

# 情報の技術 スタンダードパッケージ

## 双方向性のある コンテンツ編

教員用

The screenshot displays a digital communication interface with four characters and their messages:

- Abby** (top left): Abbyの発言内容 こんにちは
- Devin** (top right): Devinの発言内容 お腹空いた！  
A speech bubble above him says: メッセージをどうぞ  
Below his message are three icons: a yellow bell, a yellow star, and a red heart.
- Avery** (bottom right): Averyの発言内容 元気？
- Dee** (bottom left): Deeの発言内容 今日はいい天気だね

At the bottom of the interface is a long white input field with a blue checkmark icon on the right.

## 0. 問題を見いだし課題を設定する

中学校学習指導要領（平成29年告示）解説 技術・家庭編(p.53)より

- 情報の技術の見方・考え方を働かせて、問題を見いだし課題を設定し解決する力を育成
- 安全・適切なプログラムの制作、動作の確認及びデバッグ等ができるようにする
- こうした活動を通して、自分なりの新しい考え方や捉え方によって、解決策を構想しようとする態度の育成
- 自らの問題解決とその過程を振り返り、よりよいものとなるよう改善・修正しようとする態度の育成

（中学校学習指導要領（平成29年告示）解説 技術・家庭編：p.19）

生活や社会における事象を、技術との関わりの視点で捉え、社会からの要求、安全性、環境負荷や経済性などに着目して技術を最適化すること。

（中学校学習指導要領（平成29年告示）解説 技術・家庭編：p.48）

生活や社会における事象を、情報の技術との関わりの視点で捉え、社会からの要求、使用時の安全性、システム、経済性、情報の倫理やセキュリティなどに着目し、情報の表現、記録、計算、通信の特性等にも配慮し、情報のデジタル化や処理の自動化、システム化による処理の方法等を最適化すること。

「(1) 生活や社会を支える情報の技術」で気付いた技術の見方・考え方から、問題を見いだし、課題を設定する流れが大切になります。

- ①チャットを動かしてみて不便に感じた点（問題）から改善点（課題）を話し合わせましょう。既存の類似のアプリにある機能を挙げる場合、どうしてその機能がつくられたのか、技術の見方・考え方で意図や理由を考えさせましょう。
- ②生徒が設定した課題は、実際に生活や社会で主に利用されている方法で解決することが必ずしも重要なのではなく、目的に適したプログラミング言語を用いて、生徒自身がアイデアを出し、工夫し創造しながら問題を解決するプロセスを大切にしましょう。
- ③その際、特に情報通信ネットワークの基礎的な仕組みとしての通信方法は、TCP/IP通信やサーバ・クライアントモデルを扱うこととなりますが、実際の生徒が行う実習でそれを扱わなければならないわけではありません。通信の方法は様々あることを捉え、それらの違いやメリット・デメリットなどにも気づかせながら充実した問題解決の活動が促進されるようにしましょう。
- ④本指導書では、情報の技術の見方・考え方「社会からの要求」「使用時の安全性」「セキュリティ」「情報の倫理」から、目次にあるような機能を紹介します。また、それらの機能を制作する中で学習できる要素についても示しています。
- ⑤指導方法として、全員が同じ機能を制作するのではなく、生徒それぞれが見いだした課題を解決するための機能を制作することが考えられ、複数課題設定での学習活動になるので、巻末付録のヒントカードを活用してください。

## 目次

事前準備 .....	3ページ
------------	------

### 導 入

1. 双方向性のあるコンテンツとは .....	10ページ
2. 情報通信ネットワークの構築と仕組み .....	11ページ

### テーマ 1

#### 複数のPCをネットワークでつなぐ

1. ネットワークをつなぐ .....	14ページ
2. メッセージブロックを使ってお互いに信号を送る .....	16ページ
3. 変数をつくってネットワークで共有する .....	18ページ

### テーマ 2

#### チャットアプリのプログラミング

1. アクティビティ図 .....	21ページ
2. アクティビティ図に対応したプログラミング（送り手側のPC） .....	22ページ
3. アクティビティ図に対応したプログラミング（受け手側のPC） .....	24ページ
4. 繰り返しメッセージを送るプログラミングへの改良 .....	26ページ
5. 双方向コミュニケーションの実現 .....	27ページ

### テーマ 3

#### チャットアプリの問題を解決

0. ヒントカードの使い方 .....	31ページ
---------------------	-------

### ● 技術の見方・考え方：社会からの要求

1. 文字だけではニュアンスや感情が伝わりづらい → <b>スタンプ機能</b> .....	32ページ
2. 過去に行われたやりとりがわからない → <b>履歴機能</b> .....	36ページ
3. メッセージを受け取ったことに気づきづらい → <b>通知機能</b> .....	39ページ
4. 時間を気にせず使いすぎてしまう → <b>タイマー機能</b> .....	41ページ
5. 送ったメッセージが読まれたのかわからない → <b>既読機能</b> .....	43ページ
6. 複数人で同時にやりとりができない → <b>グループトーク機能</b> .....	46ページ

### ● 技術の見方・考え方：安全性の観点

[情報の技術の見方・考え方：使用時の安全性、セキュリティ]

1. プライバシーの侵害や成りすましが怖い → <b>ログイン機能（セキュリティの観点）</b> .....	48ページ
---	-------

[情報の技術の見方・考え方：情報の倫理]

2. 不適切な言葉が送られる → <b>NGワード機能（情報モラルの観点）</b> .....	52ページ
---	-------

# 事前準備

## ＜準備＞

- ソフトウェアの起動方法
- ソフトウェアの画面構成
- ソフトウェアの基本操作



## 事前準備

### ①準備

#### 1. ソフトウェアのインストール

本テキストでは「Studuino:bitソフトウェア」を使用してコンテンツの制作を行います。  
各PCに事前にStuduino:bitソフトウェアをインストールしてください。

Studuino:bitソフトウェアのインストール手順については、弊社アーテックロボ2.0ウェブサイト最新のマニュアルを掲載しておりますので、以下の手順をご参照ください。

#### ●ソフトウェアダウンロードサイト

<https://www.artec-kk.co.jp/artecrobo2/ja/software/>

- ・「ソフトウェアダウンロード」をクリックすると「Studuino:bitソフトウェア」がダウンロードされます。  
※本テキストはWindows版を元として作成しておりますが、Macintoshにも対応しています。  
※Android、iOSには対応していません。
- ・「セットアップマニュアル」をクリックすると、マニュアル（PDF形式）が開きます。
- ・Windows7/8.1をご使用の場合は、「USBデバイスドライバダウンロード」をクリックしてドライバをインストールしてください。

#### 2. ネットワークの接続

本テキストでは複数のPCをStuduino:bitのWi-Fiで接続した環境での授業を想定しています。  
接続するPCの台数分、Studuino:bitおよびUSBケーブルを準備してください。

問い合わせ窓口

**株式会社アーテック**

TEL 072-990-5656

e-mail [info@artec-kk.co.jp](mailto:info@artec-kk.co.jp)

## ②ソフトウェアの起動方法

この授業では、文字の代わりにブロックのような絵をつないでコンピュータへの指令をプログラミングできる「ビジュアルプログラミング言語」を使います。

### ① ソフトウェアの立ち上げ

キャラクターモードを選択してください。



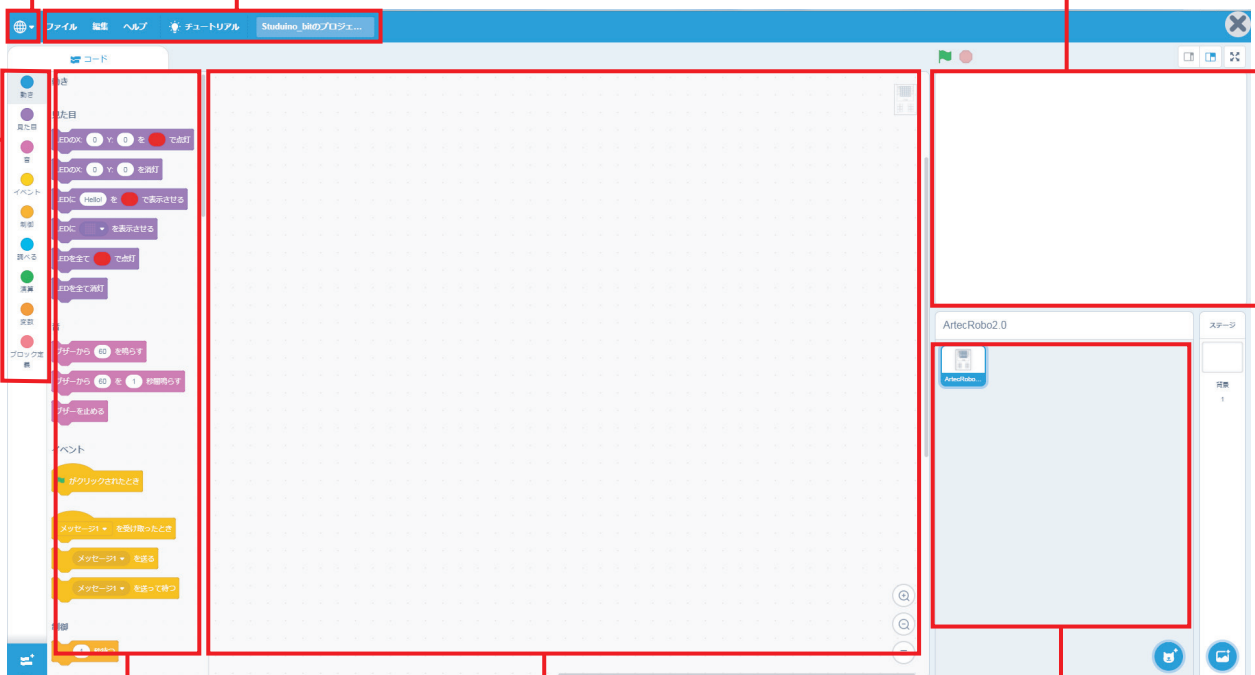
カテゴリ：命令の種類を選ぶことができます

言語の選択

メニュー

ステージ：

プログラムによってスプライトが動くエリアです



スクリプトエリア：

命令をつないでプログラムをつくることができます

[スクリプトとは？・・・7ページ参照](#)

ブロックパレット：

センサーやアクチュエータへの命令が表示されます

スプライトリスト：

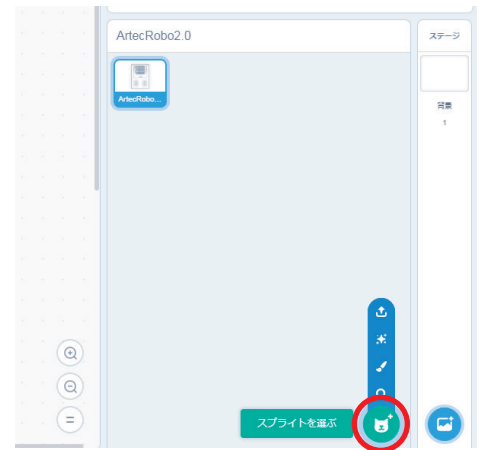
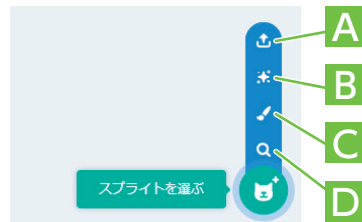
プログラム内で使用するキャラクターが表示されます

[スプライトとは？・・・6ページ参照](#)

### ③ソフトウェアの基本操作

#### ① スプライトの追加

起動時には、「ArtecRobo2.0」スプライトのみ表示されています。スプライトの追加をする場合は、スプライトエリア右下部の「スプライトを選ぶ」からスプライトを追加します。



※「ArtecRobo2.0」は、ArtecRobo2.0を制御するためのスプライトです。

#### A スプライトをアップロード

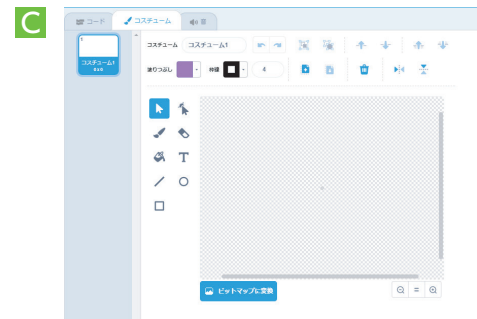
ファイル参照画面が開きます。  
ファイル保存先を参照してファイルを選択してください。

#### B サプライズ

ソフトウェアに標準で準備されている画像の中からランダムにスプライトが追加されます。

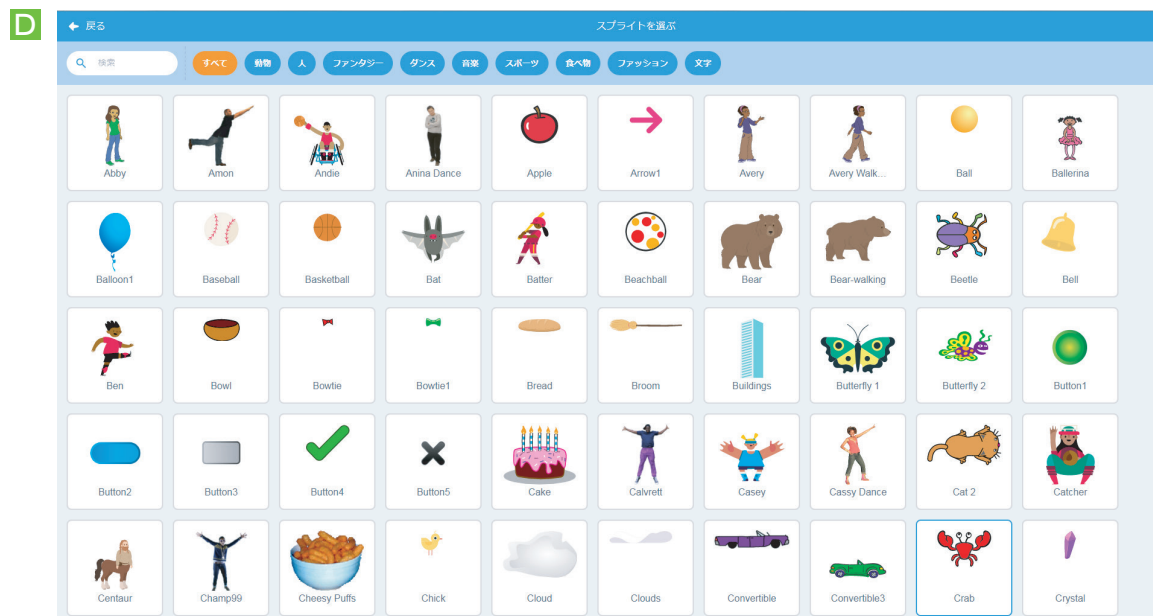
#### C 描く

ペイントエディターが開きます。



#### D スプライトを選ぶ

ソフトウェアに標準で準備されている画像の中からスプライトを選択します。



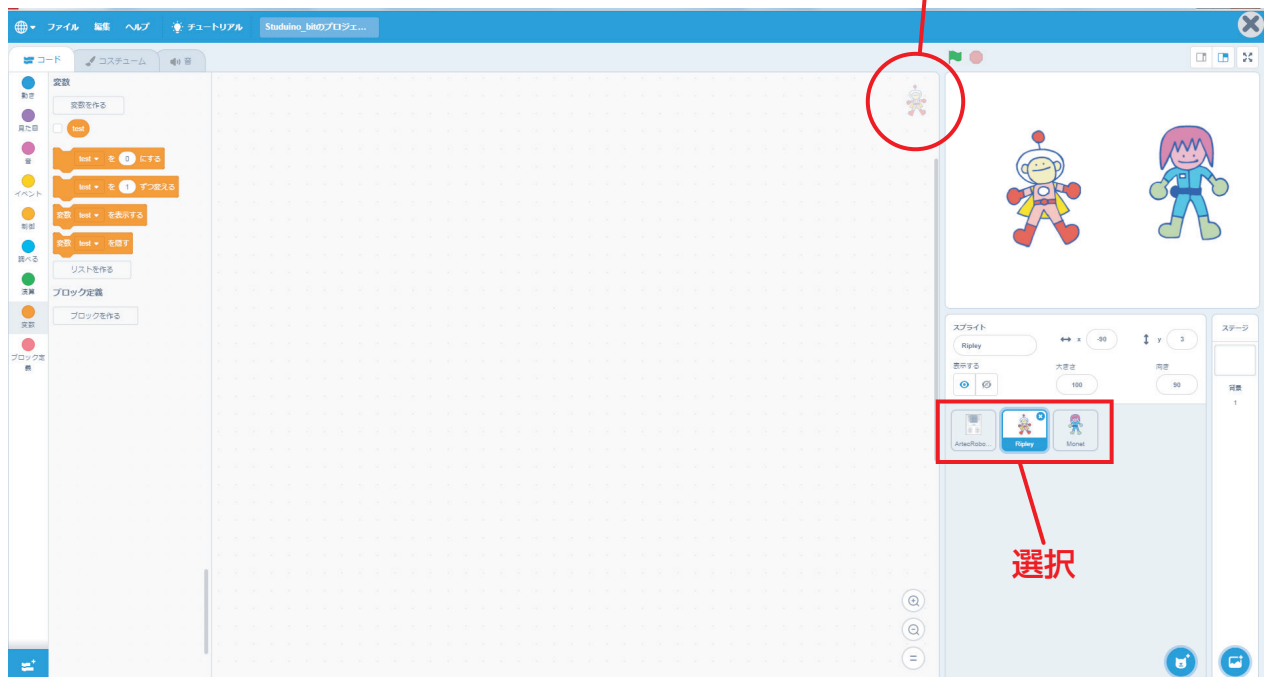
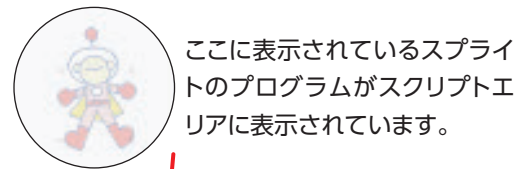
スプライトとは・・・。

コンピュータの画面上で画像や図形を動かす表現を実現する仕組みのひとつで、各画像を背景と別に作成し、それらを合成する仕組みです。この仕組みを使って表示される各画像のこともスプライトとよびます。スプライトは主に、コンピュータゲームなどで用いられている手法です。

## ② プログラムの作成方法

### ■スプライトの切り替え

プログラムはスプライトごとに作成します。  
スプライトエリアの各スプライトをクリックすることで、それぞれの  
プログラム作成画面に切り替わります。

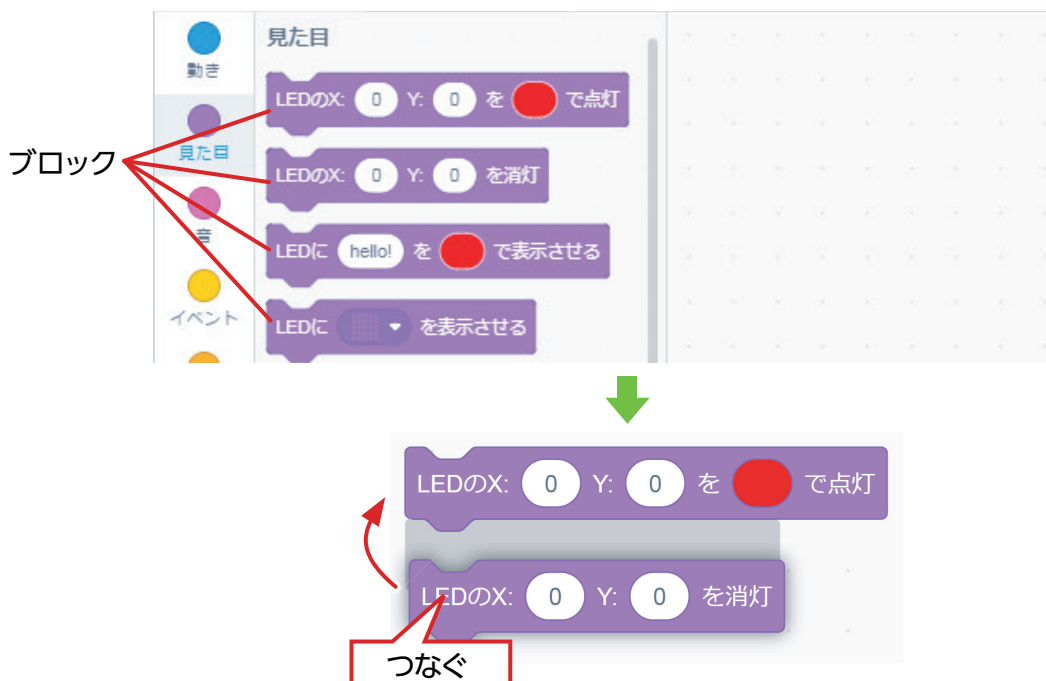


スクリプトとは・・・。

簡易的なプログラムとして用いられる用語で、スクリプトを作成するための言語を「スクリプト言語」と呼びます。スクリプト言語は機械語への翻訳を必要とせずにプログラムを実行することができ、特定のアプリケーションの動作など簡単なプログラムに用いられています。用途を限定しているため、構造自体も本格的なプログラミング言語よりも簡単に作成できます。

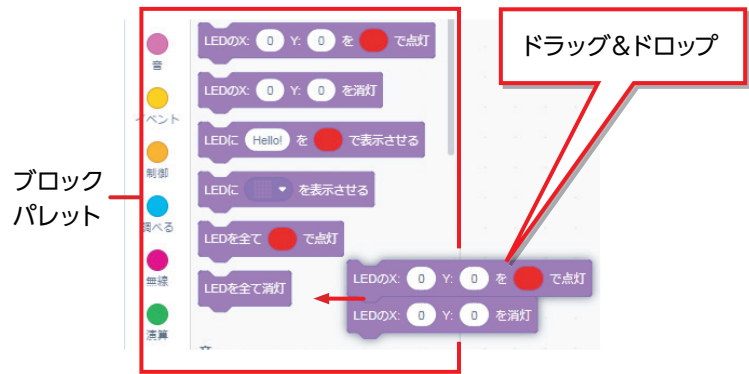
### ◆ プログラムの作成

ブロックパレットにある命令をおもちゃのブロックのようにつなぐことでプログラムをつくります。この命令のひとつひとつを「**ブロック**」と呼びます。



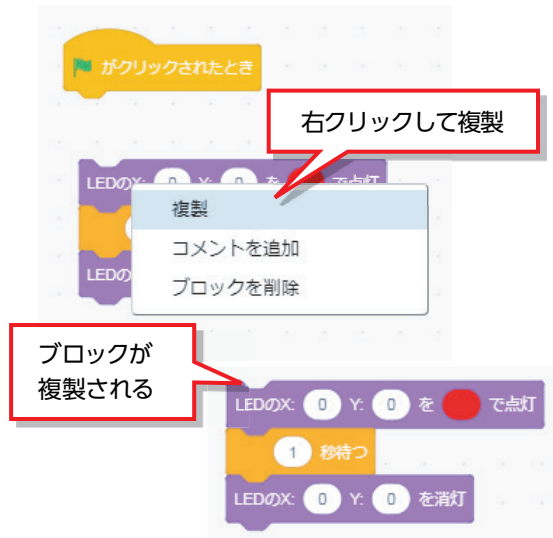
### ◆ ブロックの削除

削除するブロックをブロックパレットにドラッグ&ドロップします。



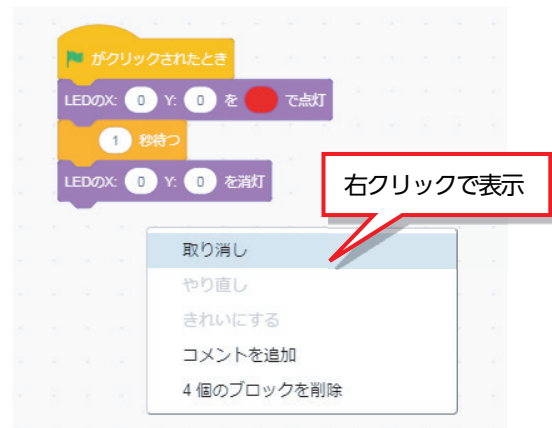
### ◆ ブロックの複製

つながっているブロックを複製できます。つなげたブロックの一部分だけ複製したい場合は、その部分を抜き出して同じ操作を行います。



### ◆ 取り消し

スクリプトエリアで右クリックをして表示されるメニューから取り消しを押すことで、一つ前の状態に戻すことができます。



### ◆ プログラムの保存方法

①「ファイル」から「コンピュータに保存する」を選択します。



# 導入

- 1.双方向性のあるコンテンツとは
- 2.情報通信ネットワークの構成と仕組み

## 1. 双方向性のあるコンテンツとは

### コンテンツとは…

英語で「内容」を意味する言葉で、情報の世界ではデジタルで取り扱うことができる情報の内容のことをさし、「文字、図形、色彩、音声、動作もしくは映像もしくはこれらを組み合わせたもの」があげられます。

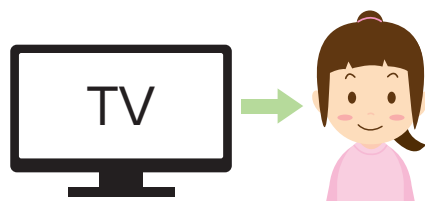
### 双方向性のあるコンテンツ

テレビ番組のようにメディアから視聴者へ向けて一方向から発信されるコンテンツに対して、テレビゲームのように操作（入力される情報）によってその場その場で発信されるコンテンツが変化するようなものを双方向性のあるコンテンツといいます。

近年ではインターネットの登場により、情報の発信者は一度に多くの人に情報発信するということが可能となったのと同時に、ネットワークを通じて情報の受け手から反応を確認するといった、発信者と受信者の間における「双方向性」のある情報のやり取りが容易にできるようになりました。

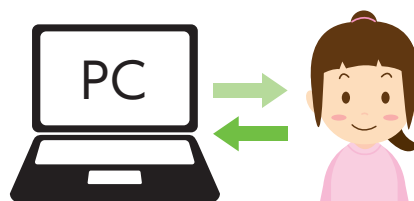
例えば、デジタル番組での視聴者参加型コンテンツのように閲覧者からの入力情報に応じて必要な情報を配信するなど双方向で必要な接点を持つことができることがネットワークを利用した双方向性のあるコンテンツの大きな特長です。

#### 従来の新聞、テレビなどのメディア



一方通行の情報

#### インターネットなどの双方向メディア



受け取るだけではなく、発信もされる情報

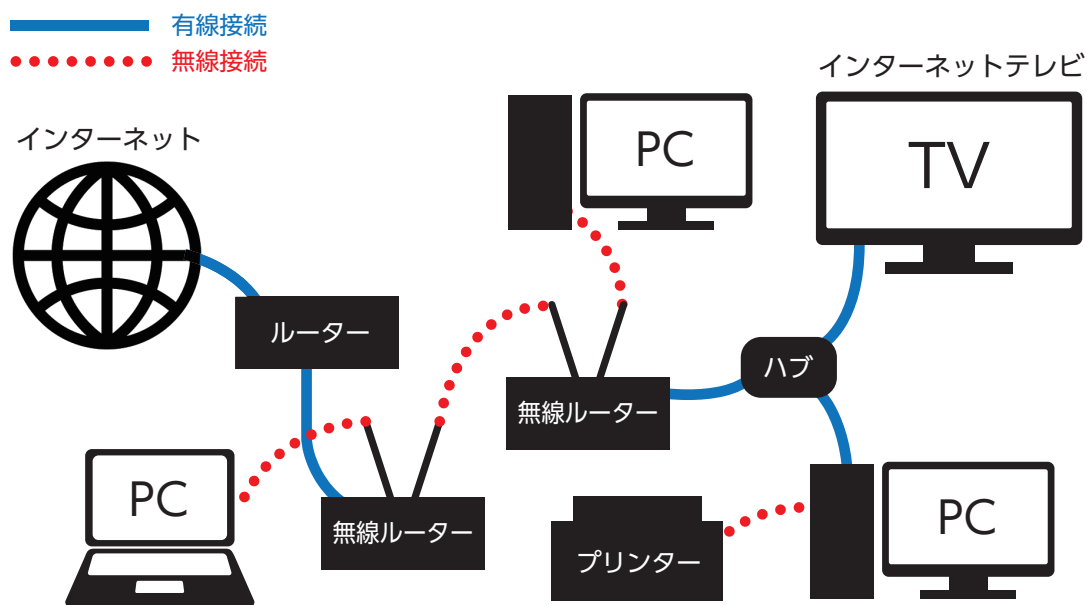
### ●一方向のコンテンツと双方向のコンテンツの例を考えよう

一方向のコンテンツ	双方向のコンテンツ
(例) 従来のテレビ番組	デジタル番組（dボタン）
映画	TVゲーム
新聞	インターネット掲示板・SNS

## 2. 情報通信ネットワークの構成と仕組み

### ネットワークとは・・・

複数のコンピュータやプリンタなどの機器を接続し、お互いに通信できるようにした状態のこと。  
接続する方法にはケーブルを使用して接続する**有線接続**と電波などを利用した**無線接続**があります。



●身の回りにあるネットワークに接続されている機器にはどのようなものがあるか調べてみましょう

有線接続	無線接続
(例) プリンター・デスクトップPC・テレビなど	(例) 携帯電話・スマートフォン・ポータブルゲーム
	Bluetoothスピーカーなどスマート家電

### 情報通信ネットワーク

テレビや電話、パソコンなど様々な機器が接続され、情報を相互にやりとりできるネットワークのことを情報通信ネットワークといい、近年ではエアコンなどの家庭用電気製品も情報通信ネットワークに接続できるようになってきています。



## LANとWAN

LAN(ラン)とは、Local Area Network (ローカルエリアネットワーク)の頭文字をとったもので、家庭や学校などの限定された範囲を接続したネットワークのことです。

WAN(ワン)とは、Wide Area Network (ワイドエリアネットワーク)の頭文字をとったもので、電話回線または専用回線などを使い、LAN 同士で構成されたより広い範囲のネットワークのことです。

## インターネット

インターネットも WAN(ワン) のひとつで、世界中のコンピュータ同士がクモの巣のようにつながった大きなネットワークになります。

## IPアドレス

手紙を送るときに相手の住所がわからないと正しい送り先に手紙を届けられないのと同様に、コンピュータ等の情報機器がネットワーク上でデータをやりとりするときも住所に相当する識別番号が必要になります。

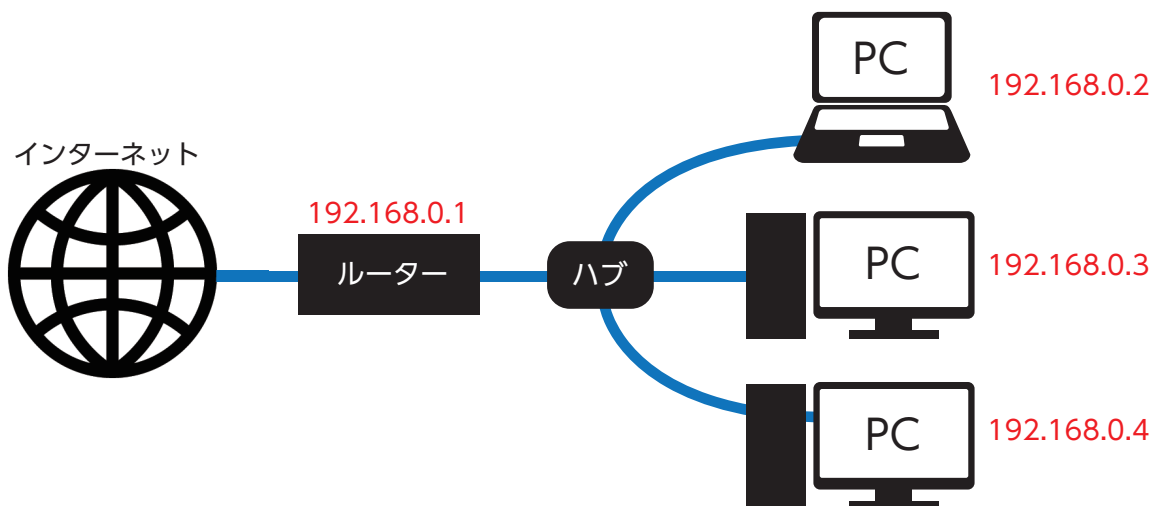
インターネットに接続されている全ての情報機器にIPアドレスという識別番号が割り当てられています。

IPアドレスはインターネットに接続された他の機器と重複しないように割り当てられます。

現在広く使われてるIPv4 という方式では 0 ～ 255 の数字4組を「.」で区切った数値として表現されます。

(近年IPv4で表現できるIPアドレスでは足りなくなっており、IPv6方式へ移行が進んでいます。)

同一LAN内でも下図のように「192.168.・・・」というように同一LAN内で接続された情報機器が重複しないようにIPアドレスが割り当てられます。



# テーマ 1

## 複数の PC を ネットワークで つなぐ

1. ネットワークをつなぐ
2. メッセージブロックをつかって  
お互いに信号を送る
3. 変数をつくってネットワークで  
共有する

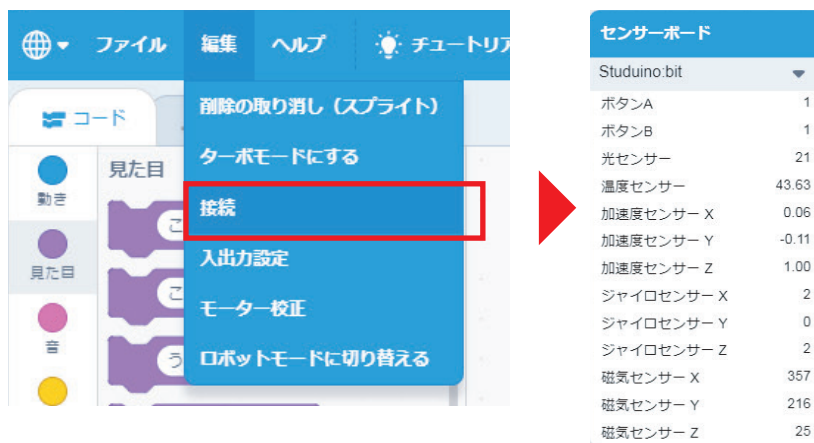
## ①ネットワークをつなぐ

複数のコンピュータをネットワークでつないでみましょう。

- ① コンピュータとメインユニット (Studuino:bit) をUSBケーブルで接続します。



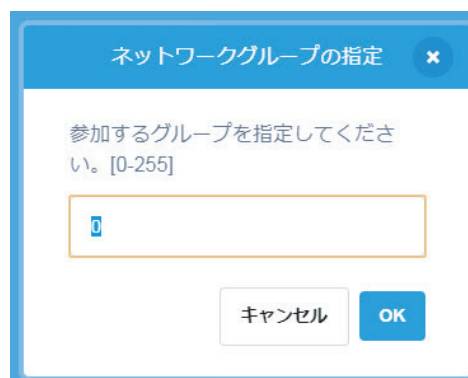
- ② 「編集」より「接続」を選択します。右図のようなセンサーボードが表示されるとコンピュータとメインユニット間の通信が正常に行われています。



- ③ 「編集」より「ネットワークに参加する」を選択します。



- ④ 下図のような画面が表示されたら、0～255の中で任意の値を入力してください。

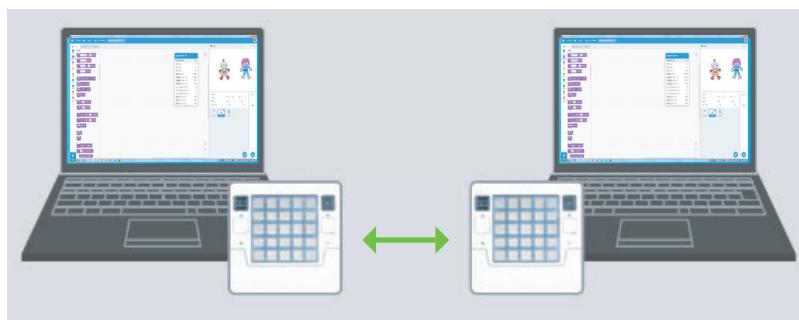


ネットワークグループの指定が完了すると、通信ランプが青に点灯します。



- ①～④の作業を各PCおよびメインユニットで行ってください。

- ④のネットワークグループが同じPC同士がWi-Fiによりネットワークでつながります。



Wi-Fi通信

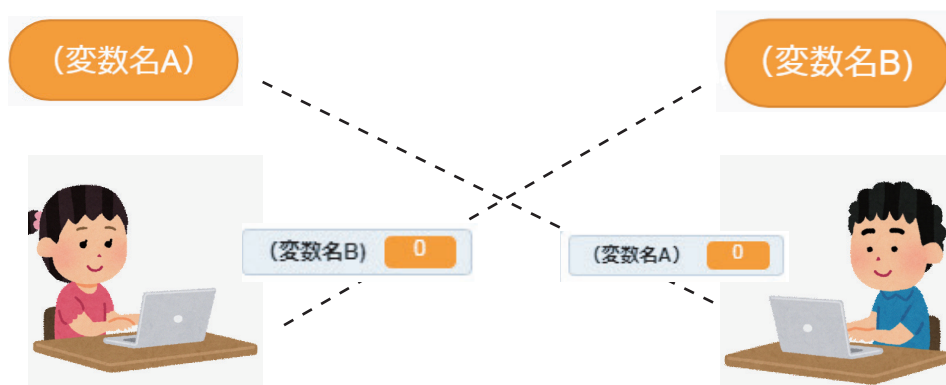
※ 3台以上のPCでもネットワークグループが同じ場合、ネットワークでつながります。

Studuino:bitソフトウェアでは、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。

①互いに信号を送ることができます。




②変数を共有できます。



### 参考 変数ってなに？

変数は数字を入れて保存しておくことができる箱と考えることができます。  
箱に名前を付けておくことで色々なところで中身を取り出して確認することができます。

 箱に7を入れると...

$$\begin{array}{ll} 5 + \boxed{7} = 12 & 2 + \boxed{7} = 9 \\ 3 + \boxed{7} \times 2 = 17 & 4 \times \boxed{7} = 28 \end{array}$$

 箱の数字を8に変えると...

$$\begin{array}{ll} 5 + \boxed{8} = 13 & 2 + \boxed{8} = 10 \\ 3 + \boxed{8} \times 2 = 19 & 4 \times \boxed{8} = 32 \end{array}$$

箱の数字を変えるだけで、たくさんの式を一度に変更できます。

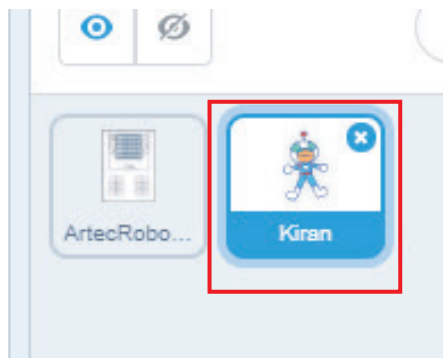
## ②メッセージブロックをつかってお互いに信号を送る

ネットワークでつながれた複数のコンピュータでお互いに信号を送る方法を確認しましょう。

①一方のPC(送信側)でイベントカテゴリから **メッセージ1 ▼ を送る** をスクリプトエリアにならべましょう。



②ネットワークでつながれた別のPC(受信側)で以下のようなプログラムを作成しましょう。

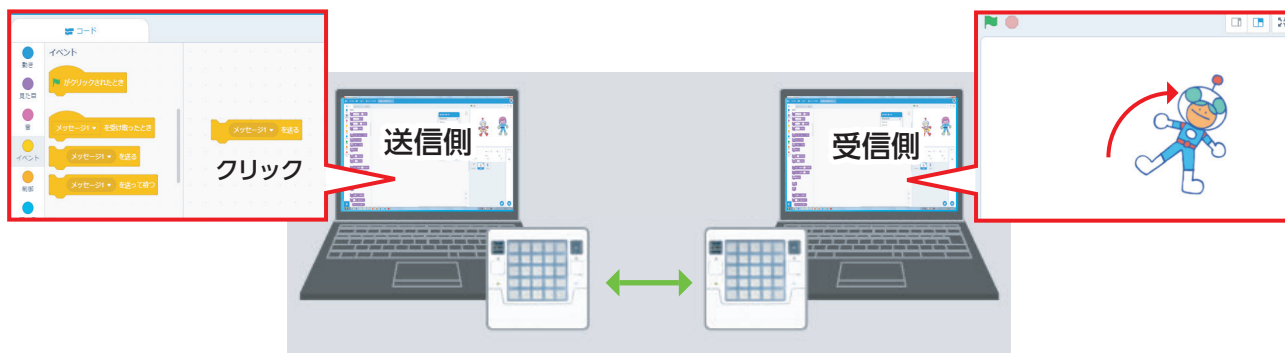


プログラムは追加したスプライトで作成してください。

メッセージ1の名前は任意に設定できますが、**必ず受信側と送信側のブロックの名前は同じにしてください。**



③送信側PCの **メッセージ1 ▼ を送る** をクリックするとネットワーク経由でメッセージ(信号)が送られ、受信側PCの **メッセージ1 ▼ を受け取ったとき** 以下につながれたプログラムが実行されます。  
ここでは受信側PCのスプライトが信号を受け取る度に15度ずつ回転します。



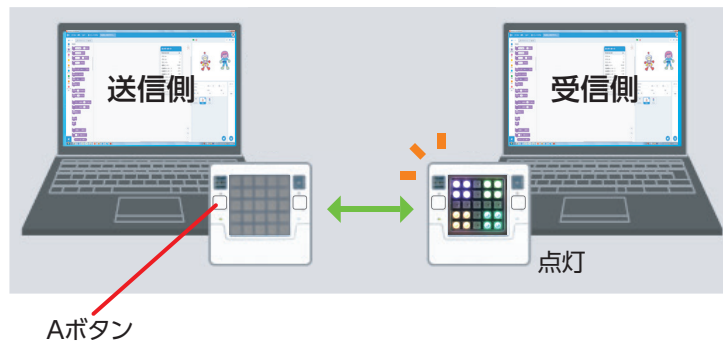
Wi-Fi通信

## ネットワークをつかった遠隔制御システムのプログラム例

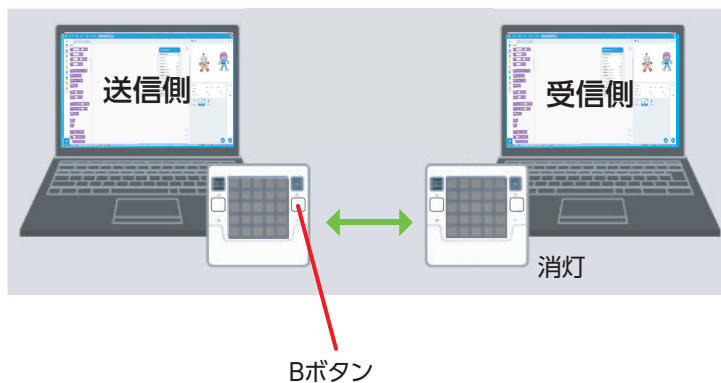
受信側PCでArtecRobo2.0のスプライトを使い遠隔でLEDを点灯・消灯させる遠隔制御システムを作成することも可能です。



送信側でAボタンを押す → 「点灯」の信号を送信 → 受信側でLEDを点灯



送信側でBボタンを押す → 「消灯」の信号を送信 → 受信側でLEDを消灯



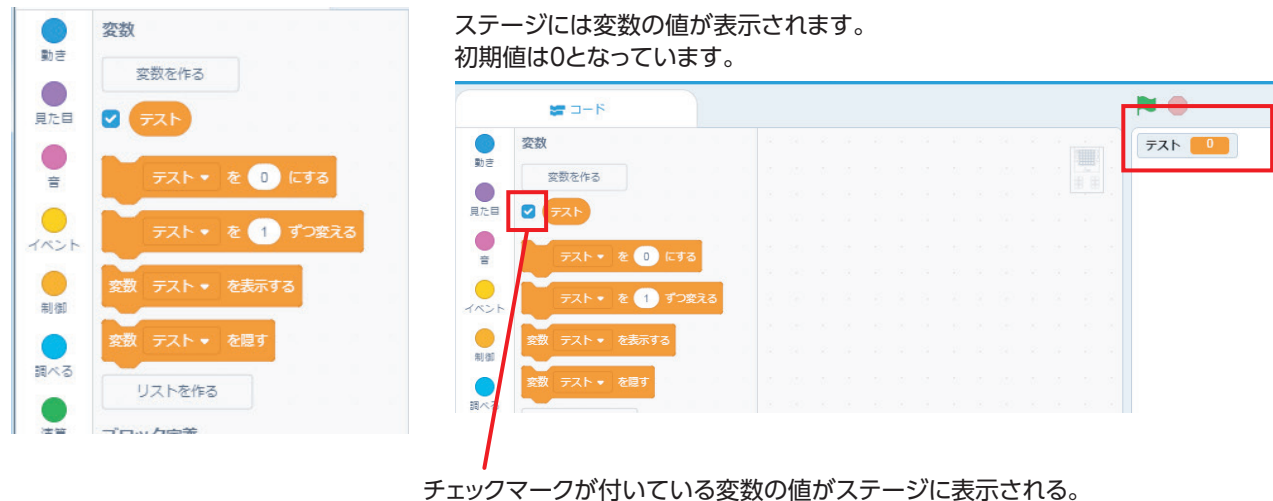
### ③変数をつかってネットワークで共有する

ネットワークでつながれた複数のコンピュータで変数を共有する方法を確認しましょう。

①変数カテゴリから「変数をつくる」をクリックして表示されたウインドウで新しい変数名を指定して作成します。



②作成した名前の変数が使用できるようになります。(ここでは「テスト」という変数名を用いています。)

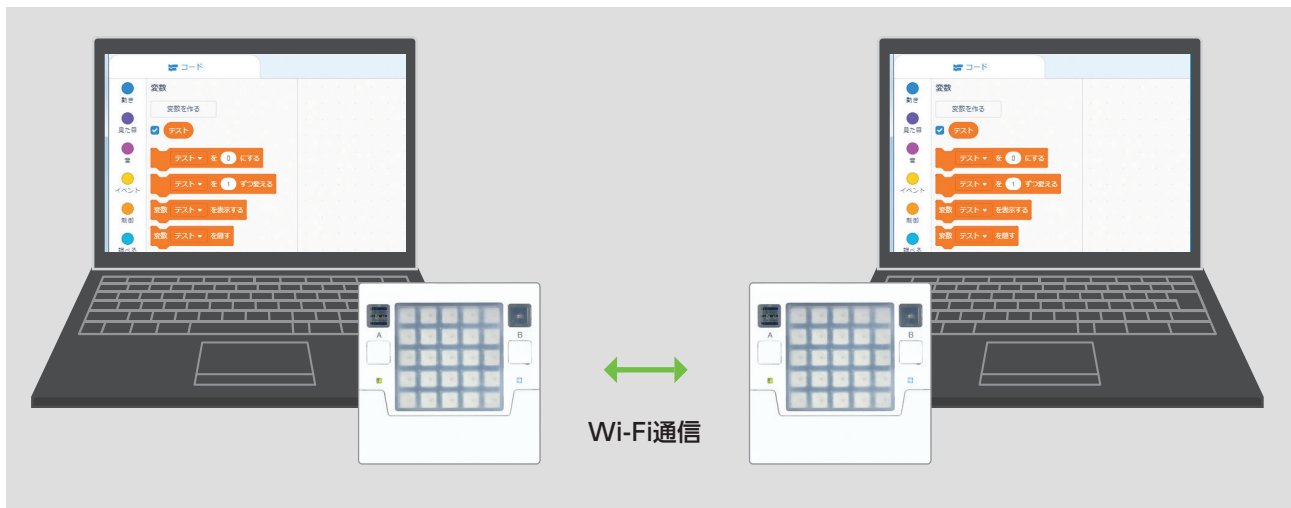


③下図のブロックに任意の値(文字列でもよい)を入力してクリックすると変数の値が変化することを確認してください。





④ ①～②と同じ方法でネットワークをつなげた別のPCでも同じ名前の変数を作成してください。

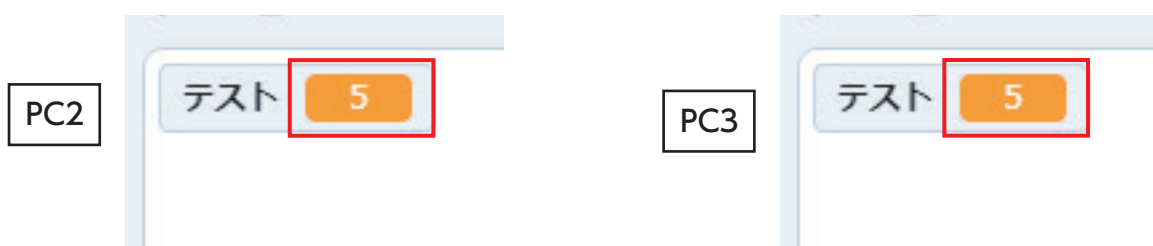


⑤ 一方のPCの変数の値を変化させると、ネットワークでつながれた他のPCの同じ名前の変数も変化することを確認してください。

ひとつのPCの変数の値を変えると…。



ネットワークでつながった他のPCの同じ名前の変数の値が変わる。



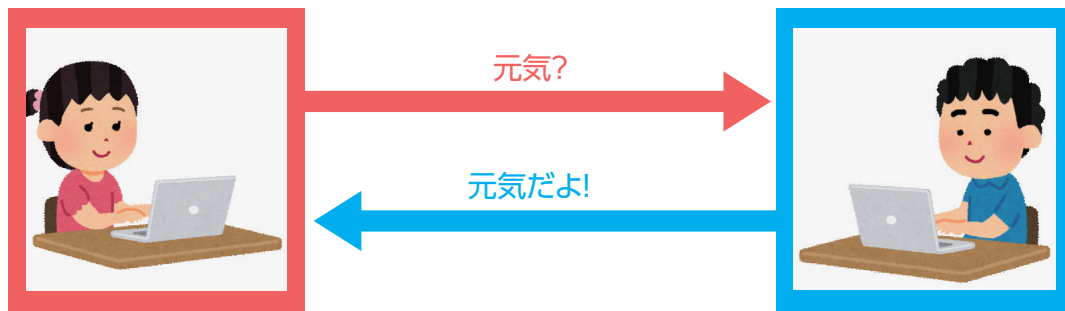


# テーマ 2

## チャットアプリの プログラミング

1. アクティビティ図
2. アクティビティ図に対応した  
プログラミング (送り手側のPC)
3. アクティビティ図に対応した  
プログラミング (受け手側のPC)
4. 繰り返しメッセージを送る  
プログラムへの改良
5. 双方向コミュニケーション  
の実現

ネットワークを利用して複数のPC間でメッセージのやり取りを行えるチャットアプリのプログラミングを学習しましょう。



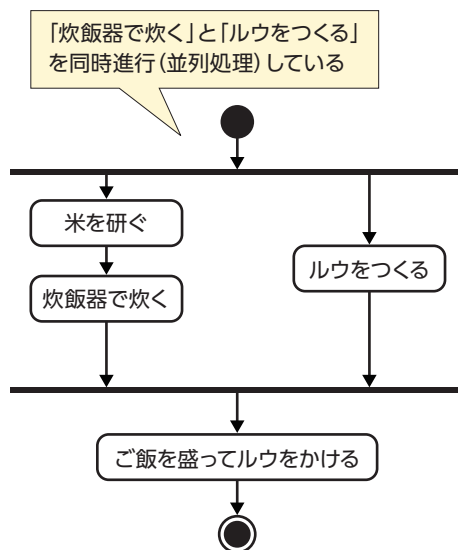
## ①アクティビティ図

プログラムを作成する基になるアクティビティ図を作成しメッセージの送信手順を分解しましょう。

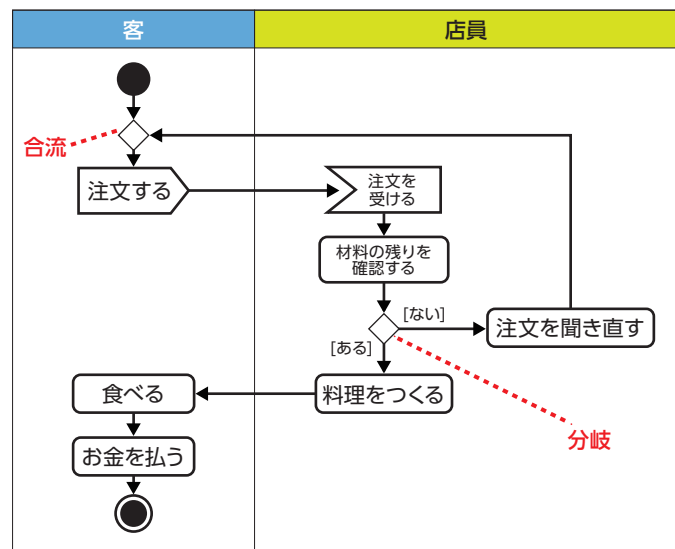
### アクティビティ図とは……

「参考事例1のような並列処理や、参考事例2のようなレーンをつくって役割間での処理を書くことができます。

#### 参考事例 1. カレーライスをつくる

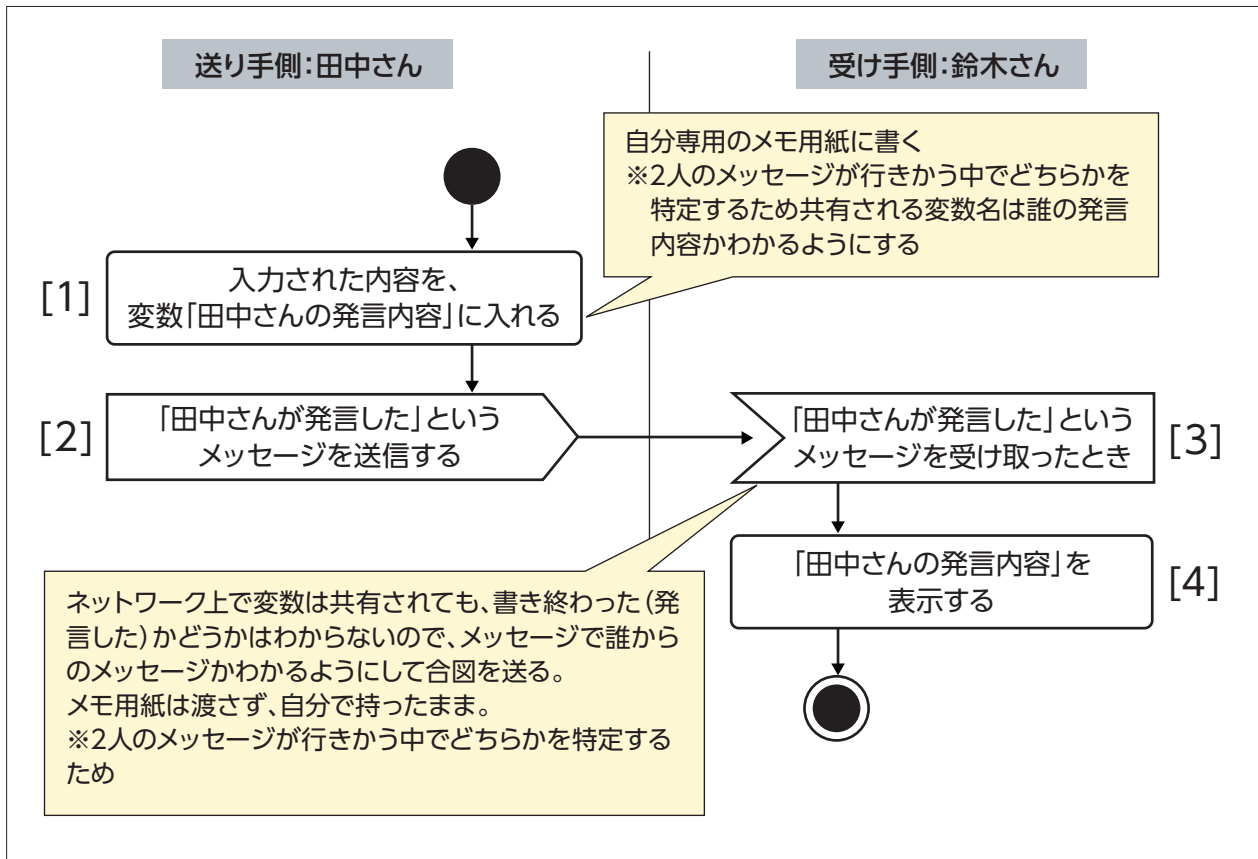


#### 参考事例 2. レストランでの食事



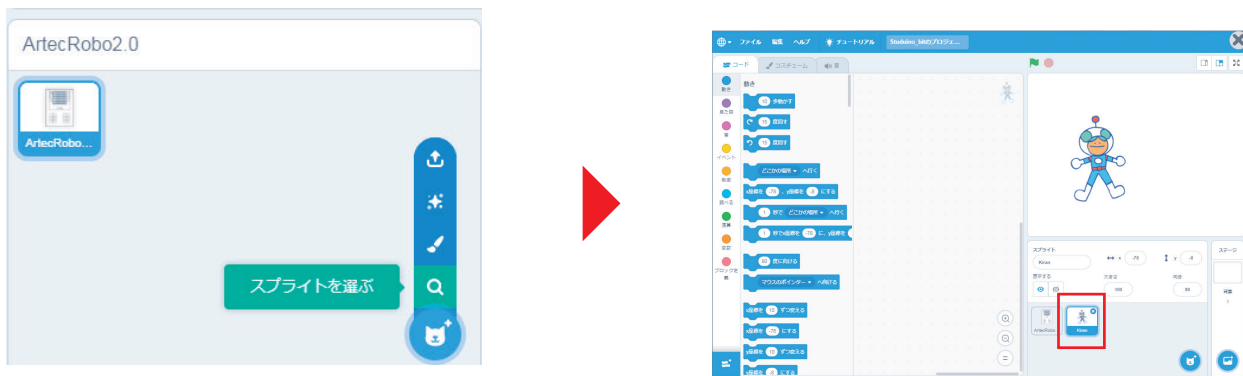
#### 凡例

●	: 処理の開始	制御名	: 出力に関する処理
●	: 処理の正常終了		
⊗	: データを保持しない終了	制御名	: 入力に関する処理
◇	: 分岐 (Decision node)、交流 (Merge node)		
——	: 並列、非同期の処理開始 (Fork node) と終了 (Join node)		



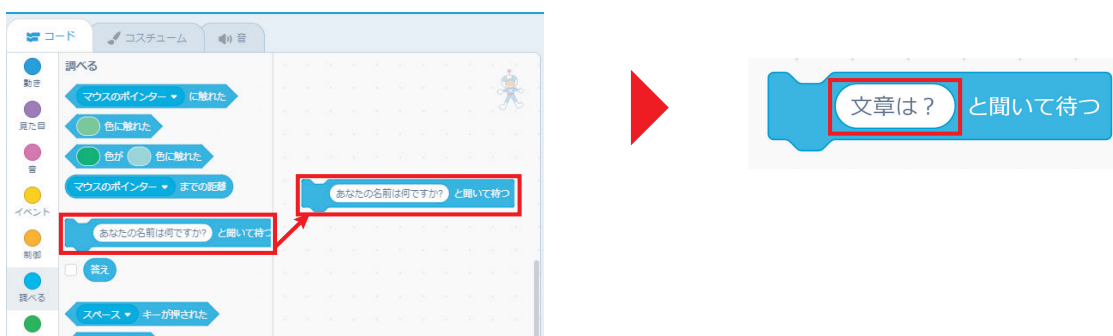
## ②アクティビティ図に対応したプログラミング (送り手側のPC)

①送り手側のプログラムを作成するために、好きなスプライトを選びます。



※ArtecRobo2.0スプライトでは作成できません。

②調べるカテゴリから 「あなたの名前は何ですか? と聞いて待つ」 をスクリプトエリアにドラッグし、「文章は?」と入力します。



- ③発言内容を入れるための変数を用意します。ここでは、発言する人の名前を付けて「田中さんの発言内容」という名前にします。  
※「すべてのスプライト用」として変数を作成してください。
- ④変数「メモ用紙」を「文章は？」と聞いて待つ の下につなげ、調べるカテゴリから「答え」をドラッグして、変数の中に挿入します。



入力欄

文章は？ と聞いて待つ

このブロックを使うと入力欄が表示され、ステージ上で好きな文章を入力できるようになります。

入力した文章は 答え に格納されます。

- ⑤動作を確認してみましょう。並べたブロックをクリックして実行すると、スプライトから「文章は？」と吹き出しが出ます。入力欄に入力した文章が「田中さんの発言内容」変数の値として格納されます。



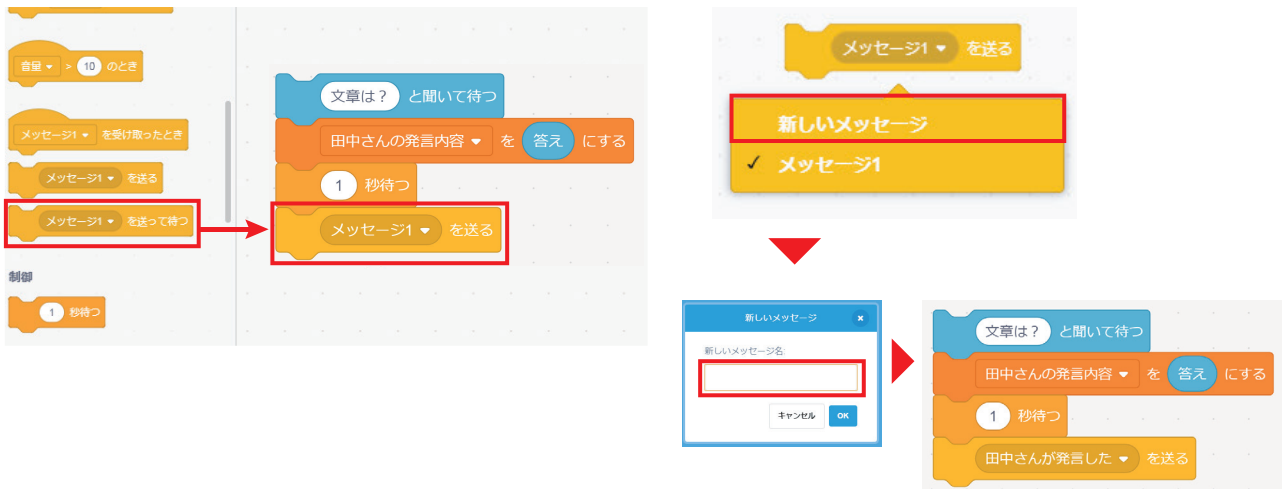
ここまでの、アクティビティ図の[1]のプログラムになります。

[1]

入力された内容を、変数「田中さんの発言内容」に入れる



- ⑥「制御」カテゴリから **1 秒待つ** と、「イベント」カテゴリから **メッセージ1 ▼ を送る** をドラッグしてプログラムの下につなげ、新しいメッセージを作成します。ここでは「田中さんが発言した」としています。



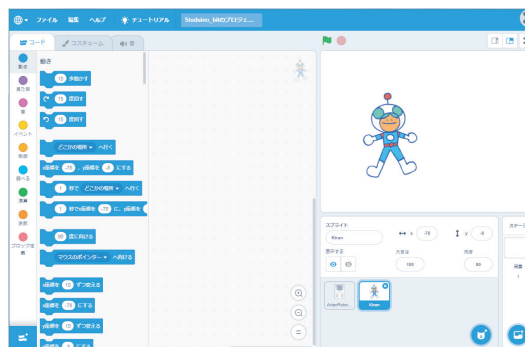
この部分がアクティビティ図の[2]のプログラムになります。 [2] 「田中さんが発言した」というメッセージを送信する

- ⑦「イベント」カテゴリから **がクリックされたとき** をドラッグしてプログラムの上につなげることで、 **がクリックしたらプログラムが動作するようになります。**



### ③アクティビティ図に対応したプログラミング (受け手側のPC: 鈴木さん)

- ①ネットワークでつながれた**別のPCを受け手側**としてプログラムを作成します。送り手側と同じスプライトを作成しましょう。



- ②「イベント」カテゴリから **メッセージ1 ▼ を受け取ったとき** をスクリプトエリアにドラッグし、送り手側で作成したメッセージと**まったく同じ文字**を入力して新しいメッセージを作成します。



※アルファベットの全角・半角が異なったり、漢字変換の際に異なる漢字に変換されていないか注意。

この部分がアクティビティ図の[3]のプログラムになります。 [3] 田中さんが発言した

③「見た目」カテゴリから「こんにちは! と言う」をドラッグして「田中さんが発言した▼を受け取ったとき」の下につなげます。



④送り手側で作成した変数とまったく同じ名前の変数を作成します。

※「すべてのスプライト用」として変数を作成してください。



※アルファベットの全角・半角が異なったり、漢字変換の際に異なる漢字に変換されていないか注意。

⑤作成した変数を「こんにちは! と言う」に挿入します。

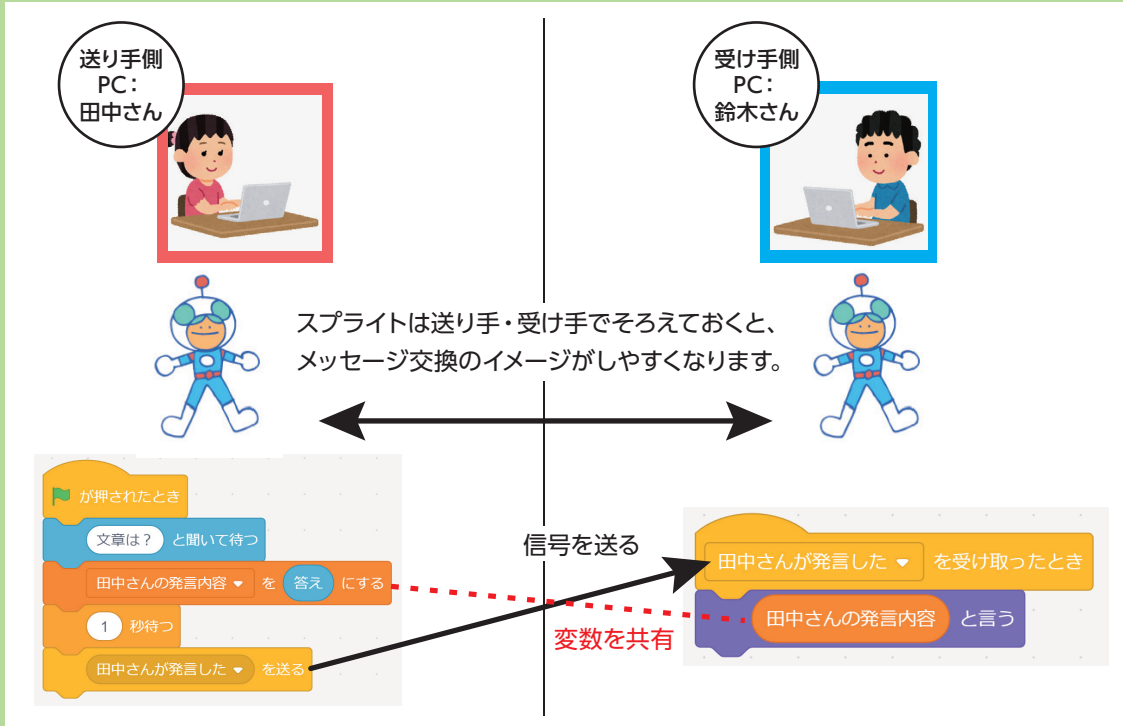


この部分がアクティビティ図の[4]のプログラムになります。

[4]

「田中さんの発言内容」を表示する

## メッセージの送信プログラムの完成イメージ



### 1 秒待つ を入れる理由

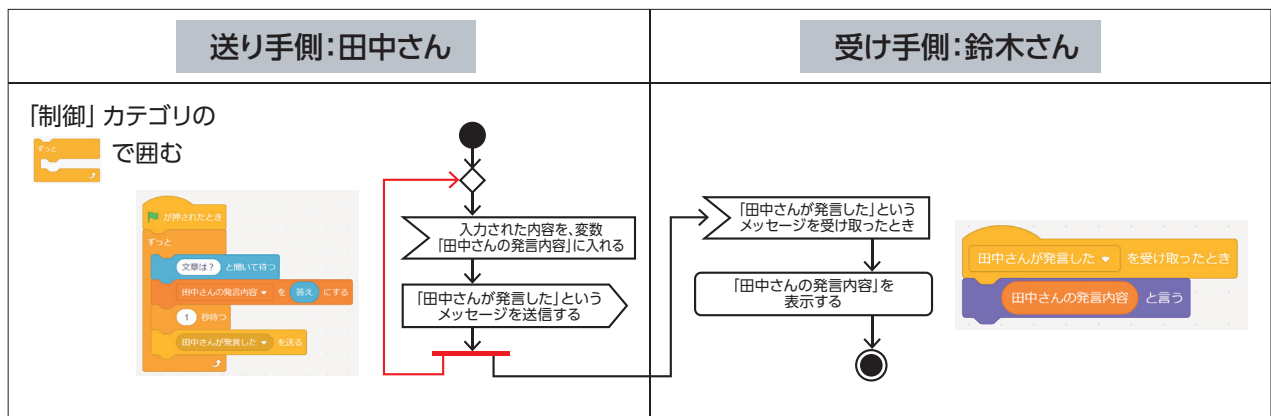
1 秒待つ を入れない右図のプログラムでも、メッセージの送信のプログラムとしては正しいです。しかし通信状況によっては 田中さんの発言内容 を 答え にする 処理で送り手側PCから受け手側PCに変数の値が共有される前に、田中さんが発言した を送る

メッセージが実行される場合があります。そうすると、通信が重複し変数の値の共有がされなくなってしまう問題が発生します。

この問題を防ぐ為に、田中さんの発言内容 を 答え にする と 田中さんが発言した を送る の間に 1 秒待つ を入れて、確実に 田中さんの発言内容 を 答え にする による変数の共有が実行されるように時間差を設けています。

## ④ 繰り返しメッセージを送るプログラムへの改良

アクティビティ図

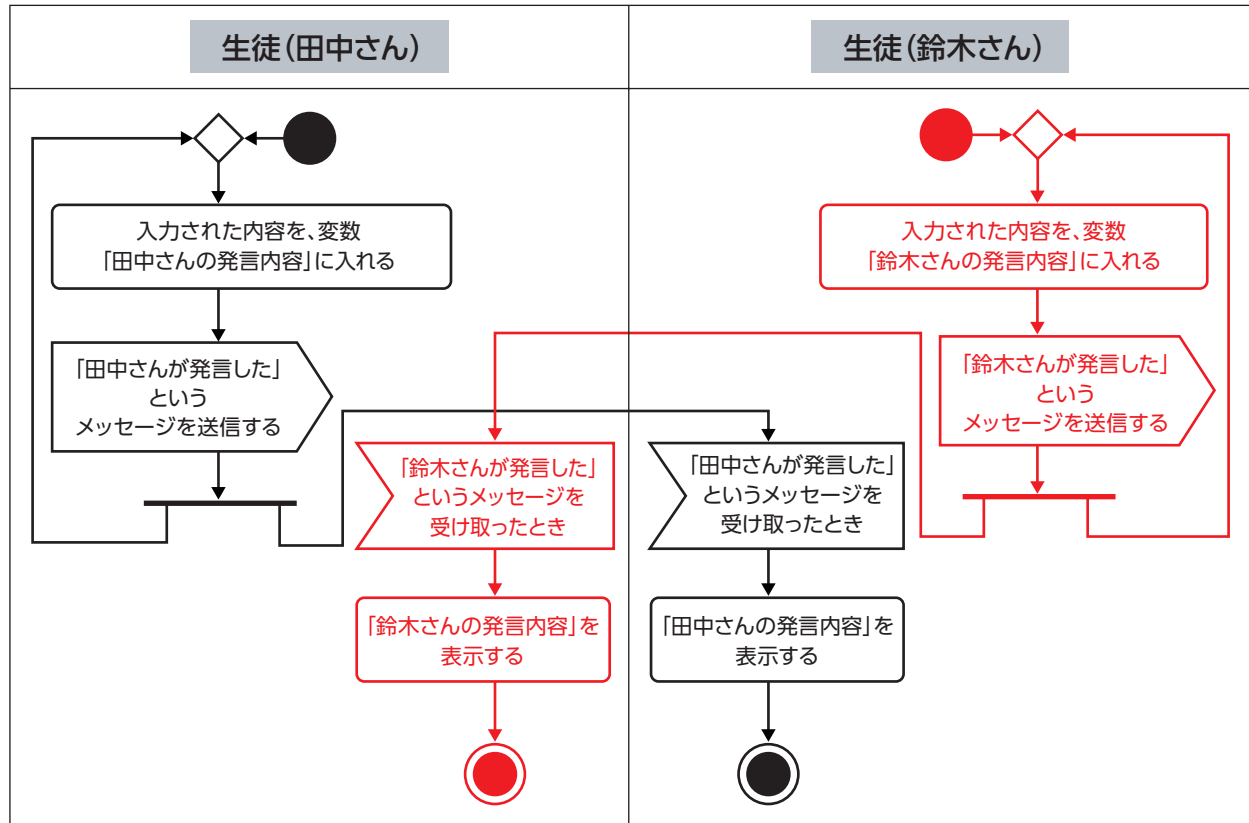


## ⑤双方向コミュニケーションの実現

ここまでのプログラムでは一方が送り手側、もう一方が受け手側となっています。

お互いでメッセージをやり取りする双方向コミュニケーションを実現できるように、それぞれのPCに送り手側と受け手側のプログラムを追加しましょう。

双方向コミュニケーションのアクティビティ図



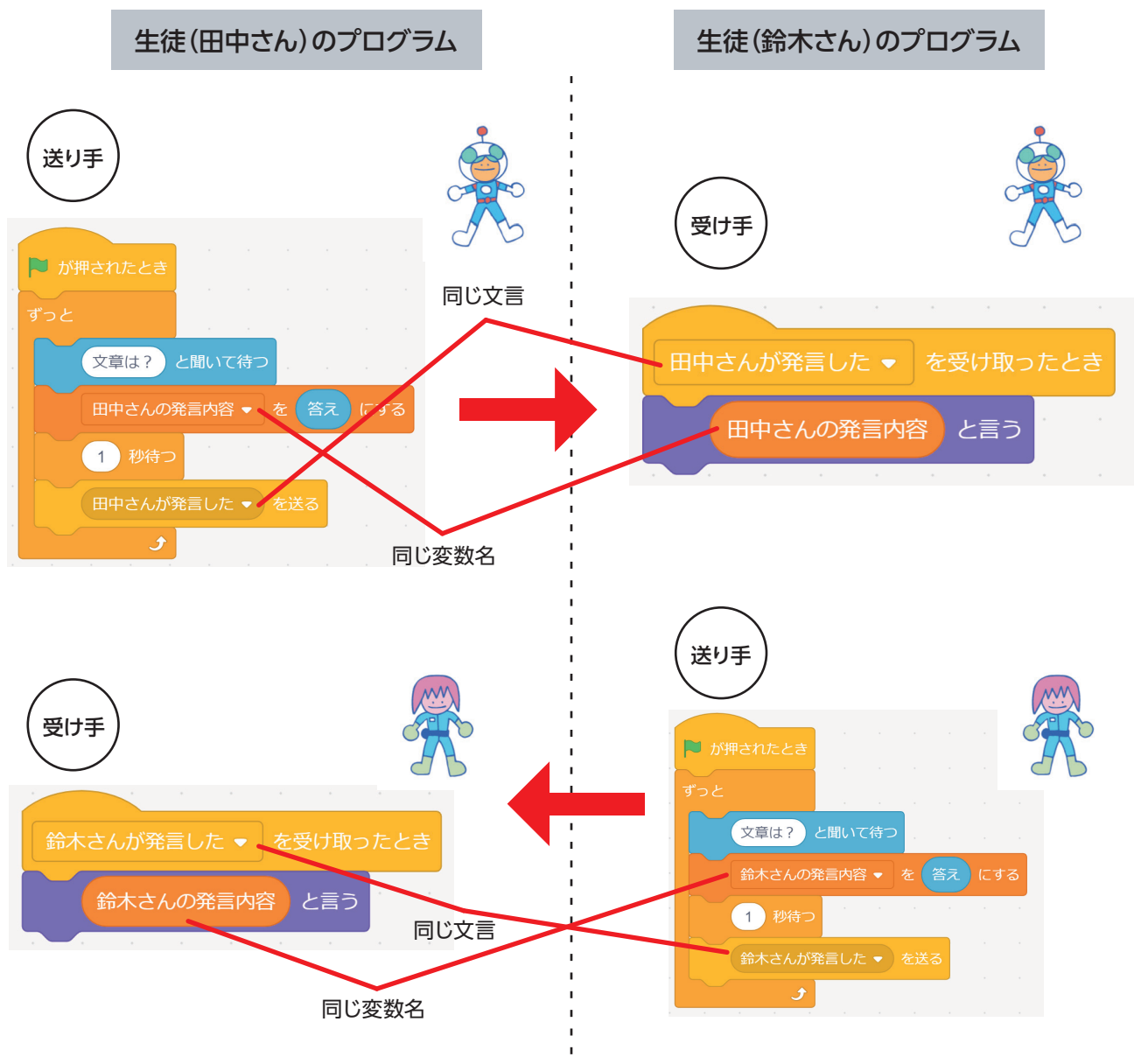
- ①プログラムを作成するために、新しいスプライトを追加します。

スプライトの名前の変更方法  
スプライトの名前は  
ここで変更できます。  
ここではそれぞれ  
「田中」「鈴木」と設定しています。





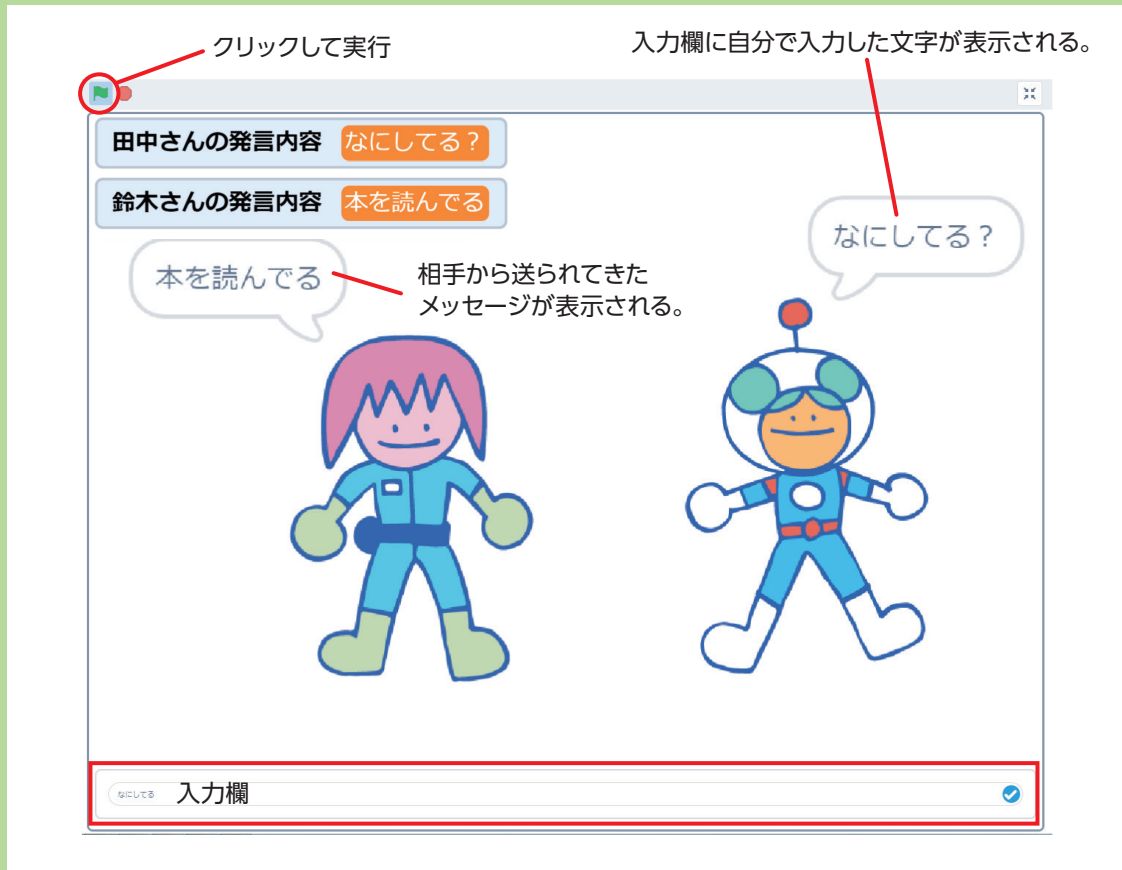
②以下のようにそれぞれ送り手側・受け手側のプログラムを作成します。それぞれの変数名・メッセージブロックの文言の  
関係に注意してください。



③送り手側のプログラムの「文章は? と聞いて待つ」に「答え」を挿入することで、自分が入力した文章(送ったメッセージ)を表示できるようになります。



## チャットアプリの完成イメージ



# テーマ 3

## チャットアプリの問題を解決

### <学習内容>

#### 0. ヒントカードの使い方

技術の見方・考え方：社会からの要求

1. 文字だけではニュアンスや感情が伝わりづらい → スタンプ機能
2. 過去に行われたやりとりがわからない → 履歴機能
3. メッセージを受け取ったことに気づきづらい → 通知機能
4. 時間を気にせず使いすぎてしまう → タイマー機能
5. 送ったメッセージが読まれたのかわからない → 既読機能
6. 複数人で同時にやりとりができない → グループトーク機能

技術の見方・考え方：安全性の観点

[情報の技術の見方・考え方：使用時の安全性，セキュリティ]

#### 7. プライバシーの侵害や成りすましが怖い

→ ログイン機能（セキュリティの観点）

[情報の技術の見方・考え方：情報の倫理]

#### 8. 不適切な言葉が送られる → NGワード機能（情報モラルの観点）

## 0. ヒントカードの使い方

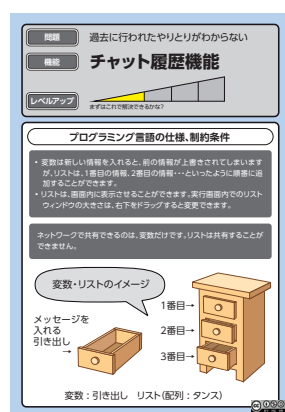
ヒントカードは、抽象度の高い解説のカードから始まり(ヒントレベル最小)、思考を促す問いや言語仕様やアルゴリズムの考え方を示すカード(ヒントレベル小)、処理の一部をアクティビティ図で示したカード(ヒントレベル中)、鍵となる部分の実際のプログラム例(ヒントレベル最大)の4段階で構成されています(下図参照)。

カードは、ヒントレベルが高くなる程、周囲の縁の色が濃くなっており、カード上部のインジケータでもヒントレベルが把握できるようになっています。生徒の中には、できるだけヒントを見ないで考えようとする生徒も出てくるのが考えられます。思考することは大いに推奨すべきことですが、実際のプログラミングにおいても、サンプルプログラムから基本的なアルゴリズムの考え方を把握し、それを自分のプログラムに合うように適応させることがよく行われます。ヒントを見ないでやるのが良いことなのだという印象を持っている生徒には、ヒントや先人の知恵に乗って、それをさらに発展させていくという考え方も紹介すると良いでしょう。

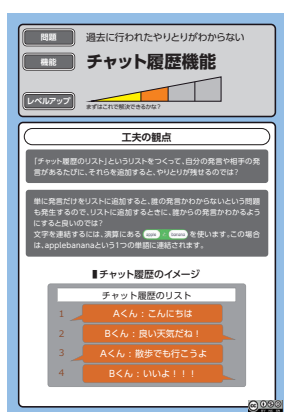
使用方法が決まっているものではありませんが、カードの趣旨としては少ない時数の中で、課題を解決する活動を充実させたいという意図で製作しています。詳細設計の部分における問題解決は、プログラミングの細かな問題になることが多いですが、生徒にとってはそのプログラミング言語で何ができて何ができないのかが不明なので、そもそも解決できないことに時間を掛けて悩んでしまい、文法的なルールを試行錯誤するなど、本質的な学習活動にならないことが懸念されます。各グループもしくは個人に印刷したヒントカードやpdf版を配布し、適宜参照させてください。

以下のサイトには、実際の中学校で活用した様子や指導過程が紹介されています。

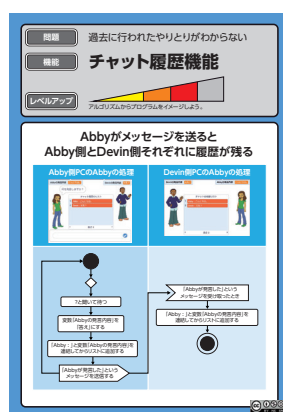
[https://www.kairyudo.co.jp/contents/02\\_chu/gijutsu/h33iko/sohoko.pdf](https://www.kairyudo.co.jp/contents/02_chu/gijutsu/h33iko/sohoko.pdf)



ヒントレベル最小



ヒントレベル小



ヒントレベル中



ヒントレベル最大

# 1. スタンプ機能

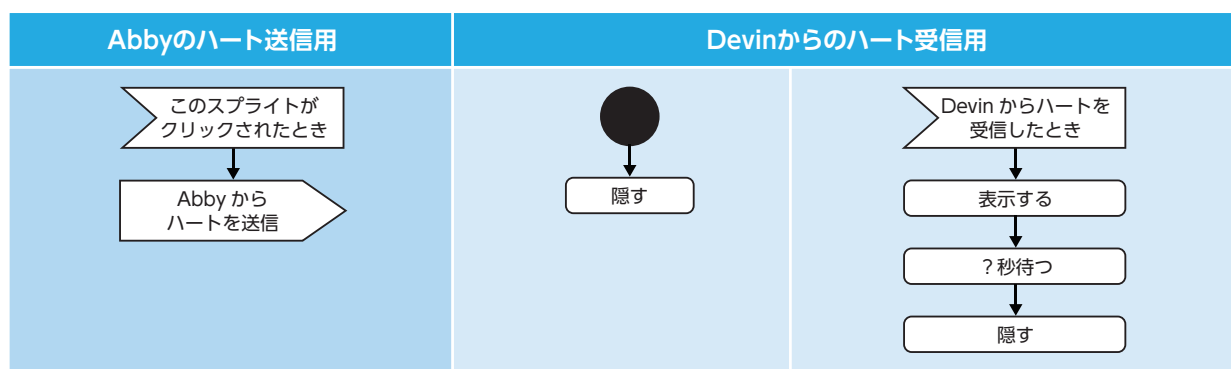
◆ 学習ポイント 中学校学習指導要領（平成29年告示）解説 技術・家庭編（p.54）

文字、音声、静止画、動画など、表現手段としてのメディアの効果的な利用

## プログラムについて

問題	文字では微妙なニュアンスを伝えづらい。
課題の設定	スタンプの絵柄によって感情を表現する。
処理	画面の中に送信したいスプライトを配置しておき、それがクリックされると、話している相手側に非表示になっているスプライトが表示される。
使用するメディア	スプライト
スプライト	ハート送信用：スタンプを送信するプログラム ハート受信用：スタンプを受信するプログラム

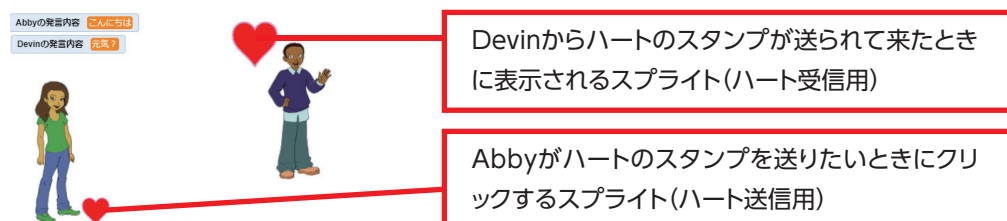
## プログラムのアクティビティ図（Abby発言用のコンピュータ側）



## プログラムの完成図（Abby発言用のコンピュータ側）



## プログラムの実行画面（Abby発言用のコンピュータ側）



## スタンプ機能のつくり方 (AbbyからDevinへハートのスタンプを送る)

①自分 (Abby) 側のハートのスプライト「ハート送信用」に対してプログラミングします。



②Abbyが発言する側のプログラムで、「ハート送信用」スプライトがクリックされたとき、イベント「Abbyのハート」というメッセージ（合図）が送られるようにします。  
(メッセージ名は自由に決められますが、送信側と受信側で同一標記にしなければなりません)



③相手 (Devin) 側からハートが送られてきた時に、Devinにハートのスプライトが表示されるように、「ハート受信用」に対してプログラミングします。



④ハートはあらかじめ送信側と受信側に用意しておき、相手からの合図が届いたタイミングで自分のハートを表示させることで、スタンプが送られてきているように見せています。最初から表示されているものではないので、緑旗がクリックされて、プログラムが開始された時にはこのスプライトを隠すようにします。

⑤イベント「Devinのハート」のメッセージ（合図）を受け取ったときに、スプライトの見た目「表示する」で表示させます。

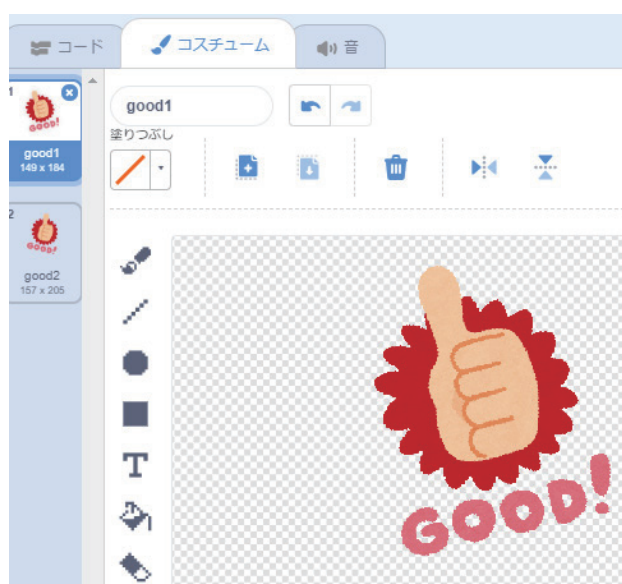
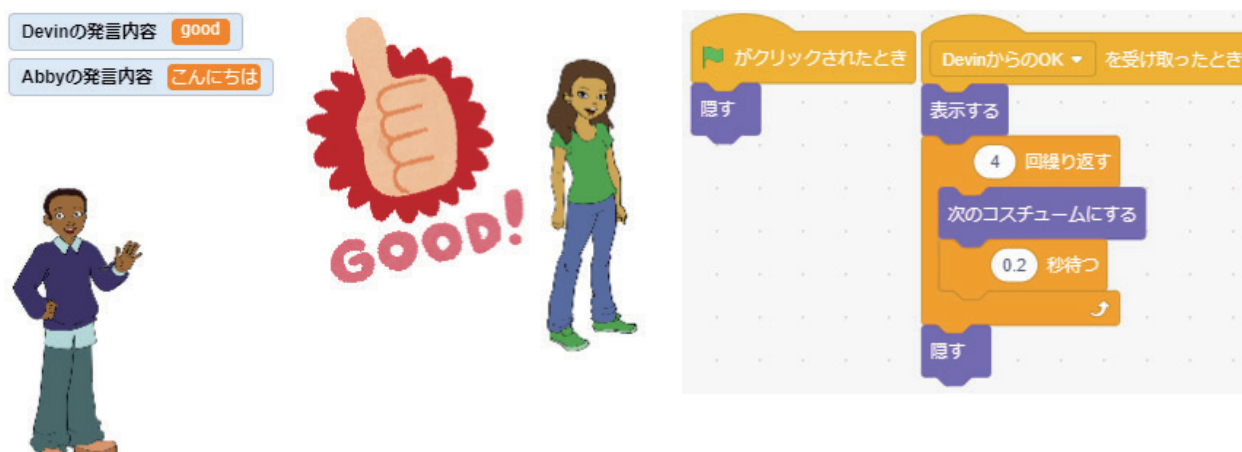
⑥イベント「〇〇秒待つ」→見た目「隠す」でスタンプは表示される時間を設定します（画面に表示し続けるスタンプではなく、一定時間表示されたら消えるスタンプを想定しています）。



⑦スプライト毎に送るメッセージを変えて設定すれば、様々な絵柄のスタンプを送り合えるようになります。

このプログラムをさらに以下のように改良することもできます。

動く（アニメーション）スタンプ



コスチュームを複数用意しておきます。



## ⑧音が出るスタンプ



## ⑨送った言葉に応じて表示されるスタンプ



生徒が日常的に利用していると思われるメッセージ交換アプリでは、相手がスタンプを持っていなくても、送信する側にさえスタンプがあれば送ることができます。このサンプルでは、画像データそのものは送信されず、チャットしている相手側に既にある画像を表示するという仕組みになっています。



## 2. 通知機能

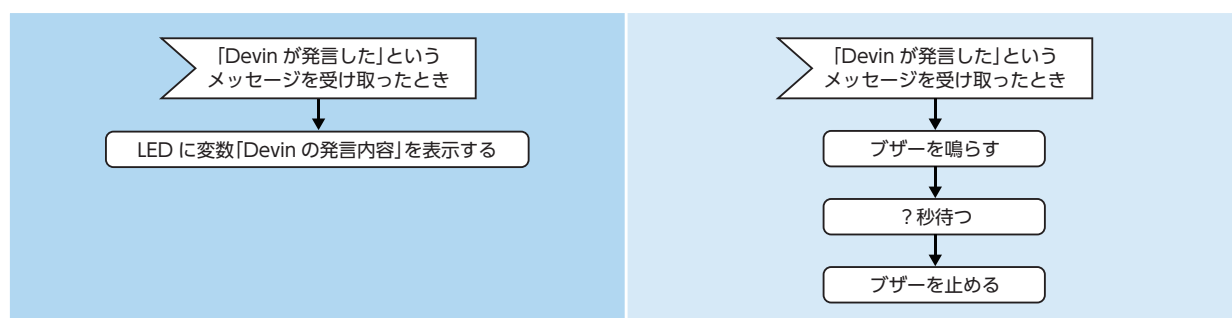
◆ **学習ポイント** 中学校学習指導要領（平成29年告示）解説 技術・家庭編（p.54）

5×5フルカラーLEDやブザーなどが搭載されていることを利用し、マルチメディアを効果的に利用して使用者に効果的に知らせる仕組みを考えさせる。

### プログラムについて

問題	メッセージを受け取ったときに気付かない
課題の設定	ArtecRobo2.0本体で光や音でメッセージを受信したことを知らせる。
処理	相手が発言したことをイベントとしてArtecRobo2.0で受け取り、LEDに表示させ、ブザーを鳴らす。
使用するメディア	5×5フルカラーLED ブザー
スプライト	ArtecRobo2.0：受信したときにLEDが光り、並列してブザーの音が鳴る

### プログラムのアクティビティ図（Abby発言用のコンピュータ側）

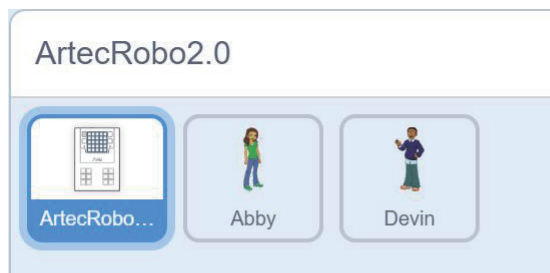


### プログラムの完成図



## 通知機能のつくり方

- ①ArtecRobo2.0のスプライトを選択します。



- ②前項同様、「Devinからのメッセージを送信」を受け取ったとき、が通知のトリガー（きっかけ）になっていることを確認します。
- ③前項で制作したプログラムに音「ブザーを鳴らす」、見た目「LEDを点灯する」等を追加し、相手からメッセージを受信したことを知らせます。



- ④ArtecRobo2.0の5×5LEDには英数字をスクロール表示させることが可能なので、見た目「LEDに（Devinのメッセージ）を表示させる」ことによってLED上に受信した発言内容をプレビュー表示させることができます。2021年4月現在では、日本語をLEDに表示することはできません。
- ブザーが鳴ってから、Devinの発言内容を表示するようにプログラムします。



- ⑤音を鳴らしながらLED表示をさせたいときは並列に処理をするようにプログラムします。

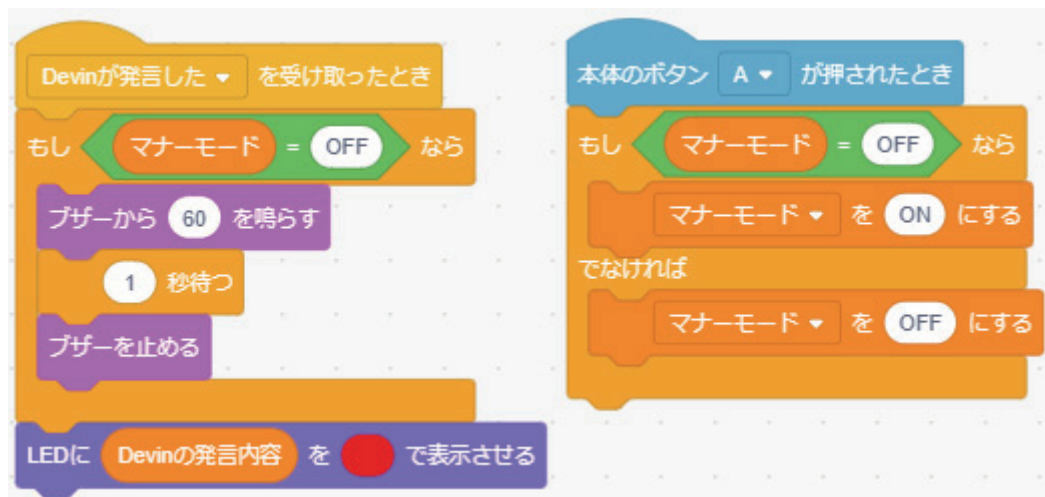


このプログラムをさらに以下のように改良することもできます。

①Aボタンを押すまで音が止まらない機能



②マナーモード機能



◆ こんな工夫をしてみました！「なんちゃってスマートウォッチ」



<https://youtu.be/d7e1c9NYCwM>

マジックテープのバンドを使用し、腕に装着できるようにしました。通知やプレビュー表示、後述のタイマー機能や、ログインのパスワードを使用することで、PC画面上のチャットと連携させることができます。



### 3. タイマー機能

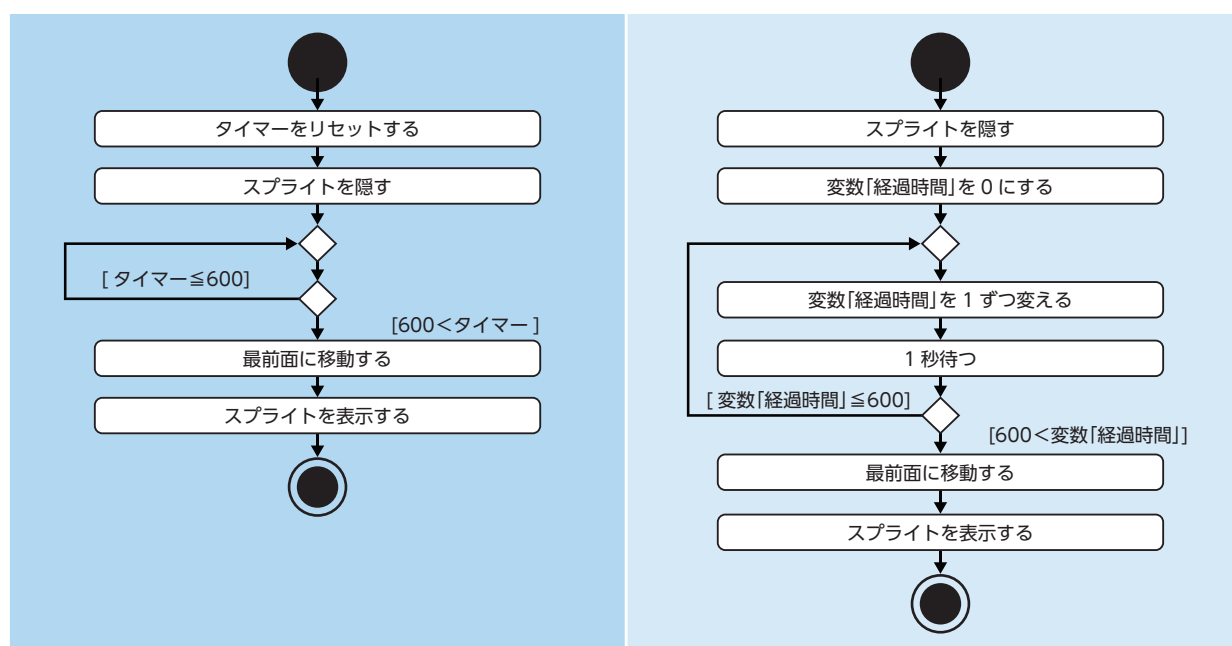
#### ◆ 学習ポイント 中学校学習指導要領(平成29年告示)解説 技術・家庭編(p.51、52)

情報の技術は使い方次第で「ネット依存」などの問題が発生する危険性があることを理解させ、情報に関する技術を利用場面に応じて適正に活用する能力と態度を身につけさせる。

#### プログラムについて

問題	時間を気にせず使いすぎてしまう。
課題の設定	設定時間になったことを知らせるスプライトを表示して使用者に知らせる。
処理	<ul style="list-style-type: none"> <li>● 計測を開始するタイミングにタイマーをリセットし、所定の経過時間が来たら処理を分岐させ、非表示にしておいたスプライトを表示させる。</li> <li>● タイマー機能を使わず並列処理として、「1秒待つ」を使って経過時間を計測する。</li> </ul>
使用するメディア	スプライト
スプライト	ArtecRobo2.0:受信したときにLEDが光り、並列してブザーの音が鳴る

#### プログラムのアクティビティ図（時間が来たときに表示するスプライト）



#### プログラムの完成図（時間が来たときに表示するスプライト）

タイマー機能と「ずっと」を使用した例



タイマー機能は使わず1秒ずつ数えて、「～まで繰り返す」を使用した例



## タイマー機能のつくり方

①設定した時間が来たことを通知するスプライトを作成します。

※メディアの特性の学習と関連させて、警告が伝わるような色や文字を入れるようにします。

### ◆ 学習ポイント 中学校学習指導要領（平成29年告示）美術編（p.110）

伝える、使うなどの目的や機能を考え、デザインや工芸などに表現する活動を通して、発想や構想に関する次の事項を身に付けることができるよう指導する。

ア 構成や装飾の目的や条件などを基に、用いる場面や環境、社会との関わりなどから主題を生み出し、美的感覚を働かせて調和のとれた洗練された美しさなどを総合的に考え、表現の構想を練ること。

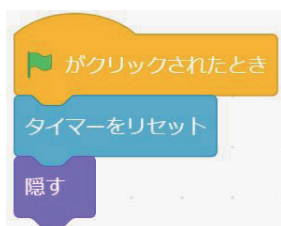
イ 伝える目的や条件などを基に、伝える相手や内容、社会との関わりなどから主題を生み出し、伝達の効果と美しさなどとの調和を総合的に考え、表現の構想を練ること。

ウ 使う目的や条件などを基に、使用する者の立場、社会との関わり、機知やユーモアなどから主題を生み出し、使いやすさや機能と美しさなどとの調和を総合的に考え、表現の構想を練ること。



②緑旗がクリックされたとき、調べる「タイマーをリセット」させ、見た目「隠す」ことで、プログラムを開始したときにタイマーがスタートし、はじめは見えないように設定することができます。

※ログイン機能と併用する場合は、ログインできたときに「パスワード認証を送る」のメッセージを送ることでログインしたことを知らせ、「パスワード認証を受信したとき」というメッセージを受け取ることで、タイマーをスタートさせます。



③制御「もし（タイマー>〇〇）なら」見た目「表示する」



このプログラムをさらに以下のように改良することもできます。

※通知音を鳴らしたり、スプライトを動かすなどの工夫ができます。

## 4. チャット履歴機能

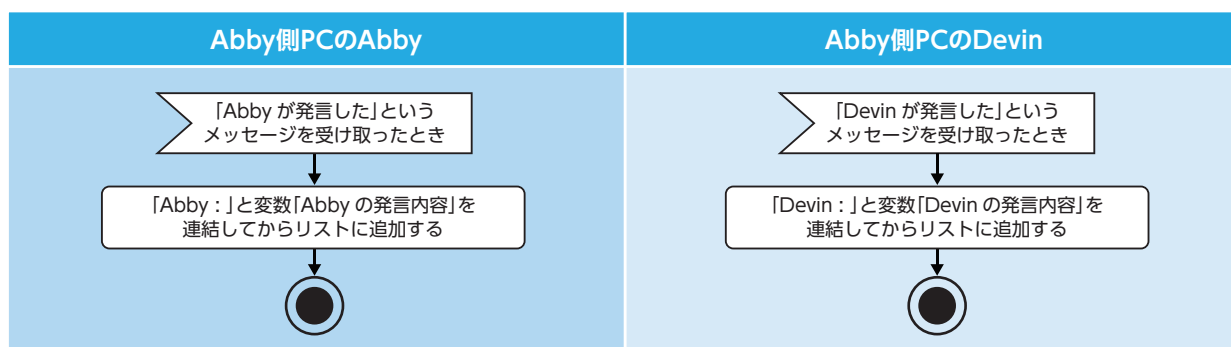
### ◆ 学習ポイント 高等学校学習指導要領（平成30年告示）解説 情報編（p.124）

リストの概念を理解し、高校情報科のデータの活用でテキストプログラミング言語を扱ったときの「配列」の学習に繋げる。

### プログラムについて

問題	過去に行われたやり取りがわからない。
課題の設定	リストにメッセージを残していく。
処理	リストを作成し、自分の発言と相手の発言をリストに追加していく。その際誰の発言かがわかるように発言内容の頭の部分に名前をつけるようにする。
使用するメディア	スプライト
スプライト	Abby：送信メッセージを履歴に追加するプログラム Devin：受信メッセージを履歴に追加するプログラム

### プログラムのアクティビティ図（Abby側のPC）



### プログラムの完成図





## チャット履歴機能のつくり方

- ① AbbyのSpriteを選択し、変数「リストを作る」より、「チャット履歴のリスト」というリストをつくります



- ② Abbyからメッセージが送信されたことの合図として「Abbyが発言した」を使用しているため、イベント「Abbyが発言した」を受け取ったとき、に変数「リストに (Abbyの発言内容) を「チャットの履歴リスト」に追加する、ようにします。
- ③ 同様にDevinからメッセージが送信されたことの合図として「Devinが発言した」を使用しているため、イベント「Devinが発言した」を受け取ったとき、に変数「リストに (Devinの発言内容) を「チャット履歴のリスト」に追加する、ようにします。
- ④ 誰の発言かがわからないので、演算「(Abby:) と (Abbyの発言内容)」のようにすると送信者付きで、発言内容を表示するようにできます。



- ⑤ 以上のプログラムによって、発言の履歴を残すことができ、かつ、誰が送ったメッセージなのかもわかるようになります。



## 5. 既読機能

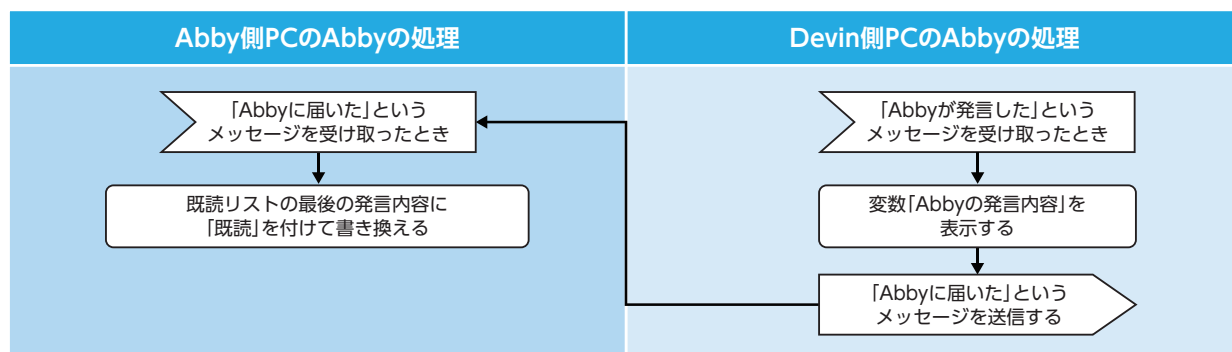
### ◆ 学習ポイント 中学校学習指導要領（平成29年告示）解説 技術・家庭編（p.55）

フィードバックという働きによって、正しくプログラムが処理されていることを使用者に知らせることで、不便さについて改善し、利便性を向上させたり、より確実に把握することで見落とさないように安全性を高める課題を設定し、その解決に取り組ませる。

### プログラムについて

問題	送ったメッセージが読まれたのかわからない
課題の設定	リストのメッセージに「既読」と付ける。
処理	自分の発言が相手に届いたら、相手に届いたことを示すメッセージを送り返すようにして、そのメッセージが届いたら、自分が発言した内容の末尾に「既読」をつけた内容に置き換える。
使用するメディア	スプライト
スプライト	Abby：履歴のメッセージに既読を付けるプログラム Devin：履歴のメッセージに既読を付けるプログラム

### プログラムのアクティビティ図（一部）



### プログラムの完成図（一部）





## 既読機能のつくり方

- ①これは履歴機能の応用編になります。
- ②受信したことを相手に伝えるプログラムを制作します。
  1. 受信用のプログラムを選択します。※ここでは自分側のプログラムのDevinのスプライトのことです。



2. 受信したメッセージを表示するプログラムの後ろにイベント「Devinに届いた」を送って待つ、を繋げます。



- ③送信したメッセージが読まれたことを知らせるプログラムを制作します。
  1. 送信用のプログラムを選択します。※ここでは自分側のプログラムのAbbyのプログラムのことです。



2. イベント「を受け取ったとき」を用意します。

3. 変数「チャットの履歴リストの\_\_番目を\_\_で置き換える」を繋げ、変数「チャットの履歴リストの長さ」を「\_\_番目」に挿入します。

※リストの長さというのは、リストに入っているデータの個数なので、リストの最後は「リストの長さ番目」ということになります。

相手のPCにあるAbbyに発言内容が届くと「Abbyに届いた」を送ってくる仕組みなので、自分側では「Abby」に届いた」を受け取れたらメッセージが届いたということなので既読された、とします。

4. 演算「(チャットの履歴リストの(チャットの履歴リストの長さ)番目)と(<既読>)」を用意し、前項の「\_\_で置き換える」に挿入します。



5. これで、自分が最後に送ったメッセージが読まれれば<既読>が付くようになります。



## 6. グループトーク機能

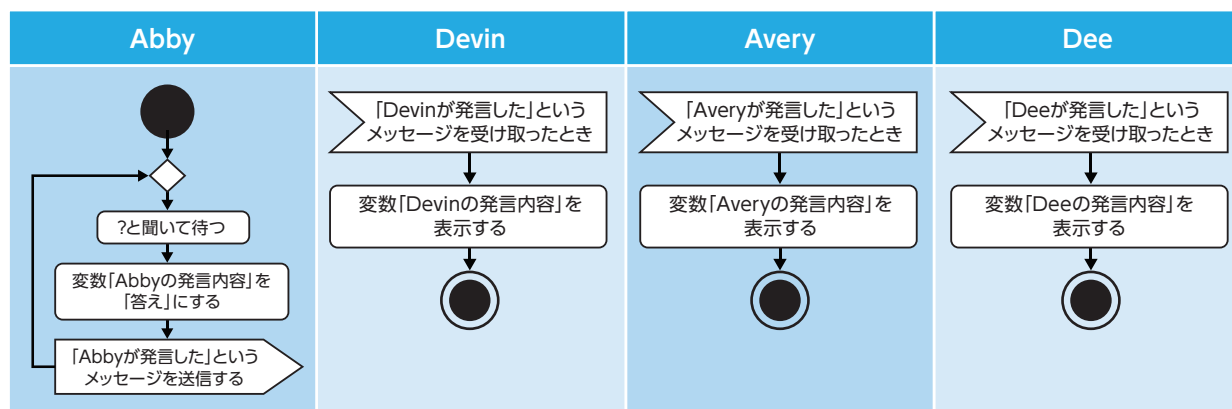
◆学習ポイント 中学校学習指導要領（平成29年告示）解説 技術・家庭編（p.53-54）

1対多でPC同士を接続して情報通信ネットワークの構成し、情報を利用する仕組みについて理解させる。

### プログラムについて

問題	複数人に同時にメッセージが送られない
課題の設定	複数人でのグループチャットを構成する
処理	発言する方法は、これまでと同様。発言したことをメンバー全員にメッセージとして送り、そのメッセージが届いたタイミングで、その人の発言内容の変数を表示させる。
使用するメディア	スプライト
スプライト	Abby：「Abbyが発言した」を送信するプログラム Devin：「Devinの発言内容」を受信するプログラム Avery：「Averyの発言内容」を受信するプログラム Dee：「Deeの発言内容」を受信するプログラム

### プログラムのアクティビティ図（Abby側PC）



### プログラムの完成図（Abby側PC）



## グループトーク機能のつくり方

- ①トークに参加する人数に合わせてスプライトを追加します（ここでは4人でのチャットをする方法を紹介します。）
- ②3人目のスプライトAveryを用意します。Averyの発言内容が表示されるように「Averyが発言した」というメッセージを使用して、それを受信したときに、変数「Averyの発言内容」を表示させます。



※4人目のDeeの場合も同様に変数「Deeの発言内容」をイベント「Deeが発言した」を使用してプログラミングしていきます。



## 7. ログイン機能

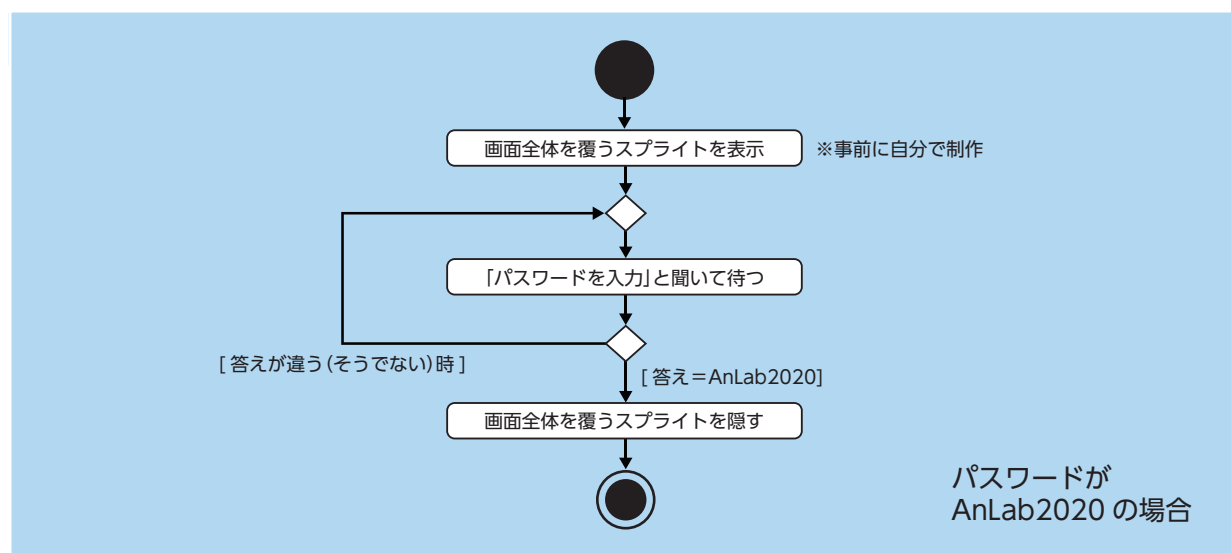
### ◆ 学習ポイント 中学校学習指導要領(平成29年告示)解説 技術・家庭編(p.52)

デジタル化された情報を安心して使用するための手立てとしてID・パスワードなどの個人認証について理解を深めさせる。

### プログラムについて

問題	プライバシーの侵害やなりすましされる危険性がある
課題の設定	パスワード認証でのログインにする。
処理	順次処理、条件分岐処理
使用するメディア	スプライト
スプライト	ログイン画面：パスワードを認証するプログラム

### プログラムのアクティビティ図



### プログラムの完成図



## ログイン画面の作り方

①新しいスプライトを制作し全画面に大きさを合わせます。

※「ロックを解除してください」等のメッセージがあると、わかりやすくなります。

※大きさはスプライト制作時に大きく制作するか、大きさの倍率を変更することで画面全体にあわせることができます。



②緑旗をクリックしたとき、このスプライトが最前面に表示するようにします。

③パスワードを使用する場合(難易度★☆☆)

1. 調べる「(パスワードを入力)と聞いて待つ」

2. 制御「もし(答え)=(AnLab2020)だったら」見た目「隠す」にすることで、そのパスワードの時ロック画面のスプライトが消え、ログインできるようになります。

※セキュリティの観点から、アルファベットの大文字と小文字、そして数字を使用することで、セキュリティレベルを上げています。

3. イベント「(ログイン成功)を送る」ことでロックが解除されたことを知らせます。

4. それと同時に、AbbyとDevinのスプライトの「緑旗がクリックされたとき」を「(ログイン成功)を受け取ったとき」に変更しておく。



### ③ ボタンを使用する場合(難易度★★☆)

1. p.38「なんちゃってスマートウォッチ」で触れた内容です。
2. ArtecRobo2.0のスプライトに対してプログラミングします。
3. ArtecRobo2.0のスプライトを選択してください。
4. 変数「リストをつくる」より「押されたボタンリスト」をつくる。
5. 調べる「ボタンAが押されたとき」変数「(A)をパスコードのリストに追加する」
6. 調べる「ボタンBが押されたとき」変数「(B)をパスコードのリストに追加する」

※ボタンが押されたときにLEDを点灯させることでボタンが正常に押された事を確認するフィードバックとして機能します。



7. プログラムが開始されたときにリストを空にしてこれまで入力されたものがリセットされるように、変数「押されたボタンリストのすべてを削除する」
8. 制御「(押されたボタンリストの1番目=A)かつ(押されたボタンリストの2番目=B)かつ(押されたボタンリストの3番目=B)まで待つ」イベント「(ログイン成功)を送信する」



### 9. ロック画面のスプライトを選択します。



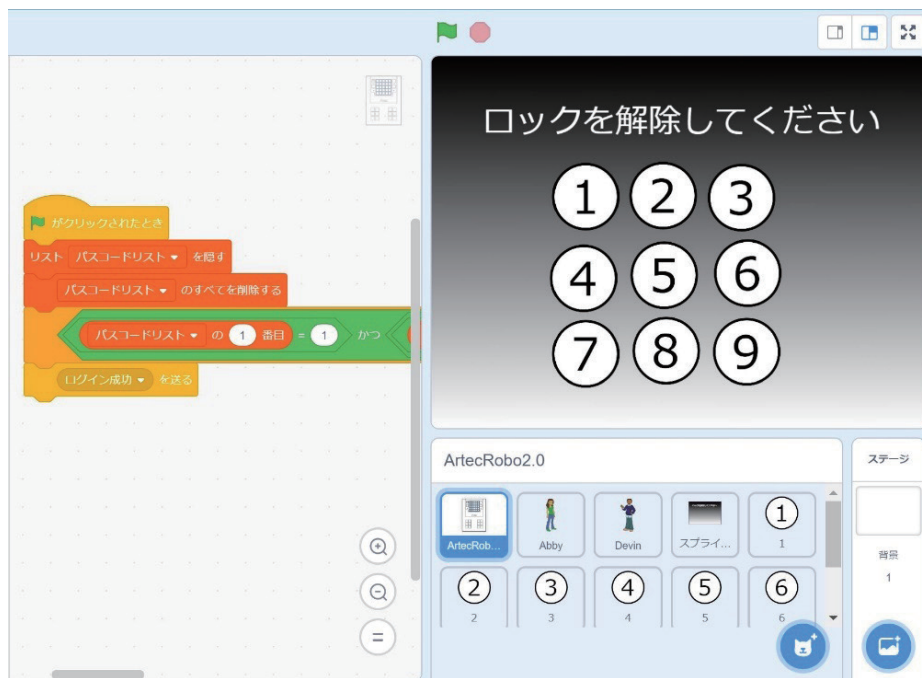
10. イベント「(ログイン成功)を受け取ったとき」見た目「隠す」とすることで、ArtecRobo2.0のボタンをA→B→Bで押したときにロック画面が消えます。





### ③パスコードを使用する場合(難易度★★★)

画面上に1～9のボタンとなるスプライトを用意し、任意の4桁の数字の時にログインできるようにすることもできます。



### ◆ こんな工夫をしてみました!「磁石でパターンロック解除」



<https://youtu.be/GKppaQWoGvc>

内蔵されているxyzの3軸に対しての磁力センサを利用して、ArtecRobo2.0本体の上で磁石を移動させるパターンでロックを解除するように工夫しました。

磁石の場所を変えて、各場所でのxyzの磁力センサーの値をメモしておいて、どの順でどのセンサーが反応したかをチェックしています。反応したセンサーをリストに追加し、その組み合わせで正しい順番か、そうでないかを確認しています。



## 8. NGワード検出機能

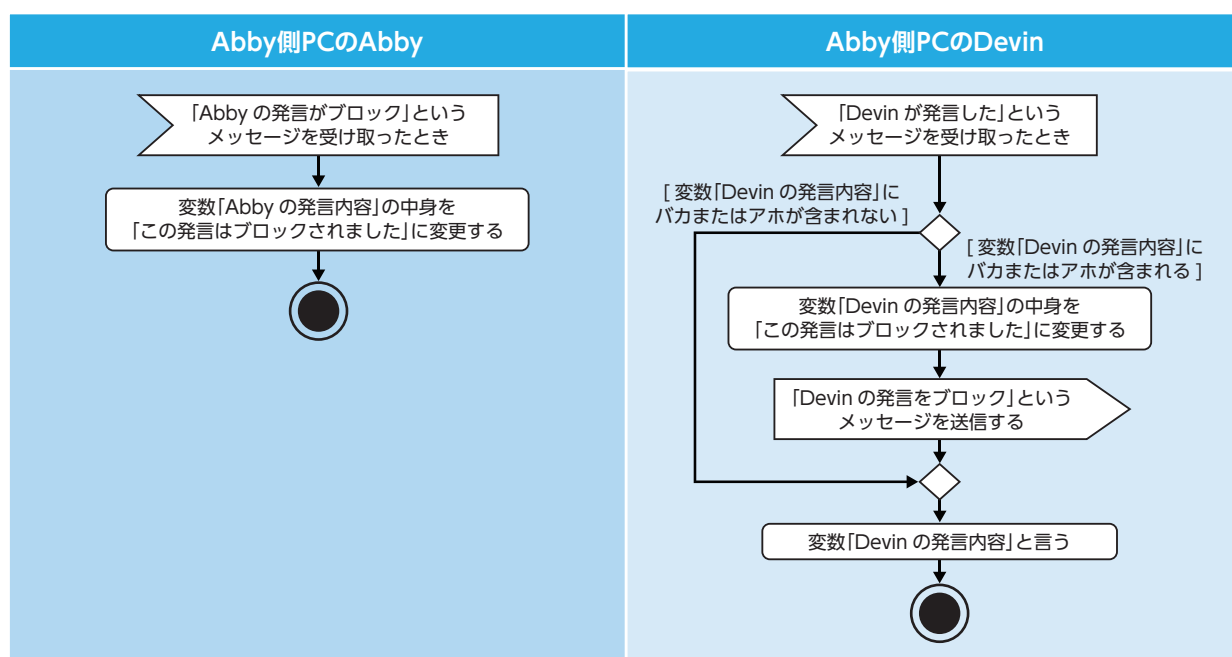
### ◆ 学習ポイント 中学校学習指導要領（平成29年告示）解説 技術・家庭編（p.55）

問題の機能的な解決を図りながら、送り手と受け手の両方の役割を持つようになることを理解させ、情報の技術に関する倫理観の育成に努める。

### プログラムについて

問題	不適切な言葉が送られてしまう。
課題の設定	NGワードに設定した言葉を表示しない。
処理	変数の内容に特定の言葉が含まれるかを調べる。 順次処理 条件分岐処理
使用するメディア	スプライト
スプライト	Abby：メッセージがブロックされたことを表示するプログラム Devin：メッセージをブロックするプログラム

### プログラムのアクティビティ図（一部）



### プログラムの完成図（一部）



## NGワード検出機能のつくり方

①自分側の受信用のスプライトを選択します。



※ここではDevinを選択しています。

②イベント「Devinが発言した」を受け取ったとき、制御「もし（Devinのメッセージに〇〇が含まれる）なら」、変数「Devinの発言内容」を「この発言はブロックされました。」にします。

③イベント「Devinのメ発言をブロック」を送って、メッセージをブロックしたことを相手に知らせます。



④演算「Devinの発言内容」に〇〇が含まれる」において、〇〇に設定した言葉がNGワードに該当することになります。上記では、2つの言葉を「または」でつないでいます。



- ⑤NGワードを増やしたい場合は、演算「〇〇または〇〇」によって増やすことができます。演算「Devinの発言内容に〇〇が含まれる」の〇〇にそれぞれNGにしたい言葉を設定してください。



※3つ以上のNGワードを登録する場合は、演算「〇〇または〇〇」をさらに連結します。

- ⑥自分が送ったメッセージがブロックされたことを知るプログラムをつくるために、自分のスプライトを選択します。



※ここではAbbyを選択しています。

- ⑦「Abbyの発言をブロック」を受け取ったとき、Abbyの発言内容を「この発言はブロックされました。」にする。

