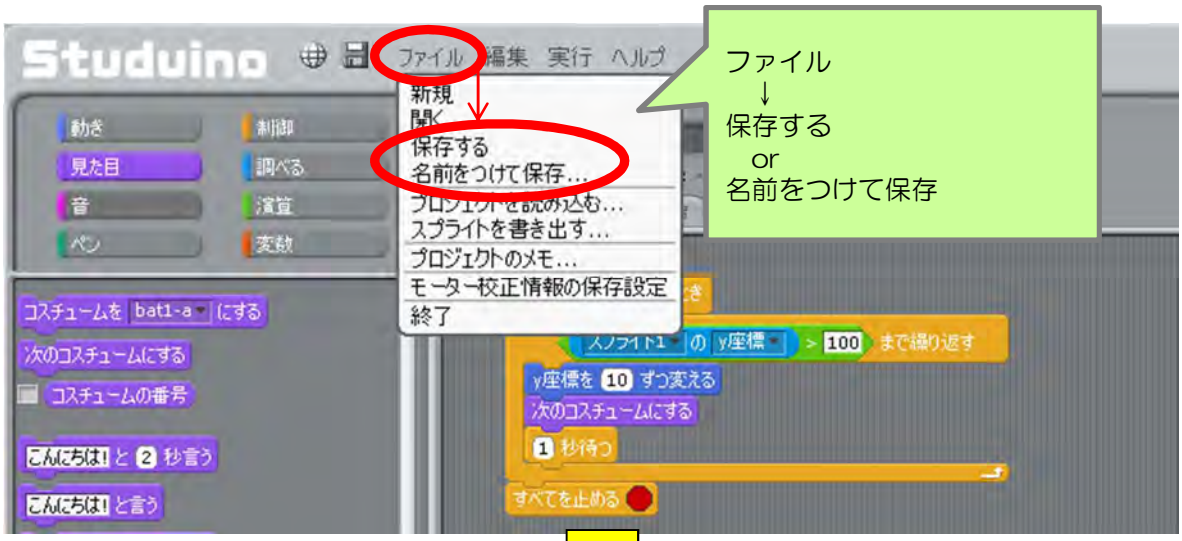
↑「正解数は」は手入力

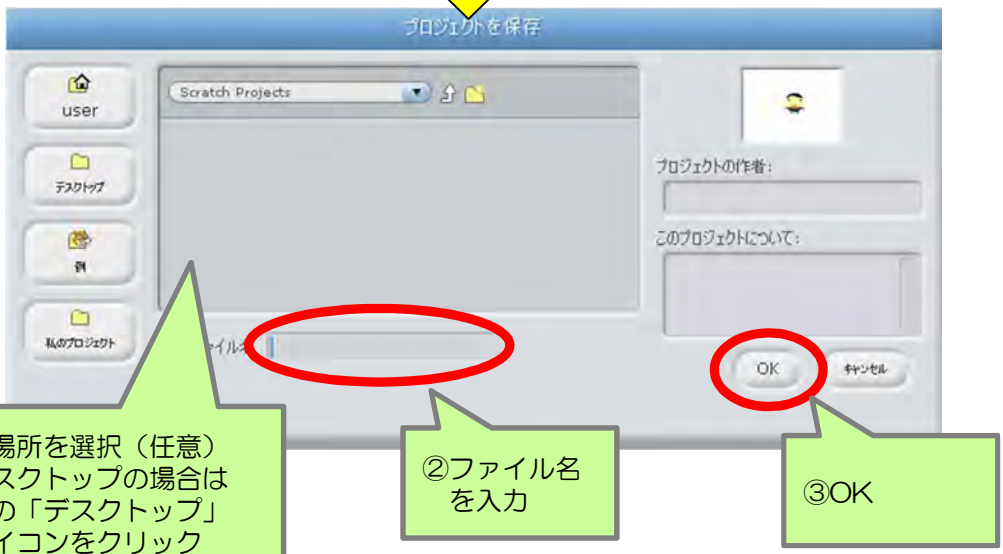
＋

「変数」グループ

⑥プログラムの保存

・プログラムを保存しておき次回以降の授業の際に読み込むことで、保存時のプログラムを使用することができる



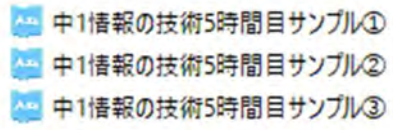


ファイルが作成される

５時間目：分かりやすく・使いやすくしよう（Studuino）

①サンプルプログラムの実行（指導過程１）

①－１　サンプルプログラム①の読み込み

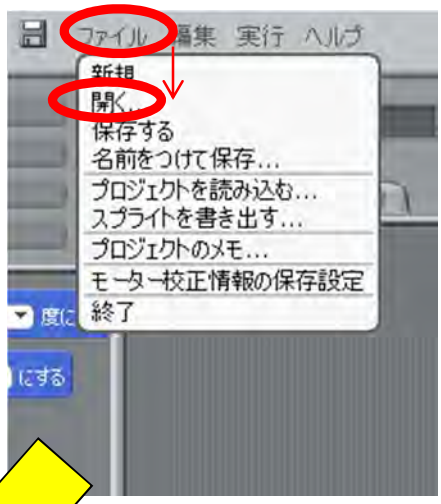


①サンプルプログラムをパソコンに保存しておく（保存場所は任意）

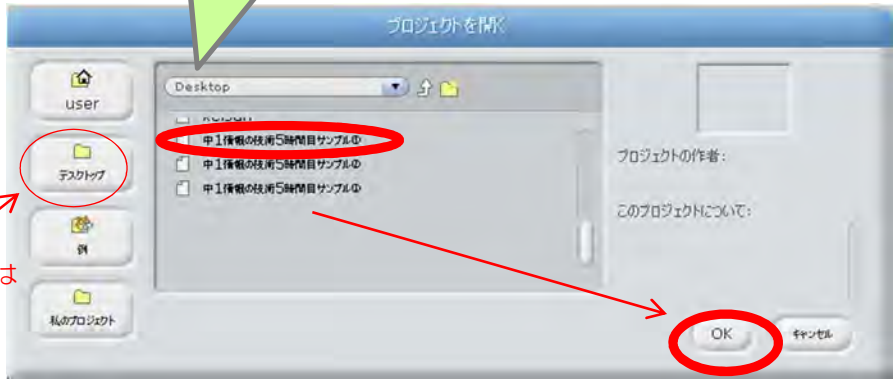
サンプルプログラムは、教育センター学びの丘の
「きのくにeラーニング」システムの「きのくにICT教育（中学校）」に掲載

②スタディーノを立ち上げ、「ブロックプログラミング環境」→「キャラクター」を選択

③「ファイル」→「開く」



④サンプルファイル①を選択して「OK」



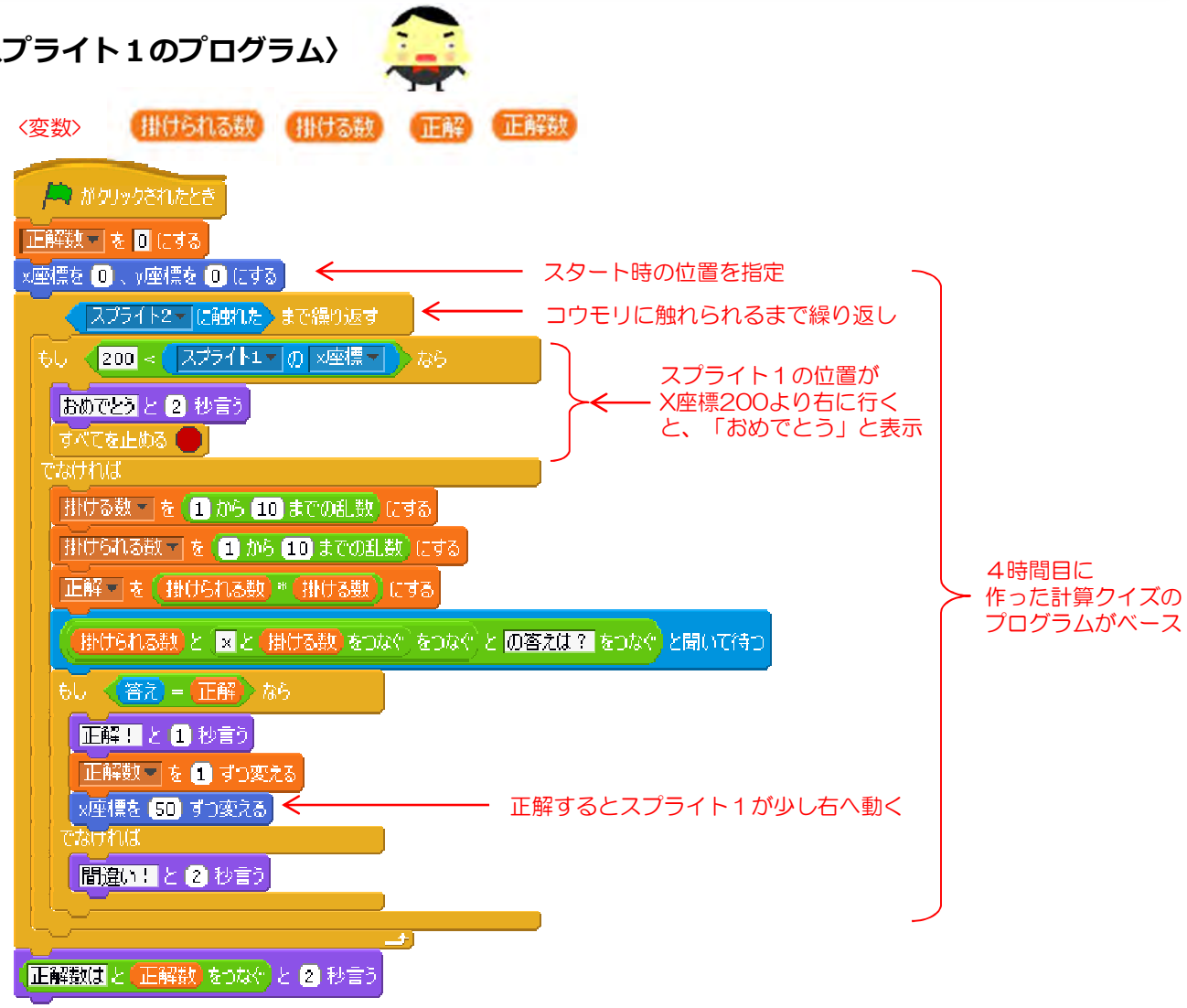
デスクトップの場合は
ここをクリック

①－２　サンプルプログラム①

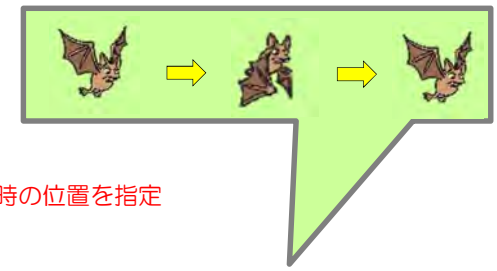
●実行すると計算クイズが出される
→　正解するとRoundman（人間のキャラクター）は右へ移動
→　コウモリが近づいてきて、触れられたらゲームオーバー

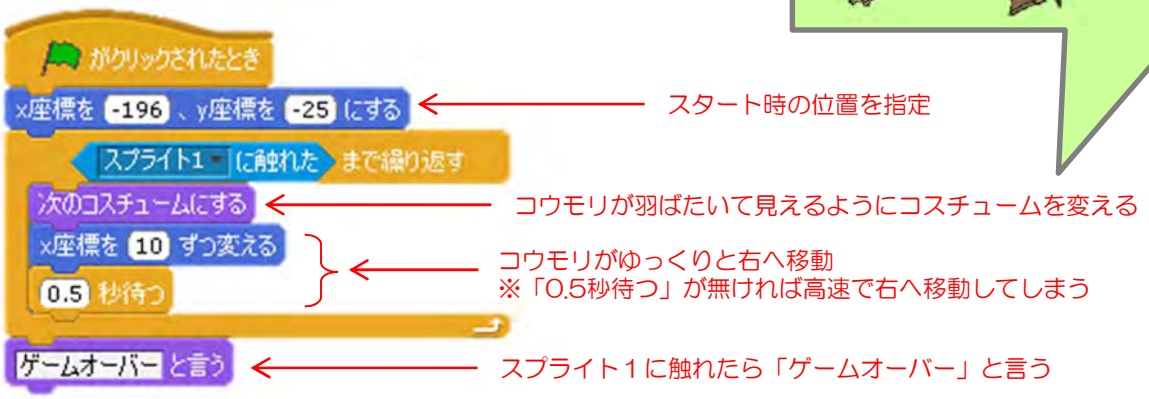
・実行して、これまで作成してきたプログラムとの動作の違いを見つけさせる

〈スプライト１のプログラム〉

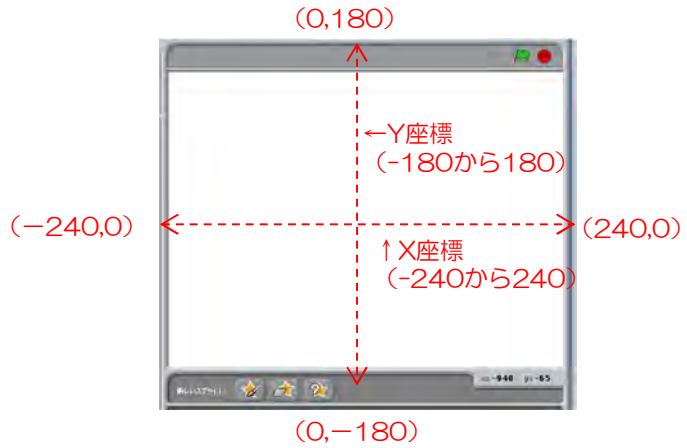


〈スプライト２のプログラム〉





〈実行エリアの座標〉



②制限時間をわかりやすくする工夫（指導過程２）

②－１　サンプルプログラム②の読み込み

※ ①－１ 参照

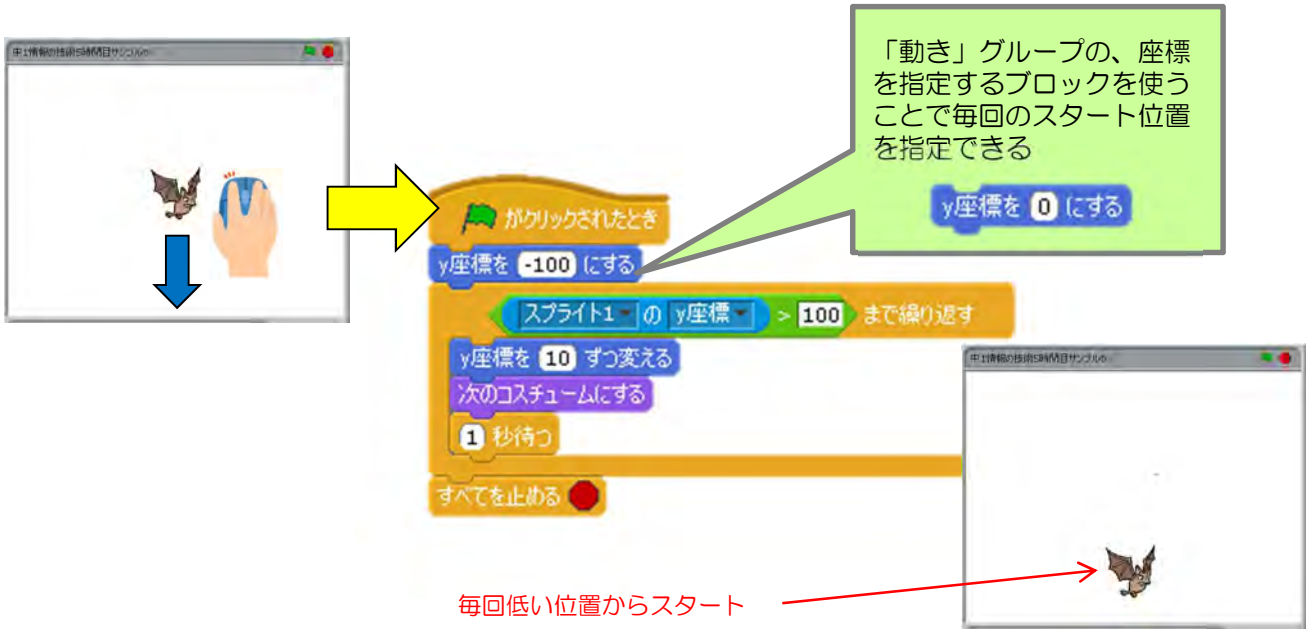
②－２　サンプルプログラム②

- 実行するとコウモリが時間とともにゆっくり上に移動
  → 上に着いたら（y座標が100になったら終わり）

・プログラムの数値などを生徒に変更させ、移動距離や速さの設定が難易度を意味することに気付かせる

〈参　考〉

- 上のプログラムのままだと、コウモリを下に戻すにはマウスでドラッグするしかないが･･･



③正解不正解のわかりやすさの課題解決（指導過程３）

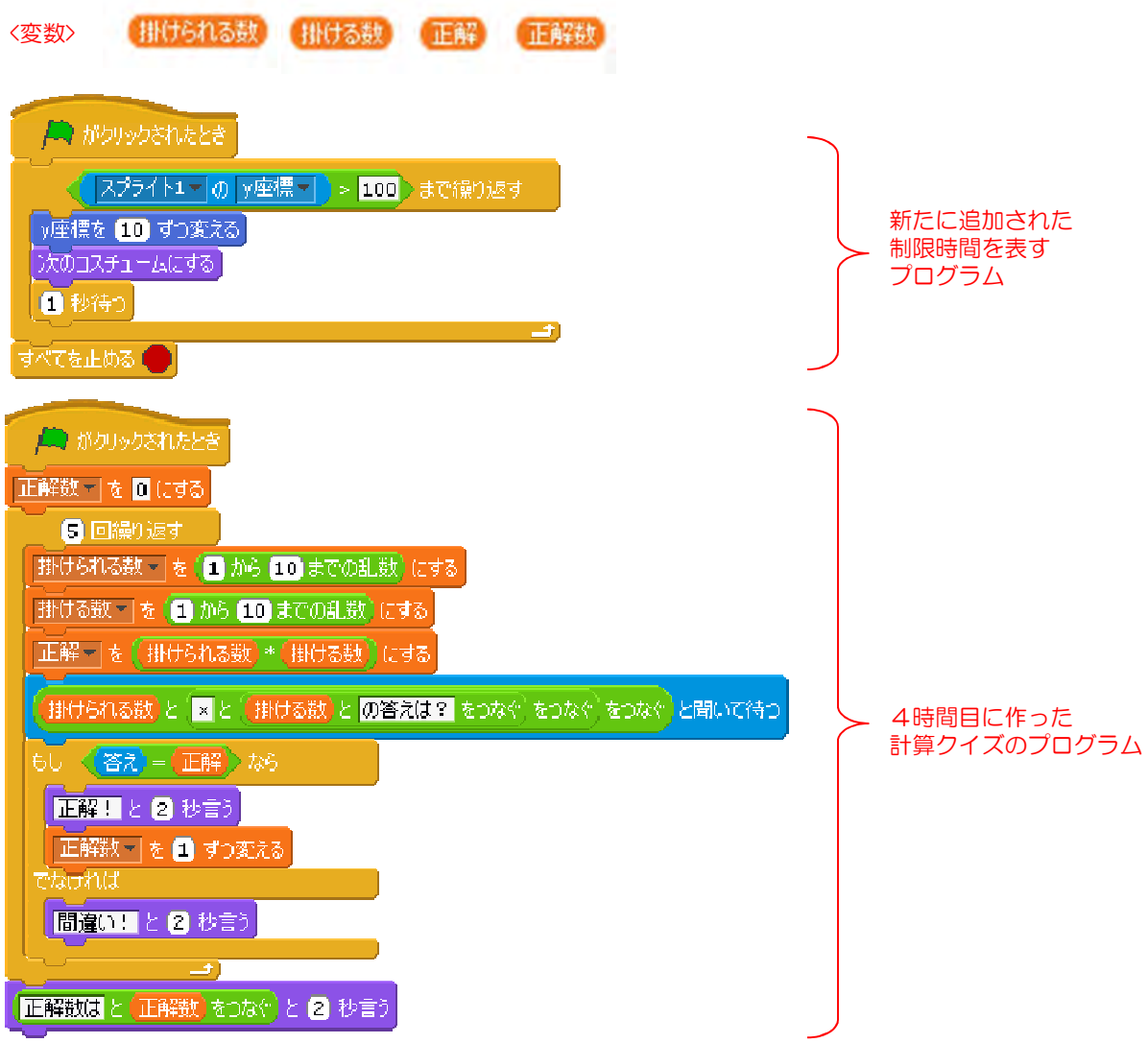
③－１　サンプルプログラム③の読み込み

※ ①－１ 参照

※生徒の状況に合わせて、サンプルプログラム①や、４時間目に保存したプログラムを読み込んでも良い

③－２　サンプルプログラム③

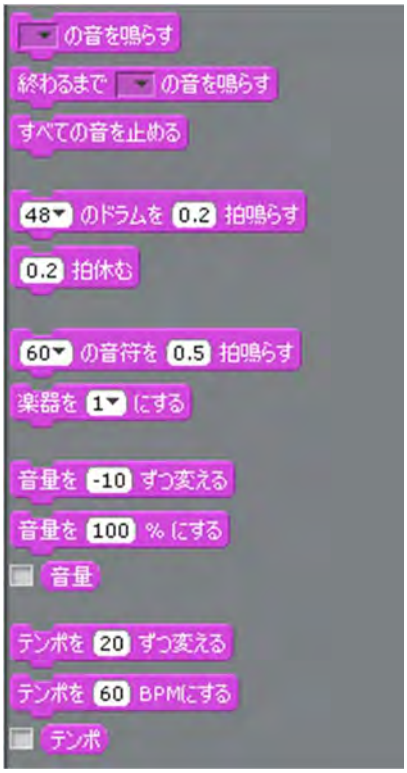
- ４時間目に作った計算クイズのプログラムにサンプルプログラム②（制限時間のプログラム）を加えたプログラム



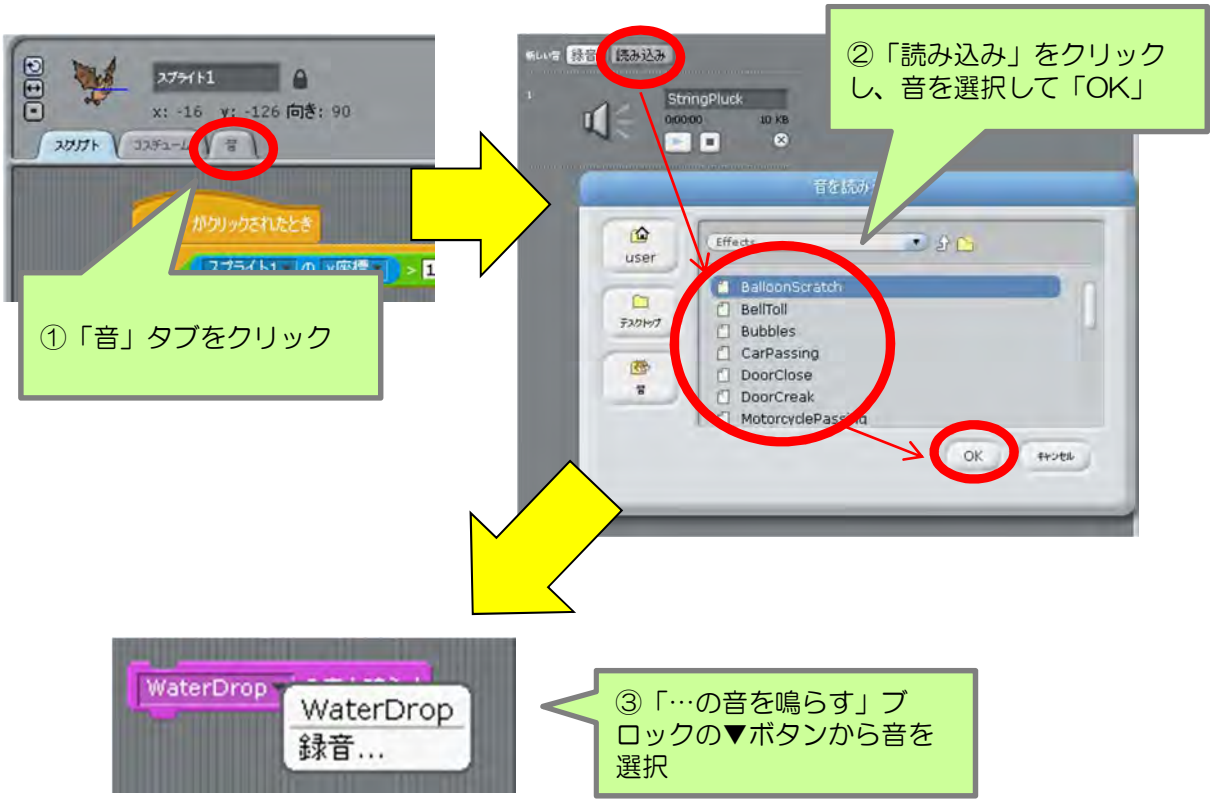
③－３　プログラムの変更例

〈例１：正解・不正解の音を追加〉

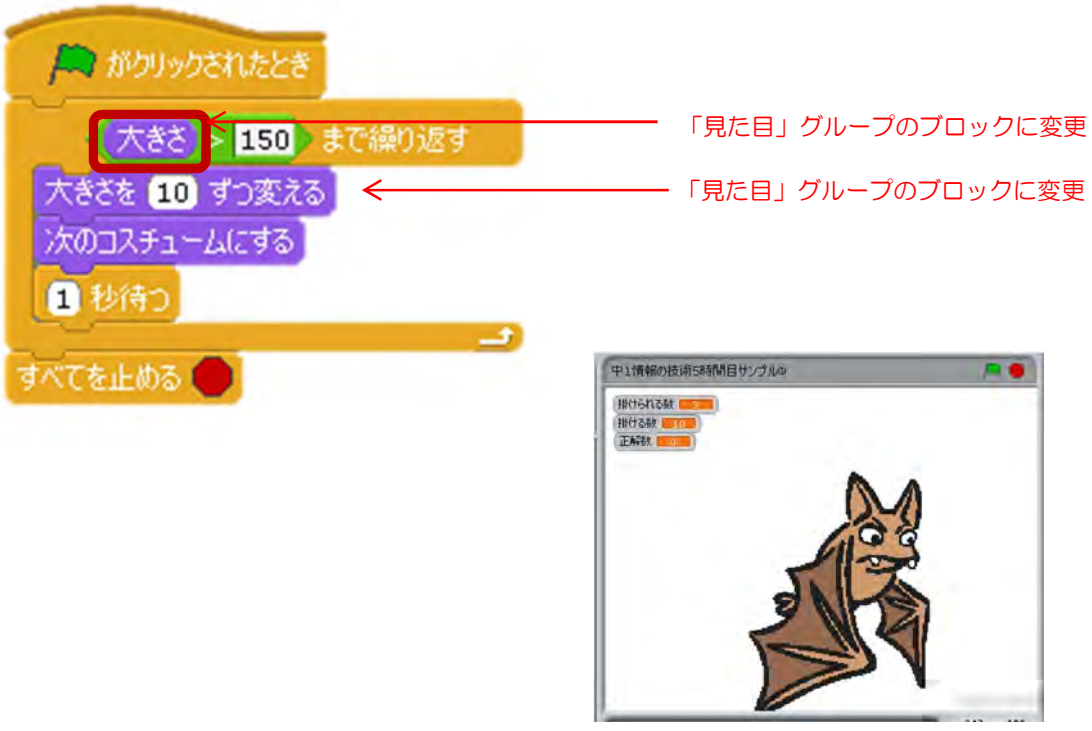
● 「音」グループのブロック



●音の変え方



〈例２：スプライトの大きさを変更〉

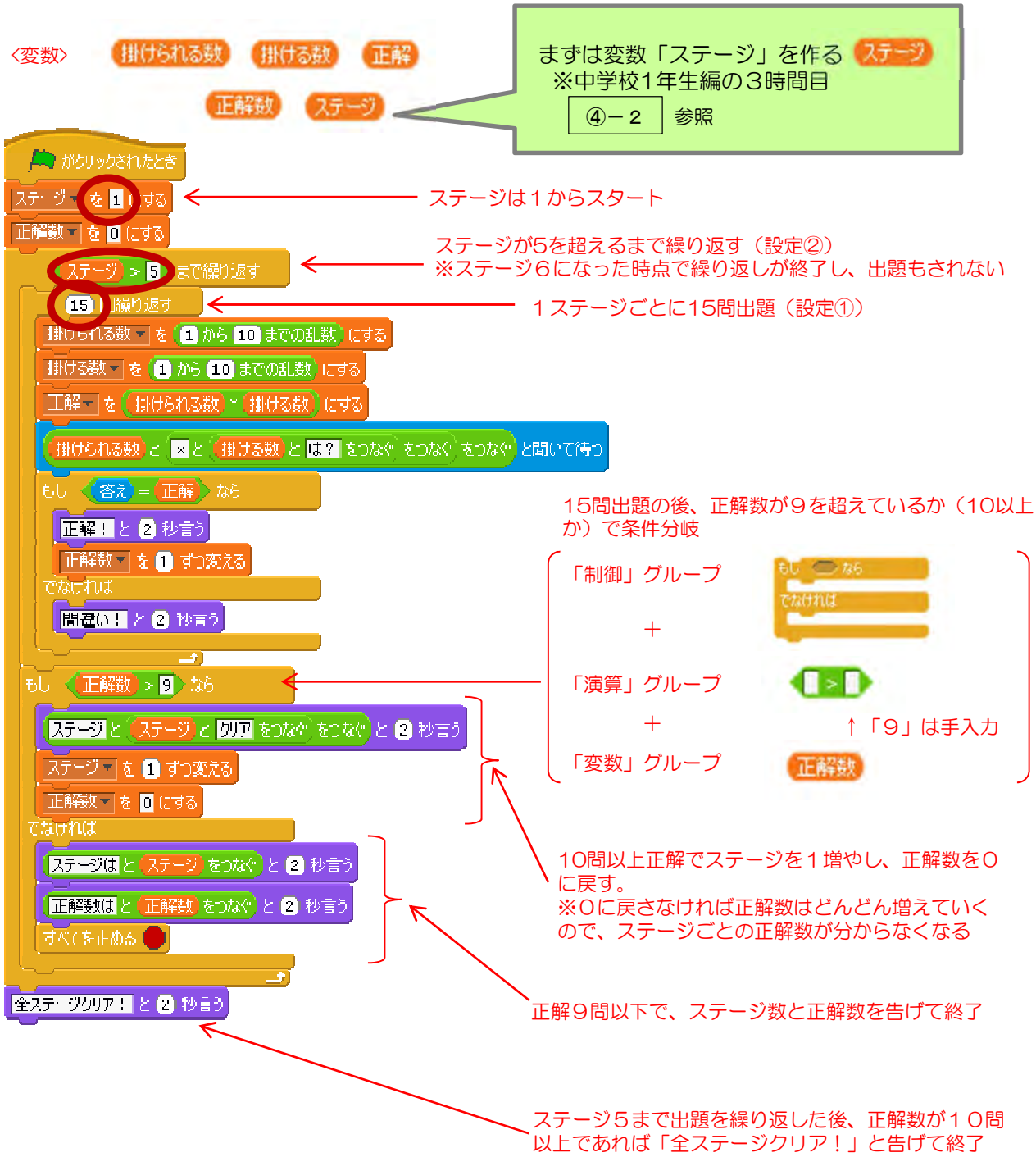


④利用者がもっとやりたいと思えるような工夫をしよう（ワークシート解答例）

④－１　ステージクリア方式（10問クリアしたら、ステージ１クリア）

● 設定
①１ステージごとに15問出題し、10問以上正解なら次のステージへ
②ステージは５まで

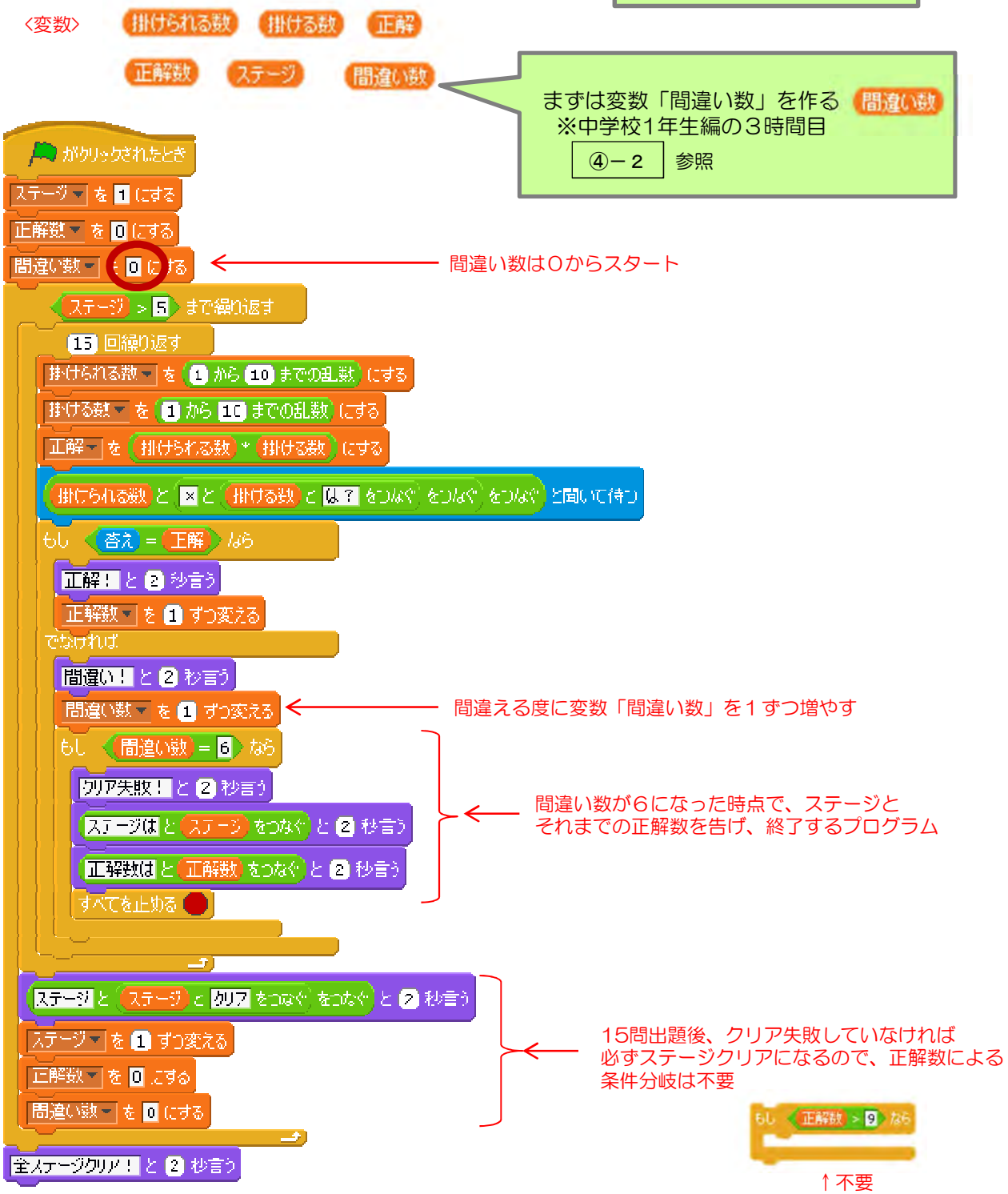
● ４時間目に作成した計算クイズのプログラムにブロックを追加



● ④－１ ステージクリア方式の別解

●設定
①１ステージごとに15問出題し、10問以上正解なら次のステージへ
ただし、１ステージで６問間違えた時点で終了
②ステージは５まで

赤字部分の設定を追加

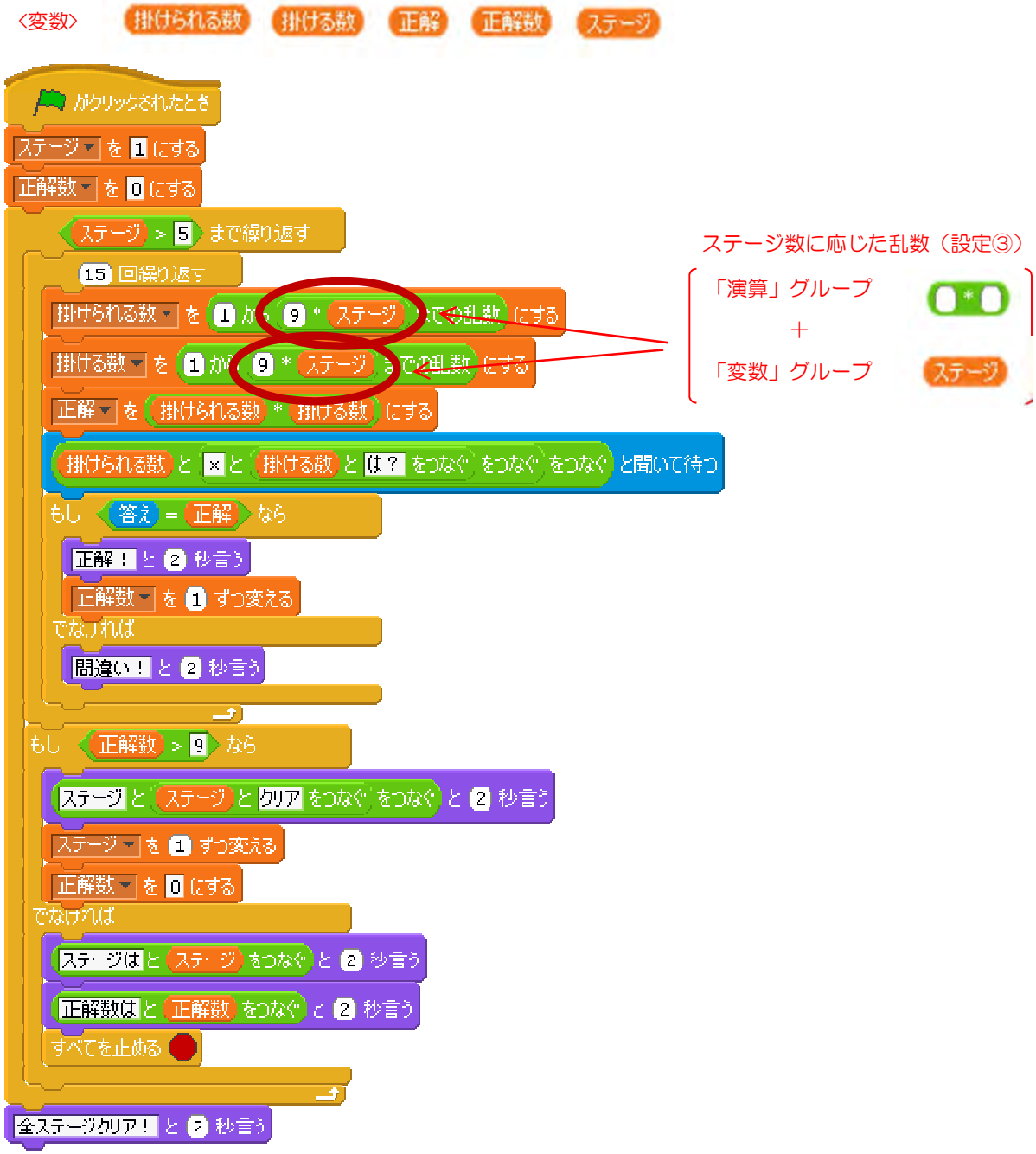


④－２　ステージが上がる毎に、問題が難しくなる（例：ステージ１は一桁の計算、ステージ２は二桁の計算）

●設定　※ ④－１ の設定に③を追加

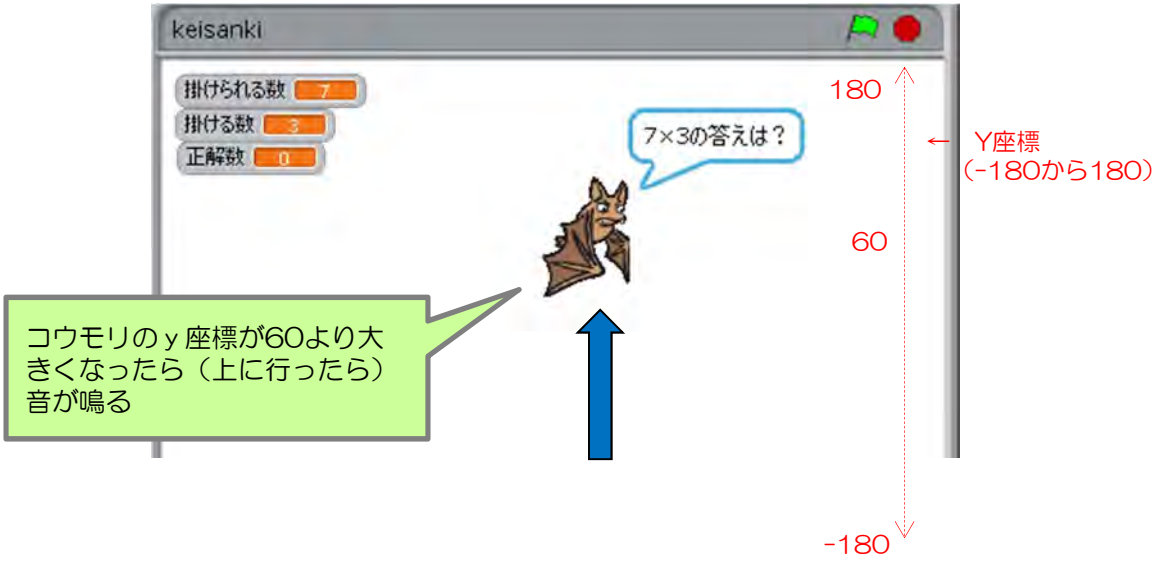
①１ステージごとに15問出題し、10問以上正解なら次のステージへ
②ステージは５まで
③ステージごとの出題は「１から（9×ステージ数）までの数」から「掛けられる数」と「掛ける数」をランダムに選択する
※ステージ１は「１～９」、ステージ２は「１～１８」、ステージ3は「１～２７」、…

● ④－１ で作成した計算クイズのプログラムにブロックを追加



④－３　制限時間が少なくなると、警告音がなる

●サンプルプログラム③にブロックを追加



④－４　ハイスコアを残す仕組み

● ４時間目に作成した計算クイズのプログラムにブロックを追加

中学校２年生編

単元の構成

１時間目～５時間目：
生活や社会を支える技術（情報通信技術について調べる活動）

1時間目：デジタル化された情報を扱うコンピュータ

2時間目：デジタル化された情報を伝えるネットワーク

3時間目：ネットワークで情報を表現するWeb

4時間目：プログラムで処理を自動化する仕組み

5時間目：デジタル化された情報と知的財産、サイバーセキュリティ

6時間目～１５時間目：
技術による問題の解決（問題発見・課題設定、設計の活動）

6時間目：SNSの文字コミュニケーション技術をより良くしよう

7時間目：アクティビティ図で主要な機能と全体設計しよう

8時間目：基本設計の情報処理の手順を分解して考え、プログラムで
アルゴリズムを実現しよう（Studuino）

9時間目：より使いやすくするための機能を考えよう（Studuino）

10時間目：グループで話すための機能を考えよう（Studuino）

11時間目：社会からの要求に応えよう(Studuino)

12～15時間目：

安全・適切なプログラムにしよう(Studuino)

１６時間目～１９時間目：
情報を利用するための基本的な仕組みと技術の評価

16～19時間目：

自分たちが制作したプログラムの紹介webページの制作

20時間目：
成果の評価と技術の概念

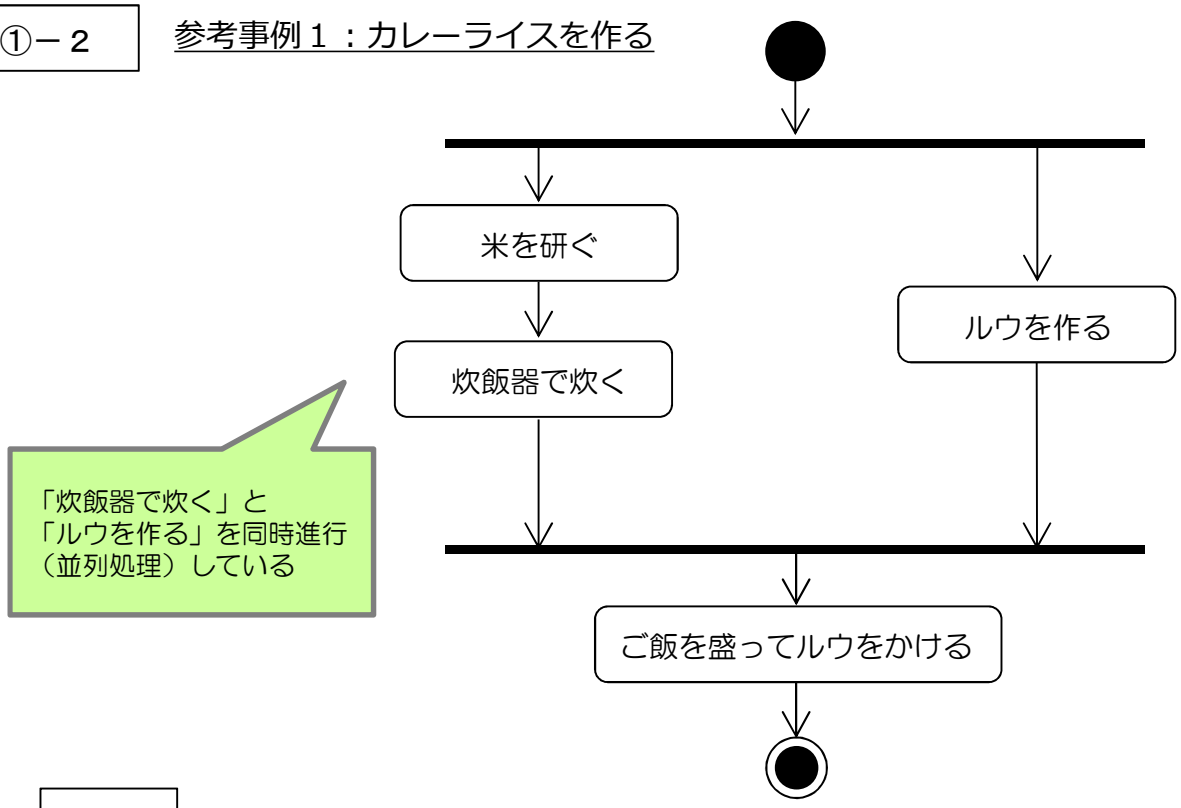
4/20時間目：プログラムで処理を自動化する仕組み

①アクティビティ図

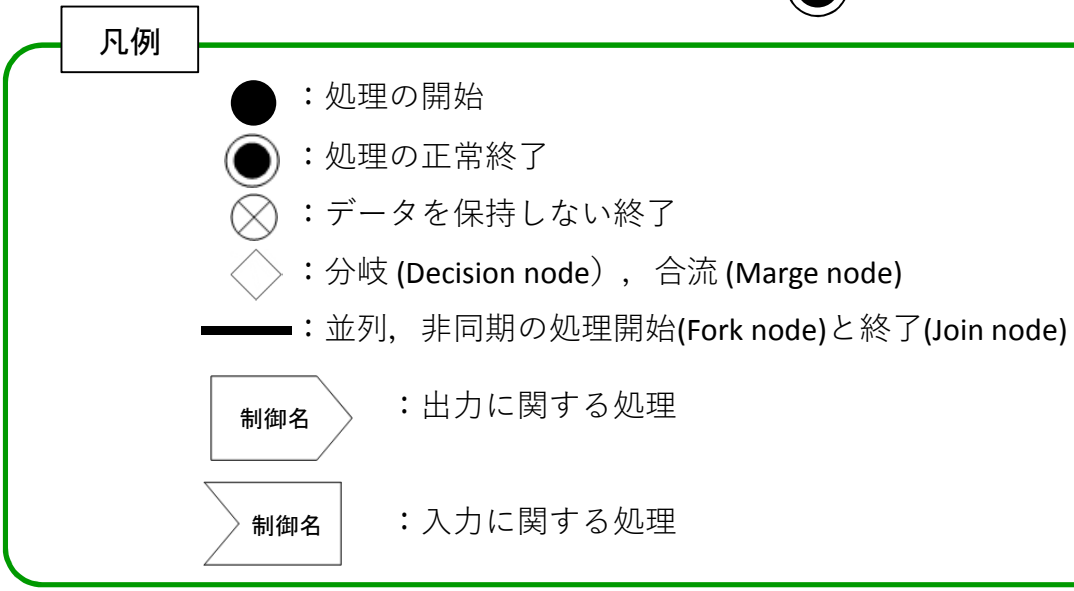
①－１　フローチャートではなくアクティビティ図を使う理由

- アクティビティ図では「並列処理」が表現できる
- また、後に学習するメッセージ交換などは、「並列処理」がなければ表現できない

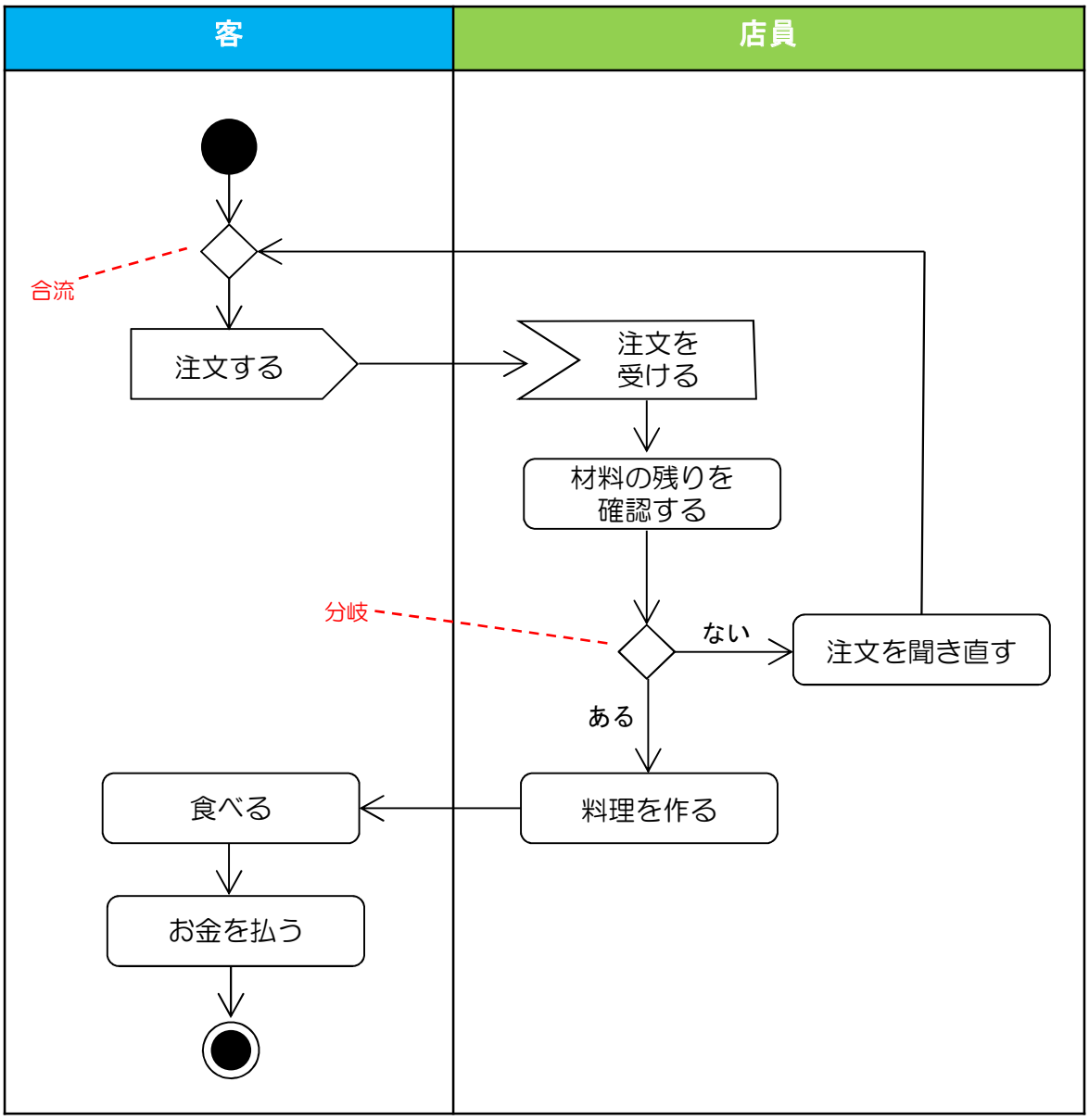
①－２　参考事例１：カレーライスを作る



凡例



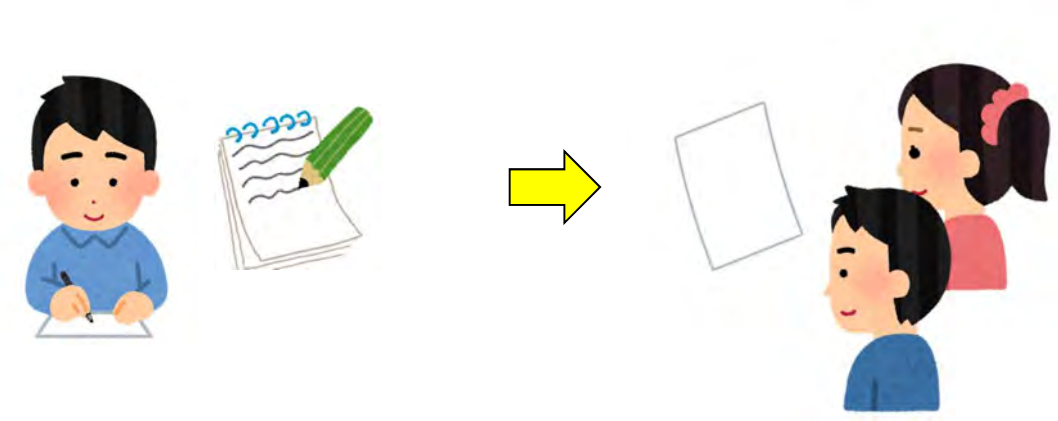
①－３　参考事例２：レストランでの食事



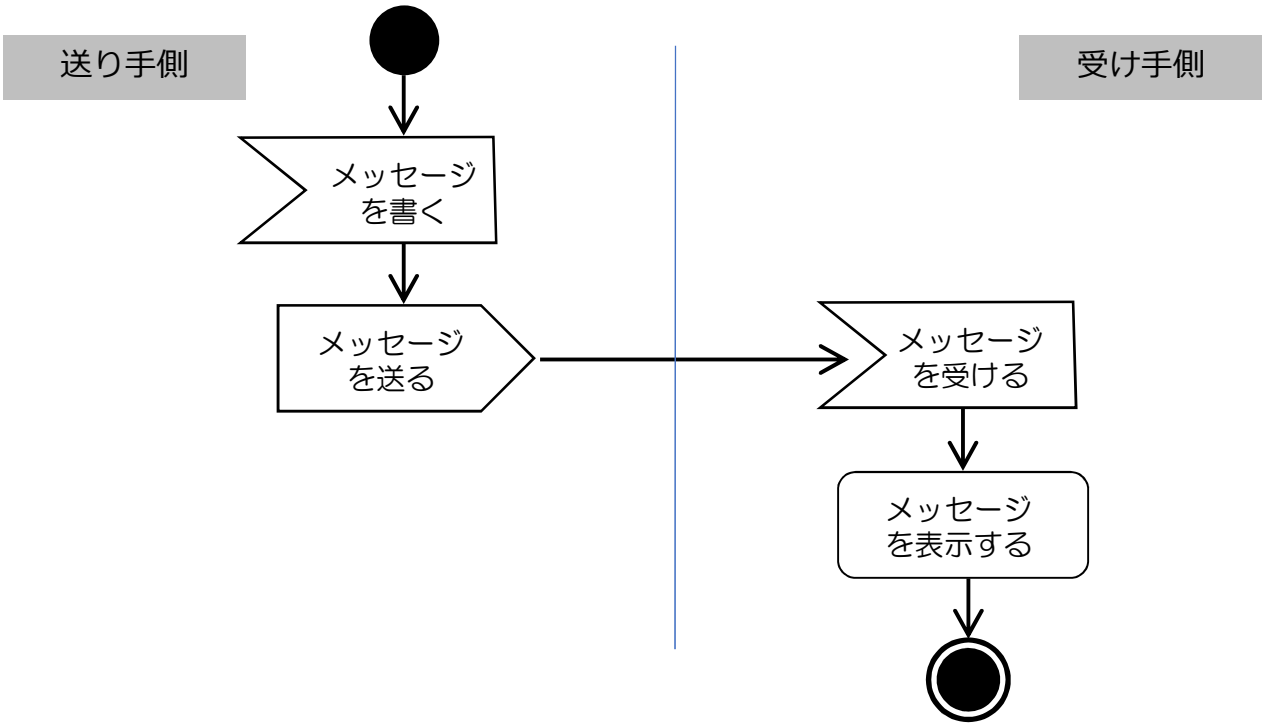
②ＳＮＳ等の文字コミュニケーションアプリの仕組み

②－１　実際の人間による、記録が残る多人数コミュニケーションの手順

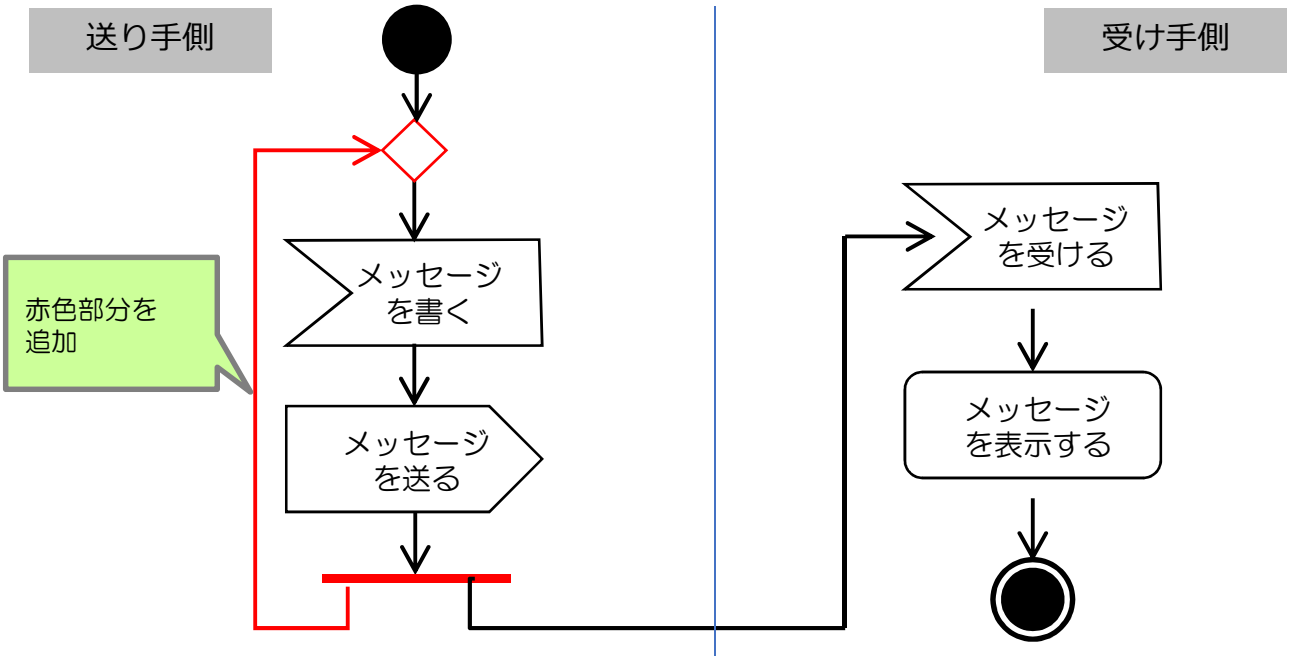
●メッセージを書いて，相手に渡す



②－２　②－１ をアクティビティ図で表現



②－３　メッセージを繰り返し送れるアクティビティ図



6/20時間目：ＳＮＳの文字コミュニケーション技術をより良くしよう

①問題解決のための課題の設定（指導過程５）

〈技術で解決・解消・軽減できる課題〉

[1]使い過ぎ防止のためのタイマー機能
[2]他人に使用されないための個人認証機能

など

②機能の詳細・精緻化（指導過程６）

[1]使い過ぎ防止のためのタイマー機能

次の時間にアクティビティ図を書くので、手順が分かるように分解して考えさせる

〈詳細・精緻化の例〉

- ●指定の時間になったらどうなるか
  - ・画像で知らせる　・音で知らせる　・強制的に電源を切る
- ●知らせる方法
  - ・どのような画像か　・どのような音か　・音の大きさは

など

[2]他人に使用されないための個人認証機能

〈詳細・精緻化の例〉

- ●パスワードの入力方法は
  - ・数字を手入力　・0～9の数字をクリックした順番（４桁）
- ●パスワードを間違った時
  - ・強制終了　・繰り返しパスワードを入力できる

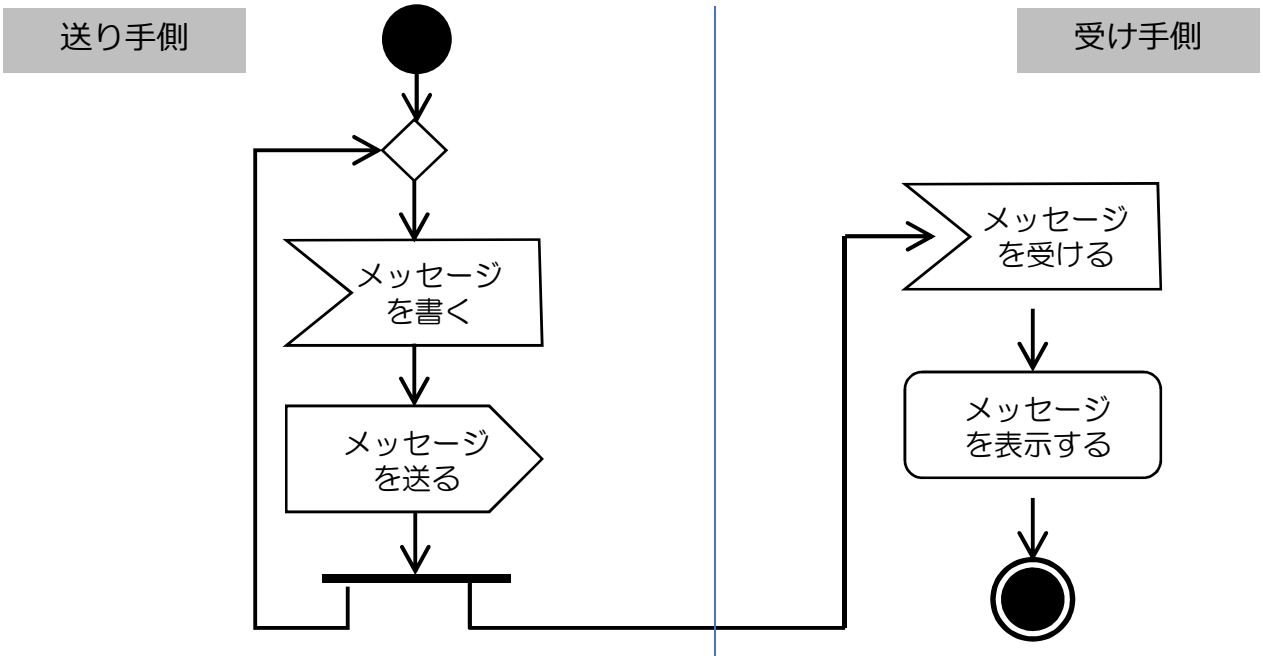
など

7/20時間目：アクティビティ図で主要な機能と全体設計しよう

①メッセージ交換の処理の確認（指導過程１）

〈文字コミュニケーションの基本手順〉

※　中学校２年生編の4/20時間目　②－４　参照

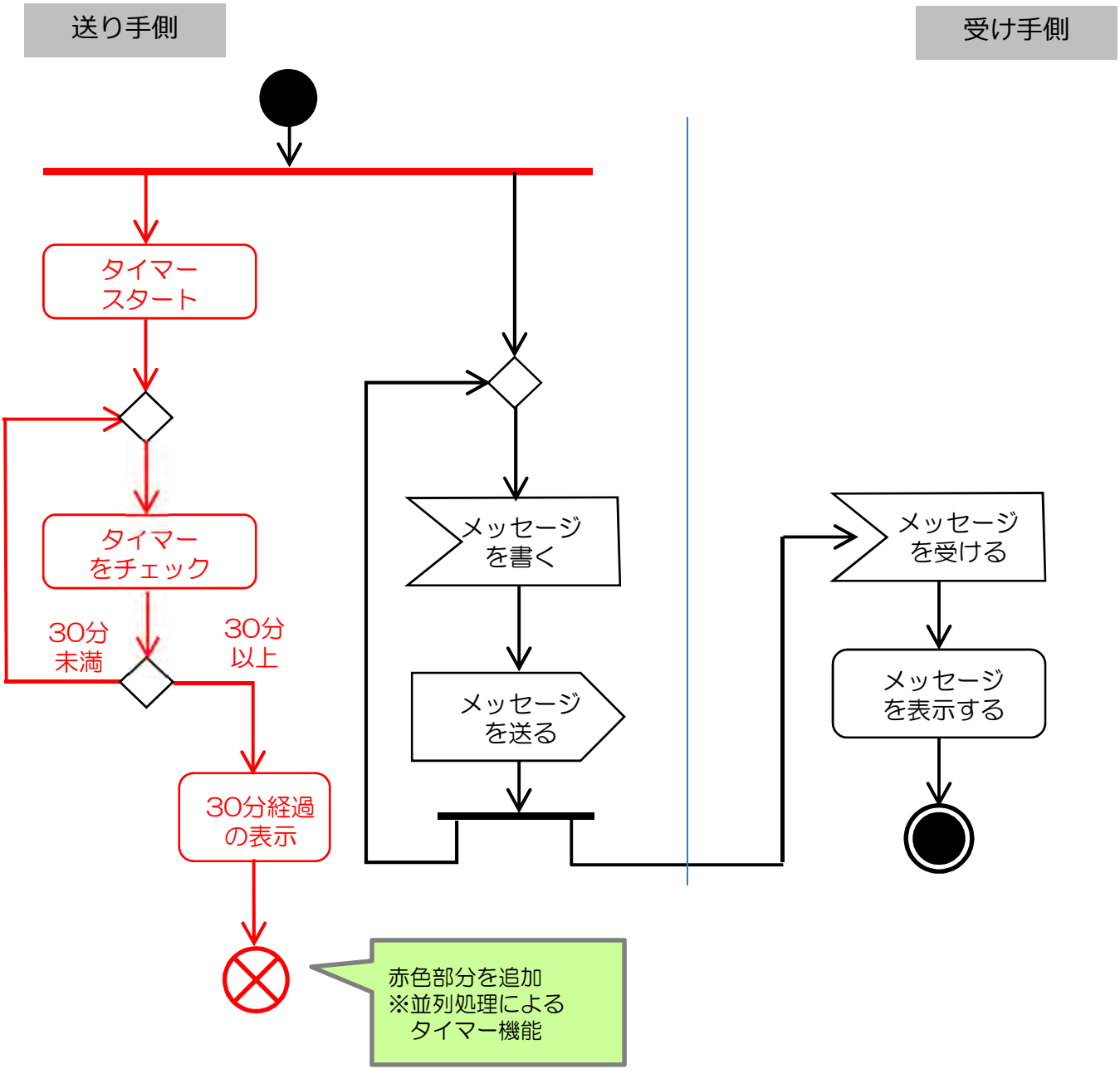


②機能の詳細・精緻化（指導過程４）

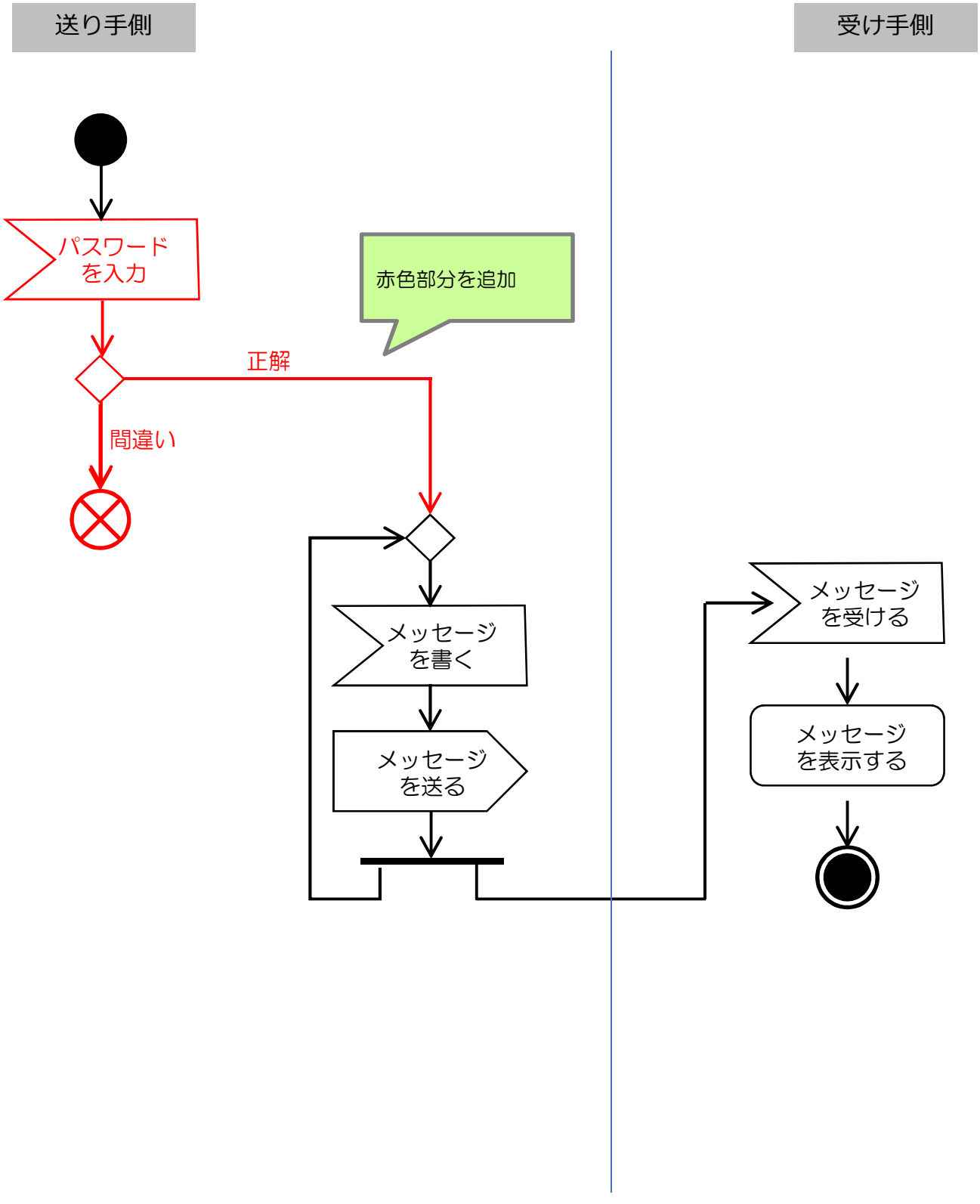
②－１　アクティビティ図の書き方

- 中学校２年生編の4/20時間目を参照

②－２　アクティビティ図の例（タイマー機能：並列処理）



②－３　アクティビティ図の例（個人認証機能）

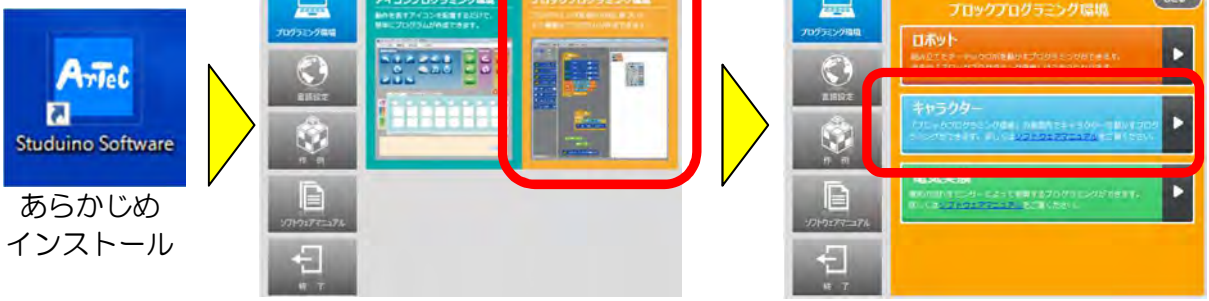


8/20時間目：基本機能の情報処理の手順を分解して考え、プログラムでアルゴリズムを実現しよう

①ソフトウェアの準備（指導課程３）

①－１　Studuino（スタディーノ）の起動

● スタディーノをキャラクターモードで使用



あらかじめ
インストール

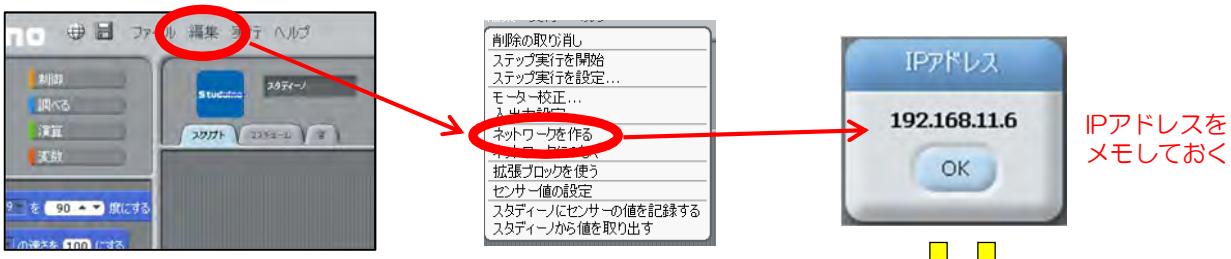
※iPad用アプリケーション「Tickle（ティックル）App for Studuino」では双方向通信はできない

①－２　ネットワークを作る

重要　ネットワークを作るためには、事前にパソコンの42001番ポートを開けておく必要がある

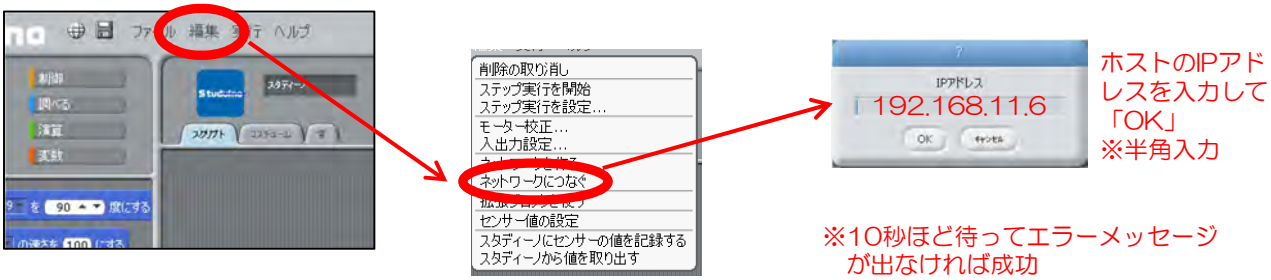
●ペアの中で「ホスト」と「ジョイン」を決める

●ホスト　→　ネットワークを作る



IPアドレスを
メモしておく

●ジョイン　→　ホストが作ったネットワークにつなぐ



ホストのIPアドレスを入力して「OK」
※半角入力

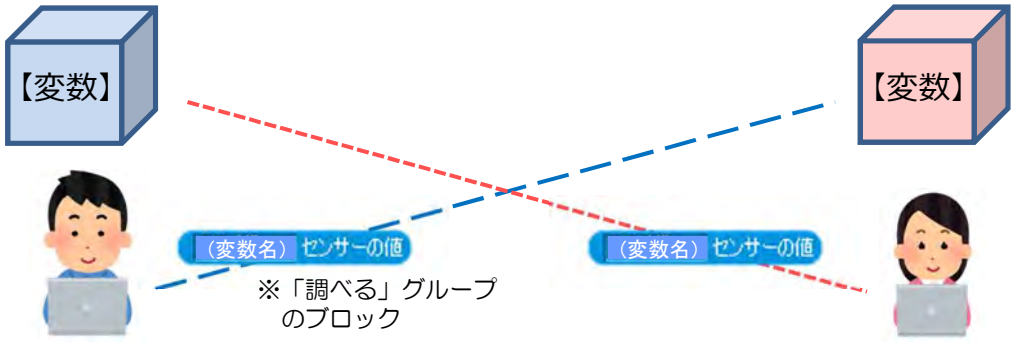
※10秒ほど待ってエラーメッセージが出なければ成功

ネットワークがつながらない場合は、ホストとジョインを入れ替えて試す

●このソフトウエアでは、ネットワークをつなぐことにより、以下のことが可能になります。

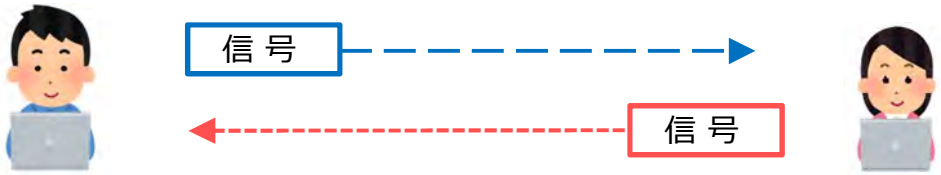
❶変数を共有できる

❷共有している変数は「センサーの値」として見ることができる



※「調べる」グループのブロック

❸互いに信号を送ることができる（例：メッセージを送る時の合図）



※「制御」グループのブロック をできる などを使用

①－３　<u>スプライトを追加</u>　※スプライト＝命令を与えるキャラクター

●Peopleフォルダのroundman（男）やsquaregirl（女）

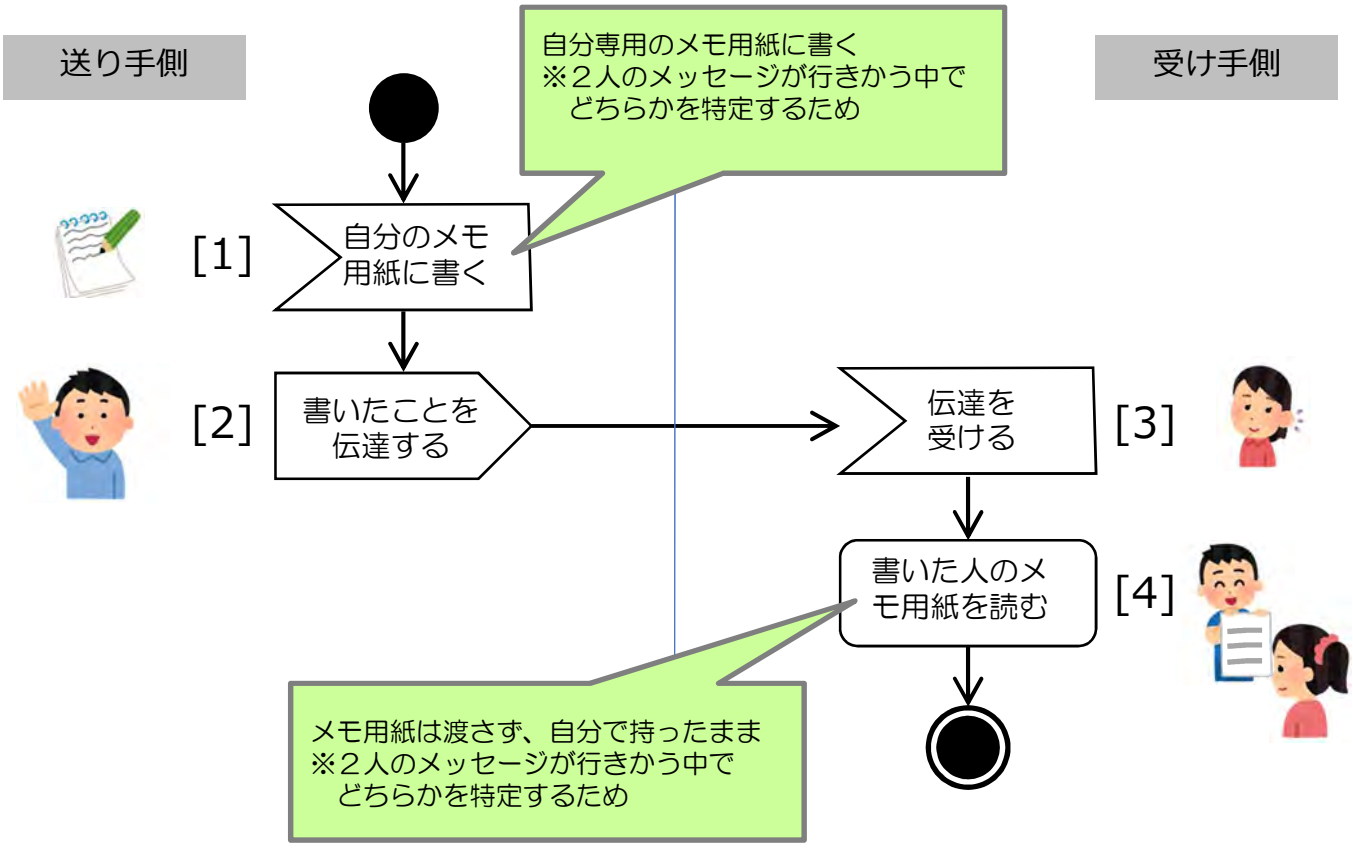
中学校１年生編の３時間目 ②－２ 参照

例：roundman

②メッセージ送信手順を分解（指導課程３）

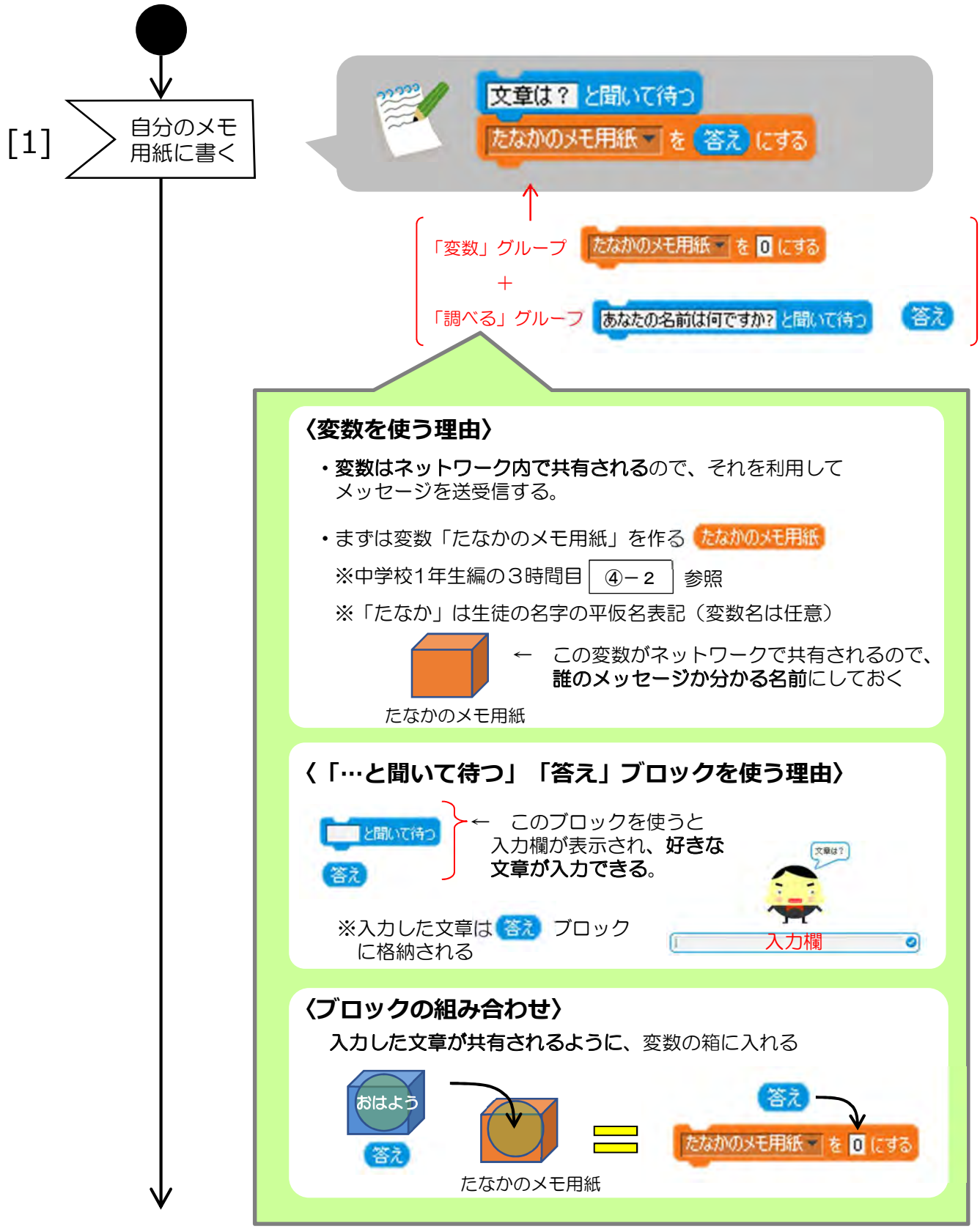
②－１　プログラムの基になるアクティビティ図

※　中学校２年生編の4/20時間目　②－２　参照



②－２　アクティビティ図に対応したプログラムブロック

送り手側（田中）



[2] 書いたことを伝達する

「制御」グループ 

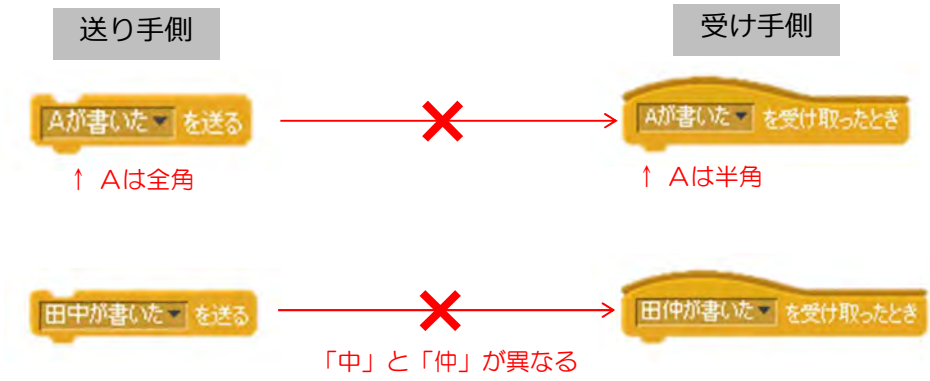
※「たなかが書いた」は手入力

・これは「書いた」という**信号を送る**ブロックで、変数ではない。

・重要なことは、四角内の文字（この例では「たなかが書いた」）を**受け手側が作るブロックと全く同じ文字を入力する**こと。

・**アルファベットの全角・半角が異なるだけでも別の信号と判断される**

・**漢字の場合、変換の際に異なる可能性がある**



・<u>よって、この例では生徒の名字を平仮名にして表記している。</u>
※同じ苗字の生徒がペアになった時は名字に「あ」や「い」を加えて「たなかあ」「たなかい」にするなど工夫する。

・四角内の文字は任意であるが、この例では、実行内容がイメージしやすいように「…が書いた を送る」としている。

受け手側（木村）

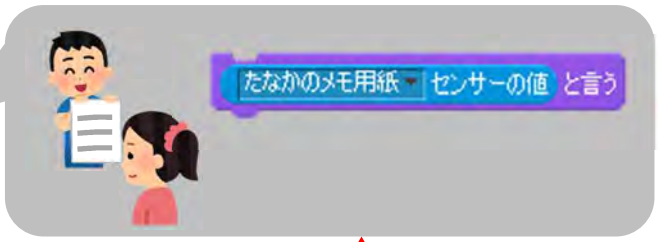
[3] 伝達を受ける

←「制御」グループ

・これは「書いた」という**信号を受け取る**ブロックで、変数ではない。

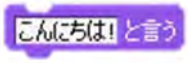
・重要なことは、四角内の文字（この例では「たなかが書いた」）を**送り手側が作るブロックと全く同じ文字を入力する**こと。

[4] 書いた人のメモ用紙を読む



※「たなかのメモ用紙」は変数

「見た目」グループ



＋ 「調べる」グループ

・外部の値を読み取るブロック

・**変数は共有されている**ため▼から「たなかのメモ用紙」を選択

②－３ ブロックの結合

スプライトは送り手、受け手でそろえておく方が、メッセージ交換のイメージがしやすい

送り手側

受け手側

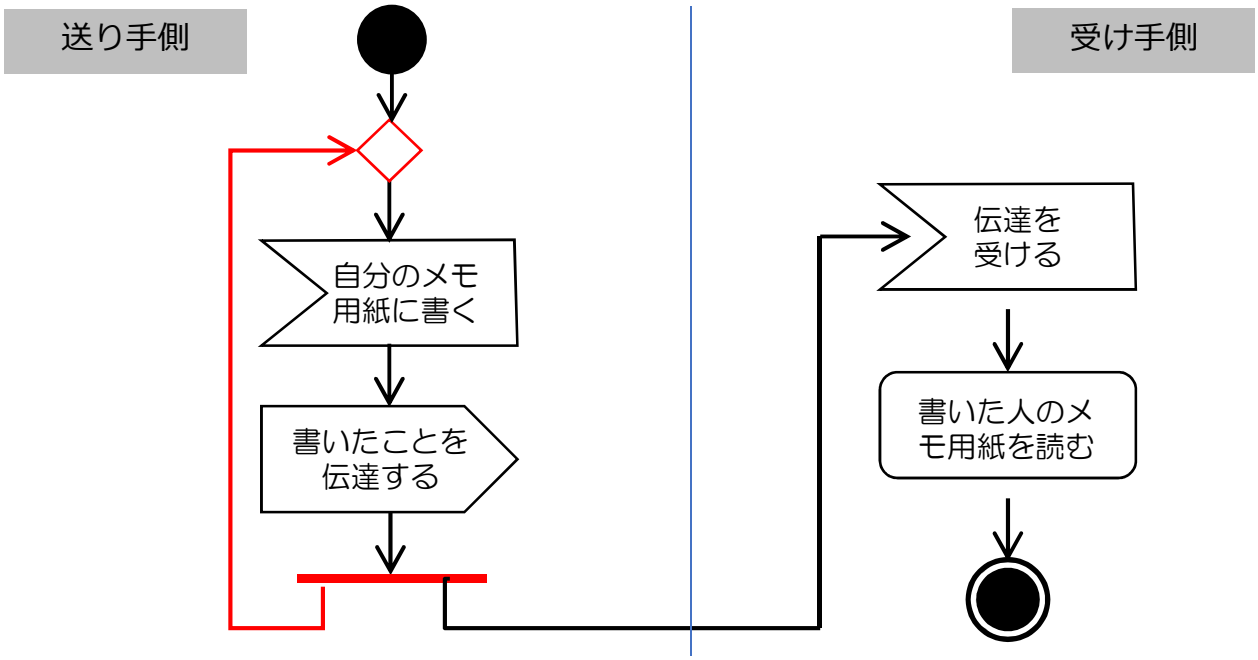
〈変数〉

変数を読み取る

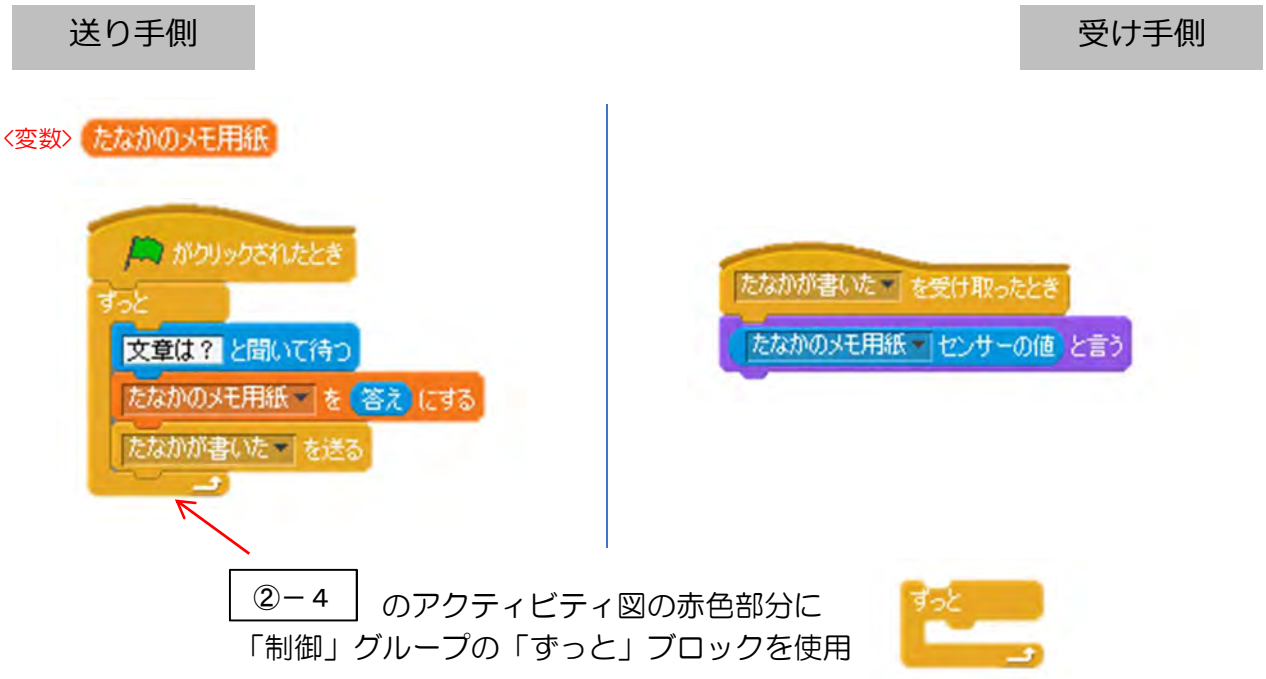
ただし、これではメッセージを送る度に旗をクリックしなければならない。 ②－４ 以降で解決を図る。

②－４　繰り返しメッセージを送れるアクティビティ図

※　中学校２年生編の4/20時間目　②－４　参照



②－５　プログラムで実現（繰り返しメッセージ送信）



②－４　のアクティビティ図の赤色部分に
「制御」グループの「ずっと」ブロックを使用

③双方向コミュニケーションの実現（指導過程４）

●**スプライトを二つ用意**して，それぞれに送り手側と受け手側のプログラムを割り付ける

③－１　２つ目のスプライトを追加

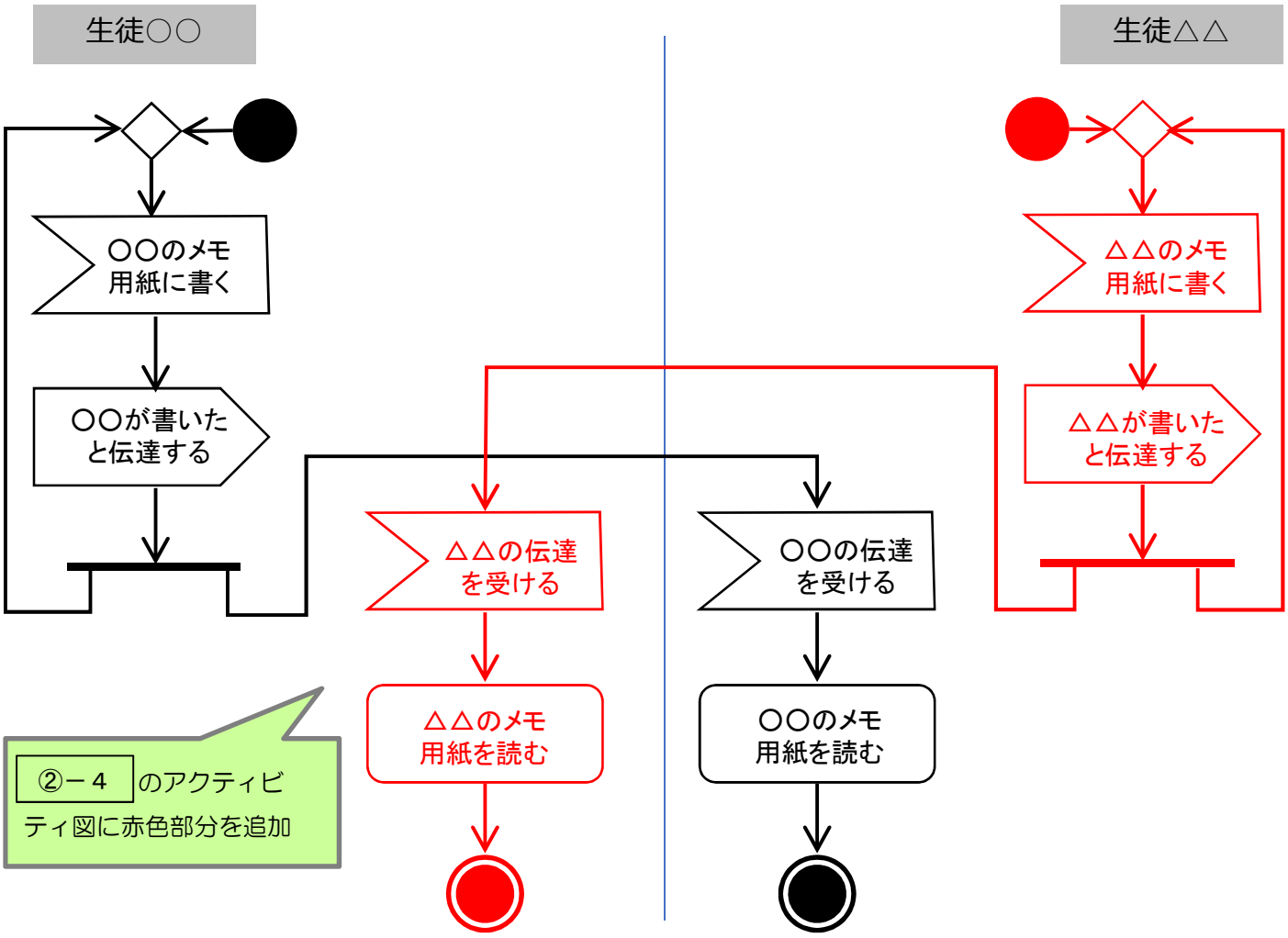
●Peopleフォルダのroundman（男）やsquaregirl（女）

中学校１年生編の３時間目　②－２　参照

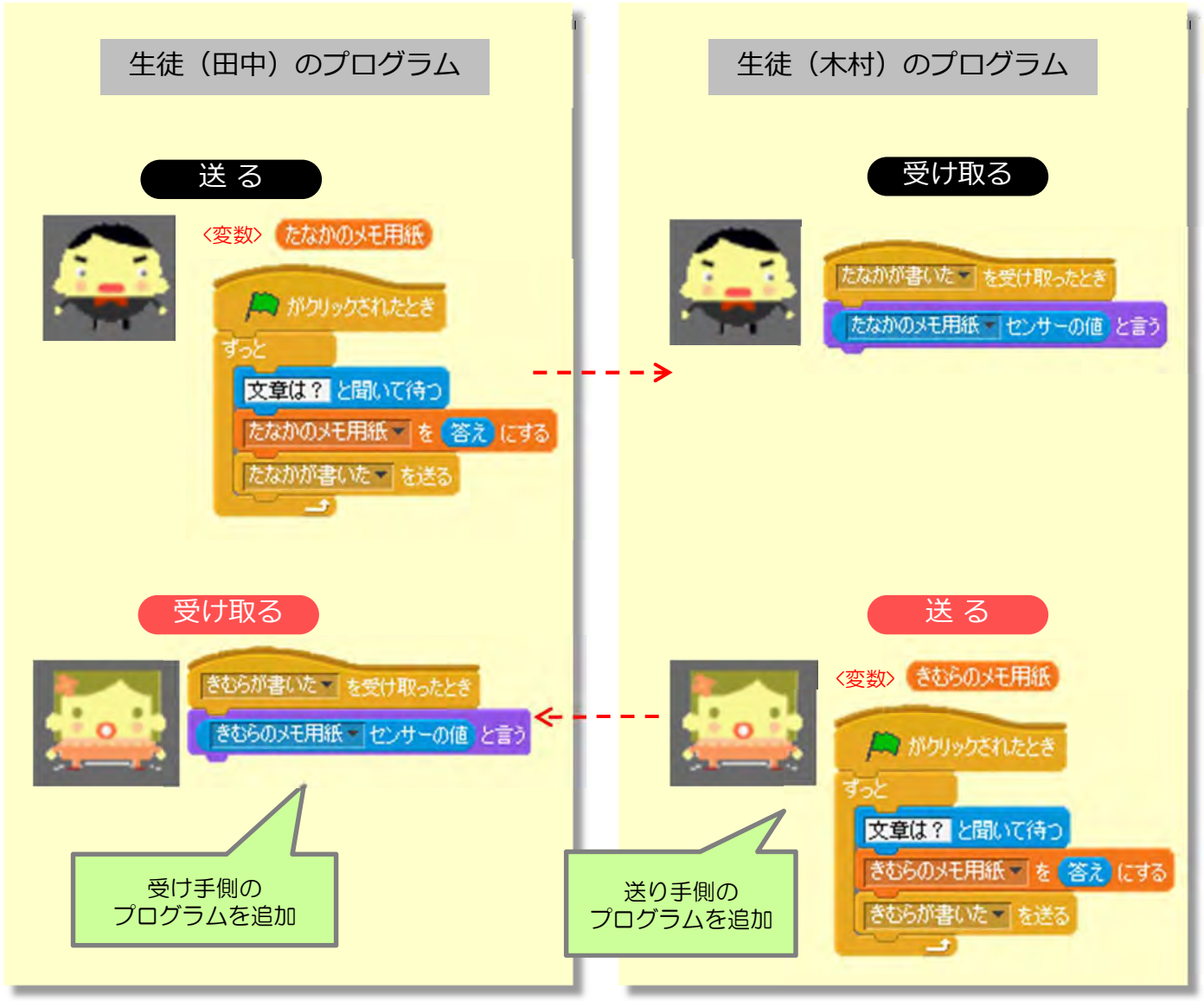
●スプライトを追加する理由

・１つのスプライトで送信・受信をすると、メッセージが表示される場所が同じになり、誰のメッセージか分からなくなる。Studuinoで解決するには、送信と受信でスプライトを分けるしかない。

③－２　双方向のアクティビティ図



③－３　アクティビティの実現（双方向）



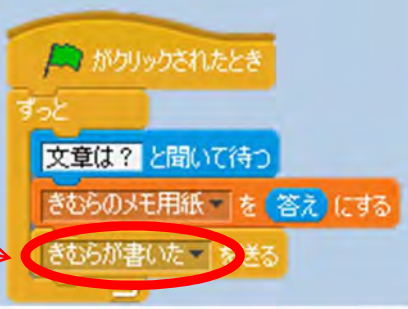
〈作成手順〉

❶生徒（木村）が変数「きむらのメモ用紙」を作る　きむらのメモ用紙

❷生徒（田中）が受け手側のプログラムを作る

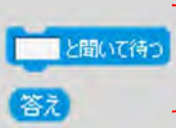
信号に使う文言は
全く同じにしておく

❷生徒（木村）が送り手側のプログラムを作る



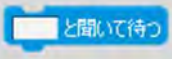
④自分が送った文章の表示（指導過程４）

●入力した文章を表示しながら次の入力を待つ



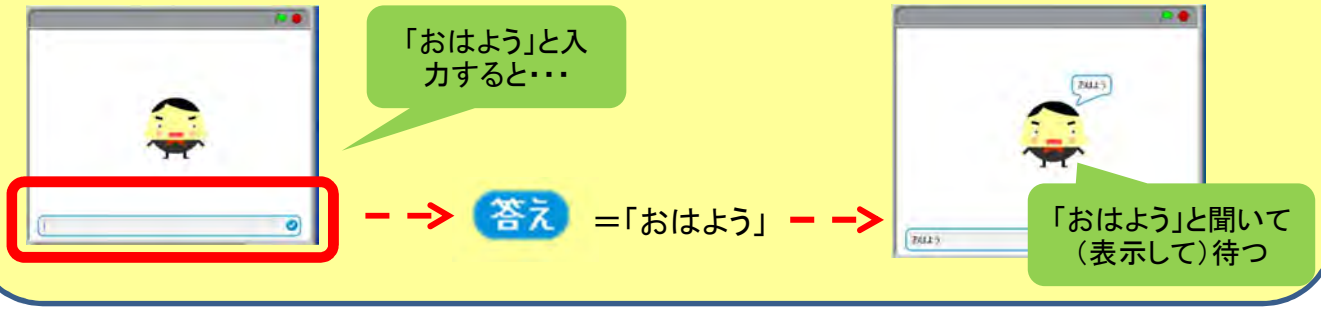
← このブロックを使うと入力欄が表示され、好きな文章などが入力できる。

※入力した文章などは 答え ブロックに格納される



「・・・と聞いて待つ」ブロックは、「・・・」に入力した文字列を吹き出しに表示する。

「答え」ブロックを入れて「『答え』と聞いて待つ」にすると、入力欄に入力した文字列を吹き出しに表示する。



⑤プログラムの保存（指導過程４）

●双方向で送受信できるプログラムを保存しておき、次の授業で拡張していく

※保存の方法は　中学校１年生編の4時間目⑥　参照

9/20時間目：より使いやすくするための機能を考えよう

①ソフトウェアの準備（指導過程１）

①－１　Studuino（スタディーノ）の起動

※　中学校２年生編の８/20時間目　①－１　参照

①－２　ネットワークを作る

※　中学校２年生編の８/20時間目　①－２　参照

①－３　双方向でメッセージを送受信するプログラムを開く

※　中学校１年生編の５時間目　①－１　参照

※　開くプログラムは、前時に保存したもの又は教員が準備したサンプルプログラム

①－４　プログラム保存時点とグループ（ペア）のメンバーが異なる場合は、変数と信号のブロックを調整する

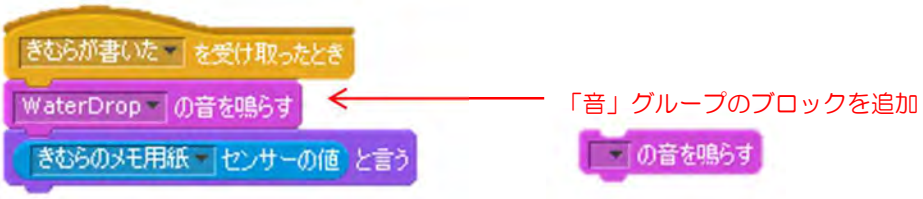
〈プログラム保存時点：田中、木村ペア〉
※田中のプログラム

〈本時：山本、木村ペア〉
※山本のプログラム

②プログラム例（指導過程４）

②－１　受信時に着信音が鳴るプログラム例

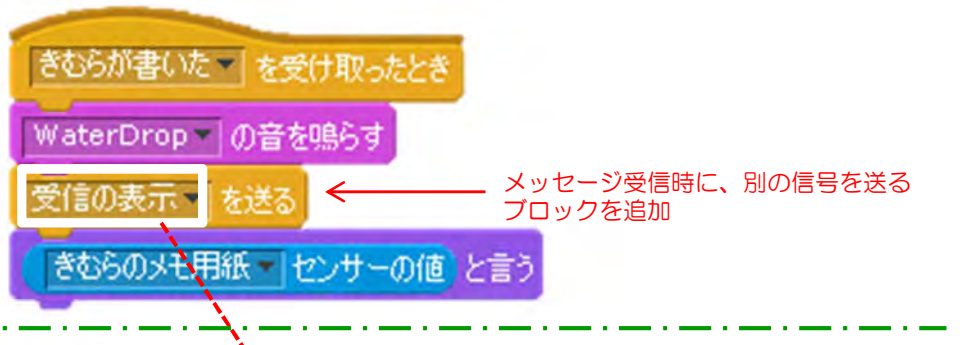
〈受信用スプライト〉



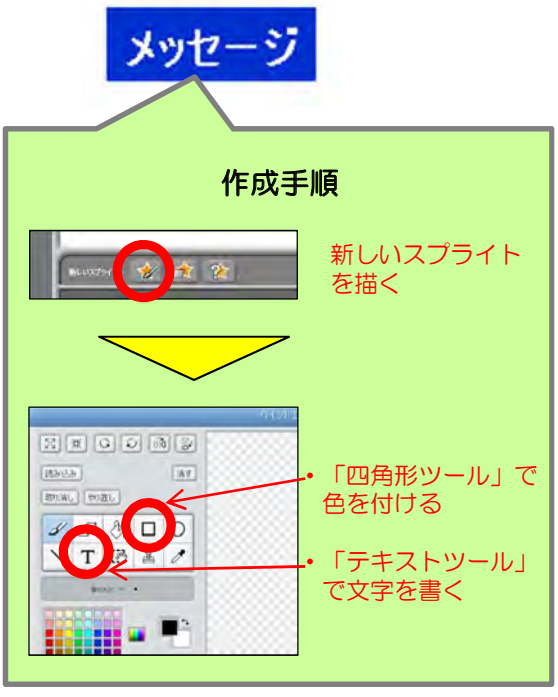
※　音の変え方などは中学校１年生編の５時間目　②－３　参照

②－２　受信時に画面が変化するプログラム例

〈受信用スプライト〉

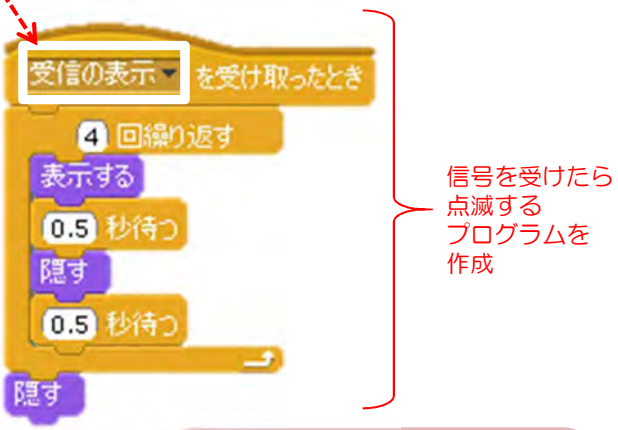


〈受信を知らせるスプライト〉
※新規作成



信 号

このように、自分から自分へ信号を送ることも可能



10/20時間目：グループで話すための機能を考えよう

①ソフトウェアの準備（指導過程１）

※　中学校２年生編の9/20時間目①参照

※　ネットワーク作成時は、以下のとおりにする。

- ●一人のコンピュータがホスト役
  - ・ネットワークを作る
- ●他の生徒のコンピュータはすべてジョイン（参加）
  - ・ホストが作ったネットワークにつなぐ

この時点では１対１の送受信ができるプログラム。以降でグループチャットに拡張していく。

②グループチャットができるプログラム（指導過程５）

●**人数分のスプライトを用意**して，自分のスプライトには送り手側のプログラムを、他のスプライトには、それぞれの生徒の受け手側のプログラムを作成する

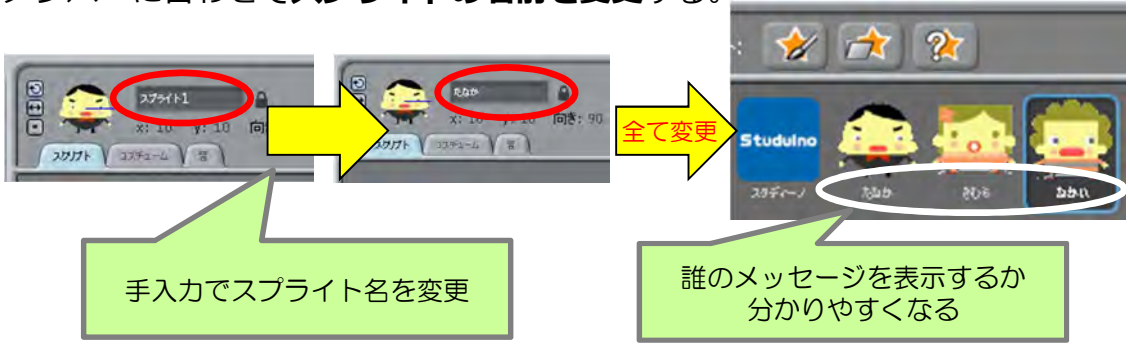
②－１　スプライトを追加

中学校１年生編の３時間目　②－２　参照

●**３人グループの場合は、３つのスプライトが必要**になるため、１つ追加する。

〈例：田中、木村、中井グループ〉
※田中のプログラム

●どのスプライトが誰のメッセージを表示するか分かりやすくするために、グループのメンバーに合わせて**スプライトの名前を変更**する。



②－２　スプライト間でプログラムをコピー

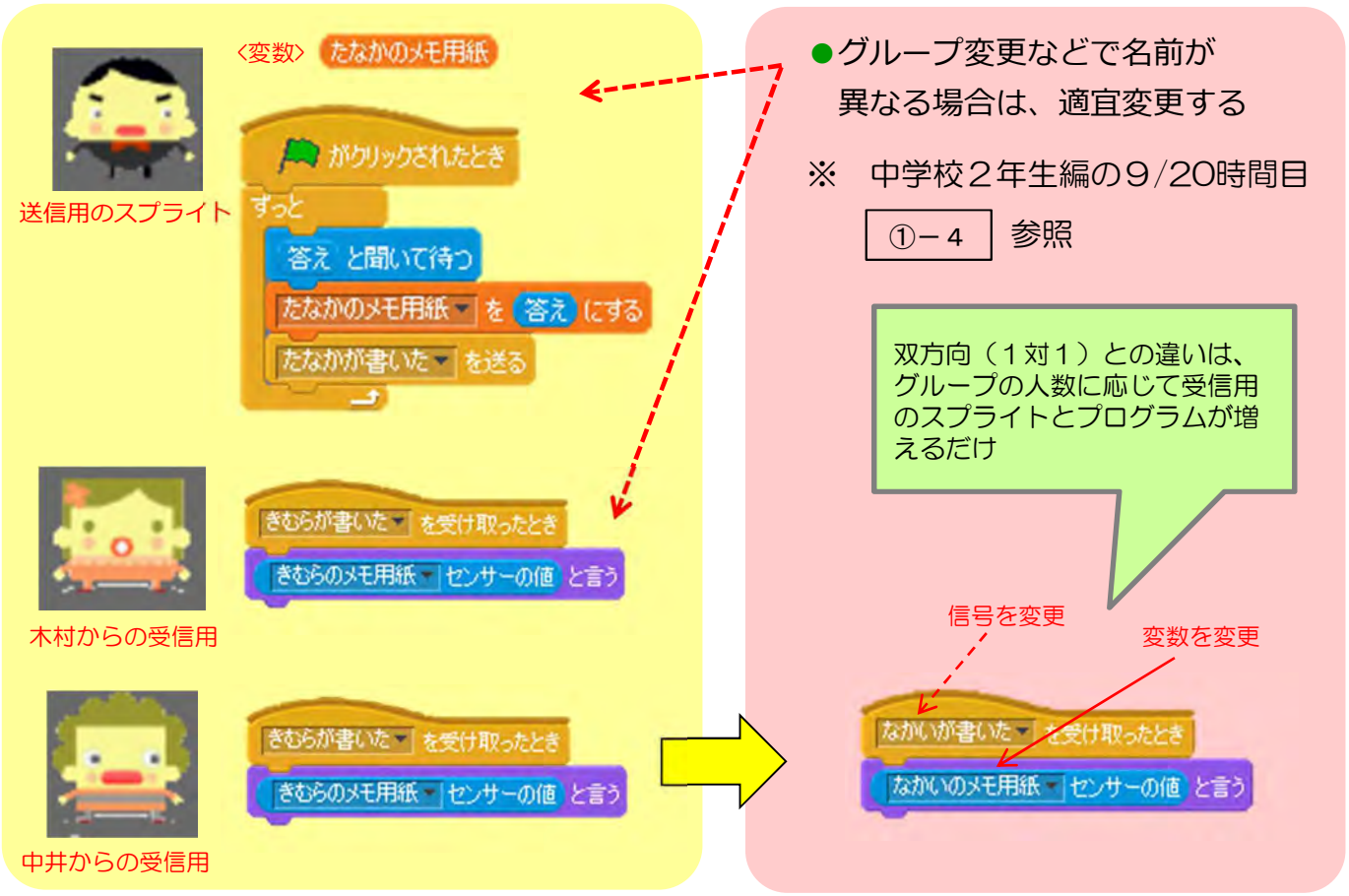
●受信用のプログラムは、ほとんど同じなのでコピー機能を使うと早い

中学校１年生編の３時間目　⑥－２　参照

②－３　変数と信号のブロックを調整

●グループのメンバーに合わせて、変数と信号のブロックを変更する

〈例：田中、木村、中井グループ〉　※田中のプログラム



③プログラムの保存（指導過程５）

●グループで送受信できるプログラムを保存しておき、次以降の授業で拡張していく

※保存の方法は　中学校１年生編の4時間目⑥　参照

④送る相手を指定（ダイレクトメッセージ）（指導過程５）

④－１　Studuino（スタディーノ）でのダイレクトメッセージのイメージ

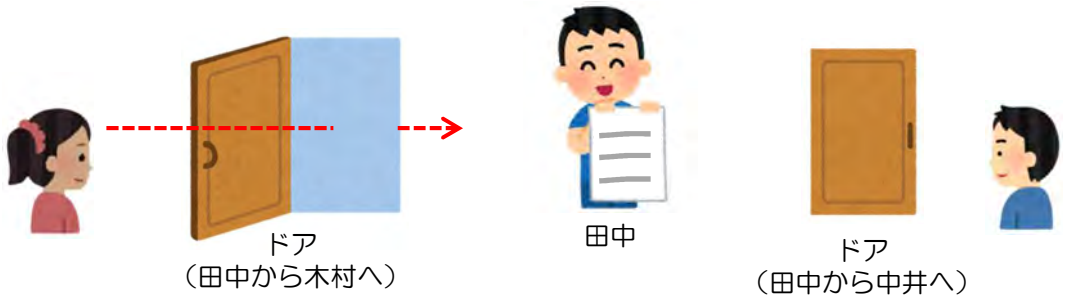
〈例：田中、木村、中井グループ〉

❶グループのメンバーとの間にはドアがある。

❷送り手は、送りたい相手のドアだけ開ける。

❸送り手は、自分のメモ用紙にメッセージを書いたら、書いたことを伝達する

❹伝達はグループ全員に届くが、ドアが開いている人だけ送り手のメモ用紙を読む



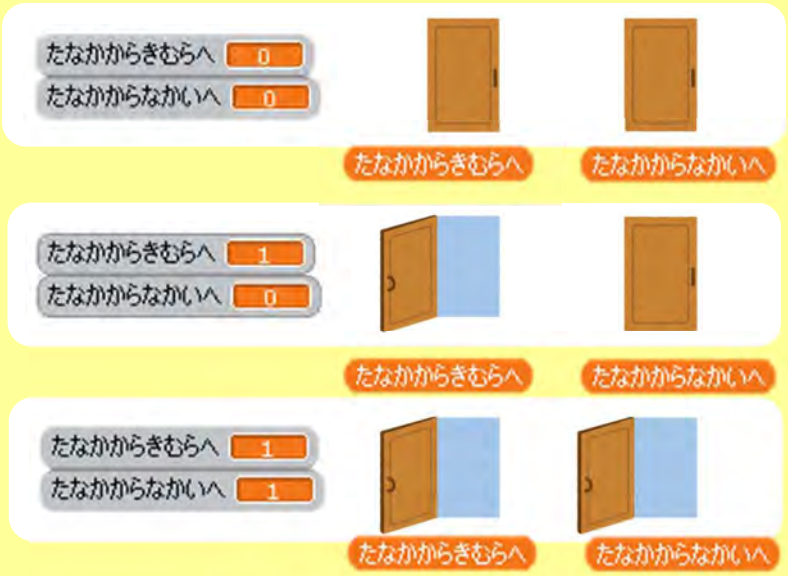
④－２　ダイレクトメッセージのプログラム　**〈例：田中、木村、中井グループ〉**
※田中のプログラム

※ ④－１ のイメージに沿って作成

❶送信する時に使うドアを変数として作る。
（ドアは開閉により状態が変化するため変数を使う）

※変数の作り方は　中学校１年生編の３時間目 ④－２ 参照

値が0なら閉まっている（送らない）、1なら開いている（送る）ことにする



最初は全てのドアを閉めておく

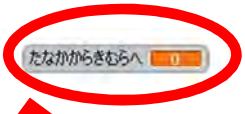
❷送り手が、送りたい相手のドアを開閉できるようにする。

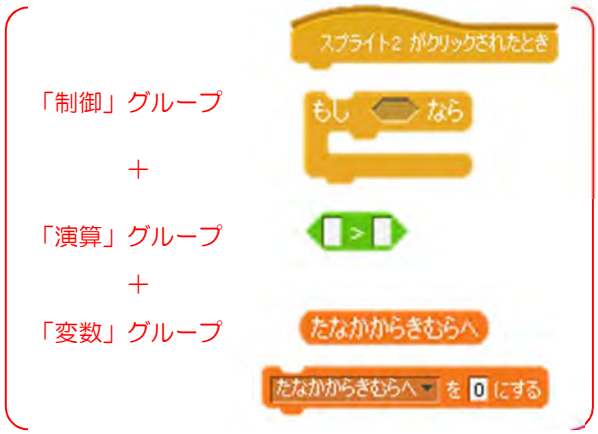
（相手のスプライトをクリックすると変数が変わるようにする）

相手のスプライトをクリック

実行エリアで木村のスプライトがクリックされたらプログラムがスタート

木村のスプライトがクリックされた時に「たなかからきむらへ」が「0」なら「1」に、そうでなければ（１なら）「0」にする





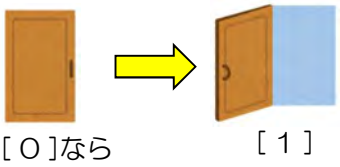
「制御」グループ

＋

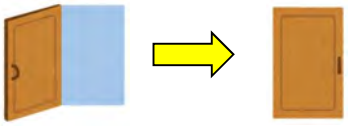
「演算」グループ

＋

「変数」グループ



[0]なら　[1]

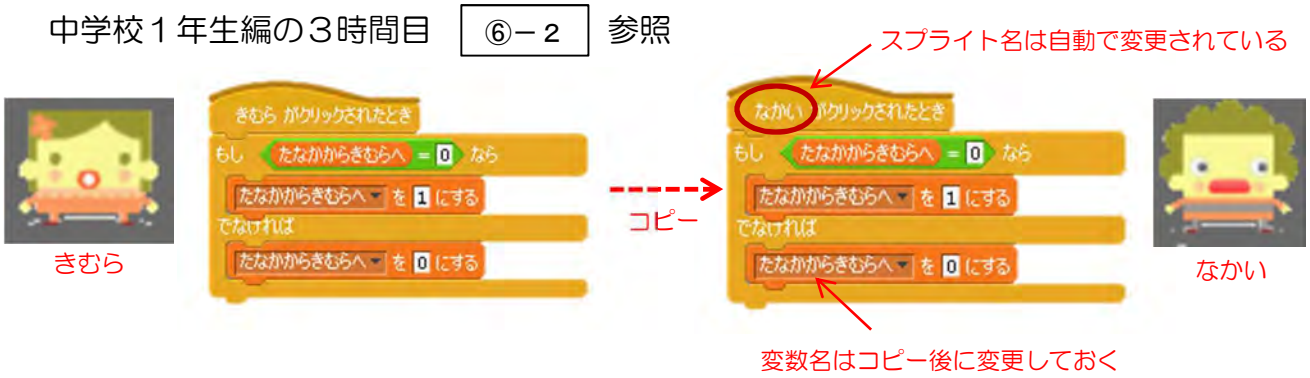


[1]なら　[0]

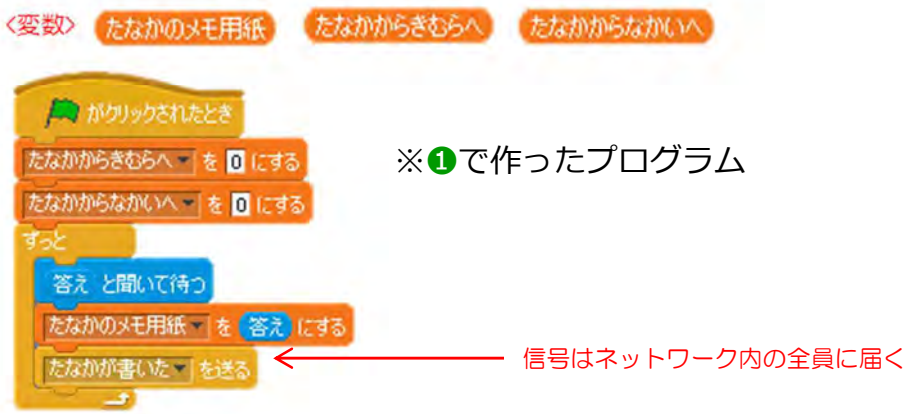
スプライトをクリックすると変数が変わる

●コピー機能を使って、別の受信用スプライトにプログラムをコピーしておく

中学校１年生編の３時間目 ⑥－２ 参照



❸送り手は、自分のメモ用紙にメッセージを書いたら、書いたことを伝達する

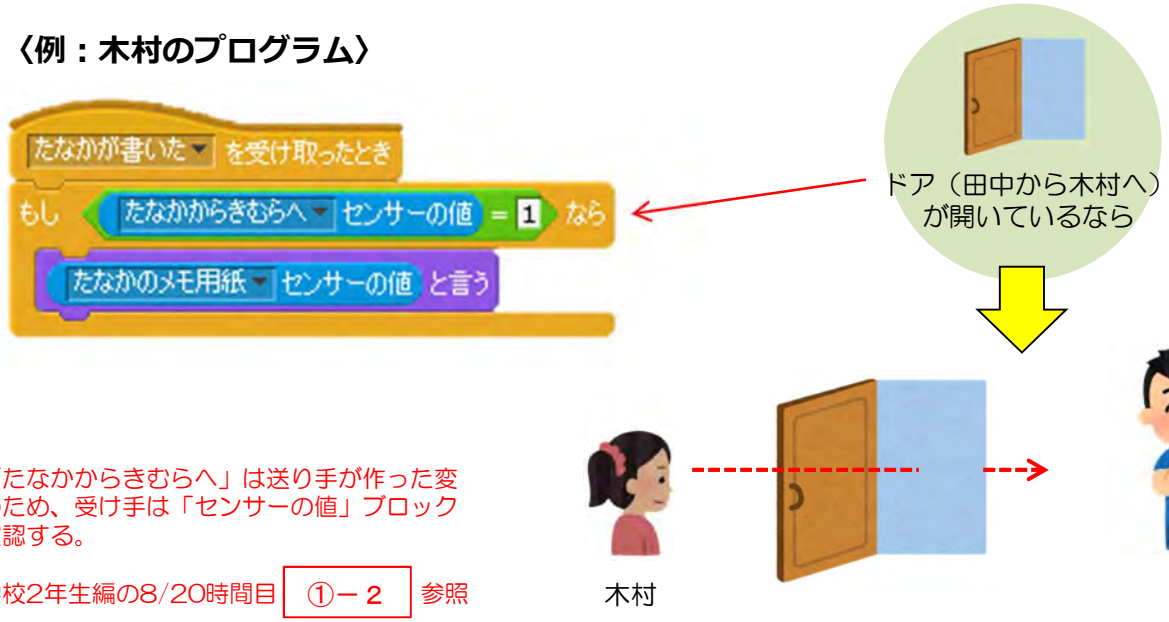


※❶で作ったプログラム

❹受け手は、ドアが開いている人だけ送り手のメモ用紙を読む

注意：このプログラムは、送り手ではなく受け手が設定しなければならない

〈例：木村のプログラム〉



※「たなかからきむらへ」は送り手が作った変数のため、受け手は「センサーの値」ブロックで確認する。

中学校2年生編の8/20時間目 ①－２ 参照

④－３　１人の生徒が作るプログラム

〈例：田中、木村、中井グループの、田中のプログラム〉

送信用プログラム	受信用プログラム

たなか

きむら

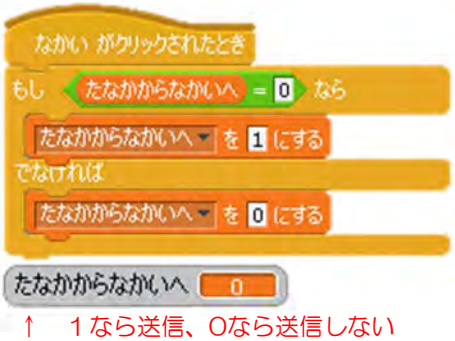
なかい

木村へ送信するかどうかの選択用

↑　１なら送信、Oなら送信しない

木村からの受信用

中井へ送信するかどうかの選択用



↑　１なら送信、Oなら送信しない

中井からの受信用

⑤コスチュームの変更（指導過程５）

●送る相手として指定されていることがわかるように

・コスチュームをコピー＆編集して，クリックするたびに切り替える

「見た目」グループのブロックを追加

⑥さらに拡張するなら（指導過程５）

●グループチャットとダイレクトメッセージの併用

・送る相手が一人も指定されていなかったら，全員に送る

送り手

まずは変数「たなかから全員」を作る　たなかから全員

※中学校1年生編の３時間目

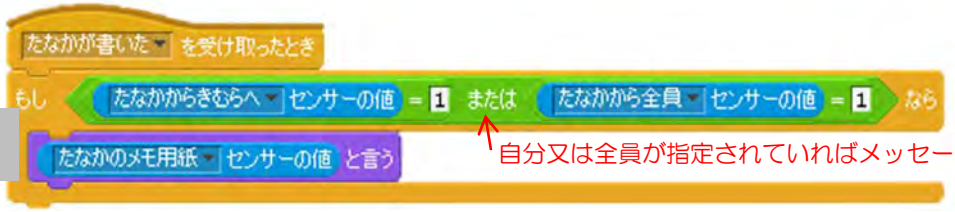
④－２　参照

送る相手が指定されているかどうかで条件分岐

誰も指定されていなければ全員を指定

誰かが指定されていれば全員指定はしない（指定済みの人のみ指定）

受け手



自分又は全員が指定されていればメッセージを読む

11/20時間目：社会からの要求に応えよう

①ソフトウェアの準備（指導過程１）

※　中学校２年生編の10/20時間目①参照

※　ただし、開くプログラムは中学校２年生編の10/20時間目③で保存した
プログラム（グループチャットができるプログラム）

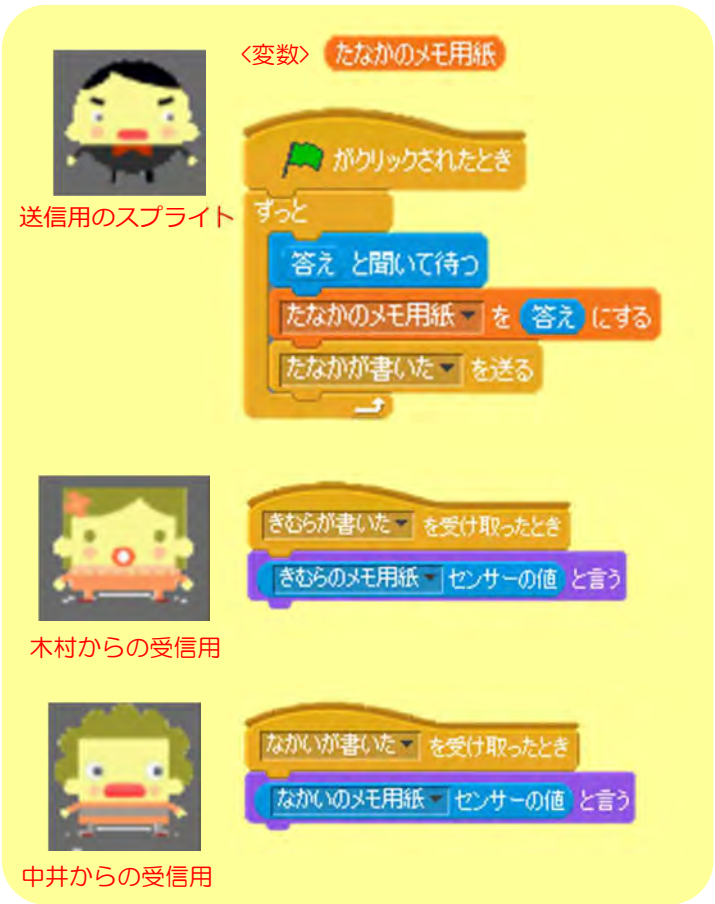
※　中学校２年生編の10/20時間目 ②－１ を参照し、スプライト名も変更する

②プログラム例：絵文字としてスプライトを表示（指導過程４）

②－１　プログラムの初期状態

●グループチャットができるプログラム（ダイレクトメッセージなし）

〈例：田中、木村、中井グループ〉　※田中のプログラム



②－２　絵文字として表示するスプライトを追加

※ 中学校１年生編の３時間目 ②－２ 参照

笑い

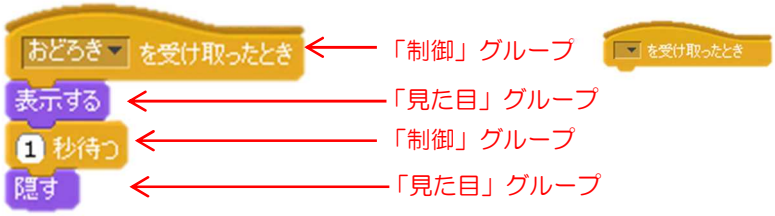
驚き

②－３　追加したスプライトに、特定の文字が送られた時だけ表示されるプログラムを作成

開始時点では表示しない

「見た目」グループ

「おどろき」という信号を受け取ったら1秒だけ表示



「制御」グループ

「見た目」グループ

「制御」グループ

「見た目」グループ

②－４　受信用スプライトに、特定の信号を送るプログラムを追加

「わらい」という文字が送られたら「わらい」という信号を送る

「おどろき」という文字が送られたら「おどろき」という信号を送る

〈イメージ〉

※自分のスプライト間で信号を送っている

信号を受けたスプライトが表示される

12～15/20時間目：安全・適切なプログラムにしよう

①ソフトウェアの準備

※　中学校２年生編の10/20時間目①参照

※　ただし、開くプログラムは中学校２年生編の10/20時間目③で保存した
プログラム（グループチャットができるプログラム）

※　中学校２年生編の10/20時間目 ②－１ を参照し、スプライト名も変更する

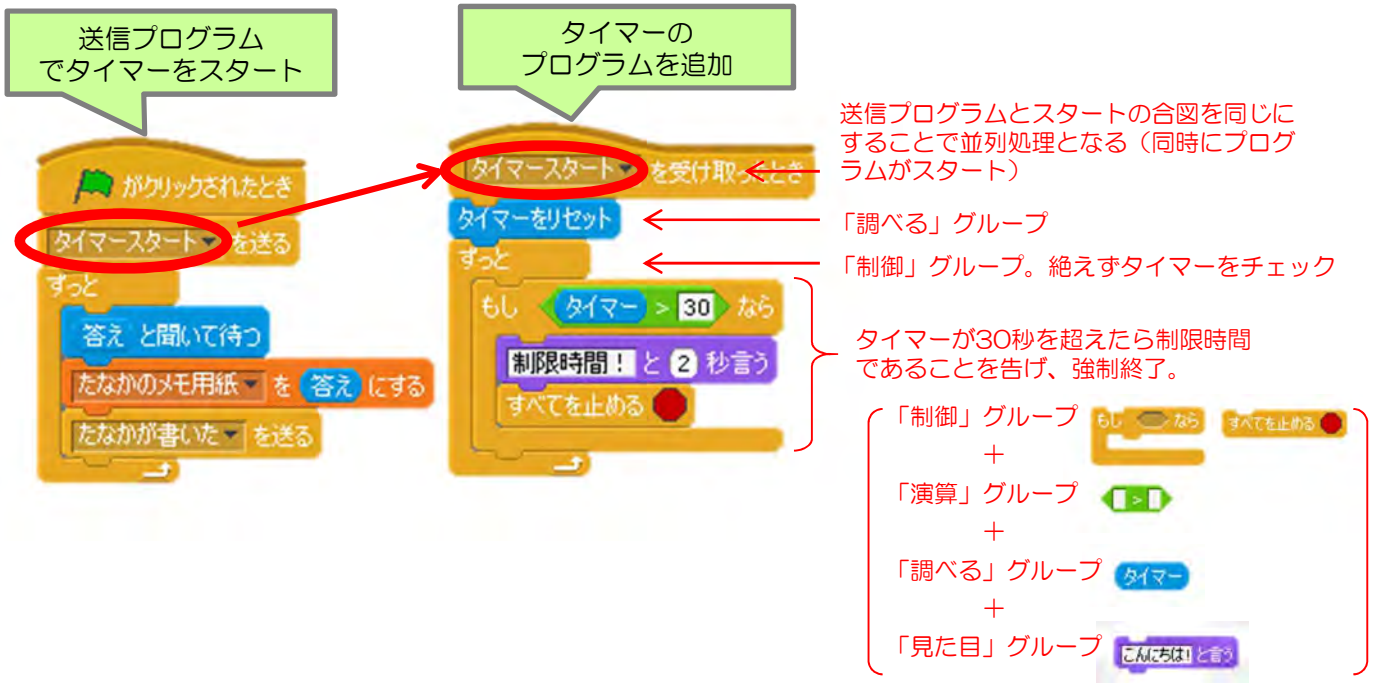
②プログラム例（タイマー機能：並列処理）

②－１　アクティビティ図

※　中学校２年生編の7/20時間目 ②－２ 参照

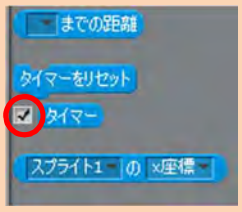
②－２　プログラム例（送信用スプライト）

〈例：田中、木村、中井グループ〉　※田中のプログラム



●タイマーを実行エリアに表示するときは

「調べる」グループの
「タイマー」ブロック
横のチェックを付ける



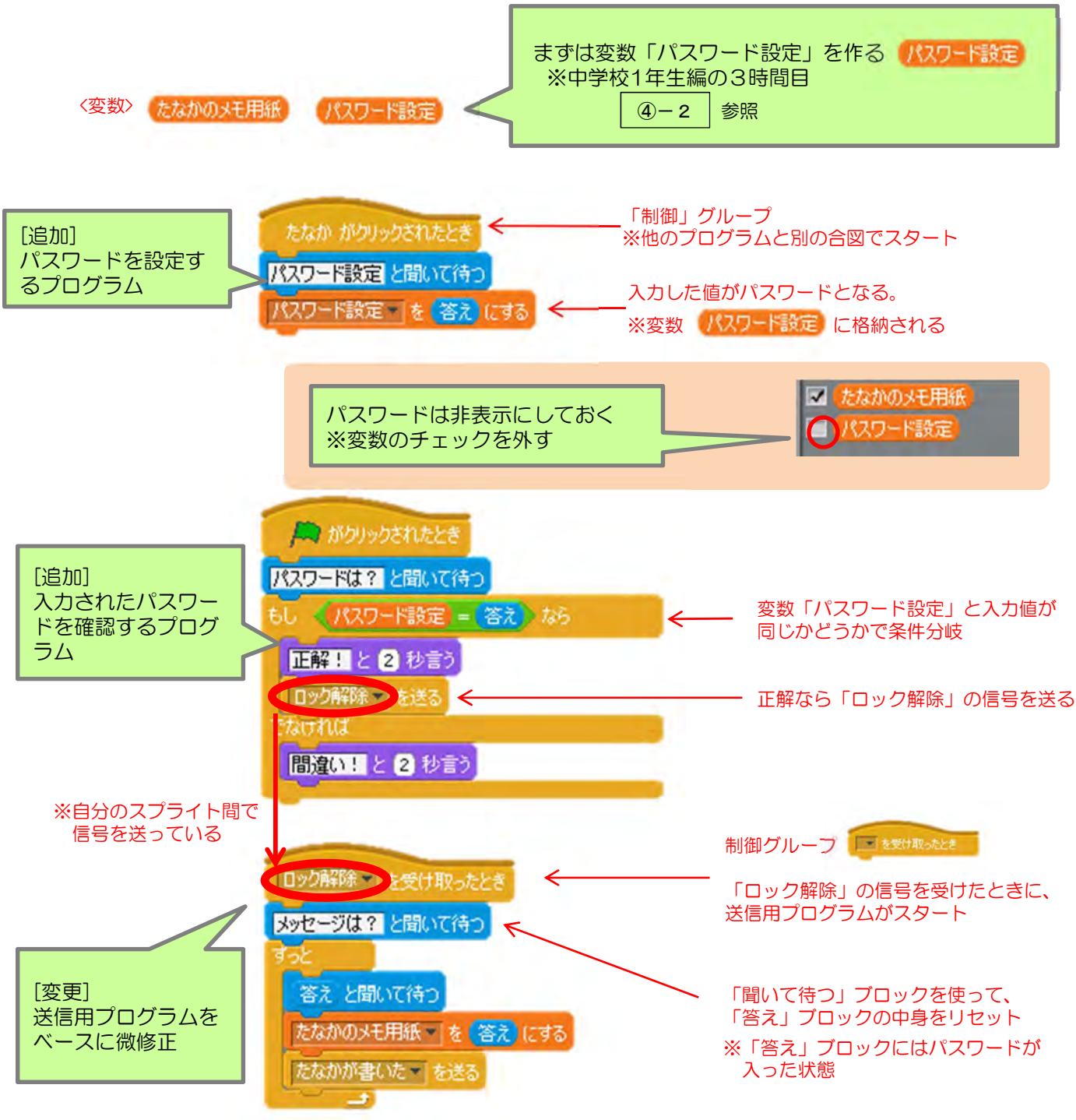
③プログラム例（個人認証機能：「聞いて待つ」ブロックを使用する例）

③－１　アクティビティ図

※　中学校２年生編の7/20時間目　②－３　参照

③－２　プログラム例（送信用スプライト）

〈例：田中、木村、中井グループ〉　※田中のプログラム



④プログラム例（個人認証機能：数字をクリックしてパスワード入力する例）

④－１　設定の整理

●1～9のスプライトを用意し、それをクリックして予め設定したPIN番号と同じ順番であるかどうかを正誤判断するプログラム

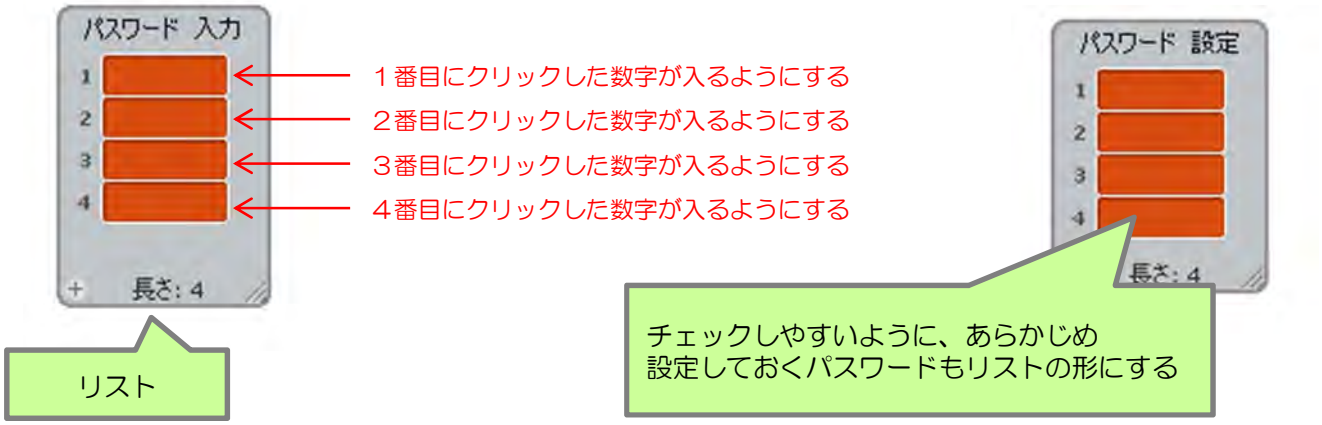
●パスワード
４桁の数字をクリックで入力

１～９のスプライト

●確認方法
パスワードは、クリックして入力するため 

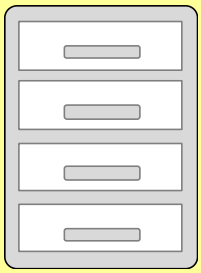
を使って入力欄に手入力する方法は使えない

●よって、「リスト」という機能を使用する



〈リストとは〉

●「リスト」とは、複数の変数を入れる引き出しのようなものです。



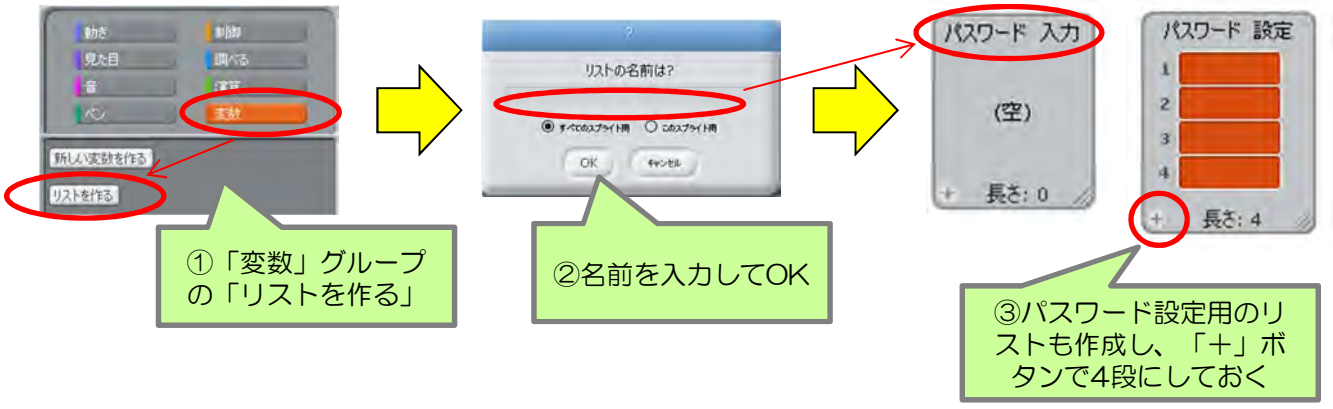
●リストはいくつも作れるため、名前を付けておきます。

●入っている変数の数も自由に変更できます。

④－２　プログラム例（送信用スプライト）

〈例：田中、木村、中井グループ〉　※田中のプログラム

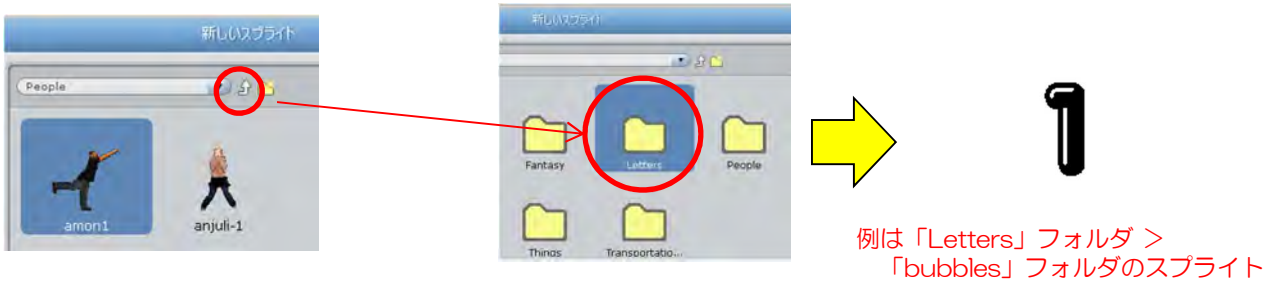
❶リストを作成



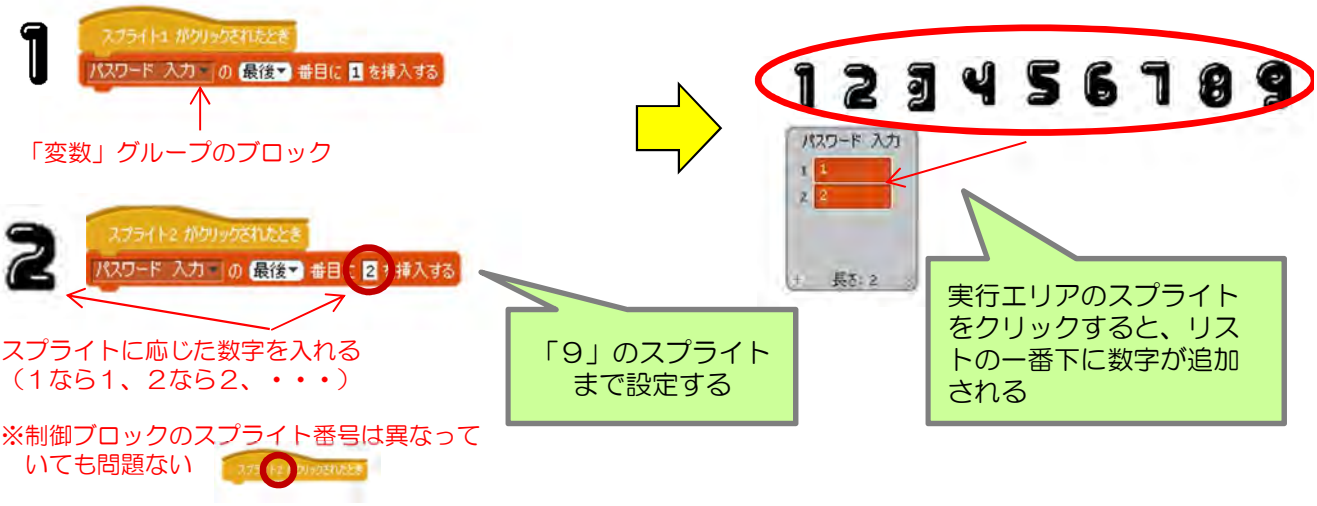
❷数字のスプライトを追加

※ 中学校１年生編の３時間目 ②－２ 参照

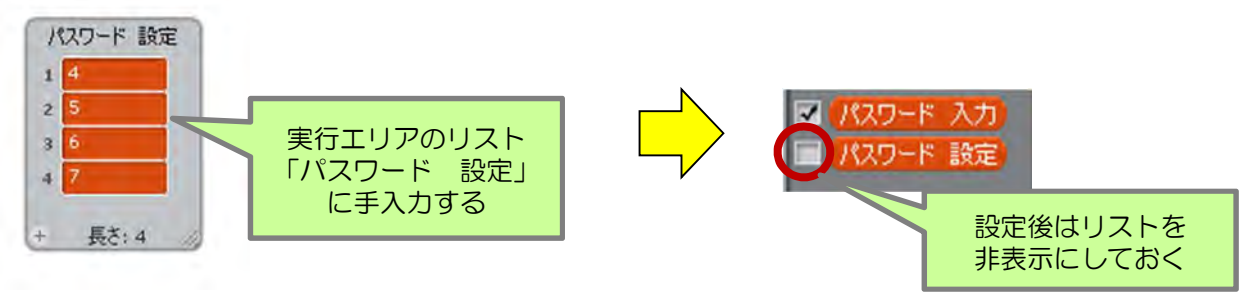
※「Letters」フォルダに数字のスプライトが格納されている



❸数字のスプライトにプログラムを追加



❹パスワードを設定

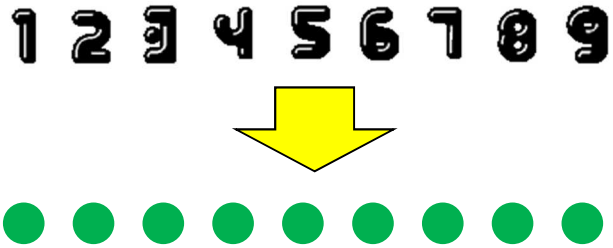


❺パスワードを正誤判定するプログラムを作成

⑤プログラム例（個人認証機能：ドットをクリックしてパスワード入力する例）

●④作成した「数字をクリックしてパスワードを入力するプログラム」を変更する

❶「1～9」のスプライトをドットに変える

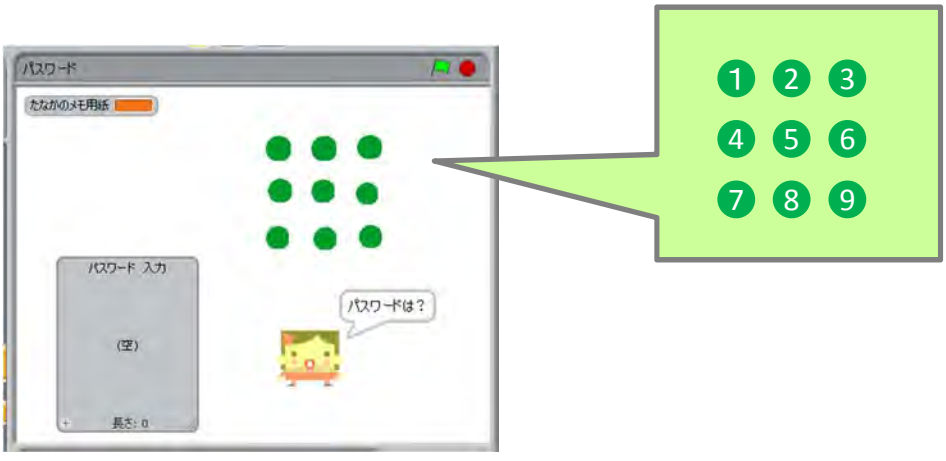


〈スプライトの変更〉

シフトを押しながら
〇を描くと円になる

❷ドットの配置を３×３に並べ替える

※プログラムは変更不要



16～19/20時間目：自分たちが制作したプログラムの紹介webページの制作

●指導例として、あらかじめ作っておいた紹介webページを**<u>生徒が写真や文章を一部編集</u>**して、工夫した点などを紹介する事例を記載する。

①イメージ

●あらかじめ以下のようなwebページのサンプルを用意しておき、生徒が写真や文章を編集する

●編集はHTMLで行う。

〈webページのサンプル〉

②指導の概要（３～４時間で実施）

❶前時までの活動について振り返る

❷プログラムの中で、ホームページで発表したい部分を３つ選ぶ（タイマー機能など）

❸内容の説明文を考える

❹画面をスクリーンショットで保存し、必要な部分を画像編集ソフトウェアで適切に編集を行う（トリミングなど）

❺HTMLについて振り返る（「きのくにICT教育」指導案では2年生の３時間目で履修）

❻**HTMLで紹介のページを作成する**（複数時間にまたがる場合は、作業の進行状況を記録させておく。※どこまで作業が進んだか、次に何から始めるか、など）

❼作成したHTMLをクラスで共有する

・作成方法は以降で説明

❽振り返りを書く

③webページの編集

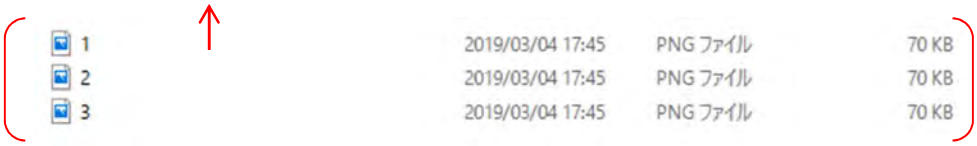
③－１　準備

●Webページのサンプルと画像データのサンプルをダウンロードしておく。

Webページのサンプル

画像データのサンプル

※ 画像データが３つ格納されている



サンプルファイルは、教育センター学びの丘の
「きのくにeラーニング」システムの「きのくにICT教育（中学校）」に掲載

③－２　webページのソースを表示

・サンプルファイルをブラウザで開き、
「表示」→「ソース」

又は

・サンプルファイルを右クリックして
「プログラムから開く」→「メモ帳」

③－３　編集

〈サンプルファイルのソースコード〉

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>技術科　プログラミング</title><!--タイトル-->
<style type="text/css"><!--
body    {background-color:#ffffff;
}
.title {
        color:#006600;
        text-align:center;
}
.text   {
        color:#333333;
        line-height:130%;
}
--></style>
</head>
<body>
<h1 class="title">○組　○班　○○○○○○</h1><!--組、班名、氏名を入れてください-->


<table cellpadding="0" cellspacing="10" border="0" align="center">
<tr>
<td width="320" height="240" valign="top">
<img src="image/1.png" width="320" height="240" alt="写真1"><!-- 写真を指
</td>
<td width="250" valign="top" class="text">
<!-- ここより下に工夫した内容を記述します。-->
工夫した点


<!-- ここまで。-->
</td>
</tr>
</table>
```

〈カラーコード〉
色を決めるコードは16進数で表される。
ブラウザで「カラーコード」等で検索すると一覧を確認できる。
「＃ｆｆｆｆｆｆ」＝白
「＃００６６００」＝濃厚な緑

氏名等を入力

写真①の大きさを指定

＜写真①を指定＞
Imageフォルダ内「１.ping」という名前の写真を表示 ※別の写真に同じ名前を付けて、同じフォルダに保存すれば、その写真が表示される。
「1.ping」を写真の名前に合わせて編集しても良い

工夫した点①を入力

〈次ページへ続く〉

〈前ページへからの続き〉

```
<table cellpadding="0" cellspacing="10" border="0" align="center">
<tr>
<td width="250" valign="top" class="text">
<!-- ここより下に工夫した内容を記述します。-->
工夫した点

<!-- ここまで。-->
</td>
<td width="320" height="240" valign="top">
<img src="image/2.png" width="320" height="240" alt="写真2"><!-- 写真を指定。大
</td>

</tr>
</table>

<table cellpadding="0" cellspacing="10" border="0" align="center">
<tr>
<td width="320" height="240" valign="top">
<img src="image/3.png" width="320" height="240" alt="写真3"><!-- 写真を指定。大
</td>
<td width="250" valign="top" class="text">
<!-- ここより下に工夫した内容を記述します。-->
工夫した点

<!-- ここまで。-->
</td>
</tr>
</table>

<table cellpadding="0" cellspacing="10" border="0" align="center" style="width:
<tr>
<td width="250" valign="top" class="text">
<!-- ここより下に感想を記述します。-->
感想<br>

<!-- ここまで。-->
</td>
</tr>
</table>

<p align="center"><a href="../index.html">戻る</a></p>
</body>
</html>
```

工夫した点②を入力

写真②の大きさを指定

＜写真②を指定＞
Imageフォルダ内「2.ping」という名前の写真を表示 ※別の写真に同じ名前を付けて、同じフォルダに保存すれば、その写真が表示される。
「2.ping」を写真の名前に合わせて編集しても良い

写真③の大きさを指定

＜写真③を指定＞
Imageフォルダ内「3.ping」という名前の写真を表示 ※別の写真に同じ名前を付けて、同じフォルダに保存すれば、その写真が表示される。
「3.ping」を写真の名前に合わせて編集しても良い

工夫した点③を入力

感想を入力

中学校３年生編

単元の構成

1時間目：ロボットが運転する利点と問題点をしらべよう

2時間目：様々なセンサを扱うコンピュータ

3時間目：問題の解決に役立つ計測・制御

4時間目：ロボットを作って動作を確かめる

5時間目：計測と制御のプログラミングでアクチュエータを動かす

6時間目：計測・制御システムの構想・設計

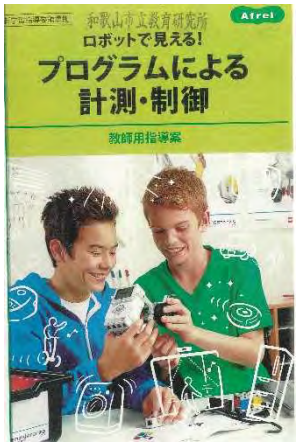
7～8時間目：プログラミング、制作

9時間目：ビジュアル言語とテキスト言語の比較

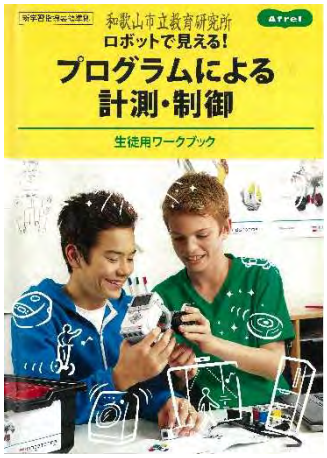
10時間目：成果の評価と技術の概念、改良と応用（社会の発展と技術）

１時間目：ロボットが自動運転する仕組みと利点をしらべよう

●本時で使用するテキスト等



「プログラムによる計測・制御」
教師用指導案（2014年版）
（株）アフレル



「プログラムによる計測・制御」
生徒用ワークブック（2014年版）
（株）アフレル

①センサとアクチュエータの理解（指導過程３）

●「プログラムによる計測・制御」生徒用ワークブック P2については、教師用指導案P9も参照

②閾値の理解（指導過程４）

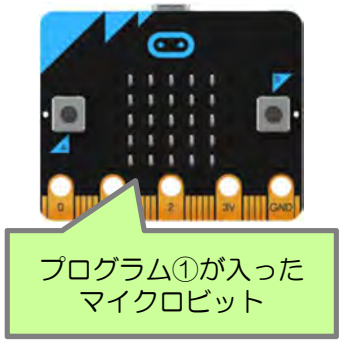
●「プログラムによる計測・制御」生徒用ワークブック P24 も参照

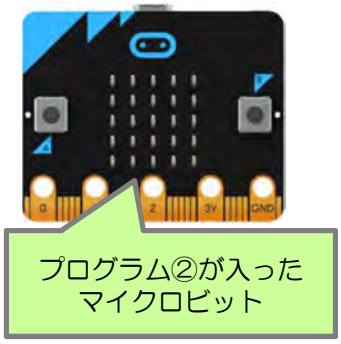
2時間目：様々なセンサを扱うコンピュータ

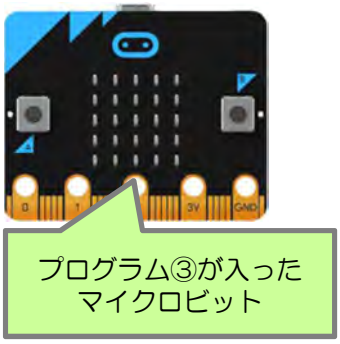
●マイクロビットの使用方法は、中学校１年生編の２時間目②、③参照

①センサープログラムの体験（指導過程３）

●あらかじめ異なるプログラムが入った３種類のmicro:bitを触って，何をしたら何が起こるかを調べる　※各種センサの位置等は、中学校１年生編の２時間目を参照







プログラム①
　https://makecode.microbit.org/_8eci15KyaAAc　又は　bit.ly/kinokuni-01
プログラム②
　https://makecode.microbit.org/_VzWFEJJ6h5jM 又は　bit.ly/kinokuni-02
プログラム③
　https://makecode.microbit.org/_AETdMjLow1b8 又は　bit.ly/kinokuni-03

●あらかじめマイクロビットにプログラムをダウンロードし、電池ボックスをつなげておく

①プログラムのダウンロード
②電池ボックスの接続

中学校１年生編の２時間目
③－２　③－３　参照

②プログラムの確認・変更（指導過程５）

②－１　エディターの起動

②－２　日本語表記への変更　※初期設定で日本語表記の場合は作業不要

②－３　プログラムの確認　※プログラム①の例

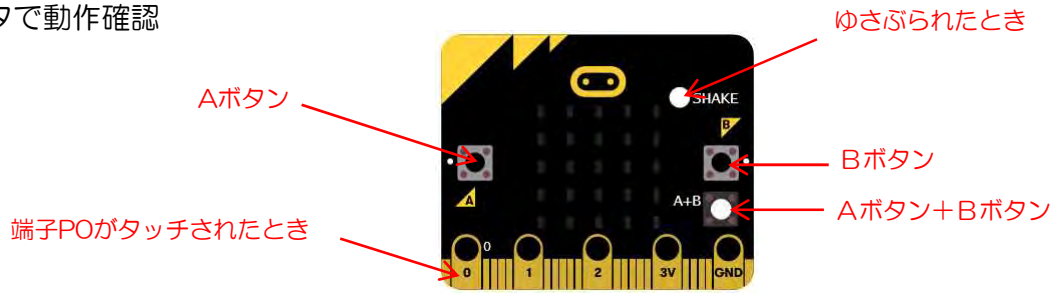
・最初は変数を０にする

・Ａボタンを押す度に変数が１ずつ増えていき、その数字が表示される

・Ｂボタンを押すと変数が０になり、０が表示される

●プログラムを確認し，センサーとアクションの対応を予想しながらプログラムを変更してみる

・シミュレータで動作確認



②－４　プログラムの変更例

Ａボタンを押す度に変数が１ずつ増えていき、その数字が表示される

変数が２ずつ増えるように変更

ゆさぶられたとき、うれしい顔を表示

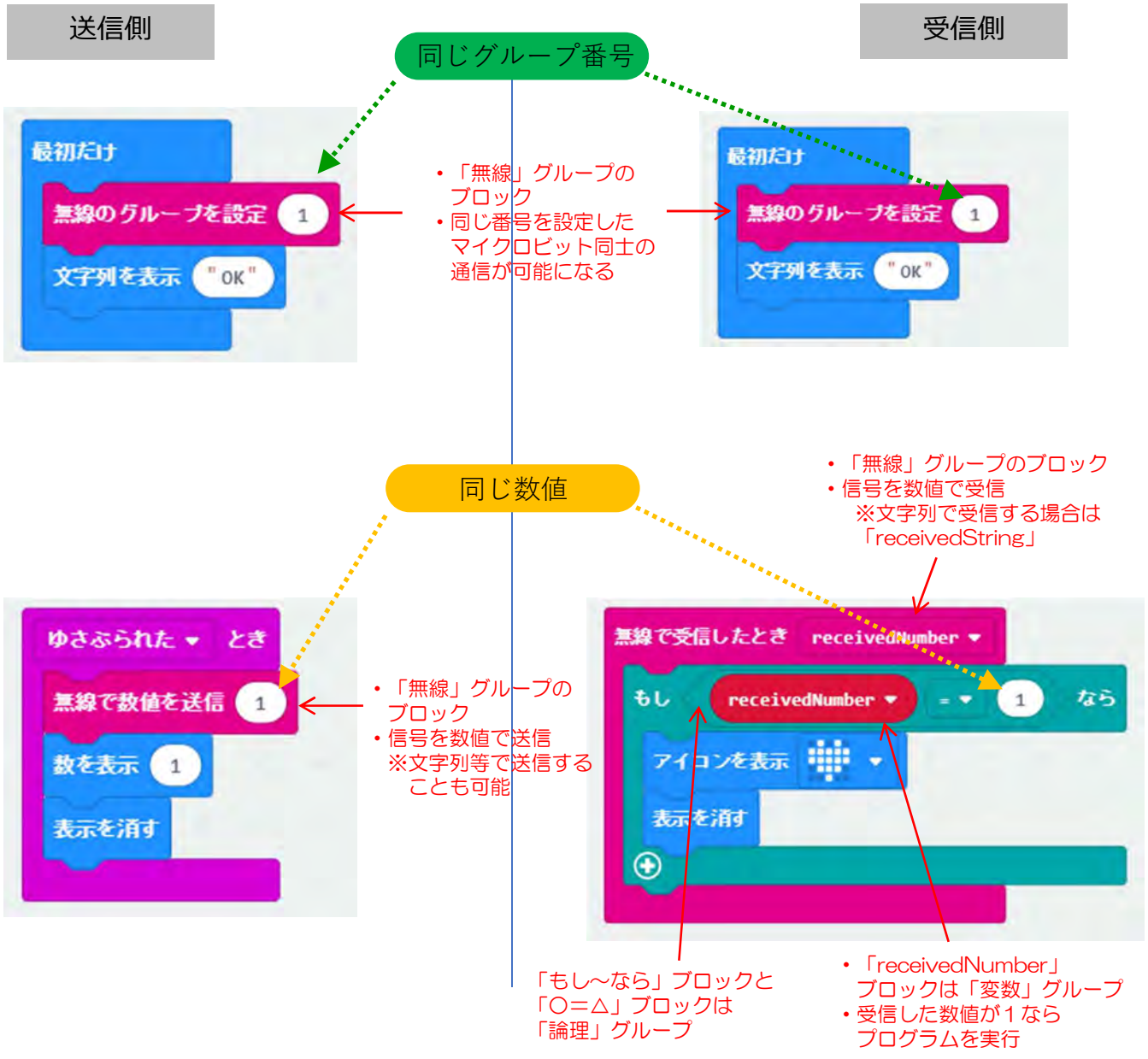
音符が表示されるように変更

③micro:bit（マイクロビット）の無線通信（指導過程６）

③－１　グループ番号の割り当て

- 生徒の各グループに、「無線のグループ番号」を１，２，３…と割り当てていく
  - 同じ「無線のグループ番号」を設定したマイクロビット同士の通信が可能になる

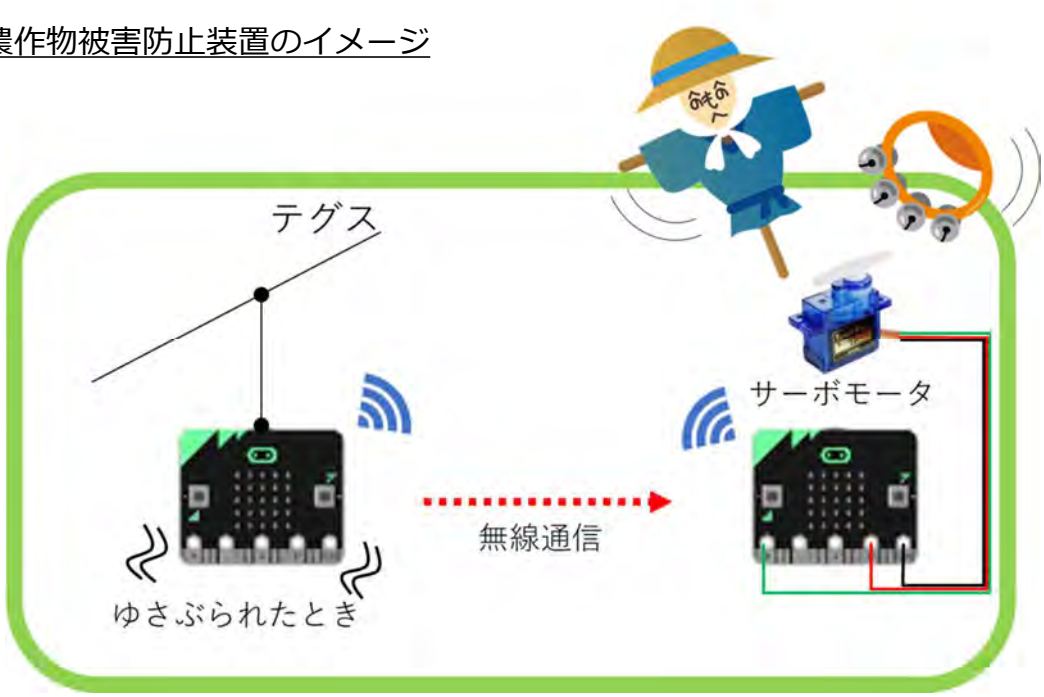
③－２　プログラム例



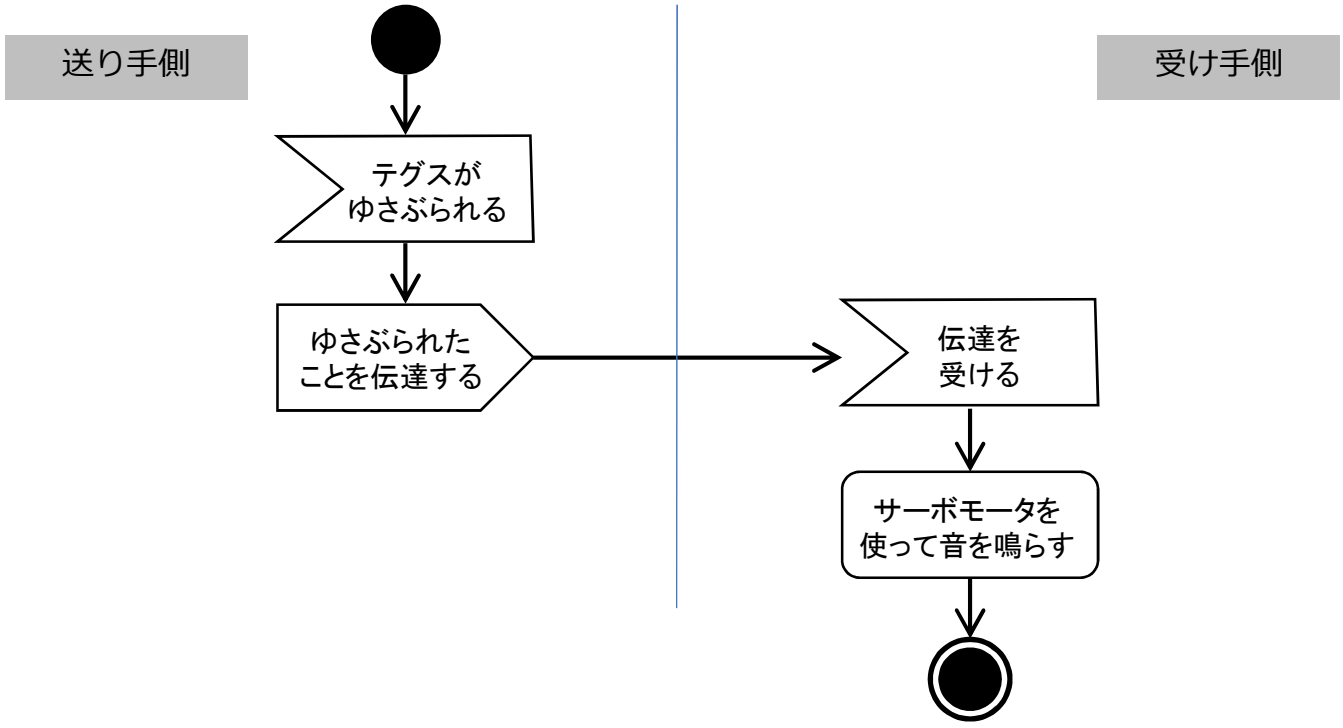
3時間目：問題の解決に役立つ計測・制御

①全体の動作をアクティビティ図で整理（指導過程２）

①－１　農作物被害防止装置のイメージ



①－２　アクティビティ図の例



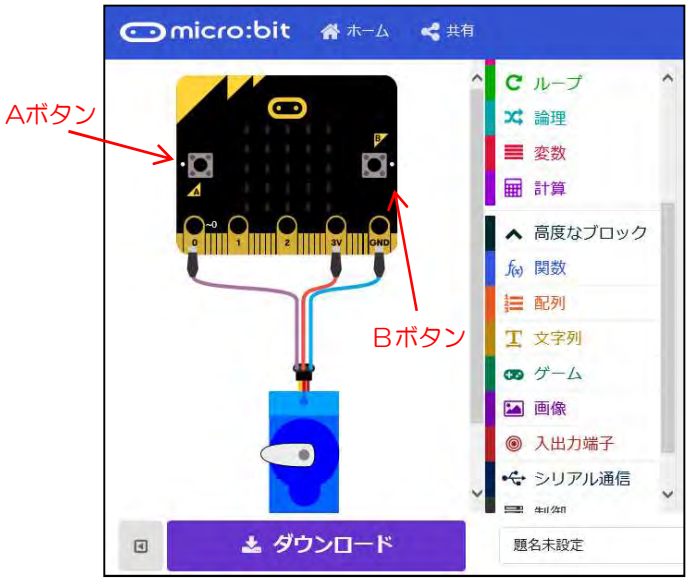
②プログラムと製作（指導過程４）

②－１　エディターを起動

中学校１年生編の２時間目　②－１　②－２　参照

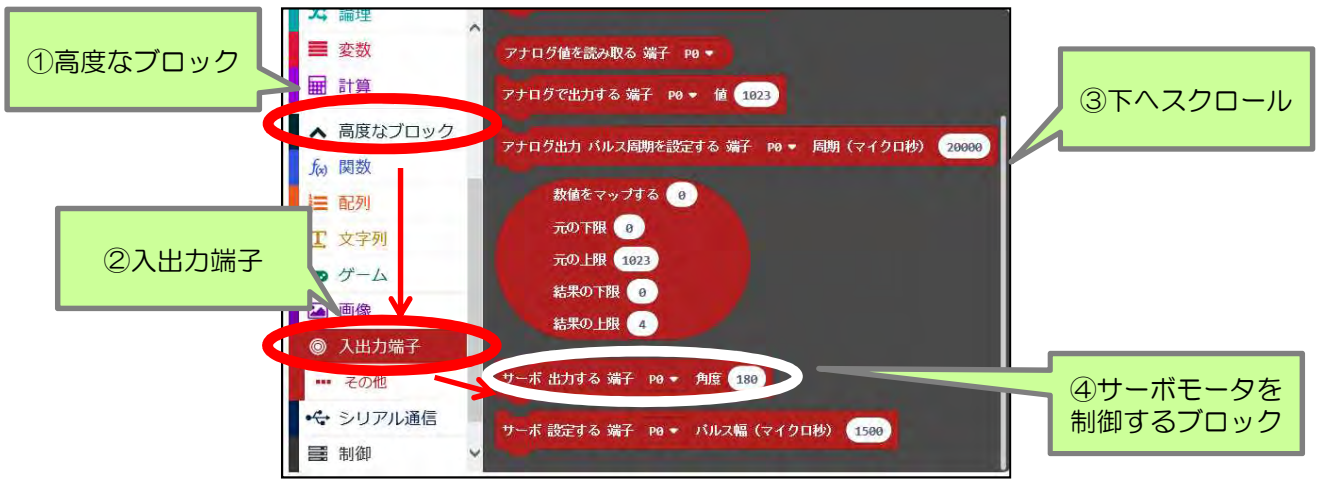
②－２　サーボモータを制御するプログラム

●まずはシミュレータ（ＰＣ上）で動作確認



〈プログラム例〉

〈サーボモータを制御するブロックの場所〉



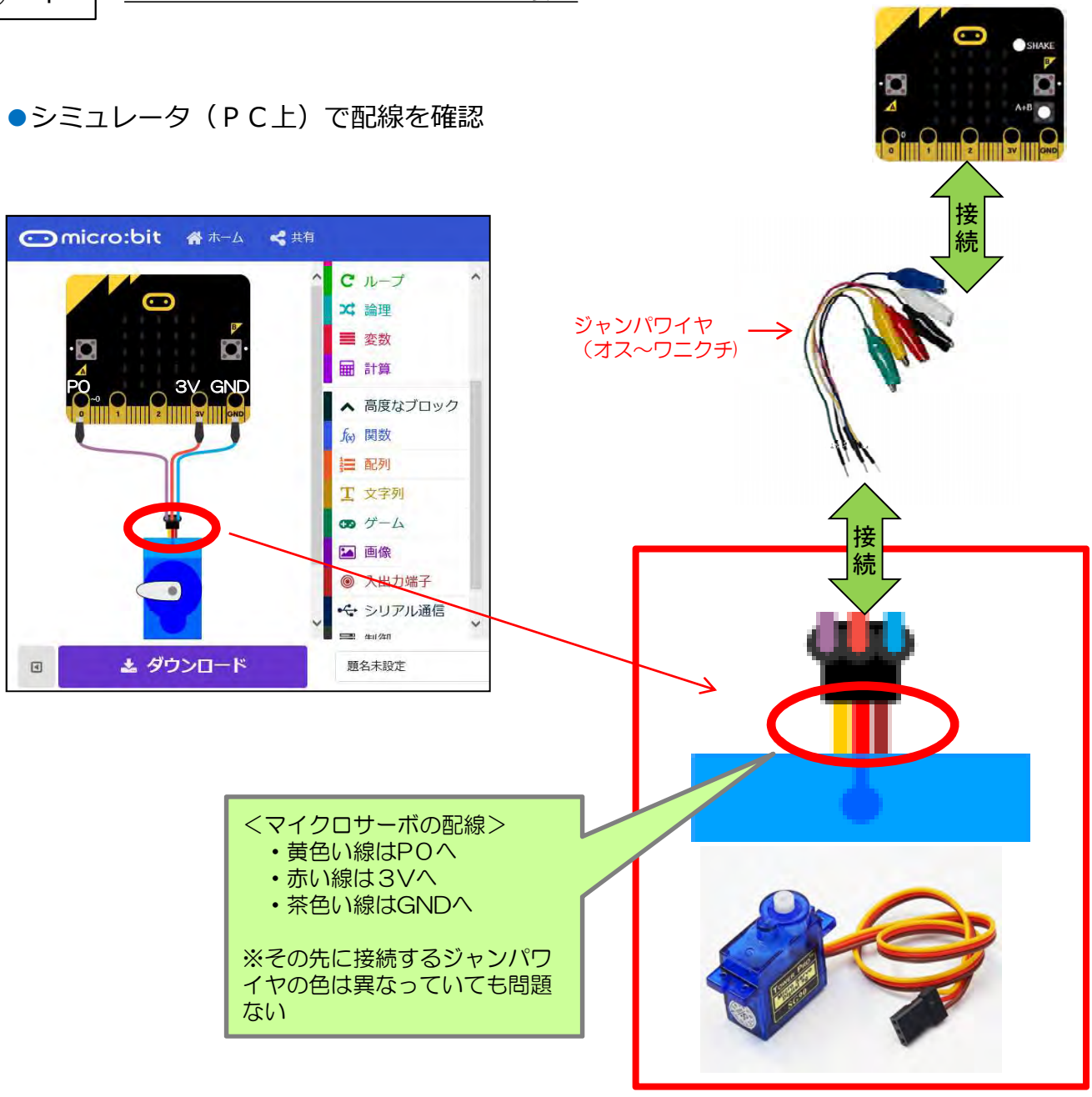
②－３　プログラムのダウンロードと電池ボックスの接続

● ＵＳＢケーブルでマイクロビットとＰＣが接続された状態だとサーボモータは動かない

①プログラムのダウンロード
②電池ボックスの接続
} 中学校１年生編の２時間目 ③－２ ③－３ 参照

②－４　サーボモータをマイクロビットに接続

● シミュレータ（ＰＣ上）で配線を確認

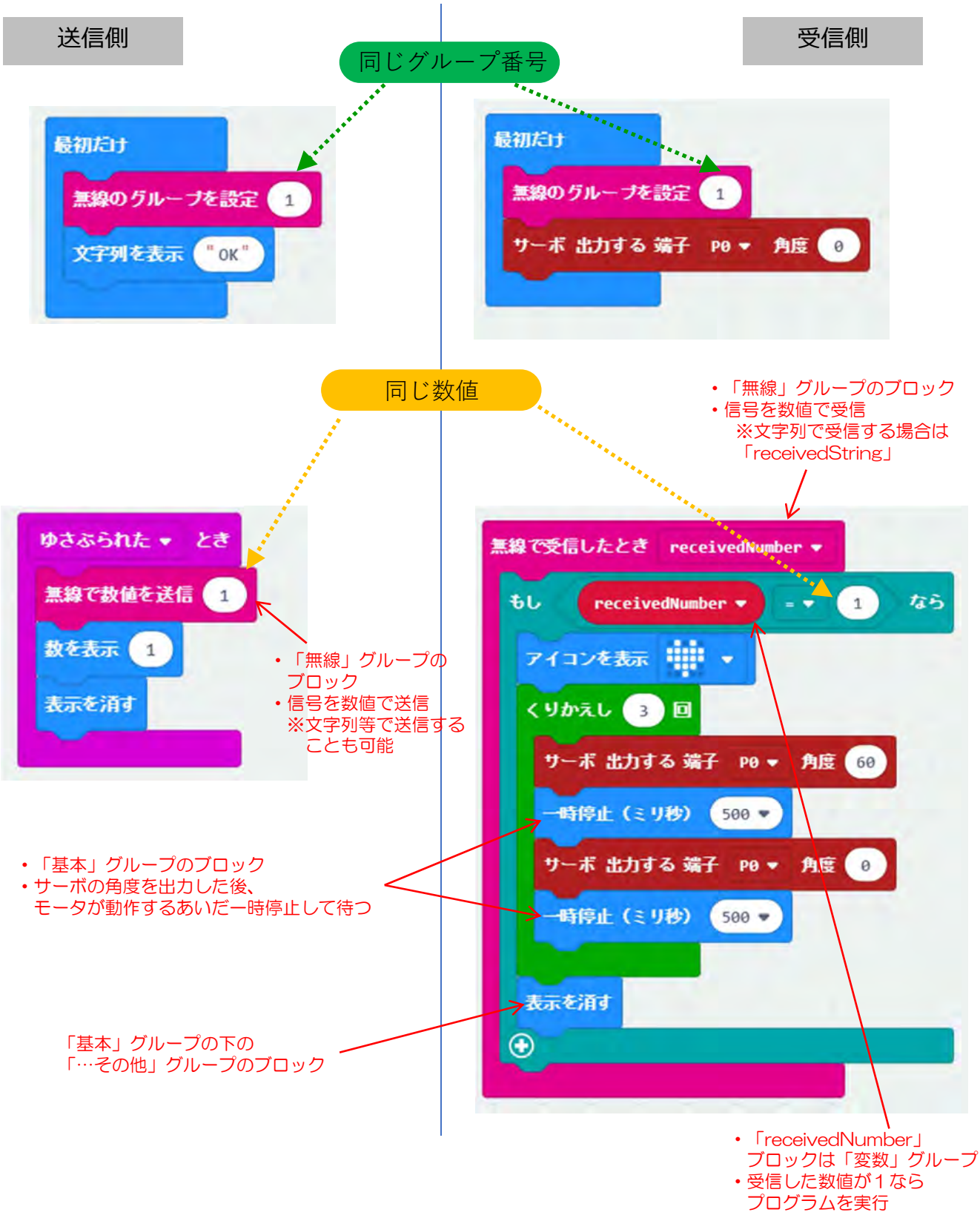


＜マイクロサーボの配線＞
- 黄色い線はP０へ
- 赤い線は３Vへ
- 茶色い線はGNDへ

※その先に接続するジャンパワイヤの色は異なっていても問題ない

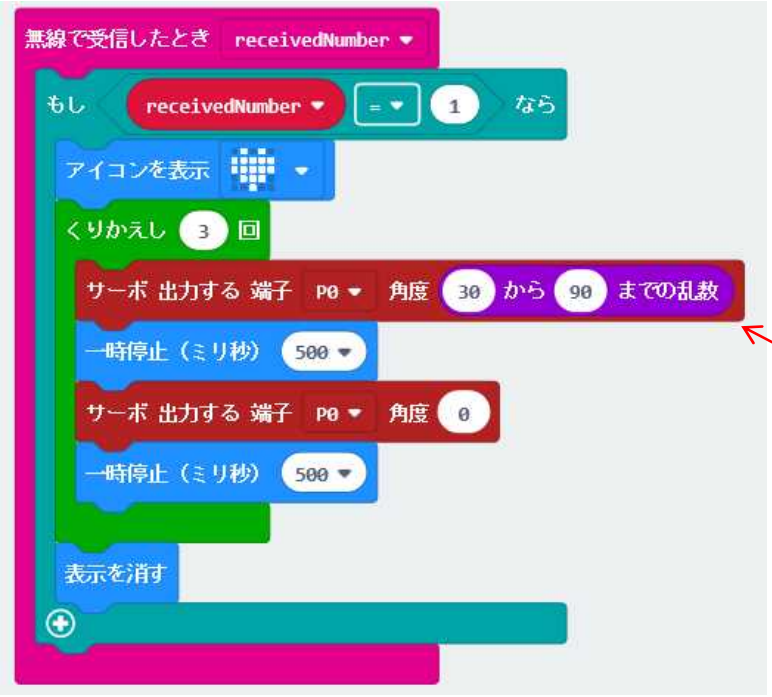
③アクチュエータ側（レシーバ側）の設計とプログラミング（指導過程５）

③－１　“無線通信で”サーボモータを制御するプログラム例



③－２　乱数を使って、毎回異なる動きをさせるプログラム

〈プログラム例〉



「計算」グループの乱数ブロックを使う

※数値を手入力で変更する

4時間目：ロボットを作って動作を確かめる

●本時で使用するテキスト等

「プログラムによる計測・制御」
教師用指導案

「プログラムによる計測・制御」
生徒用ワークブック

〈ロボット教材〉

レゴ® マインドストーム® EV3
※4,5,7,8時間目で使用

①センサーカーの組み立て（事前準備）

●「プログラムによる計測・制御」教師用指導案付属CD-ROM または
MINDSTORMSソフトウェアのロビー内を参照

・センサーカーの組み立てには時間を要するため、授業前に組み立てておくことが望ましい

②プログラミング環境（指導過程２）

●生徒用ワークブック P10-12 参照

あらかじめ
インストール

ロビー

③パソコンと切り離して走らせる（指導過程４）

- 「プログラムによる計測・制御」生徒用ワークブック P15を参照

（１）EV3インテリジェントブロックの電源が入っていることを確認し、USBケーブルでPCと接続（無線接続も可能）

（２）ハードウェアページで接続が確認できたら、ダウンロードボタンをクリック

（３）EV3から「ピポッ」と音がしたらダウンロード完了

（４）USBケーブルを外し、本体ボタンを操作して実行するプログラムを選択して実行

④ワークシートの解答例（指導過程６）

- 教師用ワークブックに記載

⑤ワークシートの解答例（指導過程７）

- 教師用ワークブックに記載

5時間目：計測と制御のプログラミングでアクチュエータを動かす

●本時で使用するテキスト等

「プログラムによる計測・制御」
教師用指導案

「プログラムによる計測・制御」
生徒用ワークブック

①使用できるセンサの説明（指導過程３）

●「プログラムによる計測・制御」生徒用ワークブック　P5

※教育版レゴマインドストームEV3に含まれているセンサ

- カラーセンサ
- タッチセンサ
- ジャイロセンサ
- 超音波センサ

②タッチセンサによるスイッチング（指導過程４）

②－１　タッチセンサの取り付け

●タッチセンサーを入力ポート1に接続

②－２　タッチセンサの数値確認

● 指導案付属のワークシート
「様々なセンサとアクチュエータを扱うシステムとコンピュータその１」
３．タッチセンサーの数値確認

EV3ブロックでPort Viewを実行し，タッチしたときの値の変化を確認
1. EV3ブロックを起動
2. 左右ボタンで「アプリ画面」
3. 「Port View」を選択
4. 左右ボタンでセンサーのポートを選択

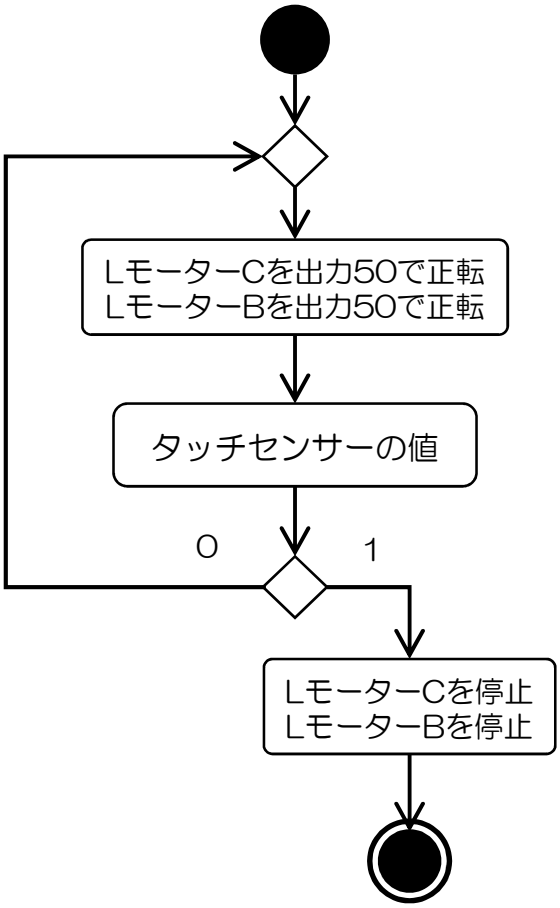

②－３　動作の整理とアクティビティ図

・実社会でより多く使われている、タッチされると「停止」するプログラムを作成する

●指導案付属のワークシート
「様々なセンサとアクチュエータを扱うシステムとコンピュータその１」
４．動作の整理とアクティビティ図

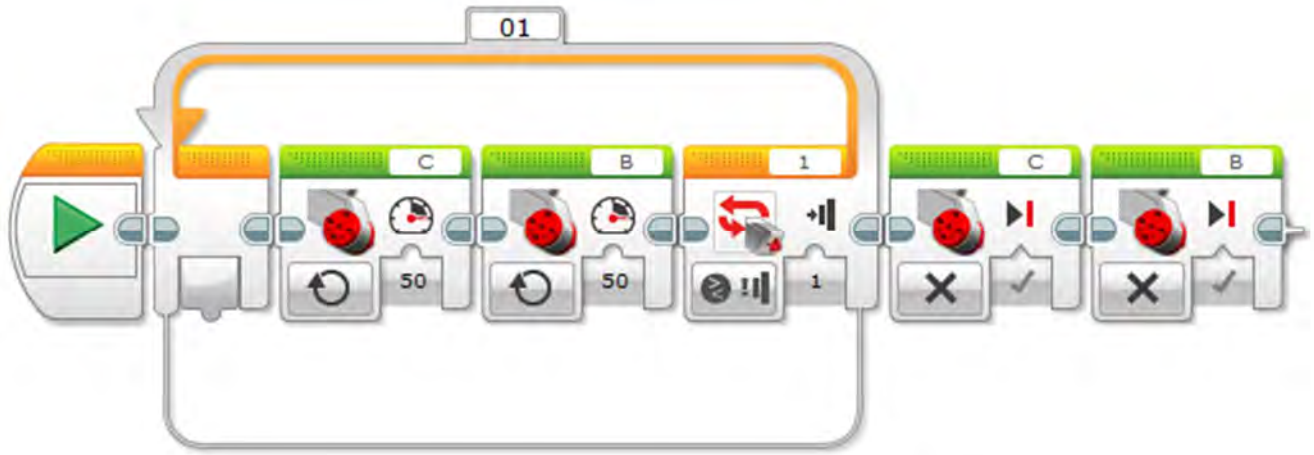
●アクティビティ図の書き方は中学校２年生編の
4/20時間目を参照

順番	動作	プログラム
①	タッチセンサーが押されるまで繰り返す	タッチセンサー(ポート1)の値＝ 1 まで繰り返す
②	前進する	Lモーターcを出力50で正転 Lモーターbを出力50で正転
③	停止する	Lモーターcを停止 Lモーターbを停止



②－４　プログラミング

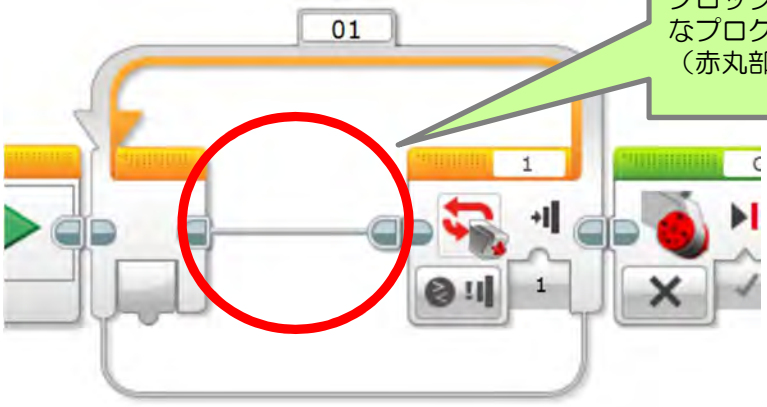
●指導案付属のワークシート
「様々なセンサとアクチュエータを扱うシステムとコンピュータその１」
５．プログラミング



②－５　参考

「…まで待つ」ブロックを使用するプログラムも考えられる。
ただし、待っている間は、さまざまなプログラムを実行することができない。（ただ待つだけ）

②－４ で示したように「…まで繰り返す」ブロックを使うことで、待っている間もさまざまなプログラムを実行することができる。
（赤丸部分にプログラムを追加できる）



③超音波センサーの利用（指導過程５、６）

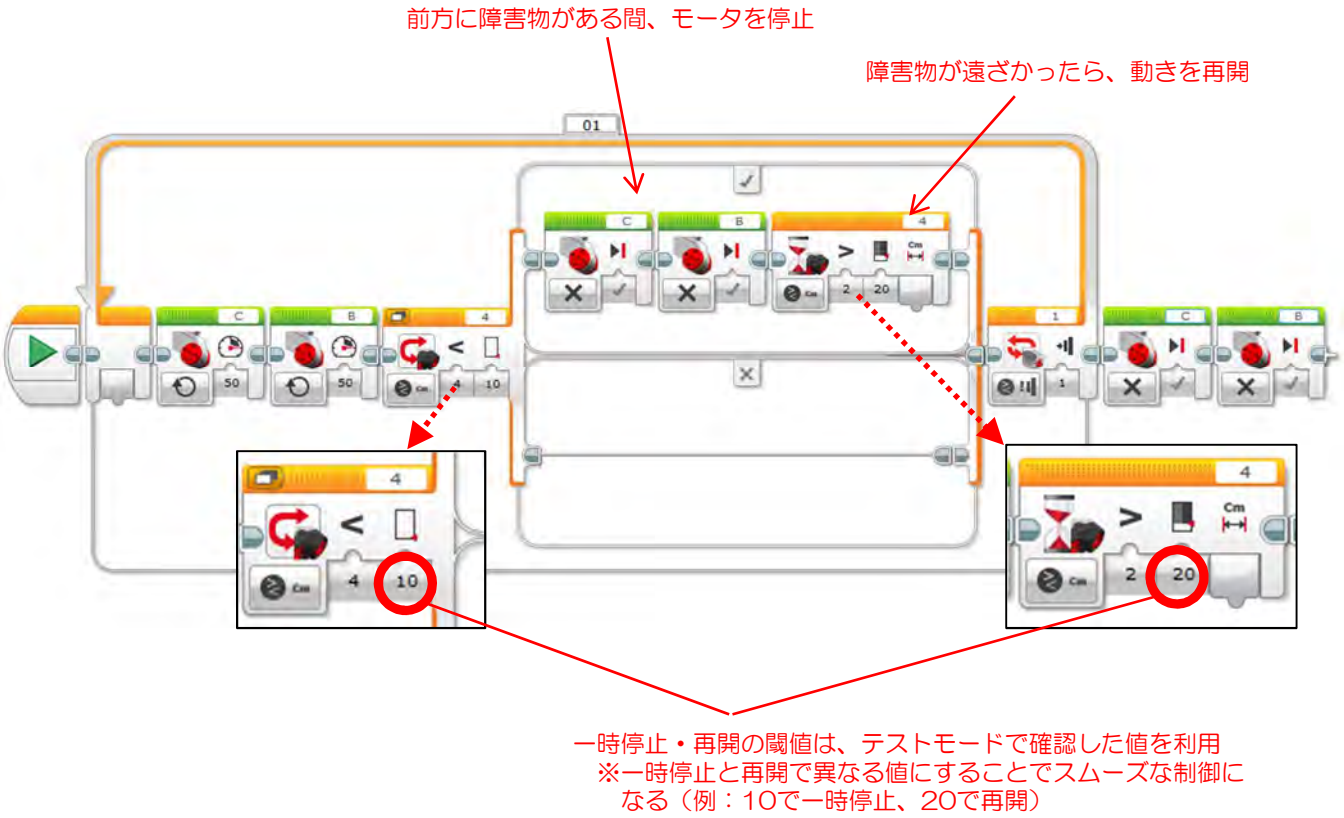
③－１　閾値の決定

- 「プログラムによる計測・制御プログラミング」生徒用ワークブック　P30参照

③－２　衝突回避カーのプログラム例

・先に作成したタッチでストップするプログラムに、障害物検知で一時停止するプログラムを追加

※超音波センサーをポート4に接続した場合

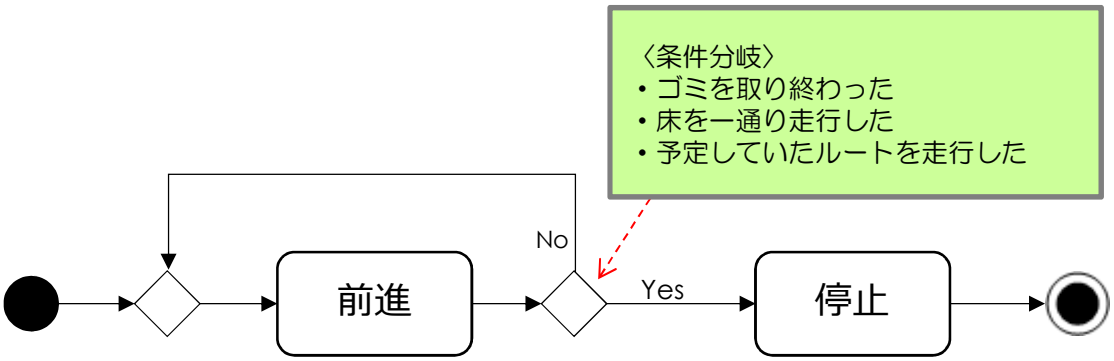


6時間目：計測・制御システムの構想・設計

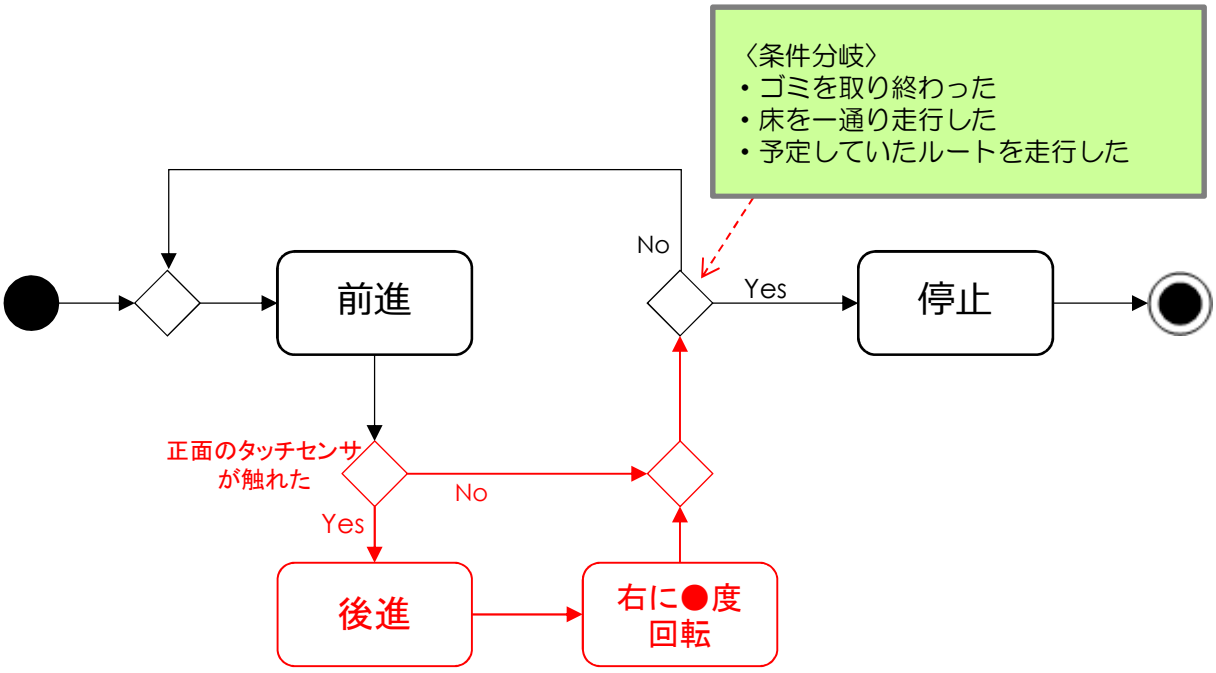
①業務用掃除ロボに求められる課題を図示（指導過程２）

● アクティビティ図の書き方は中学校２年生編の4/20時間目を参照

〈掃除ロボの基本動作〉



〈壁にぶつかったら後進、旋回、を追加〉　※赤色部分を追加



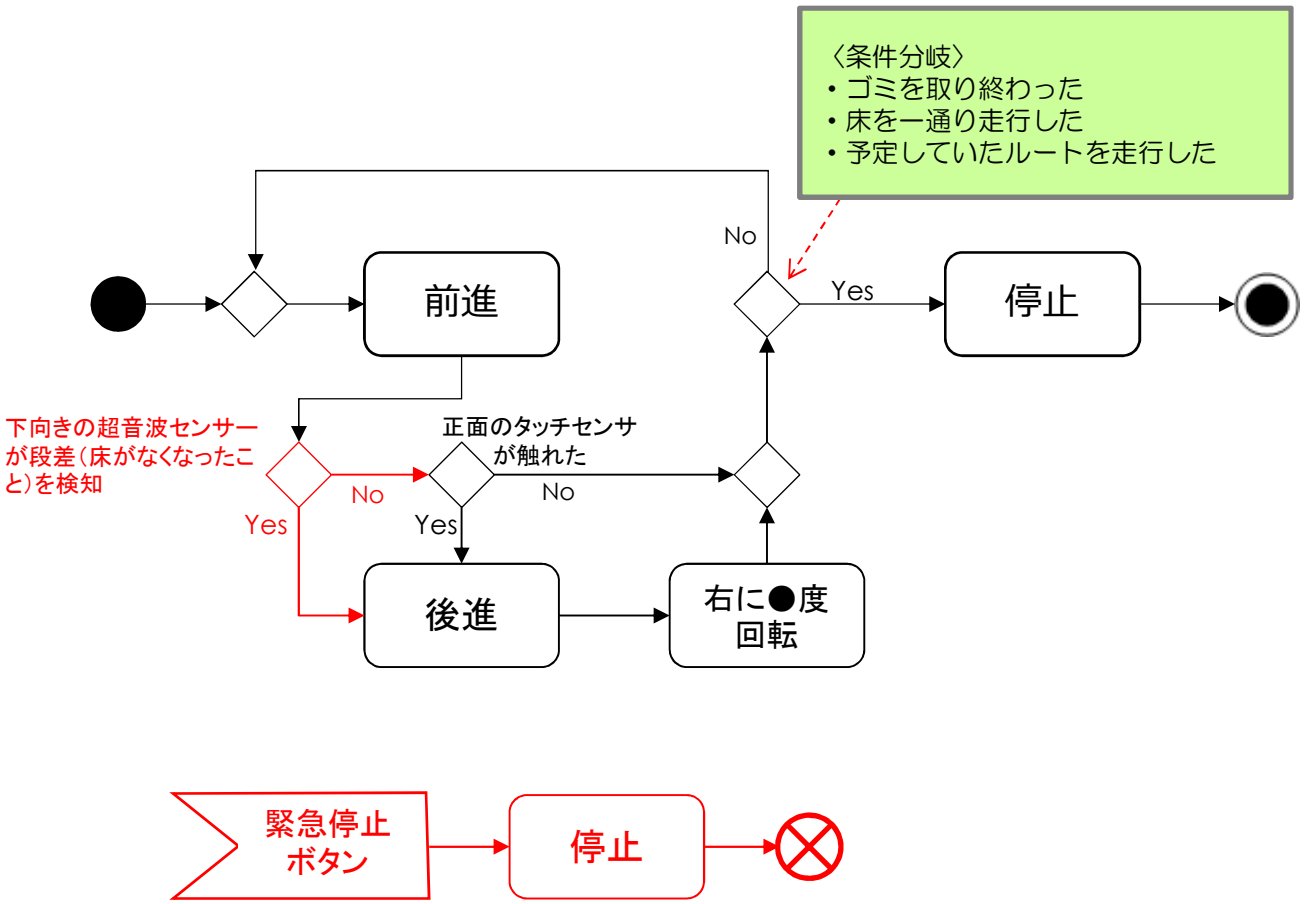
②どのような課題があるか（指導過程３）

※教育用MINDSTORMS EV3に含まれているセンサ

- タッチセンサー
- 超音波センサー
- カラーセンサー
- ジャイロセンサー

③手順分解やアルゴリズム（指導過程４）

〈凹みの段差検知、緊急停止ボタンを追加〉　※赤色部分を追加



7,8時間目：プログラミング、制作

①プログラム例（カラーセンサーやタッチセンサーの利用）

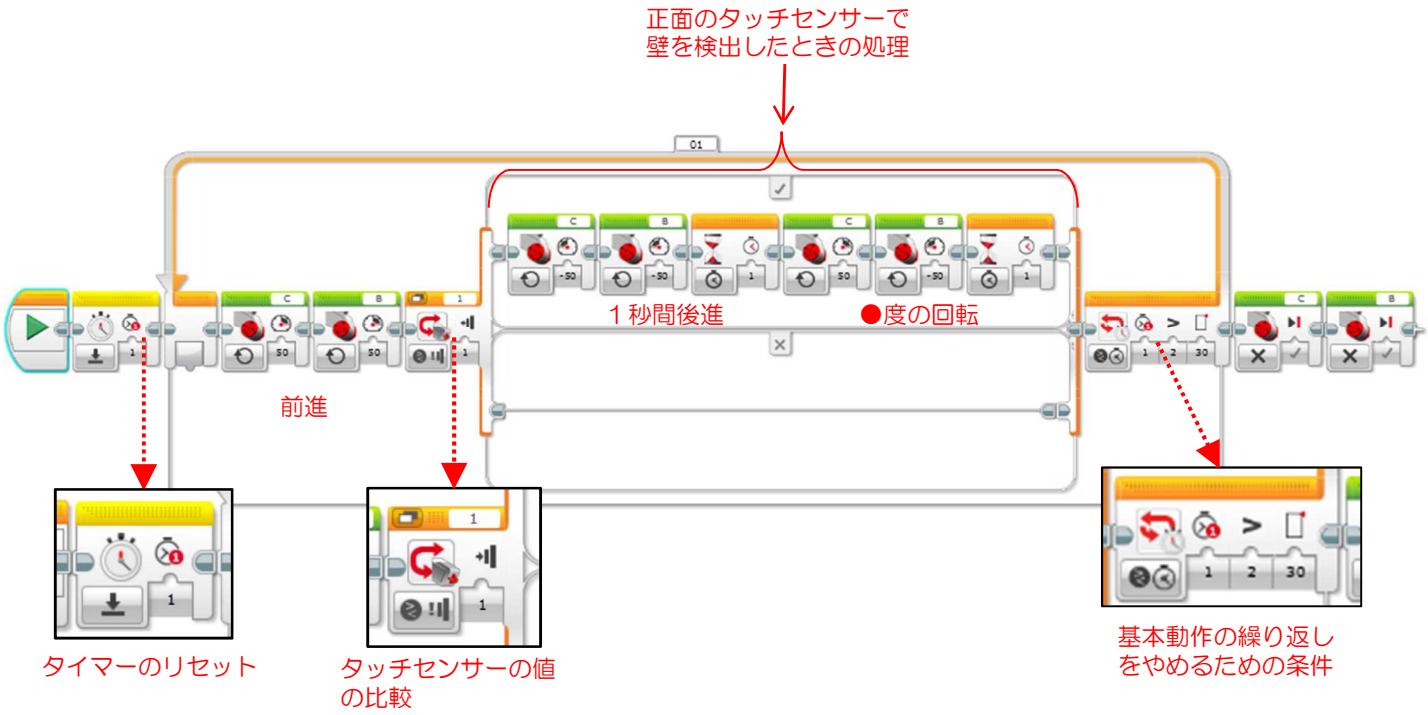
●ライントレースカー

→　生徒用ワークブックP24～29参照

●掃除ロボ

※基本動作（ここでは30秒前進して止まる）の中で，
正面のタッチセンサー(ポート1)で壁を検知したら
後進・旋回

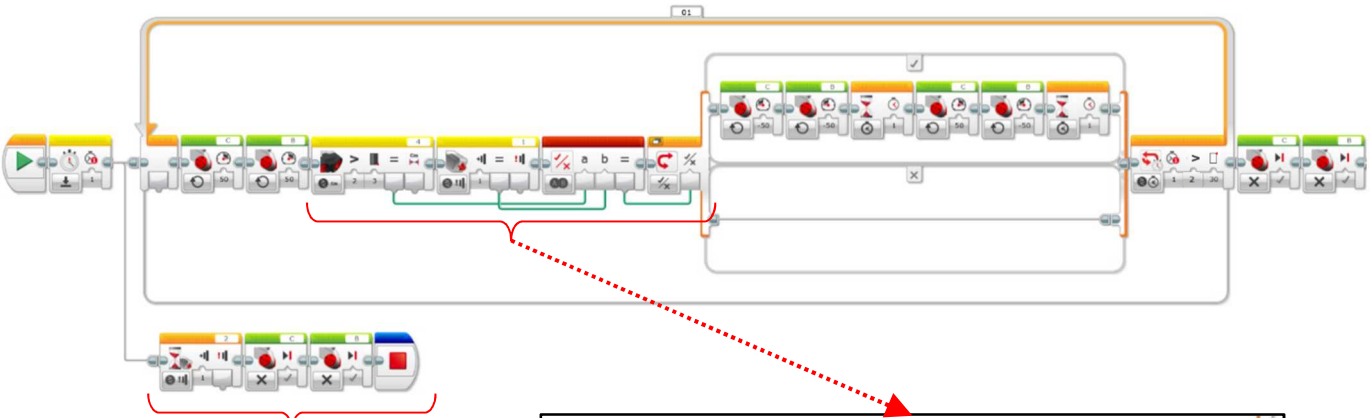
「プログラムによる計測・制御」
生徒用ワークブック

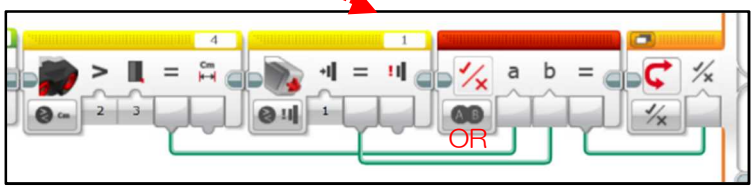


②プログラム例（超音波センサー ＋ タッチセンサー×２）

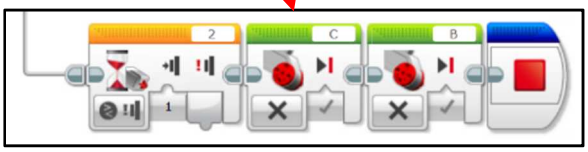
●前ページの掃除ロボに，段差検知機能と緊急停止ボタンを追加

段差検知：下向きにつけた超音波センサー（ポート4）の値が閾値より大きい
緊急停止ボタン：タッチセンサー（ポート2）が押されたら停止





段差を検知したときにも後進・旋回を行うように，ロジック操作ブロックを使って，
（超音波センサーが閾値より大きい）または（タッチセンサーが1）を表している



緊急停止ボタンが押されていたときにモーターを停止し，プログラムの実行をやめる
基本動作と並列に実行

9時間目：より効率の良いプログラミングの技術（ﾃｷｽﾄﾌﾟﾛｸﾞﾗﾐﾝｸﾞの良さ）

①課題把握（指導過程１）

①－１　世界で最も使用されているプログラミング言語

● https://www.benfrederickson.com/ranking-programming-languages-by-github-users/

〈主なプログラミング言語〉

- **JavaScript**　（ジャバスクリプト）
- **Python**　（パイソン）
- **Java**　（ジャバ）
- **C++**　（シープラスプラス）
- **C**　（シー）
- **PHP**　（ピーエイチピー）
- **C#**　（シーシャープ）
- **Shell**　（シェル）
- **Go**　（ゴー）
- **TypeScript**　（タイプスクリプト）

上位幾つかの言語を紹介し，その言語名を検索エンジンで画像検索し，全てが文字で入力されている言語であることを見せる。

②マイクロビットのエディタの起動（指導過程２）

②－１　ブラウザ（インターネットエクスプローラなど）を立ち上げる

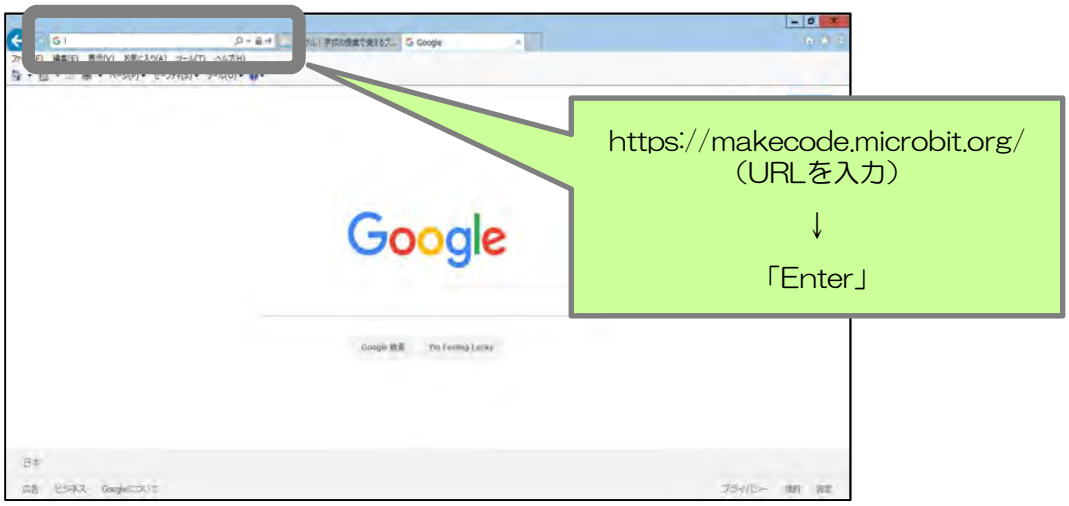
インターネット
エクスプローラ

グーグル
クローム


マイクロソフト
エッジ


※アイパッドの場合は
マイクロビットのアプリの
インストールが必要

②－２　マイクロビットのエディタへ　※Googleの例



②－３　プログラムを組む

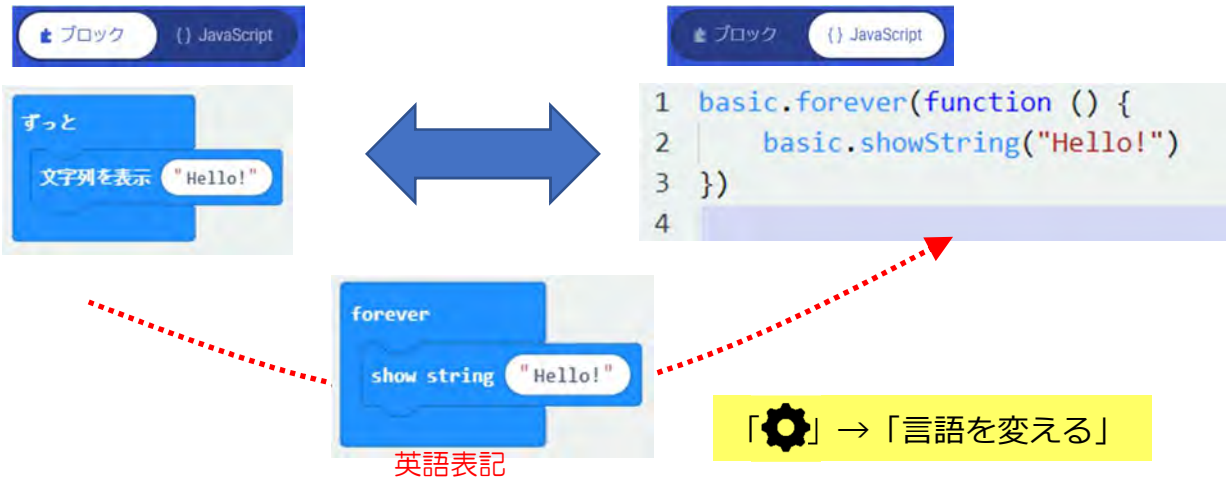
③テキスト言語との対比（指導過程３）

③－１　エディタを「｛｝JavaScript」（ジャバスクリプト）に切り替える

②プログラムがテキスト型
言語に切り替わる

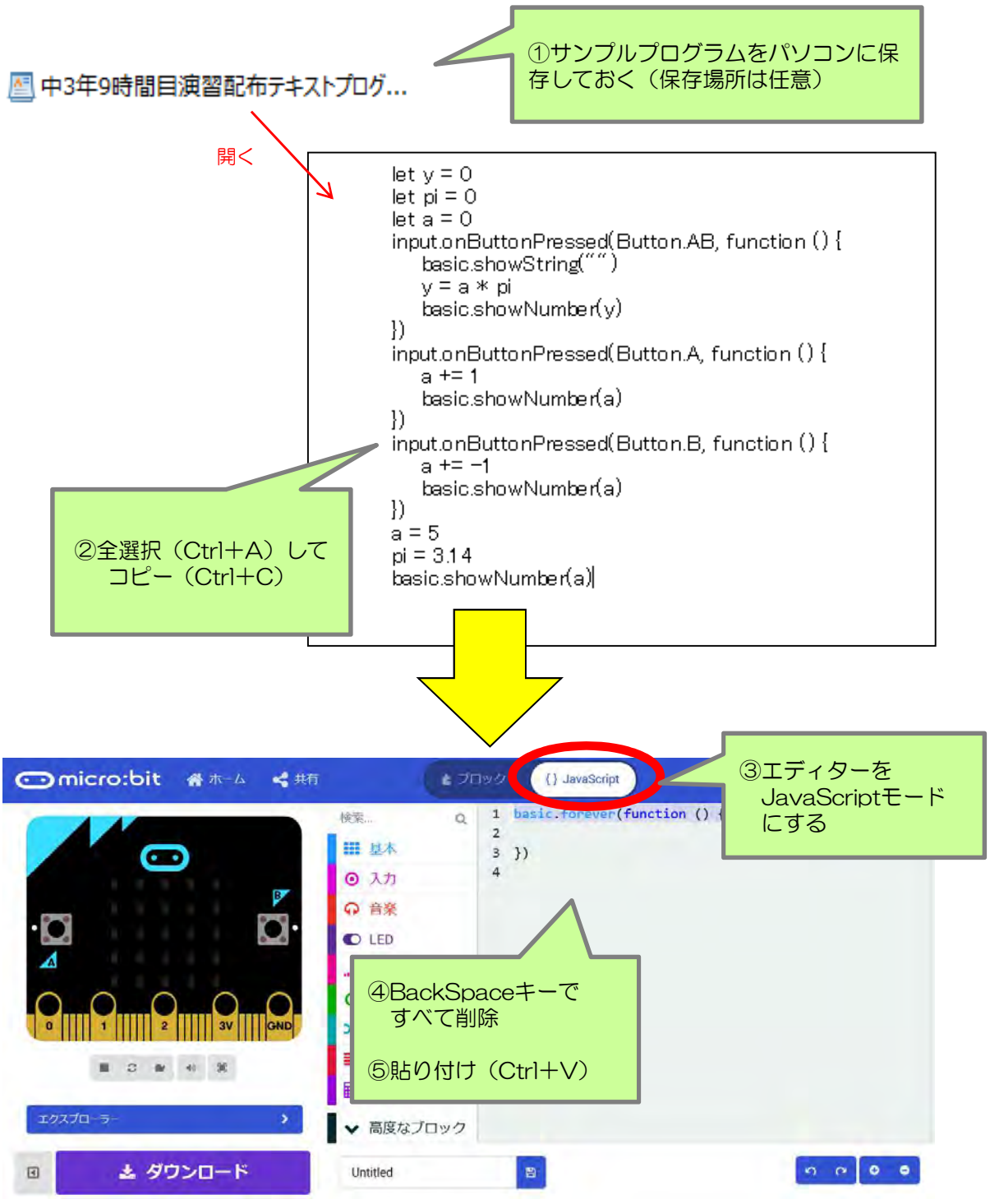
```
1  basic.forever(function () {
2      basic.showString("Hello!")
3  })
4
```

※英語のブロックを経由しても良い



④サンプルプログラムの実行（指導過程４）

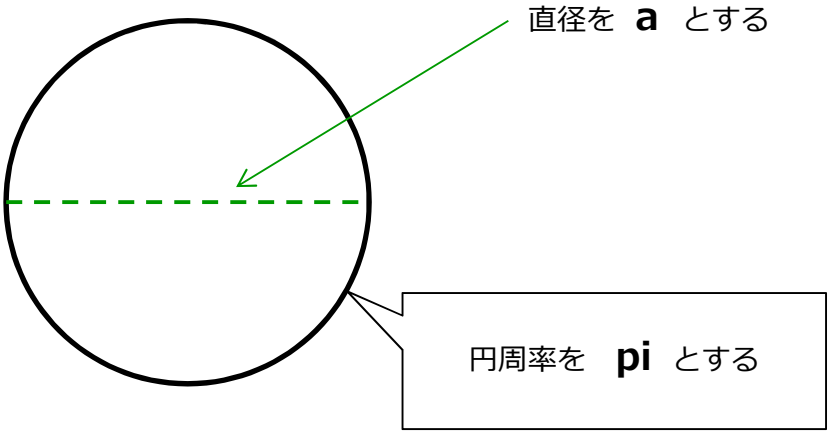
サンプルプログラムは、教育センター学びの丘の
「きのくにeラーニング」システムの「きのくにICT教育（中学校）」に掲載

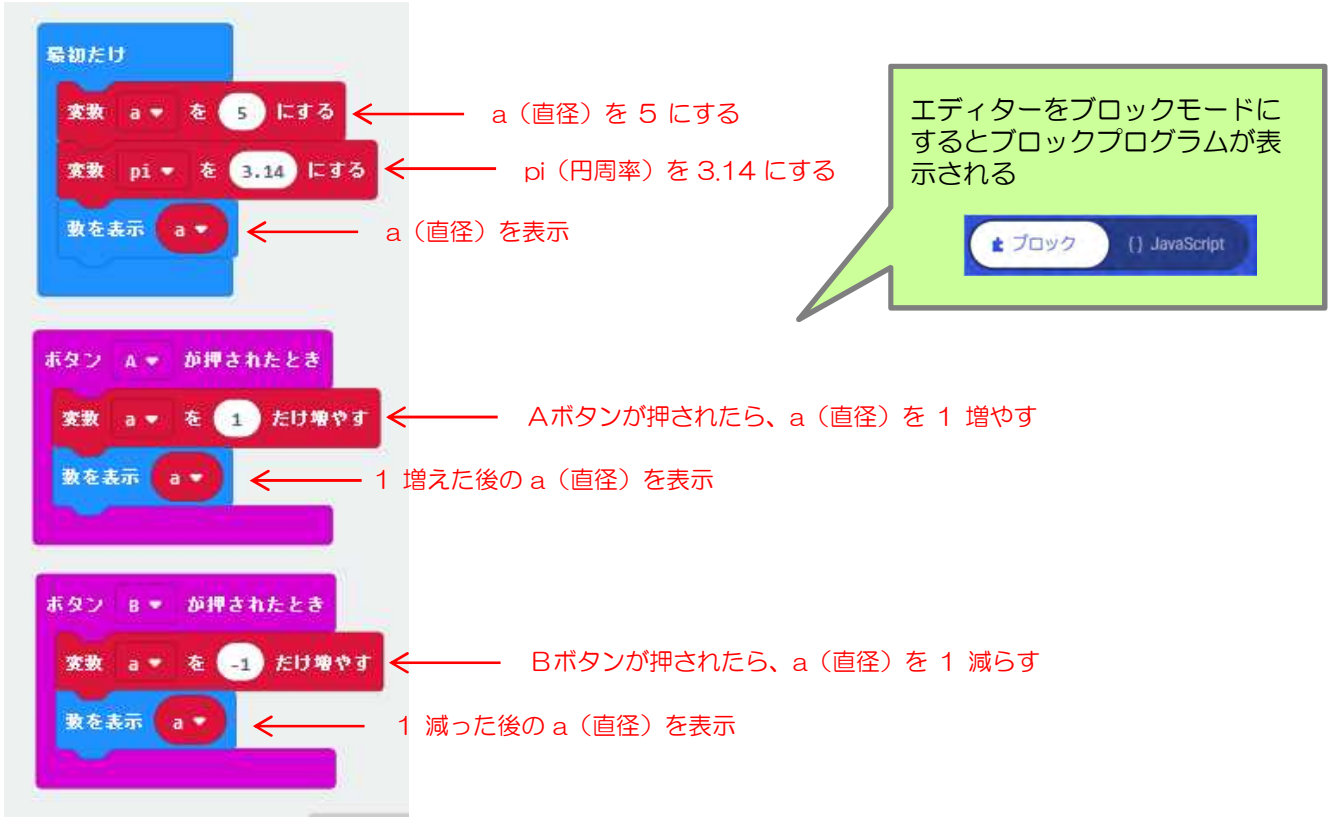


※生徒のパソコン操作の習熟度に合わせて、コピー・貼り付けを以下の方法としても良い
マウスでプログラムをドラッグ　⇒　右クリック「コピー」⇒「貼り付け」

④テキストプログラミングで数式を変更（指導過程４①）

④－１　円周を求めるプログラム（ブロックプログラミング）





④－２　円周を求めるプログラム（テキスト型プログラミング）

```
let y = 0
let pi = 0
let a = 0
input.onButtonPressed(Button.A, function () {
    a += 1
    basic.showNumber(a)
})
input.onButtonPressed(Button.B, function () {
    a += -1
    basic.showNumber(a)
})
input.onButtonPressed(Button.AB, function () {
    basic.showString("")
    y = a * pi
    basic.showNumber(y)
})
a = 5
pi = 3.14
basic.showNumber(a)
```

円周を求める計算式

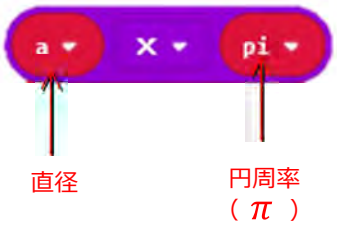
テキスト型プログラミングで“円の面積”を求める計算式に変更する

y =（ a / 2 ） *（ a / 2 ）* pi

⑤ブロックプログラミングとテキストプログラミングを比較（指導過程４②）

A 円周を求める計算式

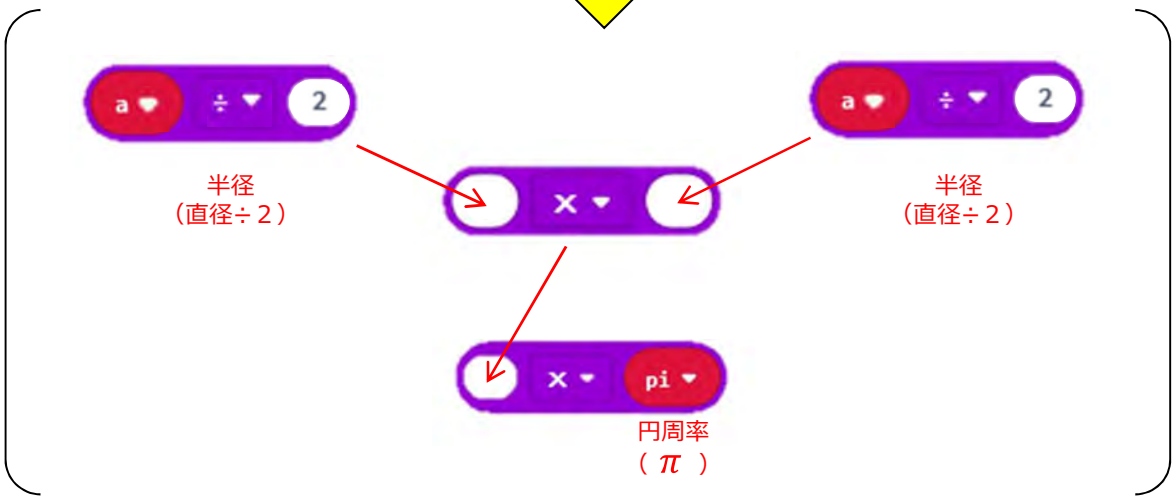
```
y = a * pi
```



B 円の面積を求める計算式

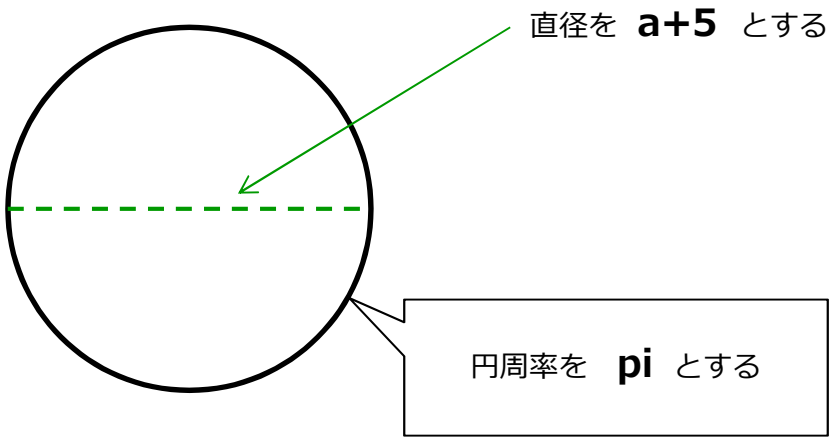
```
y = ( a / 2 ) * ( a / 2 ) * pi
```

ブロックを分解すると…

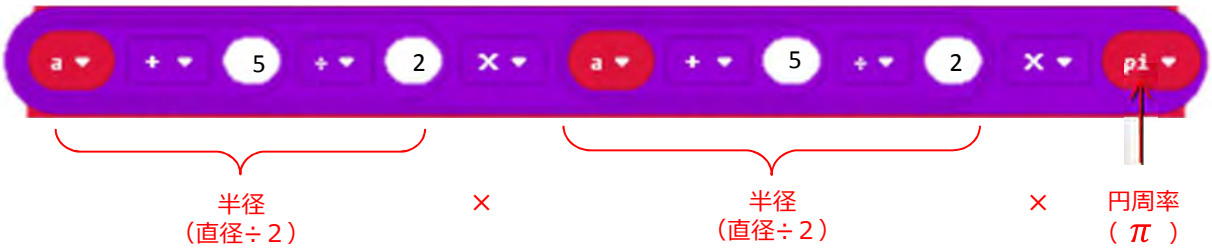


ブロックプログラミングでＡ（円周を求める計算式）からＢ（円の面積を求める計算式）に変更するには、時間と手間がかかることが分かる

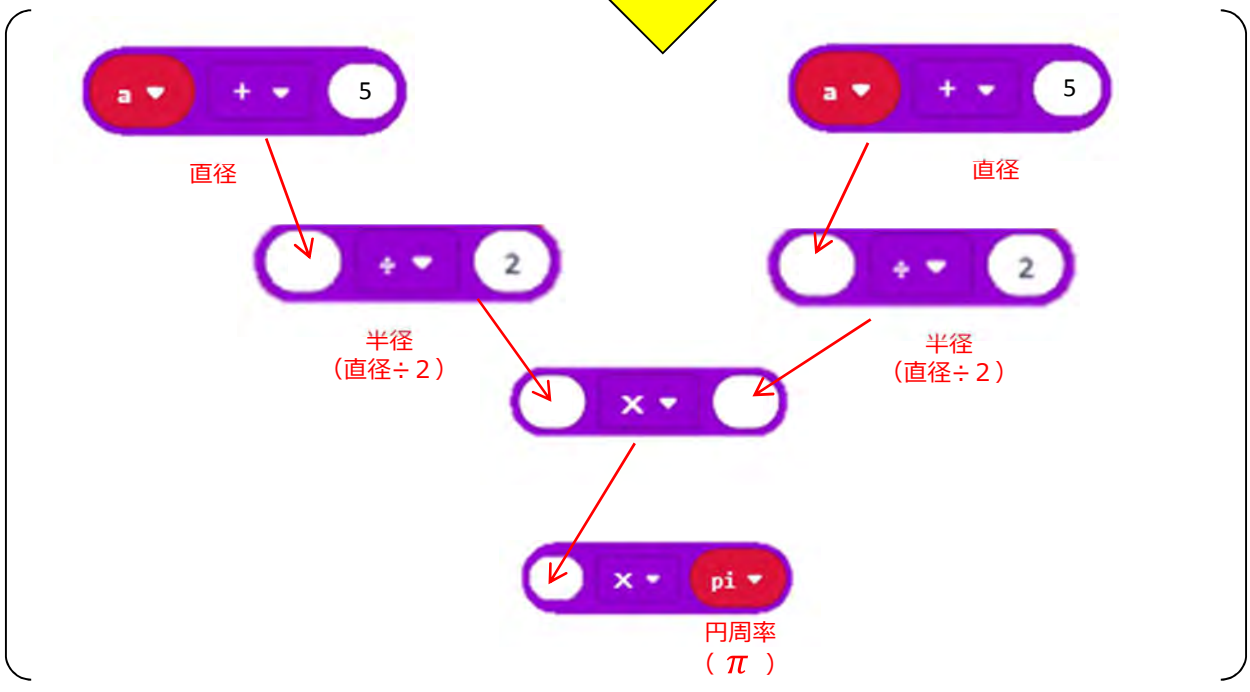
⑥ブロックプログラミングで数式を変更（指導過程４③）



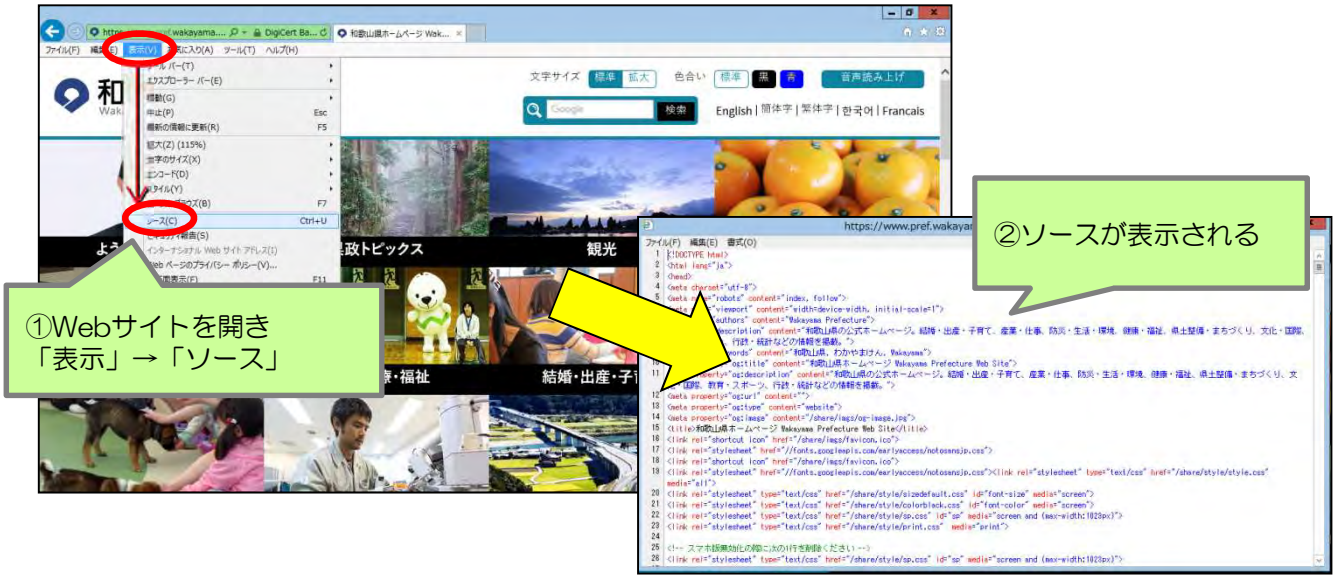
- **「直径（a+5)の円」の面積を求める計算式**



ブロックを分解すると…



⑦ブラウザのソース表示（指導過程６）

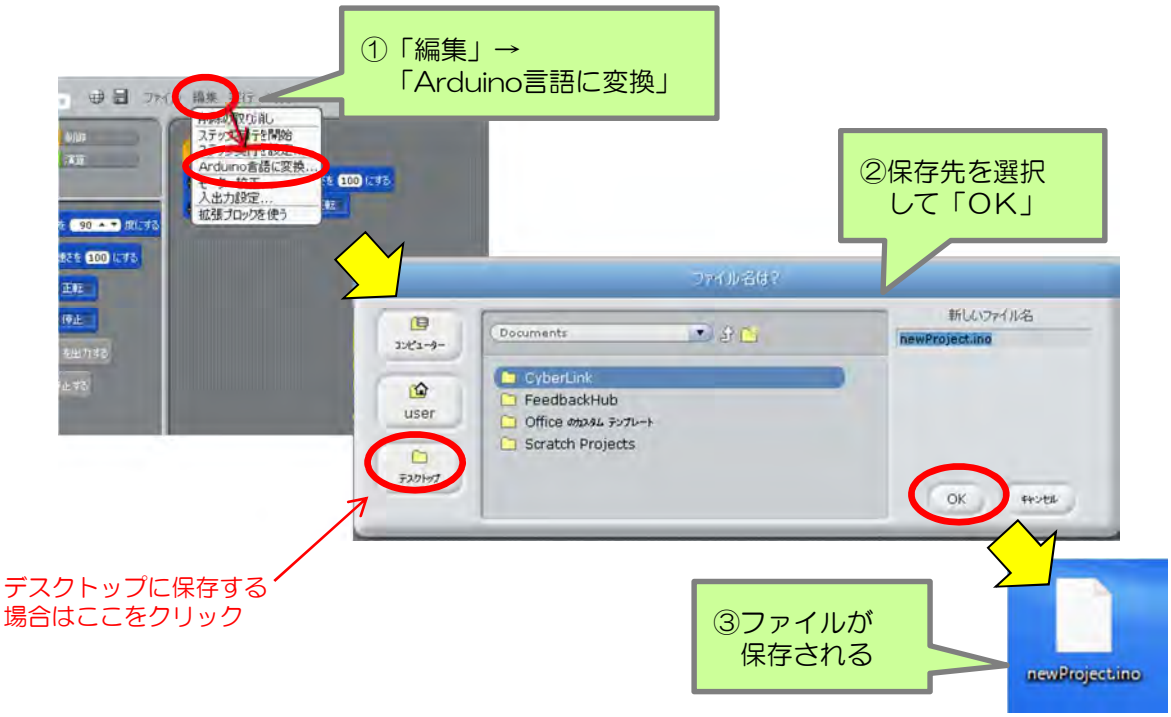


⑧ロボット制御のプログラム（参考）

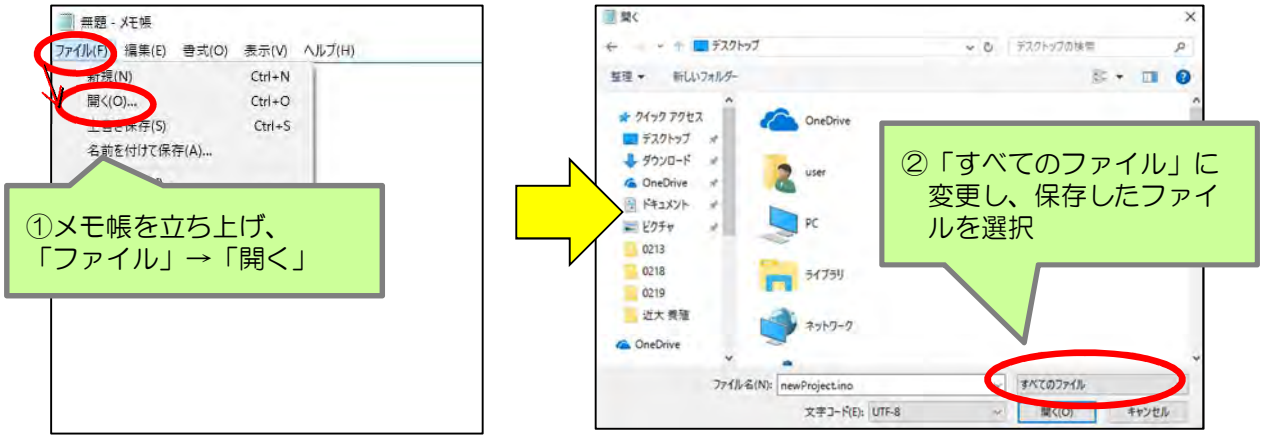
⑧－１　studuino（スタディーノ）をロボットモードで起動

⑧－２　プログラムを作成

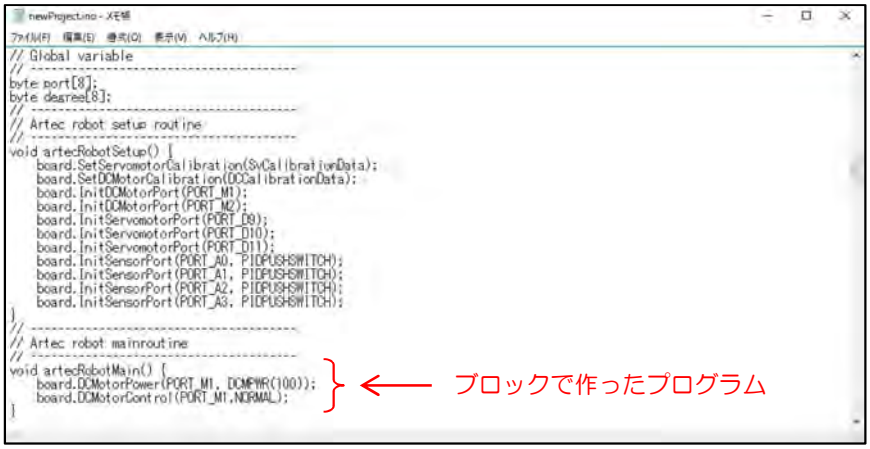
⑧－３　テキスト型言語にして保存



⑧－４　メモ帳で開く



⑧－５　プログラムを見る



県内事例集

●技術分野では、単に何かをつくるという活動ではなく、例えば、技術に関する原理や法則、基礎的な技術の仕組みを理解した上で、生活や社会の中から技術に関わる問題を見いだして課題を設定し、解決方策が最適なものとなるよう設計・計画し、製作・制作を行い、その解決結果や解決過程を評価・改善し、さらにこれらの経験を基に、今後の社会における技術の在り方について考えるといった学習過程を行うことが大切です。

●この事例集では、本県の先進的な事例を６つ集めました。それぞれの学校や地域の実態に応じ、参考にしつつ授業を行ってください。

双方向　（ネットワークを活用した双方向性のあるコンテンツのプログラミングによる問題の解決）

- 事例１：遠隔操作による鳥獣被害対策
- 事例２：遠隔医療（テレビ会議システムを利用した遠隔カンファレンス）

計測・制御　（計測・制御のプログラミングによる問題の解決）

- 事例３：地震・津波観測監視システム　ＤＯＮＥＴ（ドゥーネット）
- 事例４：農業ハウス環境の自動制御システム
- 事例５：航空レーザ測量による森林資源情報の整備
- 事例６：稚魚自動供給システム

事例１：遠隔操作による鳥獣被害対策　双方向

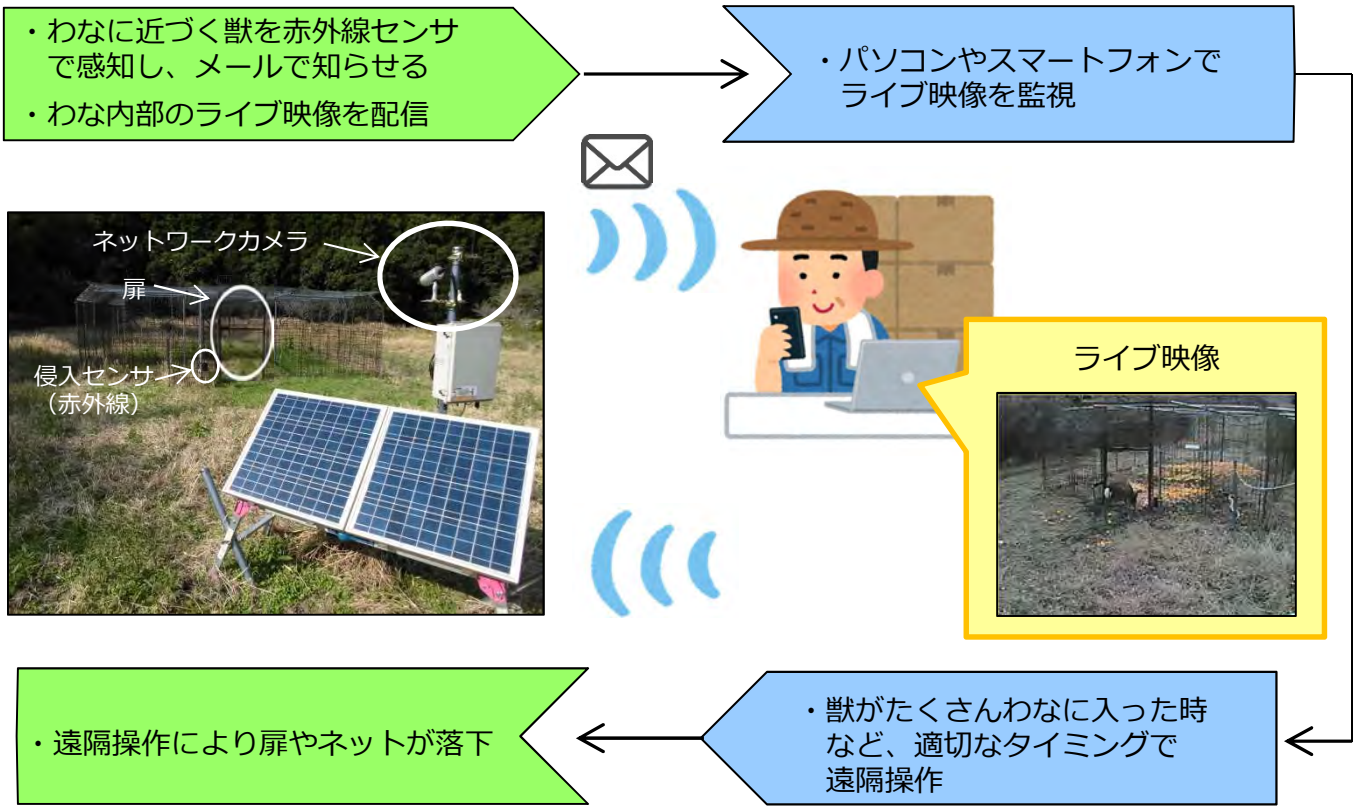
①課題

- 本県では、イノシシ、シカ、サル、アライグマなどにより、年間３億円を超える農作物被害が継続して発生しています。
- 網、わな、銃猟などにより対策を図っていますが、見回り・監視による体力的負担や大量捕獲が困難なことなど、課題があります。

②解決策

- ＩＣＴを活用した大型捕獲機の導入

  センサーやカメラを搭載するとともに、離れた場所にあるパソコンやスマートフォンと通信できる大型檻や罠が導入されています。



③効果

- 獣の映像を遠隔地で見ながら捕獲することで、監視などの体力的負担の軽減や効率的（大量）な捕獲が出来るようになりました。また、エサの置き方などの改善も可能になりました。

事例２：遠隔医療（テレビ会議システムを利用した遠隔カンファレンス）　双方向

①課題

●地方の人口減少などを背景に、医師が都市部に偏在するとともに診療科別の偏在も深刻になるなど、多くの地域で必要な医師の確保ができていません。

●患者からすると、いつでも良質な医療サービスを受けられるという信頼感が揺らいでいます。

②解決策

●遠隔医療体制の構築

・平成26年7月から、県立医科大学附属病院（医大）と県内中核病院において、テレビ会議システムを活用した遠隔カンファレンスが開始されています。

・平成２９年度からは、へき地診療所にも配備し、医大専門医への相談体制を構築することにより、へき地診療所でも診療支援を行っています。

●診断の質の向上に向けた取り組み

・現在、2020年以降の実用化を目指して、次世代通信規格「５G」を活用した遠隔医療の実証実験を行っています。鮮明なエコー動画や患部画像の送受信により、診断の質の向上が期待されています。

③効果

●患者にとって、住み慣れた地域で専門医の助言を受けることができ、遠方の医療機関への通院負担の軽減につながっています。

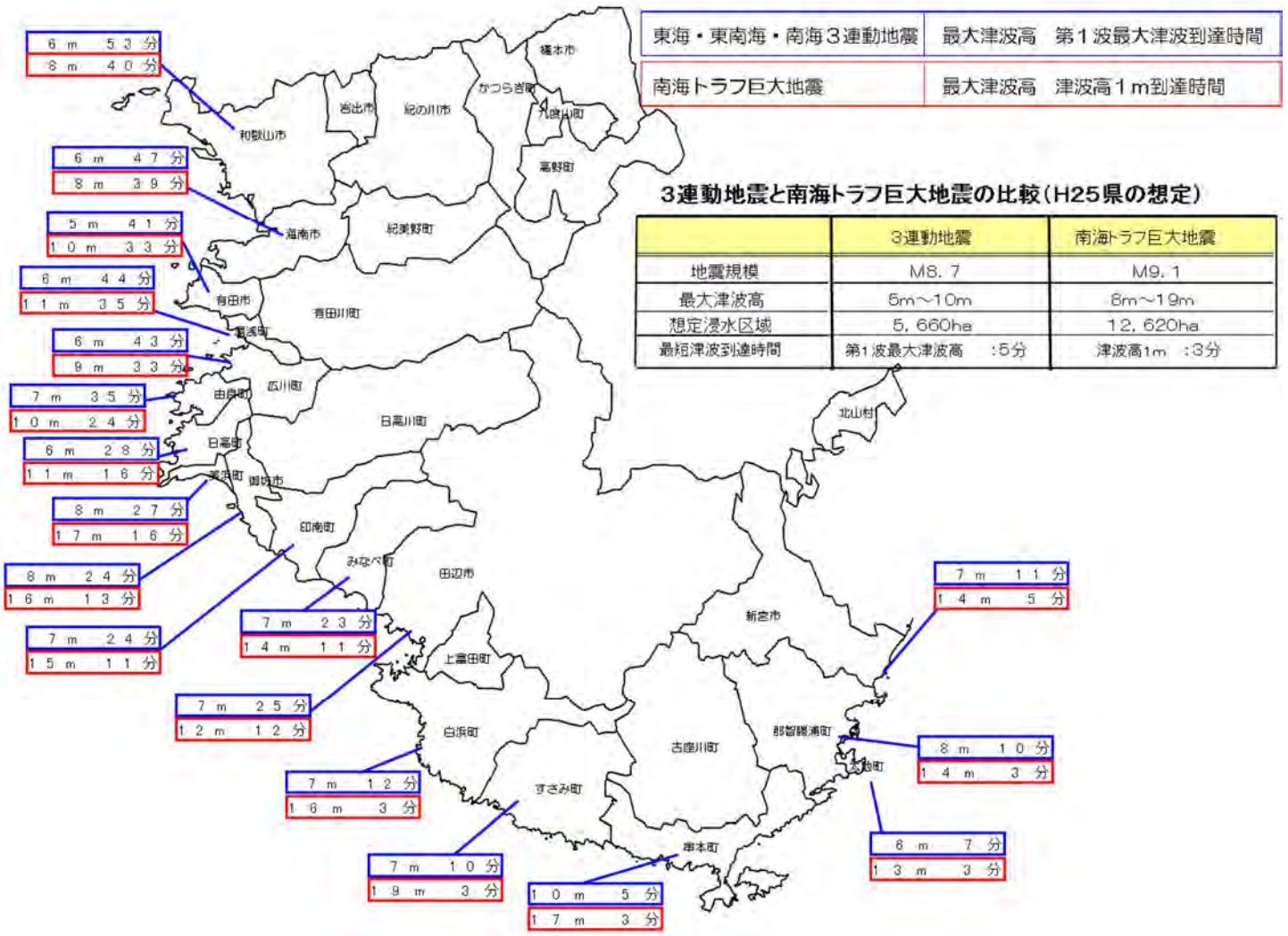
（皮膚科や神経内科（認知症）、栄養指導や言語リハビリテーションなどで活用されています）

事例３：地震・津波観測監視システム　DONET（ドゥーネット）　計測・制御

①課題

- 近い将来、発生が懸念されている南海トラフ地震の震源域は紀伊半島に近く、激しい揺れと短時間で津波が到達することが想定されています。
- 津波到達までの時間が短いことから、津波から逃げ切ることが難しい地域（津波避難困難地域）が存在します。

和歌山県の津波浸水想定（最大津波高・平均浸水深・到達時間）



②解決策

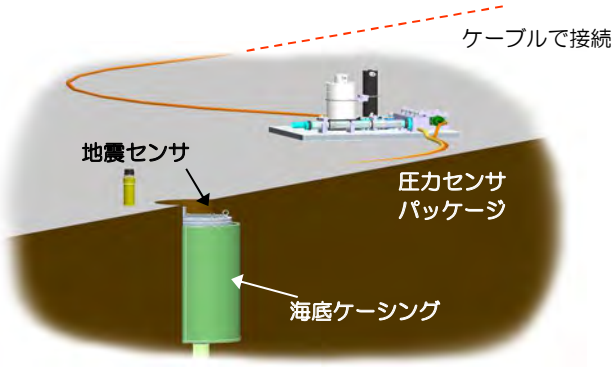
●津波予測システムの開発

・和歌山県では、国立研究開発法人「防災科学技術研究所（ＮＩＥＤ）」所管の「ＤＯＮＥＴ（ドゥーネット）」による地震津波観測網を活用して、津波の規模などを予測できる**津波予測システム**を国立研究開発法人「海洋研究開発機構（ＪＡＭＳＴＥＣ）」と共同開発しました。

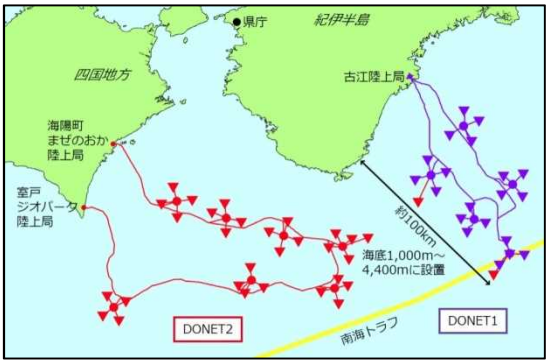
ＤＯＮＥＴ（ドゥーネット）

・紀伊半島沖合の海底に設置された地震・津波観測監視システム

＜ＤＯＮＥＴイメージ（海底）＞



＜ＤＯＮＥＴ観測点＞



＜ＤＯＮＥＴのセンサ＞

〈地震センサ〉
◇強震計：強い地震動を記録
◇広帯域地震計：地震の速い振動から非常にゆっくりとした振動まで広い周波数範囲にわたって地震動を記録

〈圧力センサ〉
◇水圧計等
◇ハイドロフォン：水中の音を記録
◇温度計

津波予測システム

・ＤＯＮＥＴ観測点からの水圧計などのデータをリアルタイムで伝送し、データ解析ソフトを使って予測対象区域の第一波到達予想時刻と最大予想津波高、浸水エリア予測、浸水深予測を即時に表示

③効果

●避難のための情報提供

・和歌山県では津波予測システムを使って、津波の規模や到達予測、浸水予測をいち早く把握し、避難のための情報提供や被害予測を行っています。

県民・県内滞在者

津波からの避難を促すエリアメール・緊急速報メールを配信

市町村・消防本部

予報データを「リアルタイム地震・津波関係情報表示システム」へ表示

※気象庁でも津波到達予報を行っています。これは、地震の観測データから津波の到達時刻などを予測したもので、浸水予測はありません。

事例４：農業ハウス環境の自動制御システム

計測・制御

①課題

●本県では、果樹や野菜、花きなど多くの農産物が生産されています。

●ビニルハウスでも盛んに生産されていますが、ハウス内の環境を適切に保つために、温度計・湿度計のチェックや換気・水やりなどに労力を要したり、人によってハウス内環境にばらつきがあるなど課題があります。

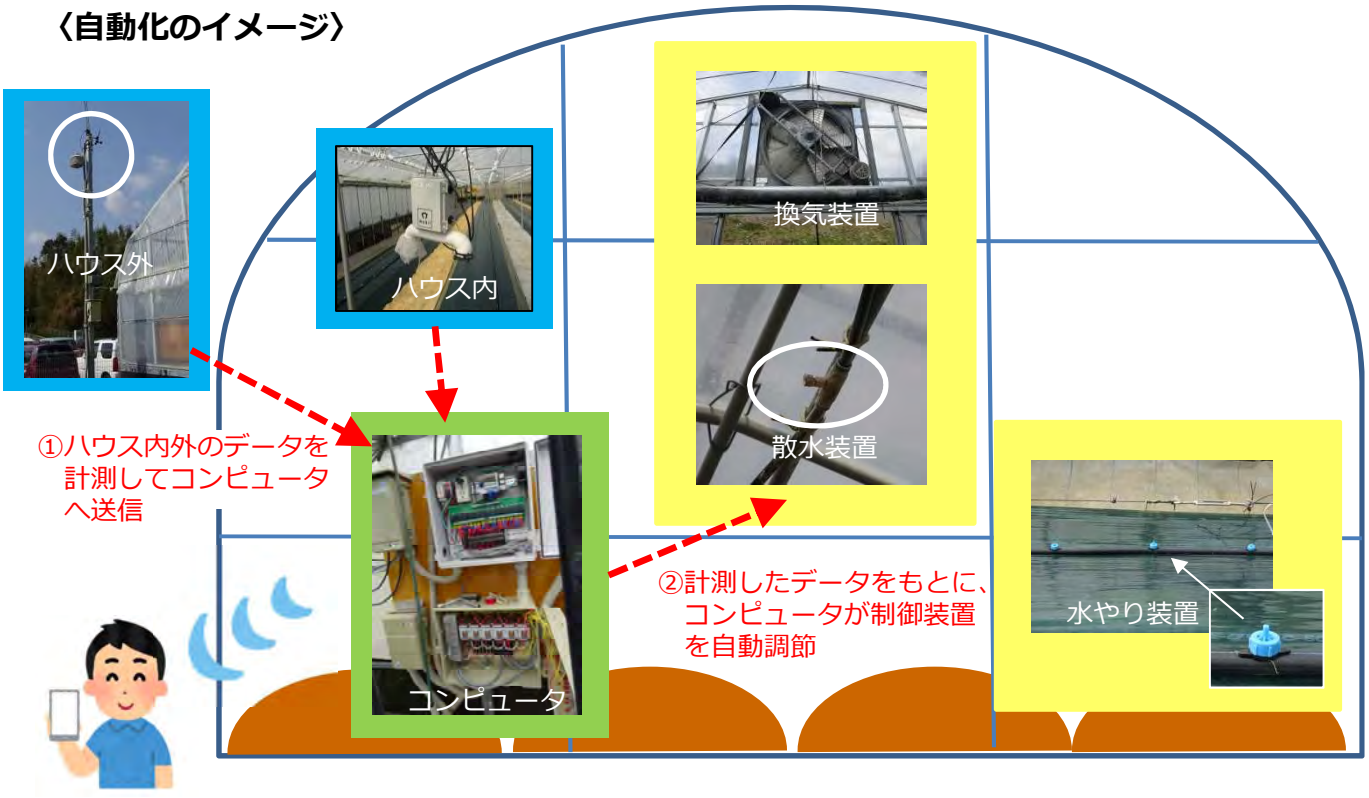
②解決策

●ハウス内環境の自動調節

①ハウス内外のさまざまなデータを計測します。
（ハウス内の温度・湿度、ハウス外の温度・湿度・雨量・日射量など）

②計測したデータをもとに、コンピュータによって水やりや温度調節などが自動で行われ、ハウス内の温度や湿度などが適切に保たれています。

〈自動化のイメージ〉



●自動調節の状況はスマートフォンやパソコンで確認することができます。換気装置や散水装置などを遠隔操作することも可能です。

③効果

●ハウス内環境調節の自動化により、体力的負担が軽減されるとともに、きめ細かな環境調節が可能になり、品質が良くなったり、育成不良が減少しています。

●ハウス内環境の「見える化」により、今後の対策が立てやすくなっています。

事例５：航空レーザ測量による森林資源情報の整備

計測・制御

①課題

●本県では林業が盛んですが、林業に必要な森林の情報は人が直接山に入って調べています。
・生えている木の種類、太さ、高さなど

●人が調べる方法では、調査する人によって精度にバラつきが出てしまいます。また、全ての森林を測ることができないので、サンプル調査になってしまいます。

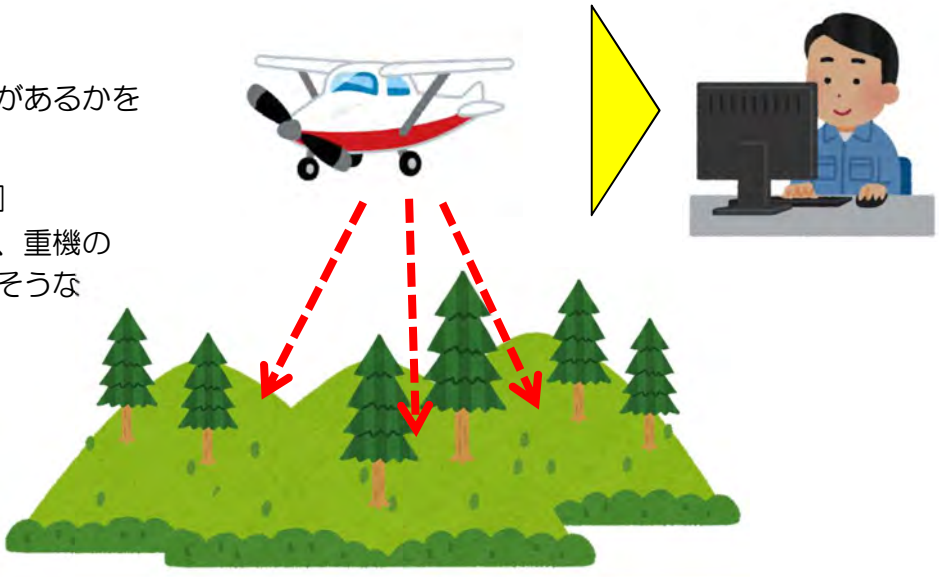
②解決策

●航空レーザ測量※による新たな森林調査手法の導入が進められているところです。
・全ての森林の調査が可能
・精度の高い調査が可能
・土地の細かな形状も同時に計測可能

※航空機に搭載したレーザースキャナから地上に向けてレーザ光を照射し、森林の上層と地上からの反射波との時間差より得られる情報などから、木の頂点の数や高さを割り出したり、地上の標高や地形の形状を求める測量方法

●収集したデータの解析
［効率的な林業］
・どこにどれくらい森林資源があるかを解析して「見える化」

［治山・林道計画の作成支援］
・地形（傾斜等）データから、重機の入りやすさや災害が起こりそうな危険な箇所などを予測



③効果

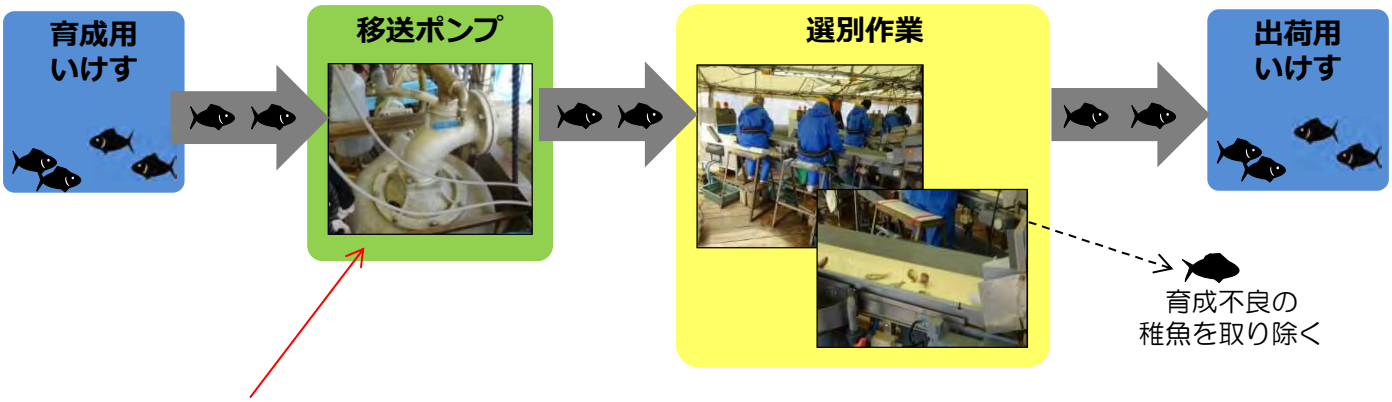
●森林調査をする人の負担が少なくなるとともに、コストも抑えられることが期待されます。
●解析したデータを活用して、効率的な林業を行うことができるようになります。

事例６：稚魚自動供給システム　　計測・制御

①課題

●近畿大学では、養殖稚魚を全国の養殖業者へ出荷する前に、専門作業員による選別作業を行い、育成不良のものを取り除くなど基準を満たす魚だけを選り分けています。

〈稚魚選別の流れ〉



●作業を効率的に行うためには、選別作業員の前を流れる稚魚が多すぎたり少なすぎたりしないように、**<u>ポンプの流量調節</u>**を行う必要があります。

●流量調節は、**<u>流れる魚の量を絶えず目視して手作業で</u>**行うため、経験と集中力が要求され、作業員への体力的負担が大きく、自動化が課題となっていました。

②解決策

●ポンプ制御の自動化

・白浜町にあるマダイのいけすでは、ＡＩやＩｏＴを活用してポンプ制御を自動化した供給作業を行っています。

③効果

●ポンプの流量調節を行っていた作業員への体力的負担がなくなり、選別作業の効率化が図られています。

レゴ® マインドストーム® EV3版

きのくにICT教育
中学校プログラミング教育
学習指導案 補助資料

発行：2019（平成31）年3月

和歌山県教育庁学校教育局義務教育課
〒640-8585
和歌山県和歌山市小松原通一丁目1番地
TEL.073-441-3651/FAX.073-424-8877