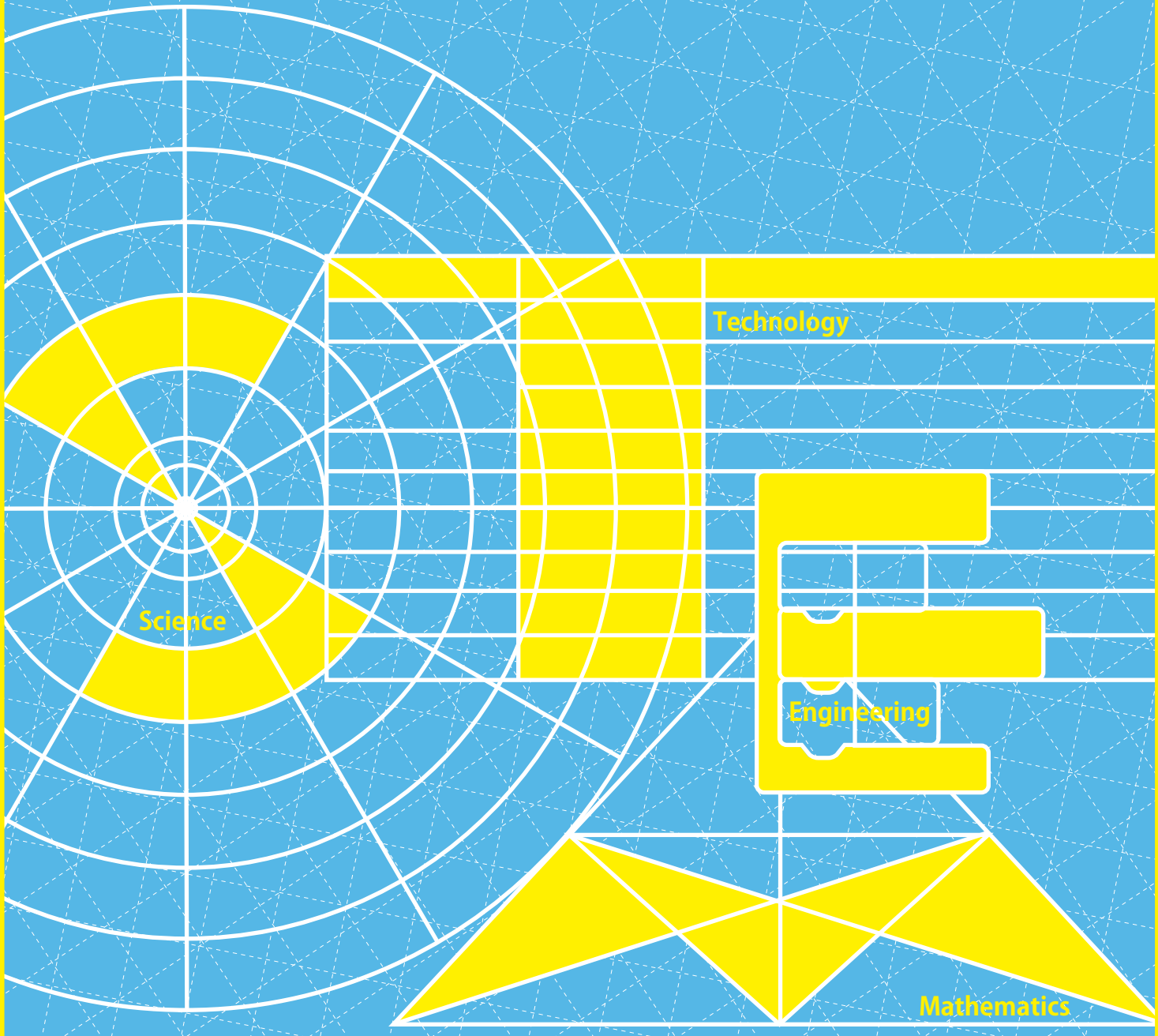
JSPS 科研費若手研究 ( 課題番号 19K14210)

# 理科教育研究グループ研究報告書 vol.4

## -STEM 教育のものづくりとプログラミングの融合 -




## 扇風機工作の型紙と micro: bit の手順書も掲載

# プロジェクトメンバー

現代教育研究所所員
昭和女子大学
准教授　白數　哲久

現代教育研究所研究員
久木田　寛直

現代教育研究所研究員
星名　由美

# サポートメンバー

現代教育研究所所員：森　秀樹・小林　広利
現代教育研究所特別招聘研究員：小川　哲男
現代教育研究所研究員：井上　文敏・小刀稱　進・酒井　隆光・佐貫　礼奈
NPO 法人ガリレオ工房:吉澤　弘（工業デザイナー）・市村　賢一（エンジニア）
（株）アルファ企画：佐々木　仁・櫻井　匡仁（撮影・編集）

# 目次

# はじめに

昭和女子大学准教授　白數 哲久

子どもたちは一人一台の端末を学習に活用するようになり、AI によるサポートによって個別最適な学びが加速しつつあります。将来的には仮想空間にある学校に自分のアバターを通わせるような未来が訪れるかもしれません。しかし、仮想の学校でクリエイティヴィティ（創造力）やコミュニケーション力を育てることは難しいでしょう。オンライン授業が増えるにつれて、その反証として人と人とが場と道具と時間を共有することで、実感を伴った記憶や理解が構築されていることが明らかになってきました。現在のオンライン技術では、伝えきれないこと共有できないことが多いのです。理科室のあの場所で、仲のいい友達とうまくいかない実験に汗をかきながら一生懸命に取り組んだ・・・といった記憶は、質感のある記憶として身体に残るようです。したがって、これからは、個別最適な学びと協同的な学びの両方を組み合わせていくことが必要でしょう。特に探究的な学習やプロジェクト型学習を行い、仲間と情報を共有していく場を提供しアドバイスをしていくことが、現実の学校の新たな役割になっていくのではないかと思います。

さて、小学校では教科学習にプログラミングを取り入れることが必須となり、特に理科では電気の学習で私たちの生活を支えている科学技術への理解を意識した教育の推進が求められています。プログラミング教育ではコーディングを学ばせるのではなく、プログラミング的思考を育成することと、生活をより良くしようとする力を身につけさせることに重きが置かれています。そこで私たち理科研究グループは、埼玉大学 STEM 教育研究センターと連携して 2018 年から研究を始め、プログラミングを探究的な学習に組み込むにはどのような教材をどのように使ったらいいか検討を進めてきました。1 年目は SONY の MESH、2 年目は埼玉大学 STEM 教育研究センター開発の STEM-DU とレゴブロック、3 年目は micro:bit を教材に用いました。授業の設計において私たちが大切にしてきたことは次の 3 点です。

**① 手を使ったものづくりとプログラミング教育の融合**
**② 子どもが自分でゴールを定めて取り組む探究的な学び**
**③ 過干渉の回避**

現在多くの学校でどのようにプログラミング教育を行うか議論されています。一人一台の端末を活用した実践がなされています。しかし、プログラミングに対して苦手意識を持つ教員が多いことから、今後、開発をする企業と学校との連携が加速し、様々な教材が学校で活用されるようになっていくことでしょう。私たちもいくつかの実践事例を見てきましたが、開発企業と協同的にプログラミング教材を学校に取り入れる場合、先に述べた３つの観点において注意すべき点があります。

まず、ものづくりとの融合の観点です。画面上で行うプログラミングにも優れたものが多数あります。しかし、子どもの発達段階を考えると諸感覚を働かせることも大切です。自分で切ったり貼ったり曲げたりすることでしか気づけないことがあります。思い通りに行かないところにも発見があるでしょう。このような経験をする過程で物をよく見て考えるようになり創造力が育つのではないでしょうか。したがって、STEM 教育の観点からも手を動かしながら頭をはたらかせることが重要だと考えています。

次に、探究的な学びについてです。完成図がある市販のキットではゴールが定められています。もちろん、完成してから更にそれを活用して新しいことに挑戦していけば探究的な学びになっていくことでしょう。しかし、学校の教育課程の枠組みの中でプログラミングに使える時間はそう多くはありません。多くの学校では完成させて終わりにせざるを得ないことも多いのではないでしょうか。そうだとすると創造力を発揮させることのできる部分は少ないことから主体的な学びになりにくく受け身の学びに留まってしまうことが懸念されます。

最後に、過干渉の回避についてです。決められた時間内に定まったゴールに辿り着かせようとするとどうしても大人は過干渉になりがちです。上手くいかない子どもに手を差し伸べたくなってしまいます。工作やプログラミングが得意な大人ほど手を出したくなりがちです。子どもは失敗から多くのことを学びますから、失敗しないように先回りをすることは子どもの学びの機会を狭くすることだと私たちは考えています。

以上のことから、教材の使用においては、手を動かすことが多く操作性の高い教材を選び、出来るだけゴールを定めず、子どもの自由な発想に委ねて、アドバイスをするものの出来るだけ手を出さないようにすることが大切だと言えます。しかし、教材キットや授業プログラムの取り入れ方によっては、プログラミング的思考を十分に育てることにつながらない場合があるのではないかと思います。例えば、ロボットを作り動かすためのプログラムの組み方を学び、人が近くに来るとお辞儀をするようなロボットの制作に留まった場合です。子どもが先回りをして設計図を手に入れてしまうとプログラミング的思考をはたらかせる機会が減ってしまいます。壁を乗り越えようとするときに人は熟考するので、乗り越えたくなるような強い想いと適度な困難さが必要なのです。今、学校で始まりつつあるプログラミング教育は、子どもの想いを重視した学びになりにくいのではないかと思っています。私たちは、どうすればこれを解決できるか考え実践的に検討を重ねてきました。問題となっているのは主に、指導者不足と、適した教材を選べないことの２点ではないかと思います。そこで、私たちは次の提案をしたいと思います。

**「教師が工作やプログラミングの指導が不得手であっても、適した教材と手順書があれば、子どもは仲間と共に学ぶことができる。」**

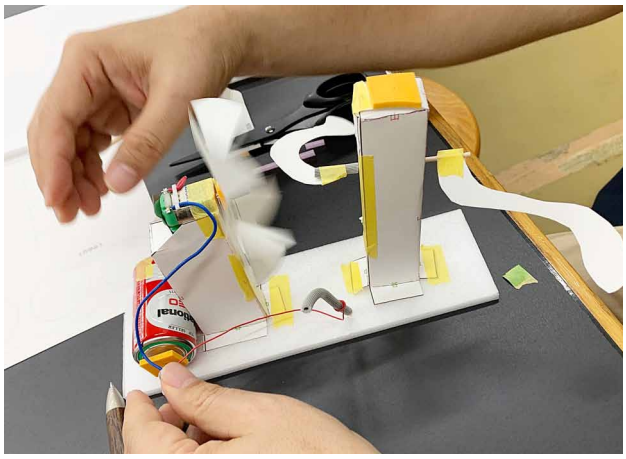
ここで言う「適した」というのは、①わくわくする遊びに満ちていて作業を進めていくと次々に発見があること、②ほとんど教師の助けを借りずにできるように作られていること、③乗り越えなくてはならない若干の困難さがあること、の３点です。私たちは、この仮説を確かめるために、数回に渡り授業実践を重ね教材と手順書を改良・改訂してきました。まず工作については小型扇風機づくりから始めることにしました。プログラミングではインプットとアウトプットを意識することは大切です。そこで、例えば「気温が上がる」というインプットで「扇風機が回る」というアウトプットが生じるとい

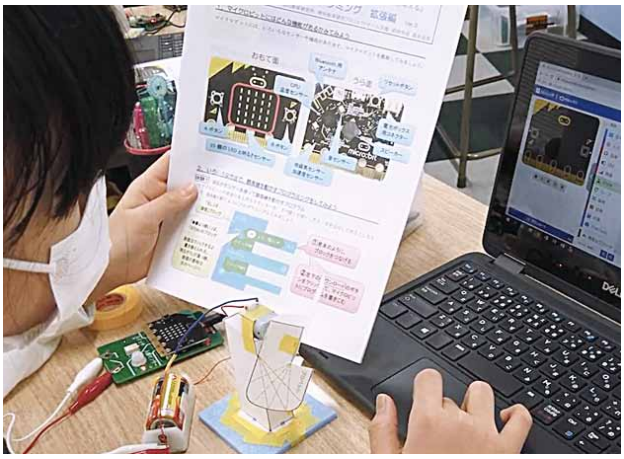
う場面設定を子どもがイメージしやすいと考えたからです。最初に用意する材料は主に学校にあるモーター、乾電池、画用紙、マスキングテープなどです。型紙はダウンロードして印刷できるようにしました。切るところはできるだけ少なくして、山折り線と谷折り線を見て直感的に組み立てられるように設計してもらいました。扇風機の羽根は大小あり角度も定まっていません。子どもはより強い風が起こるように試行錯誤します。ある程度風が出たら他の人と比べて強い風になっているか調べたくなります。そこで、風を受けて動く風車などを数種類選べるようになっています。見本の写真は用意しましたが、それは完成図ではなく自由にアレンジしてもらうことを想定しています。どうしたら風を受けて動きやすくなるか子どもたちは考えながら手を動かし形にしていきます。

一方、プログラミングでは . 多くの事例がオンラインで紹介されている micro:bit にティーファブワークス社の「STEM キット C」(https://tfabworks.com/product/)を組み合わせて用いました。そして、扇風機づくりと同様に、子どもだけでスクラッチタイプのプログラム組めるように手順書を作成しました。およそ 30 分間で、人が近づくと扇風機が回るような装置を作ることができていました。制作物のほとんどを自分で作っているので、子どもたちは創造性をはたらかせて自由に作品作りを進めていきます。そのきっかけづくりでは、「人が玄関から入ってきたら風で花粉を飛ばす装置」というような場面設定を紹介しました。これを受けて、子どもたちは、人が来たら音楽が鳴り文字が浮かび上がるような仕組みなどを作っていました。別の小学校では体に障害がある人のために役立つ仕組みを作るというテーマを設定することもありました。このような場面設定を用意することで、子どもは学びやすくなるのではないかと考えています。

ここまで 4 年間かかりましたが、ようやく皆さんに参考にしていただけるような教材と手順書が出来ました。しかし、教師用指導書が必要かもしれません。米国では実際に教師が子どもに語りかけ、それを受けて子どもがどのように学んでいるか分かる動画を教材と共に紹介している場合があります。我が国ではこのような動画はあまり公開されていません。そこで、撮影に協力してもらえる小学生に集まってもらい授業の様子を撮影しました。この動画を見ていただくことで、私たちが子どもに教えすぎることなく、子どもが主体的に手を動かしながらプログラミングを学んでいく様子をイメージしやすくなるのではないかと思っています。2023 年のうちに動画を公開する予定です。皆様からのご意見をいただければ幸いです。

本研究は、JSPS 科研費若手研究 ( 課題番号19K14210) の助成を受けています。





# 子どもの能動から始まる「拡張による学習」

昭和女子大学准教授　白數 哲久

## 1　受動から能動へ

ドイツの児童文学作家ミヒャエル・エンデは、時間どろぼうに盗まれた時間を取り返してくれた女の子を描いた物語『モモ』を 1973 年に発表ました。この作品について、臨床心理学者の河合（2020）は、主人公のモモが受動から能動へと立ち上がる物語であると述べています。この頃のヨーロッパは、民主主義教育の必要性が説かれはじめる直前で、教師による教え込みが問題視されていましたが、次の一節は子どもに受動を強いる学校教育への風刺と捉えることができます。

「で、これからどこに行くの？」
「遊戯の授業さ。遊び方をならうんだ。」と、フランコがこたえました。
「それ、なんなの？」
「きょうやるのは、パンチ・カードごっこさ。」と、パオロが説明しました。（中略）
「質問をしながらほかのカードをのけていって、さいごに一枚、目的のカードがのこるようにしなくちゃいけないんだ。これをいちばんはやくやったのが、勝ちさ。」
「そんなことがおもしろいの？」とモモは、いぶかしそうにききました。
「そんなことは問題じゃないのよ。」と、マリアがおどおどして言いました。「それは口にしちゃいけないことなの。」
「じゃ、何がいったい問題なの？」
「将来の役に立つってことさ。」とパオロが答えました。

ミヒャエル・エンデ（1973）：大島かおり訳（1976）『モモ』岩波書店

物語の学校では「遊び方をならう」という奇妙な描写があります。「遊び」は保育においては主体的な学びの場として重要な意味を持っているわけですが、学校においても同様に能動的な学びと切り離すことのできない象徴的な存在です。ところが物語の学校では「将来の役に立つ」ことを理由に子どもの砦である「遊び」を奪い、代わりの遊びを押し付け、子どもが疑問に思っても「それは口にしちゃいけない」と心に蓋をさせてしまっています。このことから 1970 年代のヨーロッパの授業では子どもは受動的であったことが想像できます。では、現代はどうでしょうか。限られた時間内に決められた内容を教えるために、子どもの思考の流れに十分な配慮ができないまま授業を進めざるを得ないことがあるかもしれません。これではいけないと、社会の変化に主体的に対応できる資質や能力を育成する学習として総合的な学習の時間が誕生しましたが、子どもの主体性は取り戻せたのでしょうか。総合的な学習の時間も各教科と同様に決められた指導計画に沿って進められるだけになってはいないでしょうか。また、新しく始まったプログラミングの授業はどうでしょうか。PC でゲームを作ることを行動目標としてしまったとしたら『モモ』の一節と同じように「遊び」を押し付けることつながってしまうかもしれません。

では、子どもの側から立ち上がる能動性はどのように現れるのでしょうか。『モモ』の後半で主人公のモモは、前ぶれなく敵対する勢力に対抗する勇気を得ます。モモの立ち上がりについて、河合（2020）は、集合的無意識という概念を提起したユング心理学を援用し「主体的な意志」と「物事がおのずからそうなること」が合致

する時を迎えたからだと説明しています。授業もこれに似ています。主体的な意志とその場の空気がぴたりと重なって教師の予想を超えた高みに届く時が訪れることがあります。子どもたちの深層にある意識が現れて空気をつくるのかもしれません。

『モモ』では「遊び」を強いられた子どもが夢の中でぺっちゃんこなカードに姿を変えられて、正しい手本どおりに小さい穴がいっぱいあけられて並べられる場面があります。今日の小学校では「キャリア教育」「外国語教育」「道徳教育」「プログラミング教育」など、次々に新しい教育の推進が迫られ教師の多忙化が続いています。教師が子どもの思いや願いに寄り添い子どもの能動性が立ち上がるまで待つ余裕を持つことは、個性豊かで人間的な厚みを持った子どもを育成するために重要なのです。

## 2　水平的カリキュラム設計

カリキュラム設計では、スパイラルと言う言葉をよく耳にします。いくつかの学習経験の後、前に学習した地点に戻ると、多角的な視点を得ているのでそれまで気付かなかったことに気づけて認知的に高いレベルに至るという考えに基づいています。一方、米国カリフォルニア大学バークレー校で開発された理科のカリキュラムである Full Option Science System( 以下：FOSS) では、子どもの認知レベルが急激に上がらないように設計されています。米国から教材を取り寄せて研究したところ、次の 2 つの知見が得られました。

・自由試行に始まる

・複数の教材が機能し水平的繰り返しがみられる

では、それぞれを見ていきましょう。まず、自由試行とは、米国のホーキンスによって提唱されたメッシング・アバウトのことで、子どもに物を与え、指示も質問もしないで、自由に組み立てたり、調べたり、吟味させたり、実験させたりすることです。FOSS では、単元の導入時に教材を渡し、子どもに自由に操作をさせて発見させ、子どもの意見をもとに知っていることと知りたいことを明確化していくことを目的として行われます。子どもの能動性を引き出す優れた教授 - 学習方法で、わが国でも自由試行に関する実践報告は多数見られるのですが、今日の理科授業ではあまり見ることはありません。

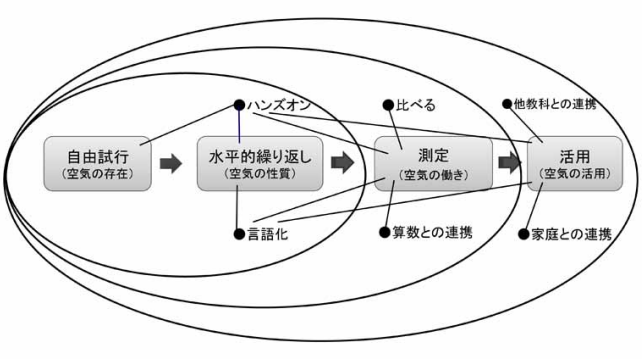



図 1　「大気」の学習における FOSS の水平的カリキュラム設計の構成（白數 ,2017）

次に水平的繰り返しについてです。FOSS「空気」のカリキュラムには図 1 のような構造がありました。自由試行で始まった学習から、パラシュートや注射器、風船など 5 種類以上の教材が登場します。我が国の空気の学習では空気鉄砲、注射器等 2 種類程度の教材が使われるのと比べると圧倒的な多さです。特に配慮されているのは、教師が科学的概念を教え込むことのないように、子どもが友達と一緒に教材を操作することで自然に科学的概念を獲得できるような機会を散りばめているところです。子どもの認識がスパイラルに上がって高まっていくモデルと異なり、水平に移行する学習では、ハンズオン活動と話し合いによる言語化を繰り返す過程で、元の経験が他の知識や経験と結びつきながら自然認識を拡張させていくようにカリキュラムが構成されています。

## 3　水平的次元を含む拡張による学習

社会的構成主義の礎を築いたヴィゴツキー（1956）は、りんご農家が果樹園の状態を明らかにしようとするとき、成熟したりんごの木だけでそれを評価するのではなく、これから成熟を迎えるりんごの木を見ることを例に挙げ、「現下の発達水準」に対して、より有能な人との協働によって行う問題解決を通して決定される水準を「潜在的な発達水準」と呼びました。そして、両者の間にある距離を「発達の最近接領域（以下：ZPD)」と名付けました。ZPD を理科教育に援用すると「現下の発達水準」は「生活知：生活的概念」、「明日の発達水準」

は「学校知:科学的概念」と置き換えることができます。ZPD における発達の解釈についついてエンゲストローム（1999）は、「発達は、個人的な転換にとどまるのではなく、集団的な転換と見なされるべきである。発達は、レベルを垂直的に超えていくことにとどまるのではなく、境界を水平的に横切っていくことであると見なされるべきである。」と述べ、「拡張による学習」を ZPD を横切る旅として特徴づけました。「拡張による学習」とは、学び手が「いまだここにない何か」について、直面する困難や葛藤を引き起こしている矛盾に迫りながら与えられた課題を超え、自ら問題を発見・創造していく学びのことです（山住 ,2017）。このことを踏まえて ZPD の場における学びの流れをモデル化したのが図 2 です。

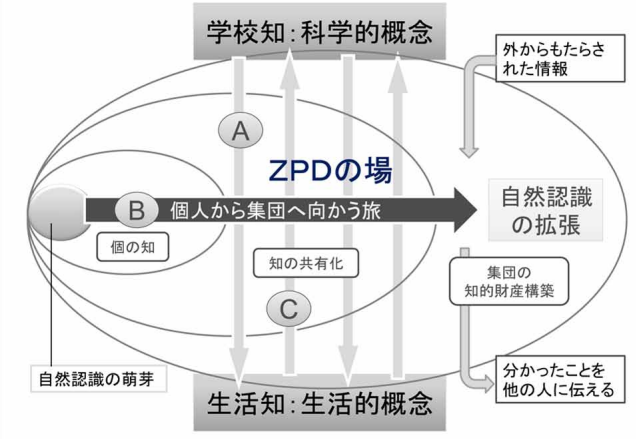



図 2　ZPD の場における垂直的相互作用と水平的相互作用　（白數 ,2017 を改変）

自然認識の萌芽は、ZPD の場において個人から集団へ向かう旅によって、個の知、知の共有化を経て集団の知的財産の構築に至ります。子どもと教師の相互作用は、生活的概念と科学的概念の往復運動として表されますが、やがて両者は自然認識の拡張によって結び付き概念構築がなされます。その際、科学読み物や専門家の助言などの外からもたらされる情報を取り入れたり、子どもが分かったことを他の人に伝えるために思考を整理したりすることによって概念構築は促進します。このような授業を創るのは簡単なことではありませんが、上越教育大附属小学校の実践から手掛かりを得ることができます（風間 ,2017）。昆虫の観察を通しての気づいたことを話し合っていく過程（図 2 Ⓐ）で昆虫のすごさに気づいた子どもが、昆虫のように「翅を作って飛びたい」という願いを持つようになります。そんな子どもの願いをかなえるため、教師はプラダンボールで子どもに大きな翅を作る機会を与えます（図 2 Ⓑ）。自分が昆虫に成り代わることによって翅を動かすためにはものすごく力がいることが分かり（図 2 Ⓒ）、昆虫の筋力がいかにすごいか実感を伴った理解に至っています。このような授業の創造は教師に柔軟性が無くてはできません。なぜなら顕在的カリキュラムが授業の脱線を阻止するからです。顕在的カリキュラムとは学術的な系統性を重視したカリキュラムを指し、これに対して子どもの論理に基づいた使用価値の高いカリキュラムを潜在的カリキュラムと言います。学校の教育は二重性のあるカリキュラムの狭間で試されているので、子どもの心に寄り添おうとする教師ほど苦悩するのです。経験を積むと過去に上手くいった授業の記憶が強化され自信をもって授業に臨むことができるようになる一方で、自身の想像を超えた先の見えない探究的な授業に足を踏み入れにくくなります。ここではどちらが優れているか論じようと言うのではありません。このようなカリキュラムの特性を受け入れ、このダブルバインドから抜け出そうとするエネルギーを原動力とすることで教育に新たな可能性が見えてくるのです。

## 4　ティンカリング

ここでもう一度自由試行について考えてみましょう。自由試行は顕在的カリキュラムと潜在的カリキュラムの狭間を埋めてくれるかもしれないからです。自由試行は STEM 教育（Science, Technology, Engineering , Mathematics を教科横断的に学ばせようとする教育）におけるエンジニアリングや、そこで必要とされるティンカリングに見ることができます。わが国の学校教育では STEM 教育の実施は始まったばかりですが、民間では習い事として急速な広がりを見せています。たとえば、ブロックやモーターでロボットを組み立ててパソコンソフトで動かす体験をさせる教室です。このような教室の多くは決められたシラバスに沿って知識と技能を習得させ

るので自由試行は限定的にしか起きません。ところが、埼玉大学 STEM 教育研究センターや世田谷ハツメイカー研究所（アザイコミュニケーションズ）の取り組みは注目に値します。子どものやりたいことに応じてカリキュラムをカスタマイズしています。両者に共通するのは次の 4 点です。

・少人数で実施している
・子どものモチベーションを重視している
・指導者による指導が過干渉にならないようにしている
・時間内に何をどこまでやるかとゴールは子どもが決める

この二つの教室は、モチベーション 3.0 とティンカリングの考えを基盤としています。ピンク（2010）は、モチベーション 3.0 における人の行動は、「活動によって得られる外的な報酬よりも、むしろ活動そのものから生じる満足感と結びついている。」と述べています。そして、モチベーション 3.0 の要素として「自律性」「熟達」「目的」を挙げています。個人の意思決定が尊重され、自分の成長が実感でき、自分の行動に重要な目的があることを自覚した時に人は活動に没頭するのです。この考え方を教育に援用すると、子どもが自ら学びたいと思うことを自ら好む方法で学び、学ぶ目的が明確で自分の成長を実感できる時に子どもは学習に没頭するのだと言えます。したがって、成績を上げることを主な目的とした学習では子どもの学習に対する意欲を持続させるのは極めて困難だと言えるのです。

また、ティンカリングとは、大昔から人が物を作るために行ってきた創造性を生み出すアプローチで、思いつくままにあれこれ工夫を重ねて改造を行っていくことです（レズニック ,2018）。ティンカリングを学校教育に取り入れると、先に述べた自由試行に近づくと考えられます。学校教育では、時間、人数、予算等の様々な制約があり、ティンカリングの要素を取り入れることは困難です。STEM 教育を推進しようとしたときエンジニアリング的な学びにおいてティンカリングのアプローチは欠かせません。伝統的な学習が、材料の機能を一つ一つ学び組み立てて作品を完成させるようなことだとすると、エンジニアリングでは、すでにある作品を部分的に分解して機能を探り、新たな部品を追加したり構造を変えたりして改良するようなことだと考えられます。エンジニアリングではいじくり回しているうちに偶然発見することも多いのです。その意味では教師は子どもが能動的に学ぶ図 2 の矢印Ⓑとⓒのアプローチの視点を持つことが重要であると言えます。

## 5　教室外の世界とつながること

総合的な学習の時間で「日本人の技術力の高さ」を調べていたクラスの学級担任にインタビューをしたことがあります。子どもたちと研究を深めていくと「日本では他国の技術を真似したものが多く日本発祥の技術と言えるものが見つからない」という解決の見えない状況に陥ったそうです。その中である子どもが、ロケットの先端を手仕事で製造する町工場があることを調べて来ました。そして、その町工場に見学に行き技術者の仕事ぶりと技術力の高さ、工場が目指している願いに触れることで解決に至ったそうです。この事例から、教師に答えられない「問い」は「拡張による学習」への動力になり得ることが分かります。

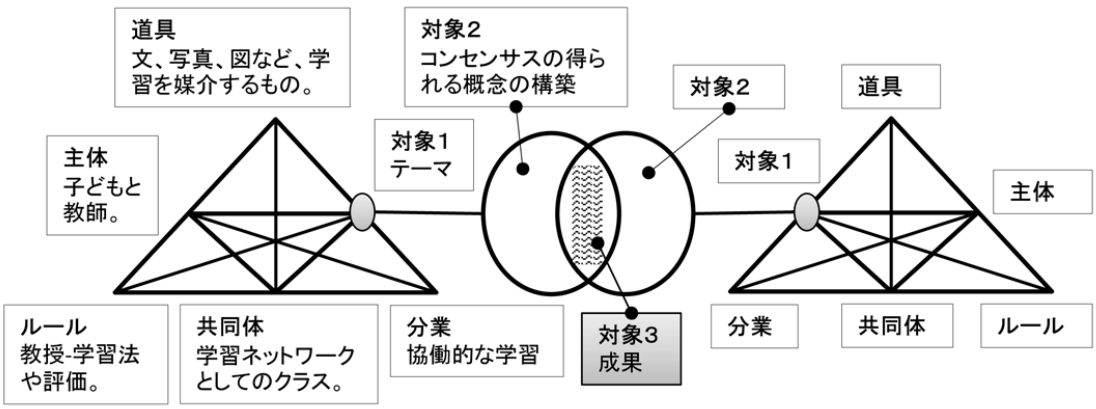



図 3　第 3 世代活動理論を援用した拡張による学習のモデル

「拡張による学習」では、第三世代活動理論のモデルが参考になります。

活動理論では授業は道具を媒介として学級のあるルール（教授 - 学習方法や評価）と分業（協働学習）に基づいて、主体である子どもと教師が学習対象である「テーマ」に向かっていく営みとして解釈できます。「日本人の技術力」の例で示したように、図 3 の対象 1 であるテーマに内的矛盾がある場合は、答えを探したいという情動によって子どもの能動性は高まるでしょう。通常は教師が答えを知っているわけですが、そうでない場合、子どもも教師も外部に助けを求めるしか方法はなくなり、教室の外にある共同体へと意識を向けていきます。このことについて山住（2006）は、「二つの活動システムが対象 1 から「対話」によって対象 2 へと拡張する。拡張を通して、双方の対象は近づき部分的に重なり合うことになる。この境界を越えた対象の「交換」において、新しい対象 3 が立ち現われてくる。そして、このような「第三の対象」は、新たな「変革の種子」（seed of transformation）を生み出していく」と述べています。第三世代活動理論の教育への援用は、これまで教師－子どもという二元論で捉えられがちだった教授－学習構造に新たな視点をもたらします。次世代の学習は、教室が他のコミュニティと結びついたり離れたりしながら子どもの潜在的カリキュラムを取り込んで進むでしょう。そして、子どもたちの中に変革の種子が生まれていくことでしょう。

（この論考は、白數哲久（2021）『教育創造』上越教育大学附属小学校高田教育研究会 ,Vol.194,10-17. に加筆修正したものです。）

## 引用・参考文献


河合俊雄（2020）『NHK100 分 de 名著 2020 年 8 月』NHK 出版 .

白數哲久（2017）『児童の科学的概念の構造と構成―ヴィゴツキー理論の理科教育への援用』福村出版 .

ヴィゴツキー ,L.S（1956）柴田義松訳（2001）『思考と言語』新読書社 .

エンゲストローム ,Y.（1999）, 山住勝弘ほか訳『拡張による学習―活動理論からのアプローチ』新曜社 .

山住勝広（2017）『拡張する学校 協働学習の活動理論』東京大学出版会 .

風間寛之（2017）「昆虫とつながり、自らの世界をつくりかえる子ども」『教育創造』上越教育大学附属小学校高田教育研究会 ,Vol.187,22-23.

ピンク ,D.(2010) 大前研一訳（2010）『モチベーション 3.0』講談社 .

レズニック ,M(2017) 酒匂寛訳（2018）『ライフロング・キンダーガーテン　創造的思考力を育む 4 つの原則』日経 BP 社 .

山住勝広（2006）『活動理論と教育実践の創造―拡張的学習へ―( 第 3 刷 )』関西大学出版部 .

# 風実験工作

NPO 法人ガリレオ工房　吉澤 弘

卓上扇風機を作るこの工作は、できるだけ身近な材料と道具で作れるようにデザインしました。ケント紙に印刷していただけると丈夫で良いかと思います。子どもが型紙に書いてある説明を読み、写真を参考に組み立てていくと、細かい説明をしなくてもおおよそ作れるようになっています。なお、作り方の手順書が詳しすぎると子どもの自由な発想力を発揮しにくくなると考え、あえて大まかに書き工夫の余地を残しました。しかし、上手く作れずに悩む子どもがいるかもしれません。その際、次の 3 点を参考にしていただければ幸いです。

・苦労しながら作ることで子どもは達成感を得る。できるだけ見守りたい。
・友達同士で教え合えるような環境があると発想は広がる。
・時間内に全員が同じゴールを目指す必要が無く、家に持ち帰って続きをしてもよい。

試行錯誤する中で、羽根が支柱にぶつからないように位置や角度を調節する必要性や、ちょうどよい穴の大きさにすることについての気づきがあることでしょう。また、支柱とプロペラの組み合わせを変えたり、余った紙で新しいパーツを切り出して貼るかもしれません。近年、子どもが自分でゴールを決め、自由に工作をする機会は少なくなっています。ぜひ、自由な工作の楽しさを子どもに味わってもらえたらと思います。

## ○用意するもの

**モーター**
FA130、配線済みのもの。

**電池ボックスと電池と電線**
2 個直列で使います。

**プロペラ部品**
A4 のケント紙にプリントしてください。
180kg(0.25mm) から 220kg(0.3mm) 程度の厚みがお薦めです。

**マスキングテープ**
ホームセンター等で売られている、黄色のものがお薦めです。100 円ショップのものでも使えますが、強度が不足する事があります。
幅 10mm から 15mm 程度のものをお使いください。

**ひっつき虫**
コクヨのものがお薦めです。配管の穴埋め等に使われる充てん剤 ( 例 : ボンドテープ状コーク : コニシ株式会社 ) を 1cm 角程度に切ったものも使えます。
100 円ショップのものは強度が不足します。

**スナップ**
10mm から 12mm 程度のものをお使いください。

**プッシュピン**

**爪楊枝**
一人 5 本程度使います。

**竹串**
直径 3mm 程度のもの。

**ストロー**
直径 4mm 程度のもの。

**網押さえゴム**
ホームセンターで売られている網戸の網を押さえるゴムです。直径 5.5mm のものをお使いください。

**糸**
ミシン糸よりも太いボタン糸 ~ 細めの刺繍糸がお薦めです。

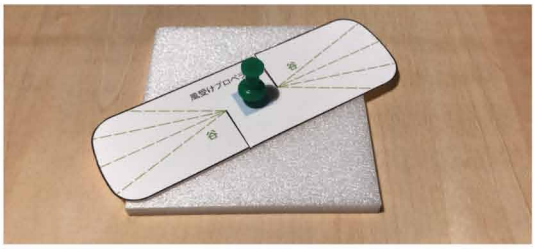
**おもり**
十円玉程度のもの。

**発泡スチロール板**
穴を開けるときの下敷きにします。百円ショップで売られている 5mm から 10mm 厚程度のものがお薦めです。10cm 角程度に切ってお使いください。
怪我を避けるため、穴を広げるときもこの上で穴に竹串を通してから、持ち上げて竹串を深く差し込み、広げるように指導ください。

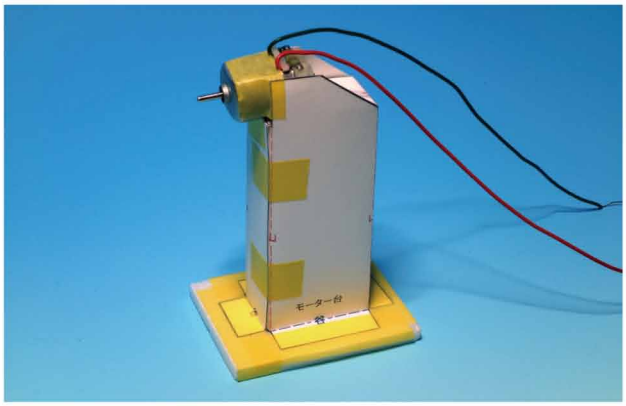
# 部品の組み立て

●使う物　　モーター (FA130)、マスキングテープ、
ひっつき虫、プッシュピン、スナップ、
つまようじ、竹串、細いストロー、

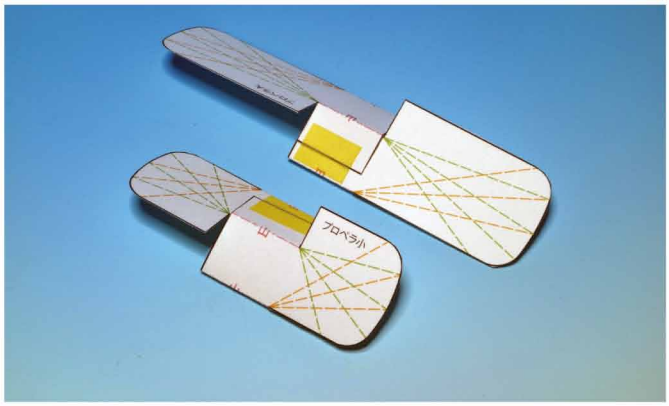
**注意：**穴を開ける時は発泡スチロールの上



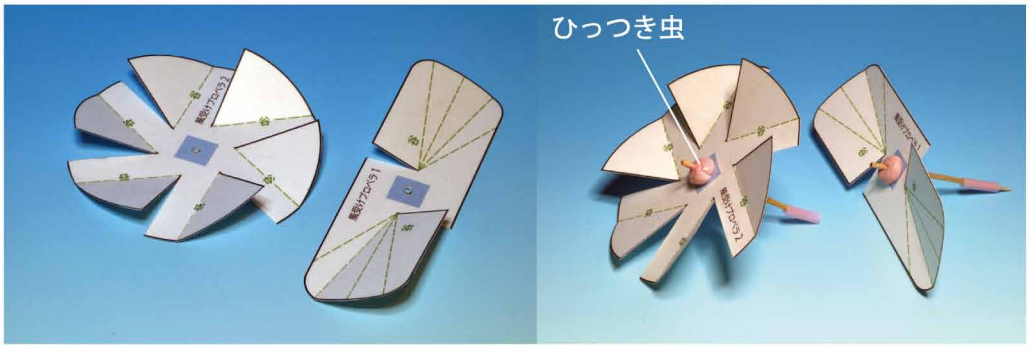
## モーター台



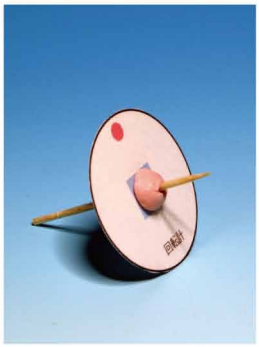
## プロペラ大 / 小



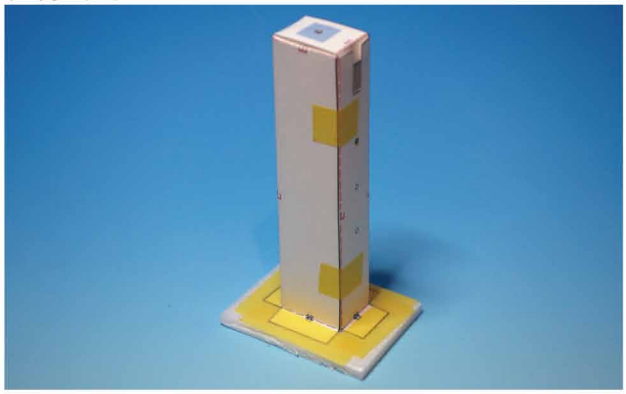

## 風受けプロペラ 1 / 2




## 回転計

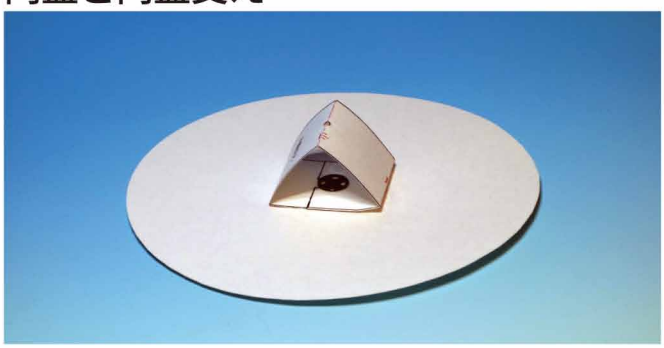


## 支柱大

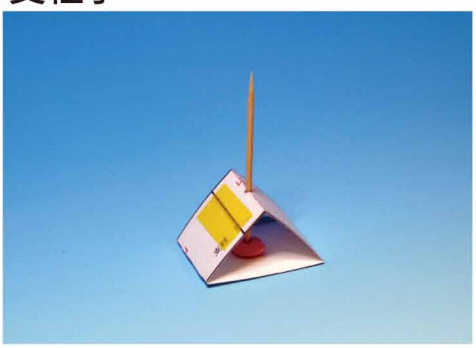


## 腕

## 円盤と円盤支え



## 支柱小

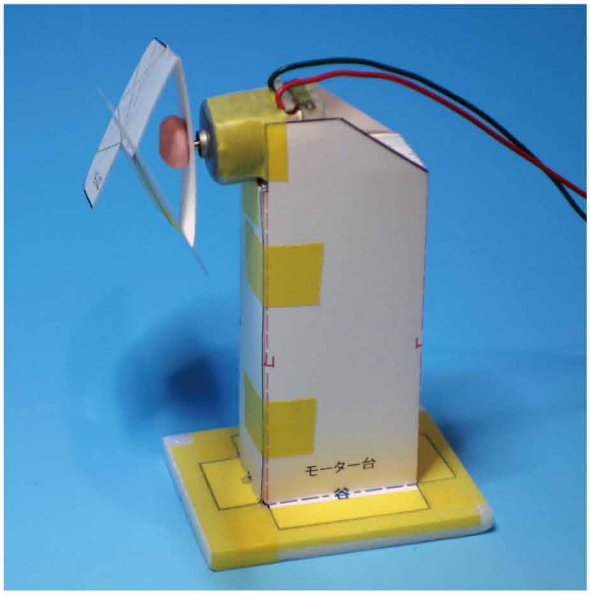


# 実験装置の組み立て

●使う物　　網押さえゴム (4.5mm)
糸、 ひっつき虫、 つまようじ
竹串、 おもり (10 円玉)

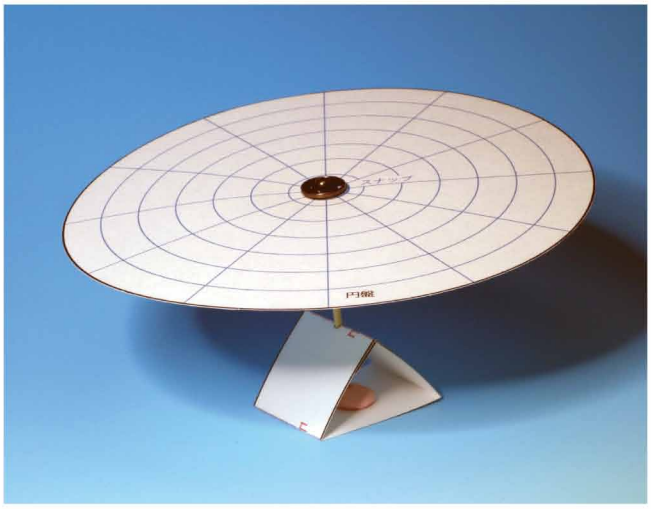
## 1 風送りプロペラ

モーターに、プロペラ大か小をつける。



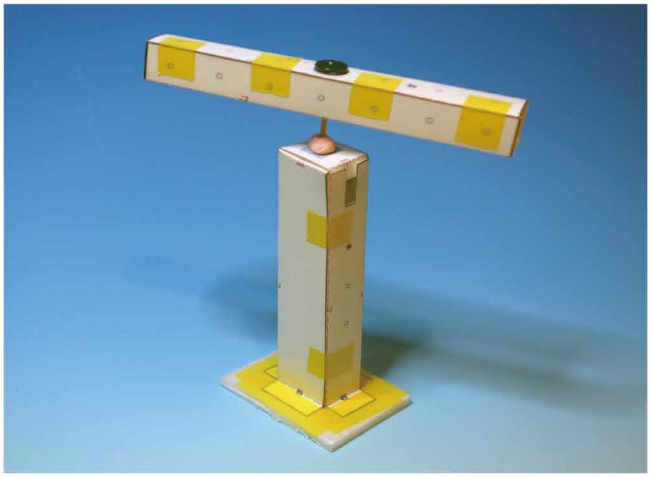

## 2 回転ステージ

支柱小に、円盤を乗せる。
円盤に、風を受ける物をテープで貼りつける。



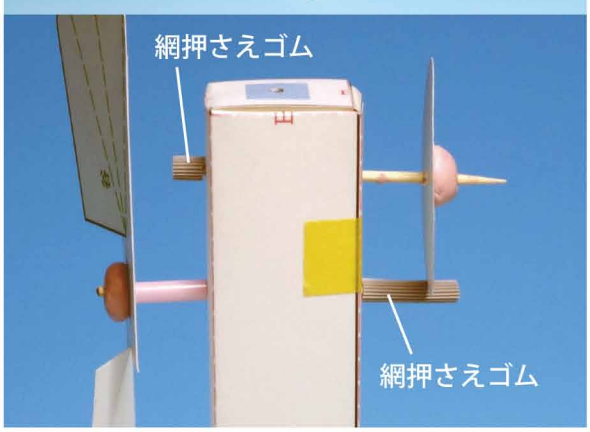

## 3 回転アーム

支柱大に、腕を乗せる。
腕に、風を受ける物をつり下げる。



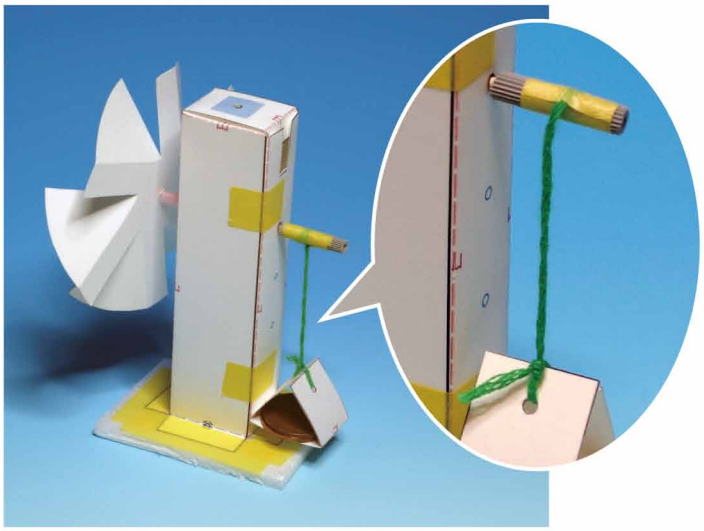
## 4 スピードメーター

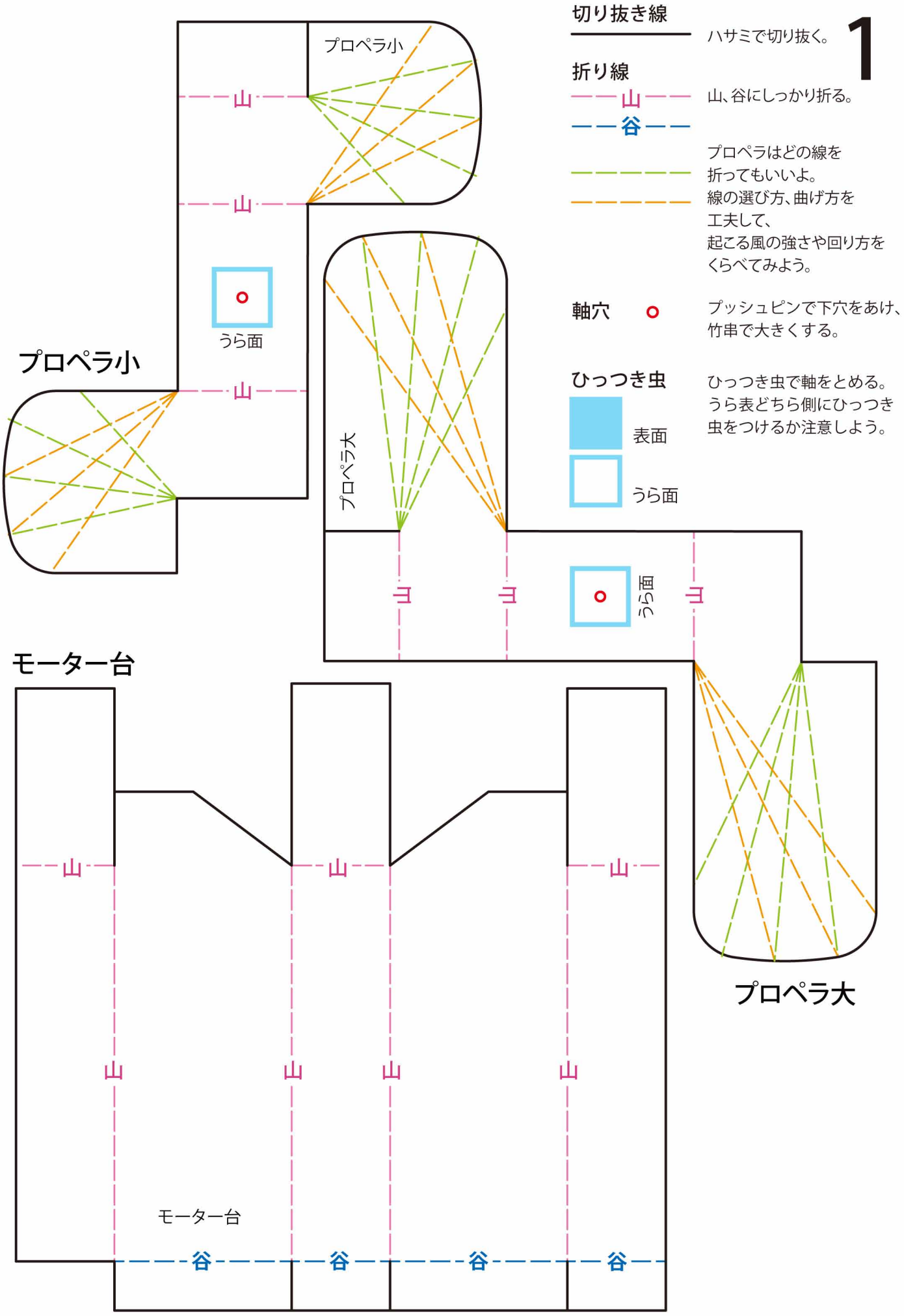
支柱大に、風受けプロペラと回転計をつける。
回転軸が入る穴は竹串で少し大きくしよう。



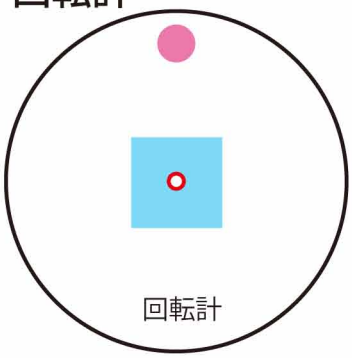

## 5 パワーメーター

支柱大に、風受けプロペラをつける。
重り受けを糸でつりさげ、重りを乗せる。

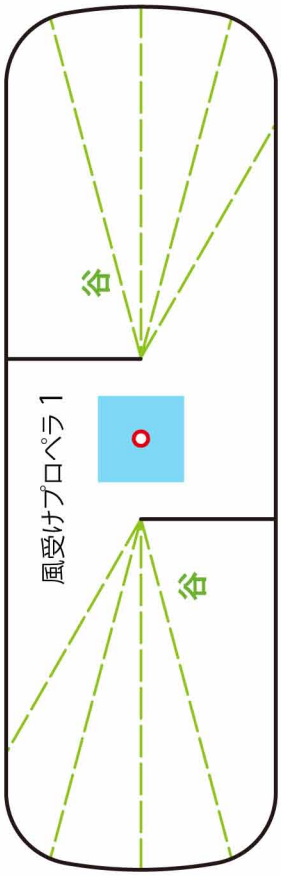



切り抜き線
ハサミで切り抜く。
1
折り線
山
谷
山、谷にしっかり折る。
プロペラはどの線を
折ってもいいよ。
線の選び方、曲げ方を
工夫して、
起こる風の強さや回り方を
くらべてみよう。
軸穴
プッシュピンで下穴をあけ、
竹串で大きくする。
ひっつき虫
ひっつき虫で軸をとめる。
うら表どちら側にひっつき
虫をつけるか注意しよう。
表面
うら面
プロペラ小
山
うら面
プロペラ小
山
プロペラ大
山
うら面
プロペラ大
モーター台
山
山
山
山
山
山
山
山
モーター台
谷
谷
谷
谷

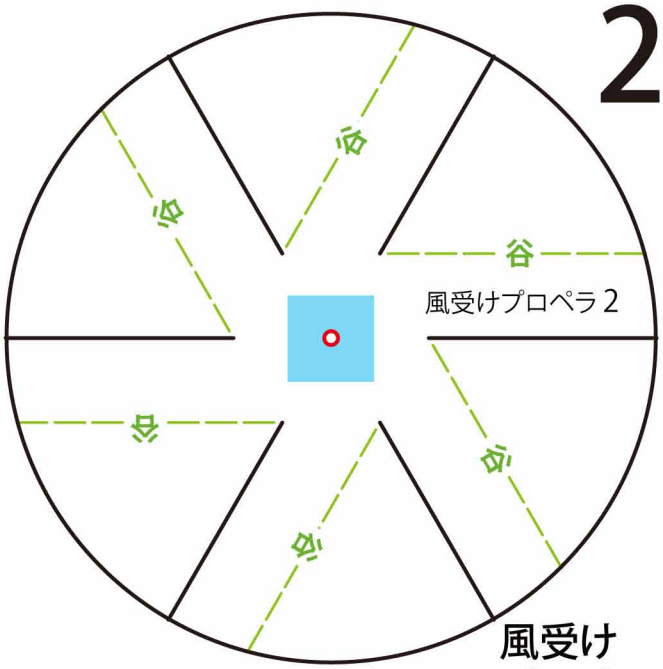
## 回転計




## 風受けプロペラ1



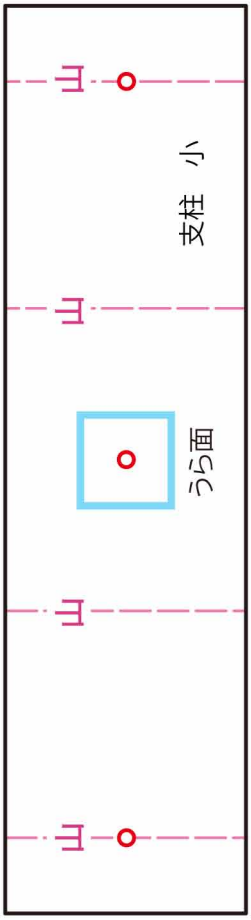

2



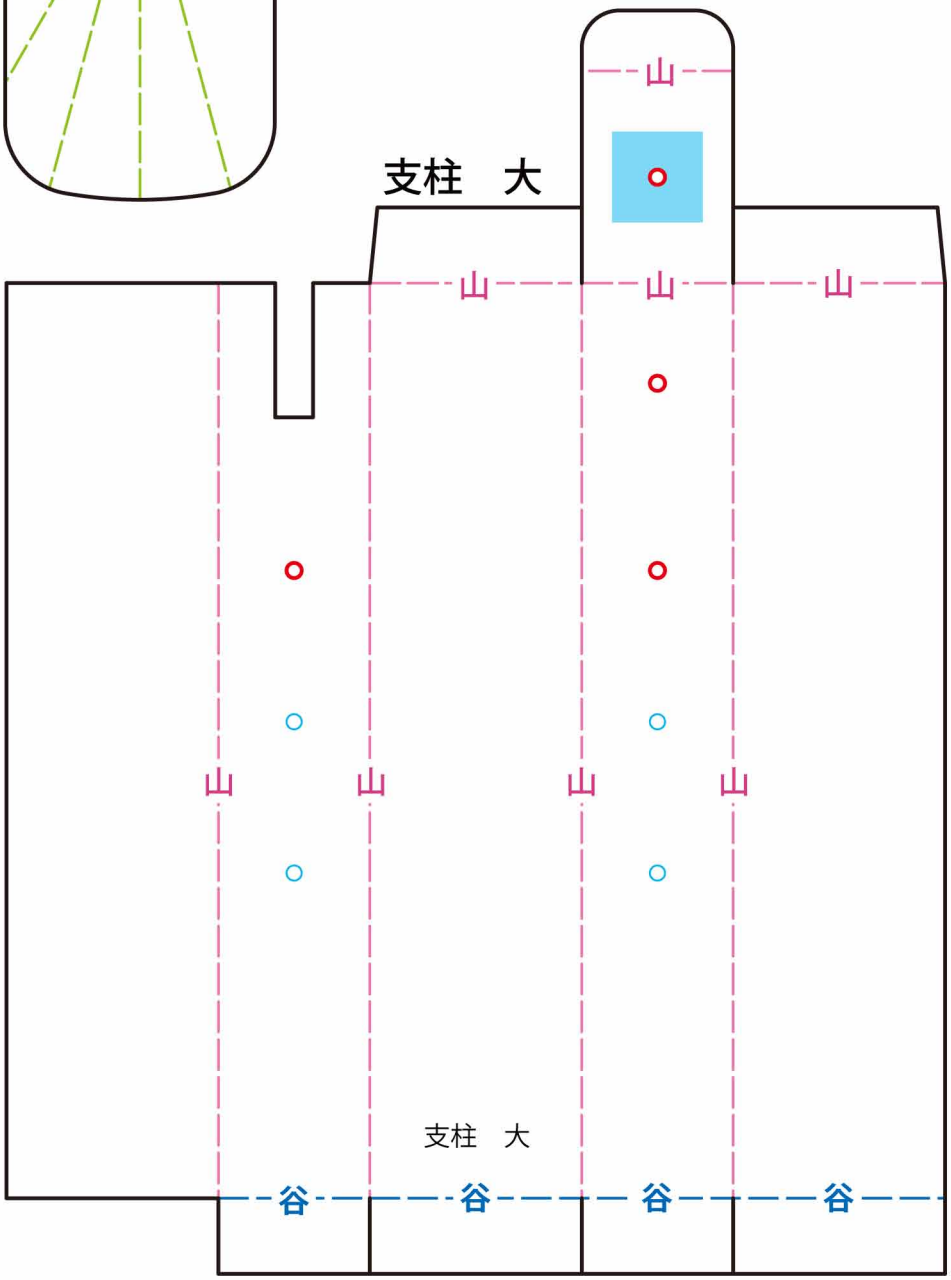

風受け
プロペラ2

## 重り受け

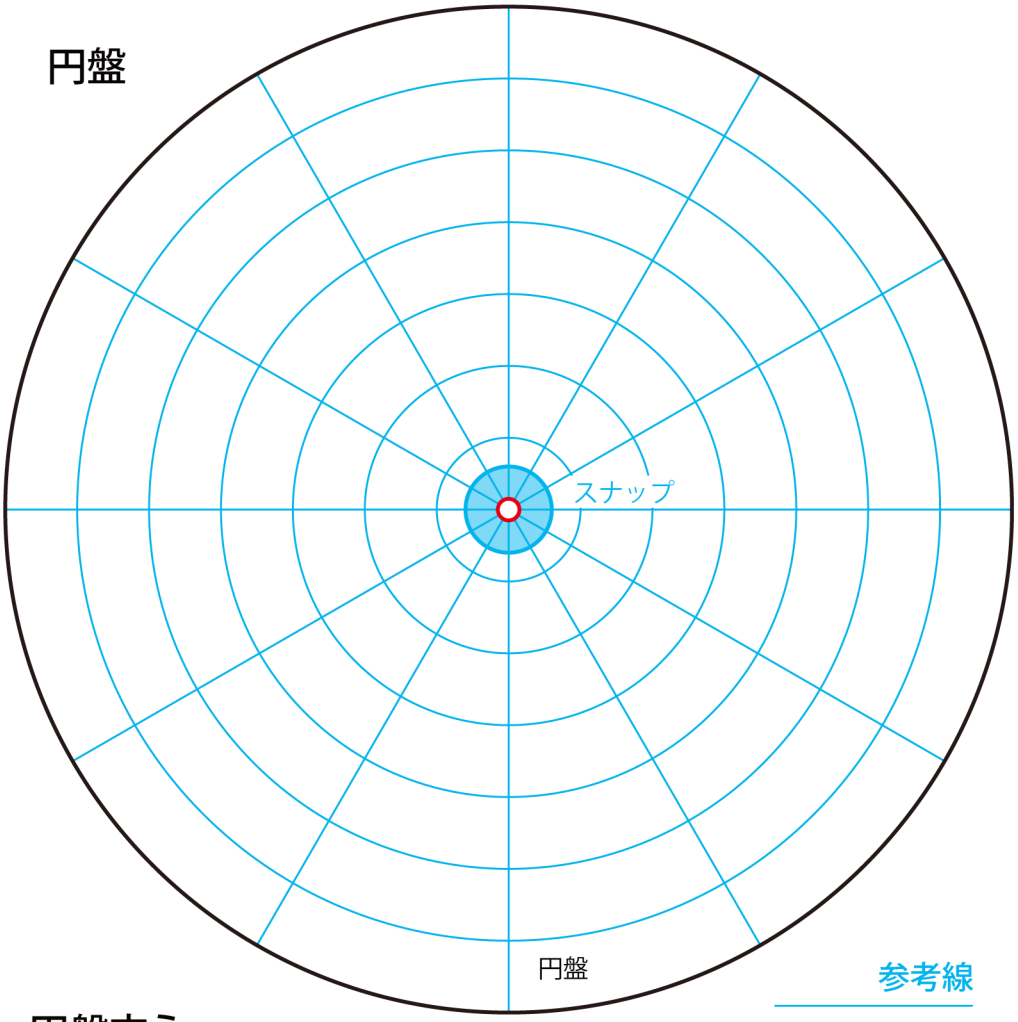

## 支柱　小




## 支柱　大

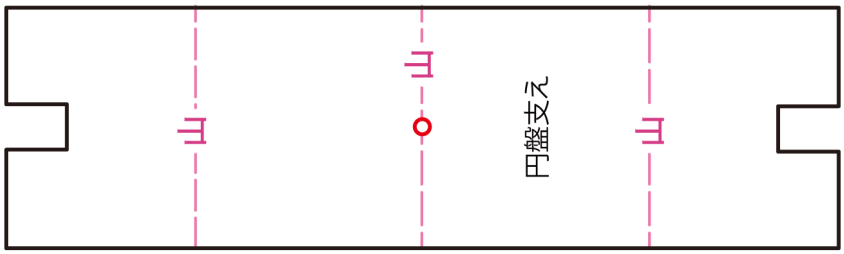


## 円盤

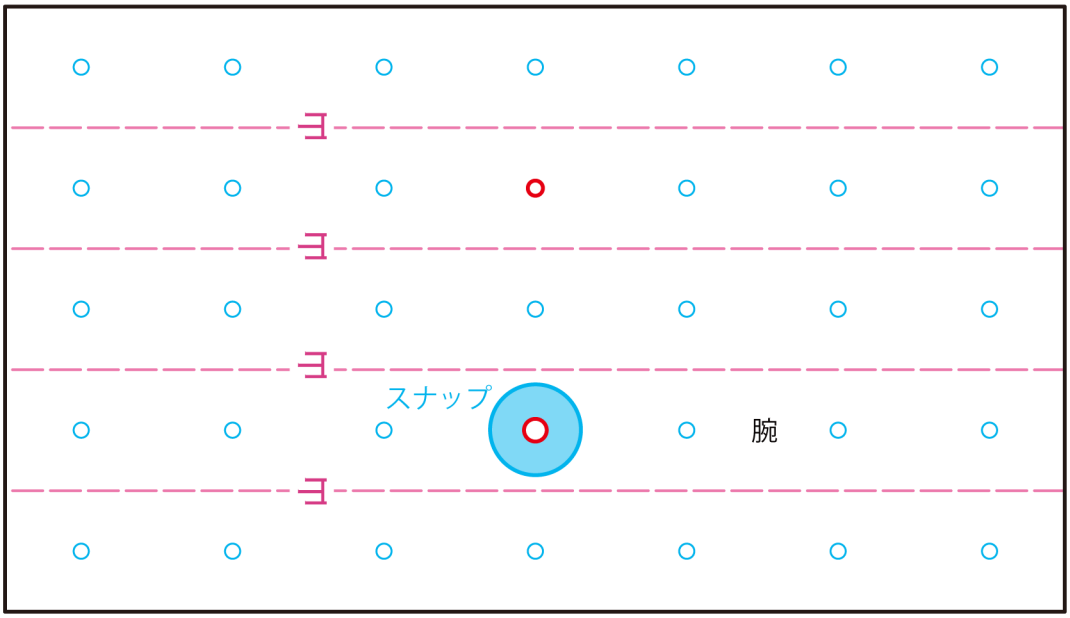



参考線

## 円盤支え



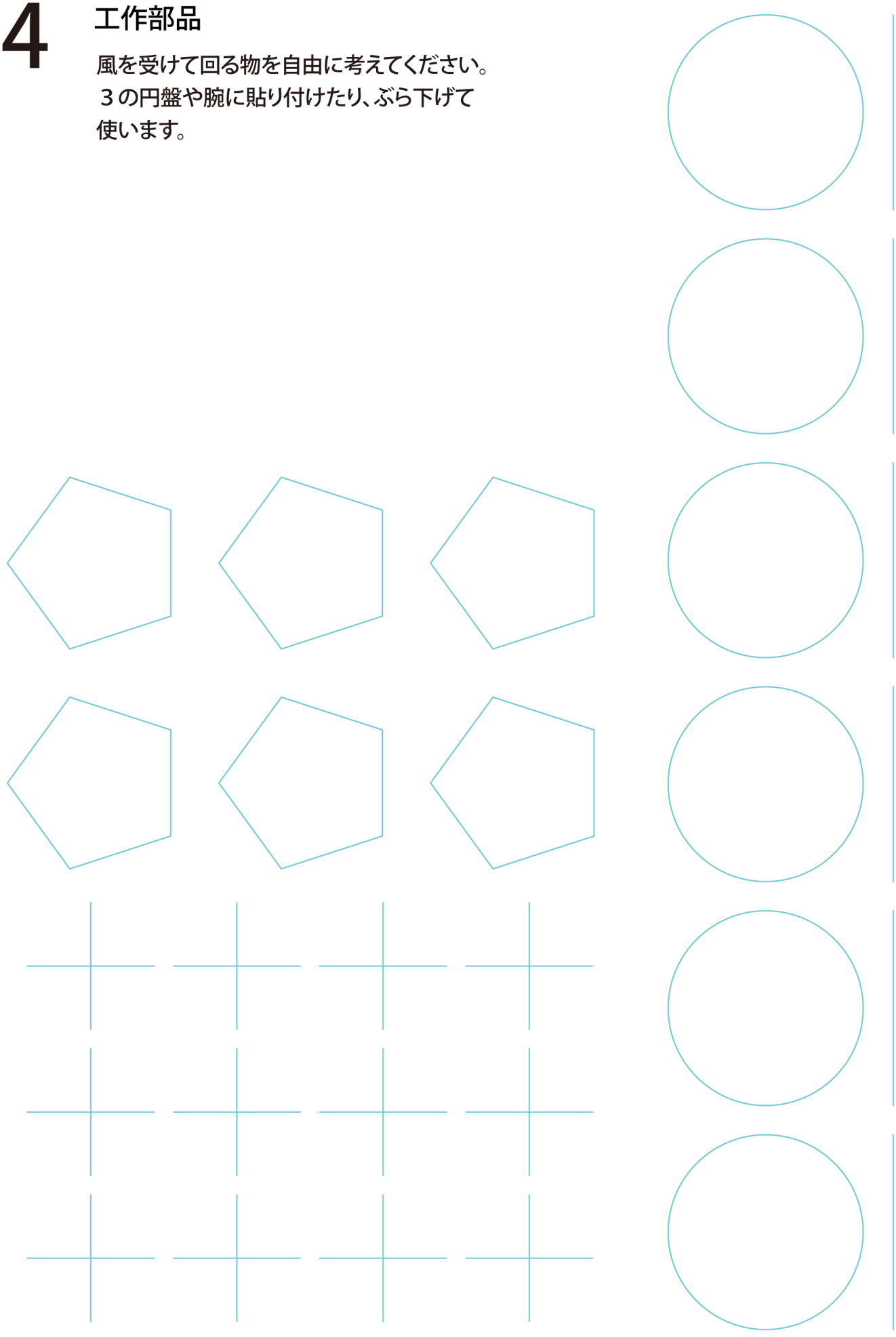

## 腕




工作部品

# 4 工作部品

風を受けて回る物を自由に考えてください。
３の円盤や腕に貼り付けたり、ぶら下げて
使います。



# プログラミングを学ぶとは？

昭和女子大学現代教育研究所研究員　久木田 寛直

「プログラミング」は言語である。
人がロボットやコンピューターと「コミュニケーション」を取るための「言語」だ。

日常生活に於いて「言語とは何か？」の問いを自ら投げかけることは、ほぼないと思う。
日本語を母国語とする親の元に生まれ、気が付いた時には日本語を話すようになっている子供には、それついて深く考える機会はあまりないだろう。
日本の文化や教えに触れ、気付かぬうちに、言葉の使い方に馴染んでいく。

母国語を扱うのと同じように、ロボットプログラミング（言語）を扱っていくことで、言葉の使い方の特性を深く知ることができる。

「言語」は本来、意見や事実を他者に伝える為に存在する。
使い方に関して言えば、その言葉を伝える伝達方法はいくつかあるが、情報ごとに、その方法（メディア）を使い分けるとより伝わりやすくなる。

「気持ち」を伝える時には、アナログのメディアを通した方が伝わりやすい。
「考え」を伝える時には、デジタルのメディアを通した方が伝わりやすい。

さらに、言語には２種類の目線があると言われている。俯瞰して見る西洋の「神様目線」と、自分の目が主体となる東洋の「虫目線」。

コンピューターを操作する言語の多くは、西洋の考え方で作られている。
なので「神様目線」という概念を理解することによって、プログラミングがより理解しやすくなる。

ちなみに日本語は「虫目線」の言語と言われている。自分自身が虫のように動くので、主語がなくてもその場の解釈で理解ができる。

プログラミング言語は、「考え」を伝える情報処理である。「考え」を伝える時には、デジタルのメディアを通した方が伝わりやすいが、「気持ち」を伝えることはないのだろうか？

AI が自我（気持ち）を持つことが有るか否か。この問いは、AI 研究に於いて、よく出てくるテーマである。
ロボットと人間の異なる点の一つに、動作を行う「きっかけ（イベント）」が挙げられる。
プログラミング言語では「きっかけ」は主に「イベント駆動型プログラミング」で実行される。つまりロボットの「イベント（きっかけ）」には、まず「設定」することが不可欠なのだ。

人間の「きっかけ」は、自らの「気持ち」や「身体の反応」から起きる。

「身体の反応」による「きっかけ」は、ロボットのセンサー

からの「きっかけ」によく似ているが、「気持ち」によるきっかけは、ロボットには存在しない。

そこに目を向ける行為は、「デザイン思考」のユーザーの行動を可視化する発想に近い。

小学生段階では、親や先生など、大人の躾により「やらされている」と感じている子が多い。その躾の奥には、親や先生たち大人の「気持ち」が隠れている。
その隠れた「気持ち」に気づくことができたなら、物事の本質を理解するようになる。

これから答えのない時代（VUCA）を生きていく上で必要なのは、自分自身の「考え」や「気持ち」から行動している＜本当のことをしている＞ところから本来の学びを行い、物事の本質に気付く力を身に着けることである。

遊びやものづくりという＜本当のこと＞から自分の身体で考え感じ、言葉を生み出し整理していく感覚。
日本語の特徴と、英語やプログラミング言語の特徴を理解し、扱うことができた時、はじめて、様々なものの奥に隠れている「気持ち」を理解していけるようになるだろう。

本来「情報」という教育は、「何かを知る」ということだけでなく、社会との関係性や感性を育てていくことである。

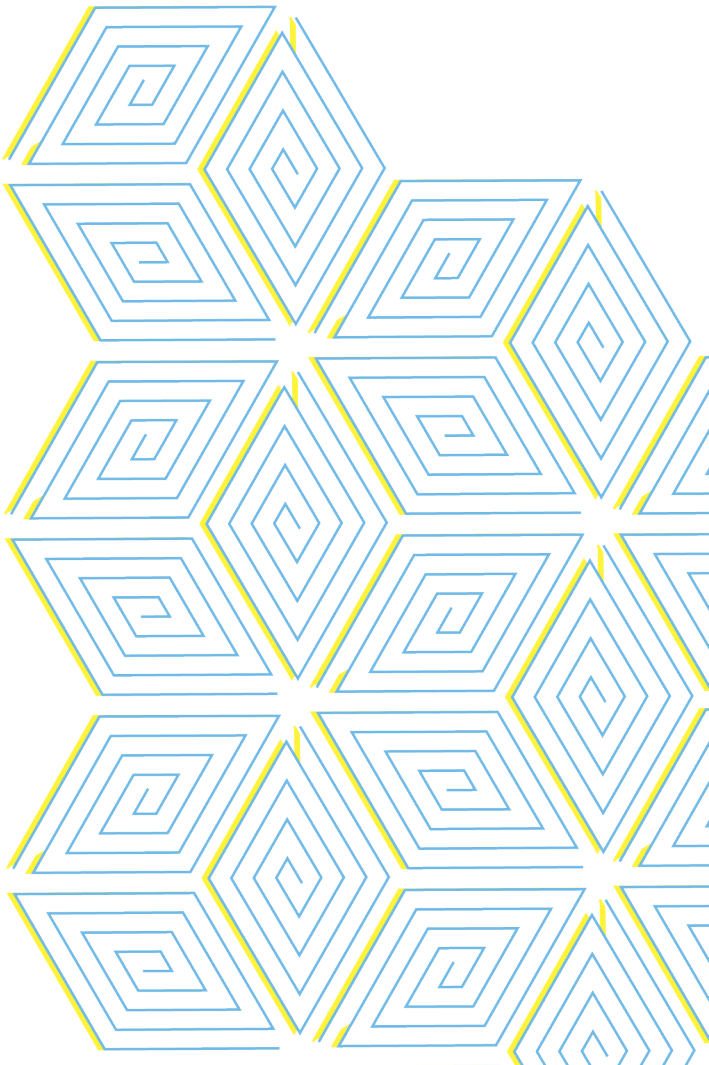
プログラミング教育、情報教育は、常に子供達によって進化し、先生自身も進化していく。様々なメディアに触れ、ものづくりを通し学んでいくことが、「何を行なっているのか」に気付くことができる情報処理能力を身に着けることに繋がるのだ。

伝えたいのは、『情報（考え）』なのか『情緒（気持ち）』なのか。

それらは全て「コミュニケーション」なのである。

参考文献


・「英語にも主語はなかった」 金谷武洋　モントリオール大学　講談社
・「木を見る西洋人　森を見る東洋人」 リチャード E ニスベット　ダイヤモンド社
・「にほんご」 谷川俊太郎　他　福音館書店
・「ことばと思考」 今井むつみ
・「アラン・ケイ」 アラン・ケイ　ASCII BOOKS
・「作ることで学ぶ」 シルビア・リボウ・マルティネス、ゲイリー・ステージャー　O'REILLY
・「AI に心は宿るのか」 松原仁　集英社 インターナショナル
・「かたちのデータファイル」 高橋研究室 東京大学 建築学科（高橋鷹志）彰国社刊
・「Sushi Science and Hamburger Science」 TatsuoMOTOKAWA（1989）

## **プログラム** とは …… コンピューターとの **コミュニケーション**

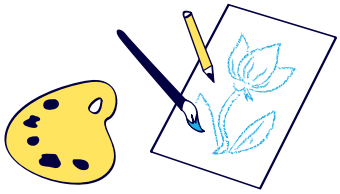

※コミュニケーション＝意思疎通

プログラムを組む事を「プログラミング」、プログラムを組む人の事を「プログラマー」といいます。

### ★ 「ことば」とコミュニケーション


わたしたちは、他人とコミュニケーションを図る時に言葉を使います。それ以外にも、表情や動作でコミュニケーションを図る時もあります。

**みんなが大好きな音楽や絵もコミュニケーションのひとつです！**



日本人に何かを伝えたい時は、「日本語」を使います。他の国の人とコミュニケーションを図りたいときは、その人の話す言葉を知る必要があります。世界共通語の一つは英語です。世界中の人とコミュニケーションを取りたければ、英語を話せるようになりましょう!

やってみよう

## 言葉について考えてみましょう！

### ★ コンピューターとことば

コンピューターに指示を出すプログラムにも色々な種類の言語があります。実際のプログラミング言語は、テキスト(文字)を書きます。多くのプログラミング言語は「英語」と「数字」で書きます。

例えば、JavaScriptと言うプログラミング言語で画面に「こんにちは」と表示したいときは、下のように書きます。

```
document.write("こんにちは");
```

これがプログラムを書くこと「**プログラミング**」です。

プログラムは **順番が大切** です！

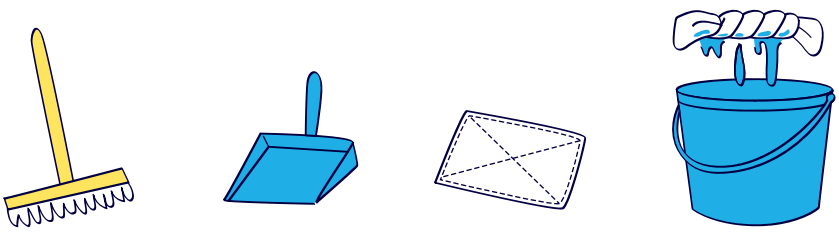
コンピューターはまじめで単純な事しかわかりません。単純な作業を丁寧に、正しい順番で命令してください。

### ★ プログラムの仕組み

わたしたちは、普段プログラムと同じ仕組みで動いています。

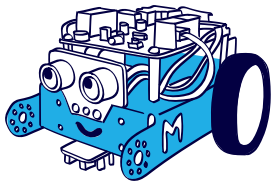
**学校のそうじの時間を思い出してみましょう！**

どんな動作が含まれますか？



その動作には必ず **きっかけ** がありませんか？

その他の場面も思い出してみましょう。
（たとえば…買い物、漢字の練習、水泳、朝学校へ出かける準備など）



きっかけから動作を実行する命令を、プログラムでは **イベント** といいます。

きっかけ

動作

動作

きっかけ

動作

きっかけ

動作

きっかけ

動作

きっかけ

動作

きっかけ

動作

きっかけ

動作

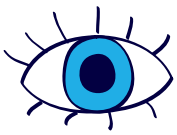
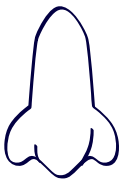
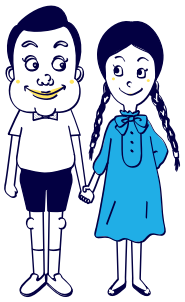
きっかけ

動作

きっかけ

きっかけを知るために、わたしたちは何を使っていますか？

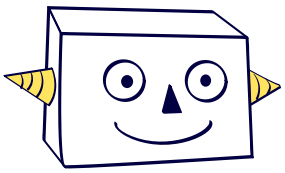
## 五感

| 目 | 鼻 | 耳 | 口 | 体 |
|---|---|---|---|---|
|  |  |  |  |  |
| みる | かぐ | きく | しゃべる<br>たべる | さわる<br>体感する |

これはコンピューターやロボットが持つ「センサー」と同じです。

**身体を通してきっかけを知ること がとても大切！**

人間 ＝ 五感が察知して ▶ 脳が反応を指示 ▶ 体（筋肉）が反応する



ロボット ＝ センサー ▶ コンピューター ▶ アクチュエーター

## 整理してみよう

プログラミングを作った時に「思い通りに動かない」と思うことがあります。それは「指示通りに動いてくれない」ではなく、「指示の仕方のどこかに間違いがある」ということです。コンピューターは間違った指示の通りに動いているだけです。

プログラミングは **「人間が考えている事を整理して、コンピューターに伝える作業」** です。

ちゃんと動かない時は一度深呼吸して、落ち着いて自分の考えている指示を整理してみてください。

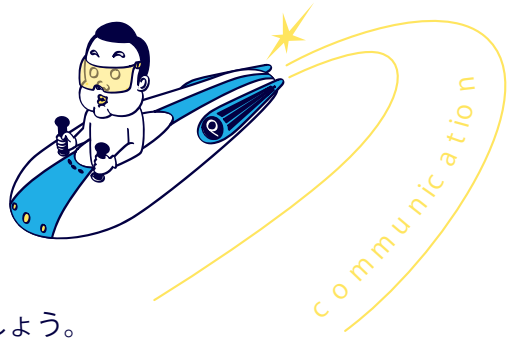
自分の出しているプログラムの指示をお父さんやお母さんなど誰か別の人に話してみましょう。自分の考えを声に出して、他の人に説明した時に、おかしな点が見つかることが多いです。焦らず落ち着いて見直しましょう。

## 大切なこと

プログラムは、人間の言葉のように気持ちを込める必要はありません。しかし、出された指示を処理するのはコンピューターですが、ロボットを見たり触ったりするのは、その先にいるわたしたち人間です。

**「人間とのつながりを意識すること」**
が良いプログラムを作る秘訣です。

**「相手の立場にたって考える」**
それができたら、あなたはそれだけで良いプログラマーになれるでしょう。




普段の生活の中にあるプログラムを見つけてみてください。そして、それはどのような仕組みで動いているか考えてみてください。

そのプログラムを作った人は
**「なぜそれを作ったのか」** **「どのような思いでそれを作ったのか」** 想像してみてください。

いろいろなプログラムがわかるようになれば、物事の大切さがわかると思います。サッカーや料理、楽器の演奏や絵を描くことの魅力もわかるようになるでしょう。

すべては **コミュニケーション** なんだということを覚えておいてください。

# 小学校における STEM 教育に基づく プログラミング教育の進め方

埼玉大学 STEM 教育研究センター / 昭和女子大学現代教育研究所 研究員 星名由美

## 1. はじめに

2020 年 4 月より小学校へのプログラミング教育の導入されたプログラミング教育は、独立した教科ができるのではなく、既存の各教科やクラブ活動などの教育課内で実施するという位置づけであるため、教員の指導計画に任されている部分が大きい。文科省「小学校プログラミン教育の手引（第三版）」（2020）や、文科省・総務省・経済産業省が連携した未来の学びコンソーシアム（2018）は、現場の教員への負担を軽減するためにも、指導体制を充実させるため、学校外部の人材派遣を含めた人的支援体制の必要性が挙げられているが、具体的な体制づくりは現在も不透明である。

プログラミング教育の導入が発表されてから、小学校の教員は各自で情報収集をしたり、研修会へ参加したりと、それぞれの教員も小学校としても、どのように日々のカリキュラムへ導入すべきか検討を重ねていた。しかし学校全体としての取り組みは学校間で格差もある。また、PC でのプログラミングを取り入れた授業においては、複数の教員が担当クラスの調整をして、該当授業の子どもたちの活動の支援をしているのが現状である。

山本ら（2016）は、プログラミング教育の特性と教育の効果と発達段階に応じた指導過程の必要性を指摘し、2019 の研究では、教員研修を通して複数のプログラミング教材の評価と課題について検討し、プログラミング教育を推進するためには、適切な教材確保と指導を支援する事例集、実践する時間の確保などの環境整備の必要性の検討を行った。また、楠見ら（2020）によるプログラミング教育の授業実践に対する調査では、授業実践に対する意欲の阻害要因として、プログラミングや教材に対する知識不足と仕事の多忙さなどが挙げられていた。またプログラミング教育について消極的な教員は 4 割だが、コンピュータ操作に不安がない教員も 4 割という結果になっている。

筆者が所属する埼玉大学 STEM 教育研究センターでは、2002 年より、STEM(Science, Technology, Engineering and Math) 教育の考え方に基づき、ものづくり活動を通した教育方法に関する研究を継続している。研究のアウトリーチ活動として実施している「ロボットと未来研究会」の活動では、教育学部生を中心に複数の大学に所属する「学生リーダー」の研修を行い、子どもたちの指導している。2018 年度より、現役を引退したエンジニアや技術者もサポーターとして参加していたが、子どもたちへの過干渉な指導が見られるという課題が生じ、その課題解決のため、プログラミング的思考力も含めた問題解決・発見力、論理的思考力、自ら考え自ら創り出す力などこれからの社会を生き抜くための 21 世紀型学力を育成するために、主体的に考えさせる指導ができる「シニアリーダー」の育成プログラムの開発と実践を行い、研究の成果を報告している（星名他, 2018，2019a，2019b）。

これまでの子どもたちのものづくりを通した 21 世紀型学力の育成につながる STEM 教育の考え方に基づく指導方法やカリキュラム開発の実践研究の結果を基に、小学校でのプログラミング教育の展開についての研究に取り組んでいる。2019 年度後半、プログラミング教育の研究校である公立小学校から、教員研修の依頼を受け、教員のプログラミング教育に対する不安に関する実態調査と不安を解消するための研修プログラムを実施した。

事前事後アンケートと研修の結果から、小学校における21 世紀型学力の育成に寄与するプログラミング教育を実践する教員の現状とプログラミング教育の展開について検討した（星名・野村，2021）。

本稿では、STEM 教育に基づく主体的な学びとなるプログラミング的思考の育成および小学校におけるプログラミング教育の進め方について考察する。

## 2.STEM 教育に基づくプログラミング教育

### 2-1 STEM 教育とはなにか

STEM(Science, Technology, Engineering and Math) 教育とは、科学、技術、工学、数学の 4 つの領域を統合的かつ総合的に学ぶ教育分野のことで、2000 年代にアメリカで始まった教育モデルである。現在では、STEM に A（Art：芸術、もしくは Arts：リベラルアーツ、教養）を加えた STEAM 教育として提唱することもある。STEM 教育は、実際の経験、ものづくりを通して問題解決力や論理的思考を育てようとするプログラミング教育と近い考え方で、21 世紀型学力の育成としても注目されている。

なにかものづくりを作るときに、工学の分野では「車輪の再発明」、つまり、「すでに確立している技術や解決法を再び一から作ること」は効率が悪いと言われているが、教育の分野では、「車輪の再発明」をしないとわからないことがあり、それが、学習の近道であり、「わかる」ことから次のモチベーションを高めることにつながる。身近なものを題材にして、例えば、「扇風機を知ってる」と子どもたちが言うときに、その「知ってる」とはなにか、仕組みを知っているのか、原理を知っているのか、また、それが作れるかと問を深め、身近なものを作る中で、観察や体験をすることも効果的である。身の回りのものがコンピュータを組み合わせたものづくりになっているのはなぜかを考え、理科の勉強にもつながることを意識させることも大切である。また、ものづくりにおいて、「よりよいものをつくる」感覚を意識させ、「1 回で成功しないもの」として、「たたき台」の感覚を説明し、他の人から意見をもらったときにも、お互いに「よりよいもの」を作るための貴重なアイデアとして、傷つかずに受け入れられるような指導も必要である。「直す」ことから、「よりよいもの」へつながる意識を持ち、トライ＆エラーの大切さと「よりよくするため」のゴールを意識させることで、よりよい問題解決につながる。

### 2-2 主体的な学びとはなにか

オックスフォード大学のズボーン准教授が 2014 年にあと 10 年で「消える職業」「なくなる仕事」を認定した。702 業種を調査した結果、あと 10 ～ 20 年程度で、アメリカの雇用者の約 47% の仕事がコンピュータによって自動化されると提言した。2021 年現在、日本においても、レジや建物の案内など、コンピュータによる自動化が加速している。この変化する世の中を生き抜くためには、子どもたちはコンピュータについて知り、コンピュータに得意なことと苦手なこと、人間に得意なことと苦手なことを整理し、既存の考えに縛られずに、主体的にものごとに取り組む姿勢が必要である。

ここで、教育のキーワードとなるのが、「主体性」、「主体的な学び」である。「主体性」とは、学習者本人が内発的動機づけにもとづき行動する様子で、ダニエル・ピンク（2010）は、これからの変化する社会を生き抜くためには持続するやる気のためには、「モチベーション 3.0」、対価を求めない動機づけ、つまり「内発的動機づけ」が必要であるとした。「モチベーション 1.0」とは、人間が持つ原始的なやる気、寝る、食べるなどの生きるため動機づけであり、「モチベーション 2.0」は、対価を求める動機づけ、つまり「外発的動機づけ」と位置付けた。高度成長時代は、対価を求める動機付けで発展できたが、賞罰・飴と鞭の動機づけは、だんだんと目的が変化し、これからの社会を生き抜くためには有効ではないとした。「モチベーション 3.0」は、自律性（自主性）・マスタリー（熟達・成長）・目的の 3 つが個人の内発的な動機づけにつながり、持続するやる気となる。

この主体性を妨げるのが、「過干渉」である。過干渉とは、学習者の活動を先取りして関わってしまう行為のことで、過干渉は、正解を求め、失敗を恐れるようにな

る。授業では時間制限もあるため、過干渉になってしまうこともある。過干渉にならないためには、全員が同じである必要はないこと、子どもの目線で見えているものは違うこと、自分で考え、周囲はそれを認めること、個性と多様性を認めることを意識するとよい。主体的でないのは、主体的でなくなる環境の結果であることを教員や支援者、保護者は認識し、間違ってもすぐに訂正しない、失敗してもいいので、そこからどう学ぶかという指導につなげるとよいだろう。

### 2-3 内部知識とはなにか

では、過干渉にならないために、すべてを考えさせればいいのだろうか。たとえば、初めて使うプログラミングソフトで、「モーターを２秒回して、３秒止めるプログラミングをしてみて」と、なにもルールを教えないまま考えさせるのは、指導法として問題がある。ここで重要なのは、その分野での基本知識、すなわち内部知識はなにかということを明確にしなければならない。内部知識にあたる基本知識は、明示的な指導が必要であり、それを基に、外部知識を収集・活用することが、変化する社会に対応する問題解決力の育成につながる。変化に対応できる力として、学び方を学ぶためにも、どこまでが内部知識として明示的に指導すべきことで、どこからが主体的に外部知識の収集となるのかを整理することが必要である。

## ３.プログラミング教育実施における問題点

### 3-1 お金がない！

プログラミング教育のための新たな教材を購入するための予算が確保できないという問題点がある。スクラッチなどの Web 上の無料のソフトもあるが、画面上のプログラミングだけでなく、自分のプログラミングで実際のものを動かす、制御するという体験は、プログラミング教育には大切な要素であると考える。そのため、理科のモーターカーなどの既存の教材に、プログラミングの要素を追加したり、素材を工夫することで、理科を学ぶ意義も見いだせる。また、大学や企業、学校間でのプログラミング教材の借用や共有なども有効な手立てといえよう。

### 3-2 時間がない！

どの教科のどの部分にプログラミング教育の要素を導入するかも問題点として挙げられる。この際、ものづくりからプログラミングまで、１つの教科で完結しようとせず、基本の知識はそれぞれの教科で学び、総合的な学習の時間を教科の発展的な時間として活用することも考えられる。学校全体のカリキュラムマネジメントの中に、発達段階に応じたプログラミング的思考の育成を目指し、教科横断的かつ学年の学びを積み重ねる内容と教材を用いた設計を組み込むことで学びが深まるであろう。

### 3-3 スキルがない！

プログラミング教育をどのように取り入れればいいのか、多くの教員は不安を感じている。ここで有効なのは、教員研修である。苦手意識のある教員にも取り組めるスタートアップ研修や、複数回の研修で、実際の授業の取り組みをイメージすることも有効である。また、Web 上での事例の情報収集や学校内外での情報共有にも積極的に取り組み、学校全体で協働し、軸となる教員を増やし、不安を感じている教員のサポートも重要であるといえよう。

### 3-4 人がいない！

PC を使ったプログラミングを取り入れた授業では、子どもたちの機器トラブルや個人差対応に、担当教員以外の支援体制が必須である。ここで、今後のサポート体制の可能性としては、教員を目指す大学生や地域住民、現役を引退したエンジニアや技術者であったシニアという多様な人材の研修も行いながら、小学校と協働することが考えられる。前述のシニアリーダー研究では、学生と社会人・シニアの大きな違いは経験の差であると考え、図１のように，現役を引退したエンジニアや技術者だけでなく，社会人としての経験を持つ現役世代を含めた社会経験者と子育てを含む社会生活経験者として広義に捉

え，その中で子どもの活動に関心のある人たちがこの活動に参加する「シニア」であると定義した。子どもたちの活動支援に協力するシニアは、教えてあげたい、困っていたら助けたいという気持ちが強いことから、過干渉な支援をしてしまうことがあるので、主体的な学びを支援するための指導についての研修を行ってから、サポートに入ってもらうという手順の検討も課題である。

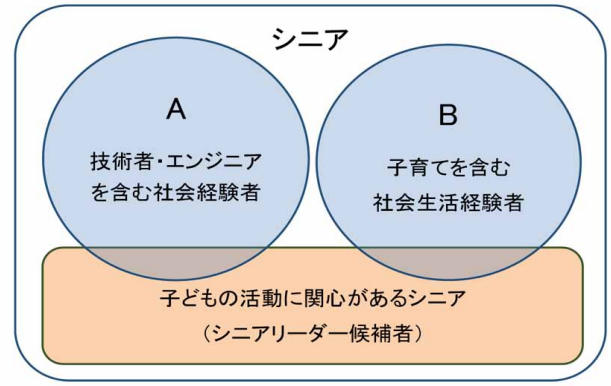



図１　シニアリーダーの分類
（星名他 ,2019b）

**4 . プログラミング的思考を育成する指導法**

4-1 プログラミング教育の進め方

プログラミング教育の進め方として、以下の 4 つのポイントが挙げられる。

①学ばせたい目的を確認する

「プログラミング」を教えることが目的にならないようにする。教科横断的なつながりを確認する。

②基本知識を指導する

目的を達成するための手段として、必要な知識はなにかを明確にする。知識の指導方法も、一斉に指導するのか、資料や動画で個人のペースで指導するのかを検討する。

③主体的に学ぶためのしかけを作る

「自由に考える」なのか､「テーマから考える」なのか、どちらが学ばせたい目的を達成するために有効なのかを考える。子どもたちが考えるプロセスをあとで追えるようなワークシートを作る。一度作ったものを、どう改善するといいのか、その結果も記録する。

④教材の提供の仕方を工夫する

どのような教材を提供すればいいのか、発達段階に適しているかを考える。子どもたちの思考に沿って、あとで自分の辞書となるような資料やワークシートを工夫する。内部知識の部分は明示的に学ばせる。子どもたちの個人差を意識する 。ものづくりの得意な子とそうでない子がいるので､「自由に」の難しさとその単元の目的にための時間配分を考える。例えば、基本形から発展させ、修正できる力を育成する。できる子どもたちには、プラスの課題を与えることも有効である。

4-2 プログラミング的思考を育成するための工夫

プログラミング的思考を育成するためには、以下の 4 つのポイントが挙げられる。

①直感で終わらせない

画面上で動かすのは得意！でも、それを説明できるのか？ワークシートや教材の工夫や仕掛けが必要

②思考のプロセスを意識させる

なにが問題で、どのように情報を集めて、どうやって改善するのか。改善した後、実験して、その問題は解決したのかを記録する。

③類推する力を育成する

類似の問題に対して、それまでの経験が活かせるのかを考えさせる。日常生活やニュースから、構造や仕組みをイメージさせる。今あるものを、もっと便利にするにはどうしたらいいかを考える。

④問題解決的に学習させるための工夫をする

内部知識を明確にし、明示的に指導する。実験してどう改善したかを言語化してまとめるワークシートを工夫する。うまくいかなかったことも大切な結果であると認識させて、次へつなげる。

**5 . プログラミング教育の教員研修**

公立小学校からの研修依頼時に、PC を使ったプログラミング教育に対する不安の強い教員が複数在籍しているとの情報を得たため、段階的な研修が有効であると仮説を立て､研修内容の検討を行った。研修の内容設計は、鈴木（2015）の研修設計のインストラクショナルデザインにおける ARCS モデルの 4 要因〈注意〉(Attention:

面白そうだ)、〈関連性〉(Relevance: やりがいがありそうだ)、〈注意〉(Confidence: やればできそうだ)、〈満足感〉(Satisfaction: やってよかった)により行い、研修プログラムの開発は、ADDIE モデルに沿って行った(星名他, 2021)。プログラミング的思考の育成につながるものづくりを導入研修とし、作ったものを自分の思うように動かしたいというプログラミングへの興味関心へつなげる発達段階も意識した授業設計の例として複数回の研修を設計した。

研修の目的は、「プログラミング教育に関する知識を得て、プログラミング的思考を育成するための授業設計を実現する」とし、目的実現のための達成したい目標は、① STEM 教育に基づくプログラミング教育について理解する、②プログラミング教育が導入できそうな科目をイメージできる、③プログラミング教育への不安を解消する、④プログラミング教育の授業実践へつなげる、という 4 つを設定した。

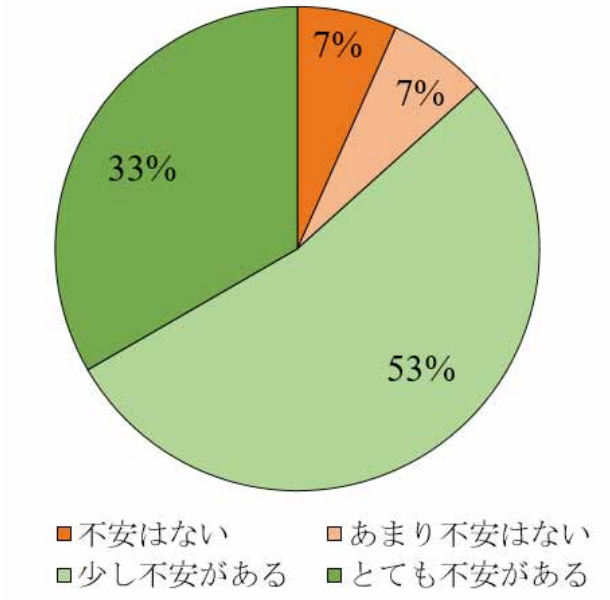



図 2　研修前のプログラミング教育への不安
(星名他 ,2021)

事前アンケートから、図 2 のように 86%の教員にプログラミング教育導入に関する不安が見られた。内訳として「不安はない・あまり不安はない」と回答したのは授業を担当していない管理職であり、実際に授業を担当する教員は、なんらかの不安があることが明らかになった。「具体的にどのように進めていったらよいのか、わからない」という【授業設計に関する不安】、「やることが多数あり、教材研究を含めて、時間的余裕がない」という【時間的な不安】、「知識も経験も足りないため」という【教員の経験やスキルに関する不安】、「お金の面(教材、教具、ソフト、ハード)」という【経済的な不安】の 4 つの観点の不安が挙げられていた。

研修後には、「どのように考え、改善していくのか、自分自身が体験できたことにより、イメージがわいてきました」、「なぜそうしたのか？・・・の説明ができるような授業展開を考えればよいのだとわかった」などの感想が得られ、教員研修は不安の軽減に一定の効果が見られ、図 3 のように教科へと発想をつなげることができた。

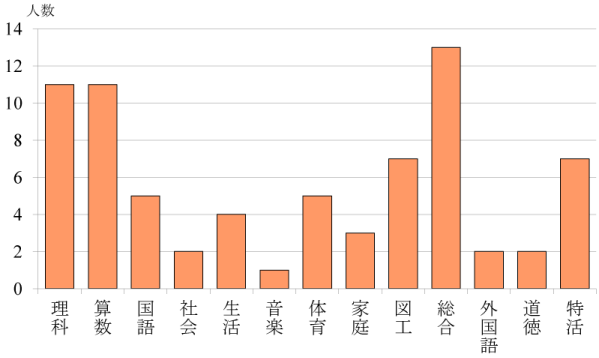



図 3　プログラミング教育の導入可能な教科
(星名他 ,2021)

2020 年度までの教員研修プログラムと授業実践に関して表 1 にまとめる。教材 A は埼玉大学 STEM 教育研究センターが監修に加わり作成された NHK 小学生ロボコンのキッドである。モーターと自作リモコンを使った生きものロボットの制作から、問題解決的な動きの改善活動を体験した。教材 B はレゴブロックと同センター開発の STEM Du コントローラを用い、PC のスクラッチソフトと連動してコントローラにプログラミングする研修を実施した。教材 C は同センター開発の鉛筆プログラマを使って、コントローラの金属部分を鉛筆で塗りつぶすことで命令するプログラミングの研修を実施した。

2021 年度は、マイクロビットを取り入れた研修と教材の開発を行った。別添の「マイクロビットプログラミング基本編・拡張編」は、マイクロビット教材を小学校で利用することを念頭に、資料だけを読めば、教員だけ

でなく、高学年ならひとりで学習を進められるように作成した。この資料を使ったワークショップや教員研修を複数回実施し､アンケートを基にした資料改善を行った。この資料の内容がマイクロビット教材を使うための基本知識である。この知識を基にして､「こんなふうに動かしてみたい」「こんなものを作ってみたい」「こんなものがあったらいいな」という主体的な活動へつなげて発展させることができる。発展させた活動の中で、プログラミングでわからないことがあったときは、教員も一緒に情報収集をしたり、外部人材の力を借りて子どもたちと考えるという進め方が、子どもたちにとってもより知識を広げる学習方法であり、教員にとってもプログラミング教育への不安を解消する手段であると思われる。

山本ら（2020）はプログラミング教育の実践をしている教員の分析を行い、学校単位で効果的に実践するためには、中心となる教員の重要性を指摘した。本稿の事例では、軸となって授業実践をした教員が翌年異動となった経験から、少数の教員のみがわかっているだけでは、継続したプログラミング教育の実践は困難であることが明らかになった。複数の軸となる教員の存在はプログラミング教育の推進には重要であるが、全教員がプログラミング教育に関する意識を高め、協働して授業設計を進める環境づくりが推進を継続するためには必要であると推察できる。今回の研修を通して､「プログラミング教育に関する知識を得て、プログラミング的思考を育成するための授業設計を実現する」という目的は、複数回の研修と授業実践からも達成できたといえよう。今後は、異動の多い現場に対応した研修の方法や情報の提供方法や支援などについても検討を進めたい。

表 1　プログラミング教育に関する教員研修プログラムと授業実践　（星名他 ,2021）

| 開催月 | 分類 | 内容 |
|---|---|---|
| 2019 年 12 月 | 研修第 1 回 | STEM 教育とプログラミング教育の講義・教材 A（パソコンなし）の研修 |
| 2020 年　2 月 | 研修第 2 回 | 教材 B（パソコンあり）の研修・授業実践①のワークシート検討 |
| | 授業実践① | 教材 B を使ったオリジナル扇風機のプログラミング（6 年理科） |
| 2020 年　7 月 | 研修第 3 回 | 教材 C（パソコンなし）の研修 |
| 2020 年 10 月 | 授業実践② | 教材 C を使った障がい者に役立つものづくりとプログラミング（4 年総合） |
| 2020 年 11 月 | 研修第 4 回 | 教材 B（パソコンあり）の高学年教員向け復習・授業実践③・④の内容検討 |
| 2020 年 12 月 | 授業実践③ | 教材 B を使ったプロペラのプログラミング（5 年総合） |
| 2021 年　1 月 | 研修第 5 回 | 教材 B（パソコンあり）の全教員向け復習 |
| 2021 年　2 月 | 授業実践④ | 教材 B を使ったプロペラのプログラミング（6 年理科） |

※研修で利用した教材は以下の通り。教材 A：いきものロボット（プラダン＋モーター＋オンオフのリモコン）、教材 B：STEM Du コントローラ＋レゴブロック、教材 C：鉛筆プログラマ＋豆電球。

**6. プログラミング教育の実践例**

実践①では、6 年理科において、教材 B を使ってレゴブロックで扇風機を作り、明るさセンサーや音センサーを使ったプログラミングを行った。資料とワークシートおよび教室のモニターに教員の PC 画面を提示して進められた。

実践②では、4 年総合的な学習の時間において、教材 C を使った福祉をテーマにした授業が実施された。車いすと白杖体験の後、ペアで目や耳が不自由な人が困っていることを考え、身の回りの箱なども使って役立つ道具を作り、鉛筆プログラマでプログラミングをした。耳の不自由な人は、玄関のブザーが鳴っても聞こえないので、ボタンが押されたら豆電球を光るようにする道具など、自由な発想が見られた。教室の掲示も児童たちが参加する形式で作られ、活用されていた。作品の発表時は、教員がタブレットで

撮影しながら、同時に教室前のモニターに投影するなど、ICT を活かした進め方となっていた（図 4，図 5）。

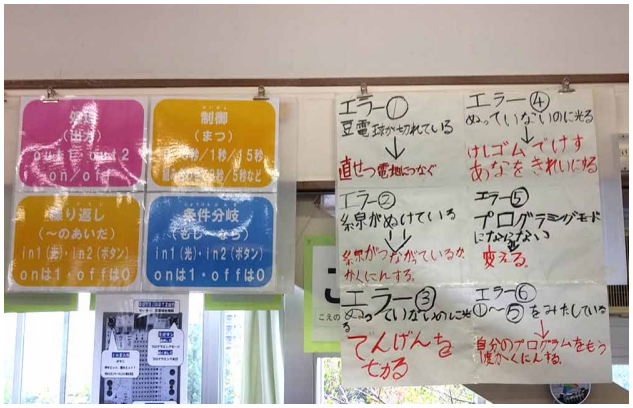



図 4　授業実践時の教室の掲示

図 5　玄関ブザーの発表

実践③、④では、プログラミングに集中させるために、プロペラを作るレゴパーツのみを用意して実施した。組み立てる途中経過の写真や、命令ブロックを拡大してマグネットで黒板に貼り付け、児童が黒板でプログラムを並べるなどの進め方の工夫も見られた。

発達段階に適した教材とどのような活動まで展開させるのかについては、導入する教科や他教科とのつながりを考えた学校としてのカリキュラムマネジメントが必要である。さまざま教材を使った事例については最終ページにて紹介する。

### 7. まとめ　～柔軟な問題解決力の育成へ～

2020 年度に導入されたプログラミング教育を推進するためにも学校や教員への支援は必須である。また、教員自身がプログラミング教育の意義を認識し、どのようにプログラミング教育の活動に意味を持たせるかが重要である。単なるプログラミングのスキルを学ぶだけにならないようにしなければならない。

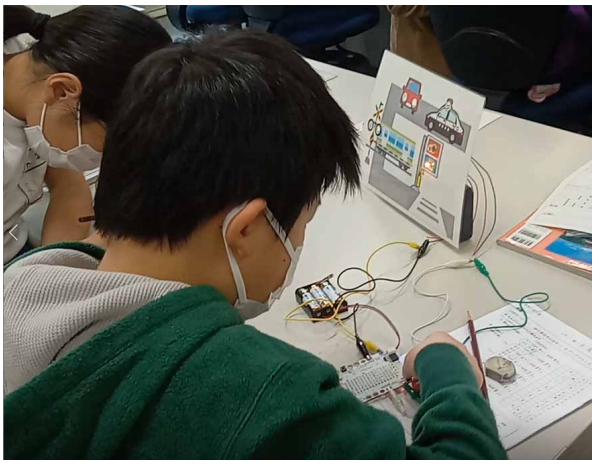
GIGA スクール構想により、2021 年 3 月に研修校にもタブレットが導入された。2020 年度の研修や授業実践では、大学から PC と教材一式を借用していたため、教員の運搬にも苦労があったが、2021 年度はコントローラやレゴブロックなどの教材のみの借用で対応できた。GIGA スクール構想は、教科横断的なプログラミング教育の推進に寄与していると思われた。しかし、プログラミング教育に対する取り組みは、学校間での差が大きい。研修校で軸となっていた教員の異動先の小学校で、タブレットと大学より借用したマイクロビットを使った研修を行った。その教員から「プログラミング教育に対する熱が違うのですが…」と事前に聞いていたのだが、事前事後アンケートからもプログラミング教育に対する情報収集を含めた取り組み方の違いが明らかになった。この学校間の差をなくすためにも、教員研修や教材や実践事例などの情報提供方法の検討が必要である。教員研修では、教材の操作方法を学ぶことが目的ではなく、プログラミング的思考を育成する意義を理解し、プログラミングの基本的な考え方および発達段階に応じた指導方法や進め方のポイントなどから、授業への導入をイメージできるようにするのが目的である。教材やソフトウエアは絶えず変化する。Web 版のプログラミング教材は、仕様の変更や機能の追加などの更新が早い。資料と実際のプログラミング画面がまったく同じでなければならないという考え方を捨て、共通点を探す視点が重要である。基本的な考え方は共通であるということを意識していれば、情報を収集して対応することができる。「常に情報は新しくなる」という意識を教員も子どもたちも持つことで、新たな問題解決場面でも、変化することへ柔軟な対応ができ、社会を生き抜くための 21 世紀型学力の育成につながるといえよう。

## 引用文献

ダニエル・ピンク，大前研一訳（2010）モチベーション 3.0，講談社

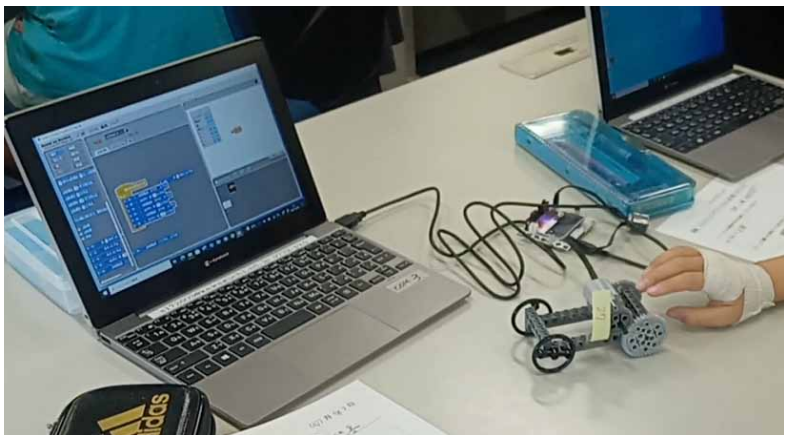
J.M. ケラー，鈴木克明監訳（2010）学習意欲をデザインする―ARCS モデルによるインストラクショナル

デザイン―，北大路書房
ユーリア・エンゲストローム, 山住他訳(2013)ノットワークする活動理論―チームから結び目へ，新曜社
鈴木克明（2015）研修設計マニュアル―人材育成のためのインストラクショナルデザイン―，北大路書房
文部科学省 . 小学校段階における論理的思考力や創造性，問題解決能力等の育成と プログラミング教育に関する有識者会議（2016）小学校段階におけるプログラミング教育の在り方について（議論の取りまとめ），http://www.mext.go.jp/b_menu/shingi/ chousa/shotou/122/attach/1372525.htm（閲覧日 2018.9.20）
山本利一・本郷健・木村猛能・永井克昇（2016）初等中等教育におけるプログラミング教育の教育的意義の考察，教育情報研究，第 32 巻第 2 号 3-11
文部科学省・総務省・経済産業省（2018）未来の学びコンソーシアム　小学校プログラミング教育必修化に向 け て，https://miraino-manabi.jp/assets/data/info/miraino-manabi_leaflet_2018.pdf（閲覧日 2018.9.20）
星名由美・小山航太・野村泰朗（2018）プログラミング教育における子どもの主体的な学びを指導するシニアリーダー育成プログラムの開発－シニアの経験を活かしつつ子どもに過干渉しないシニアリーダーの構えの検討－，第 19 回計測自動制御学会システムインテグレーション部門講演会（SI2018），714-717
星名由美・峯村恒平・小山航太・野村泰朗（2019a）プログラミング教育における子どもの主体的な学びを指導するシニアリーダー育成プログラムの検討－シニアの経験を活かしつつ子どもに過干渉しないシニアリーダーの構えとは―，埼玉大学紀要　教育学部，68（1）253-260
星名由美・峯村恒平・西原舞・野村泰朗（2019b）プログラミング教育におけるシニアリーダー育成と協働に関する検討～子どもに過干渉しない指導法の研修プログラム開発の試み～，日本科学教育学会研究会報告，33 巻 4 号 27-32
山本利一・鈴木航平・吉澤亮介（2019）小学校情報教育担当者向け教員研修を通したプログラミング教材の評価と課題, 教育情報研究, 第 35 巻第 1 号 49-58
文部科学省（2020）小学校プログラミング教育の手引（第三版），https://www.mext.go.jp/content/20200218-mxt_jogai02-100003171_002.pdf（閲覧日 2020.3.5）
山本朋弘・三井一希・木村明憲・大久保紀一朗・堀田龍也（2020）小学校プログラミング教育の推進に関する個人別態度の構造分析，日本教育工学会論文誌，44(1) 145-154
楠見孝・西川一二・齊藤貴浩・栗山直子（2020）プログラミング教育の授業実践に対する小中学校教員の期待と意欲，日本教育工学会論文誌，44(2) 265-275
星名由美・野村泰朗（2021 印刷中）STEM 教育に基づくプログラミング教育の教員研修プログラム開発，埼玉大学紀要　教育学部，70（2）269-277

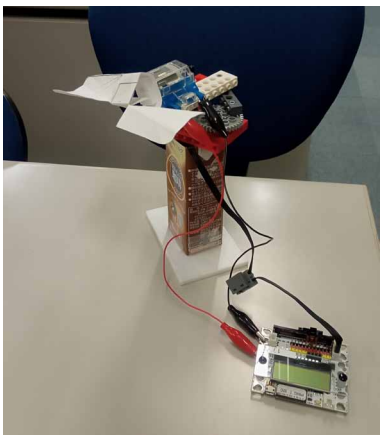


鉛筆プログラマ＋豆電球
歩行者用信号機や踏切の点滅など

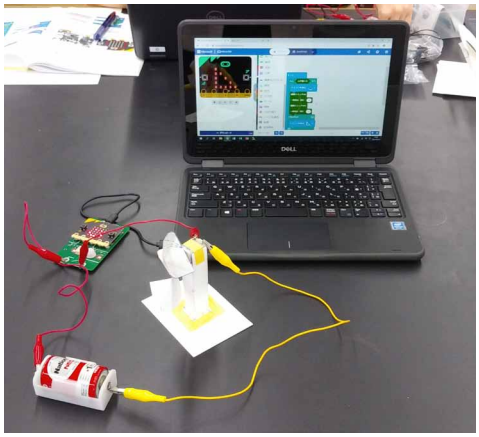
STEM Du ＋レゴの扇風機



STEM Du ＋レゴの車



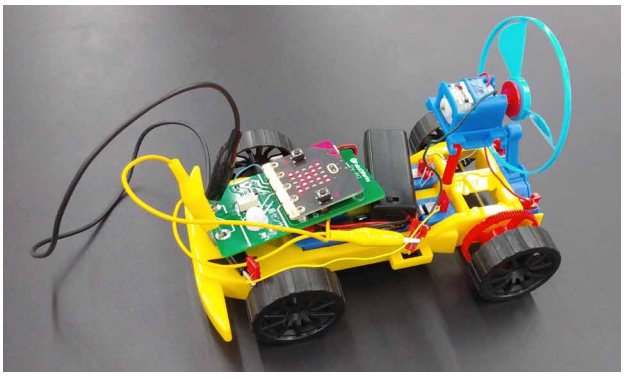
STEM Du ＋レゴ＋紙工作プロペラ
身近な箱などを活用してオリジナル扇風機



マイクロビット＋紙工作扇風機

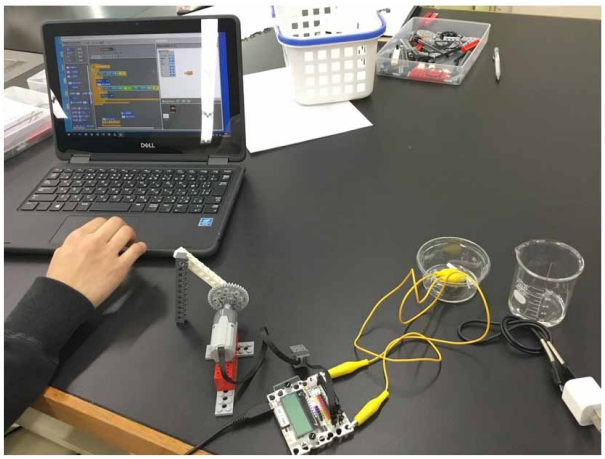
マイクロビット＋紙工作扇風機と風力の実験

マイクロビット＋理科モーターカー



マイクロビット＋理科風力カー

STEM Du ＋プラダンの車



STEM Du ＋レゴ＋抵抗センサー
雨が降ったらドアが閉まる装置

「マイクロビットプログラミングを子どもに教える教え方の考える」

# マイクロビットプログラミング　基本編 Ver.3

昭和女子大学現代教育研究所　理科教育研究プロジェクトチーム　資料作成：星名由美

**◇◇　本日の目標　◇◇**

（1）マイクロビットとパソコンをつなぐことができる
（2）マイクロビットのプログラムをするページで、モーターを動かすための「STEM」ブロックを追加できる
（3）マイクロビットの基本的なプログラミングの手順がわかる
（4）人感センサーを使って扇風機を動かすプログラミングができる
（5）音センサーや明るさセンサーなどを使ったプログラミングを体験する

## 1．マイクロビットで扇風機を動かす準備をしよう

①パソコン


②マイクロビット（micro:bit V2.0）

③USB ケーブル


④理科ボード

⑤プロペラ付きモーター回路

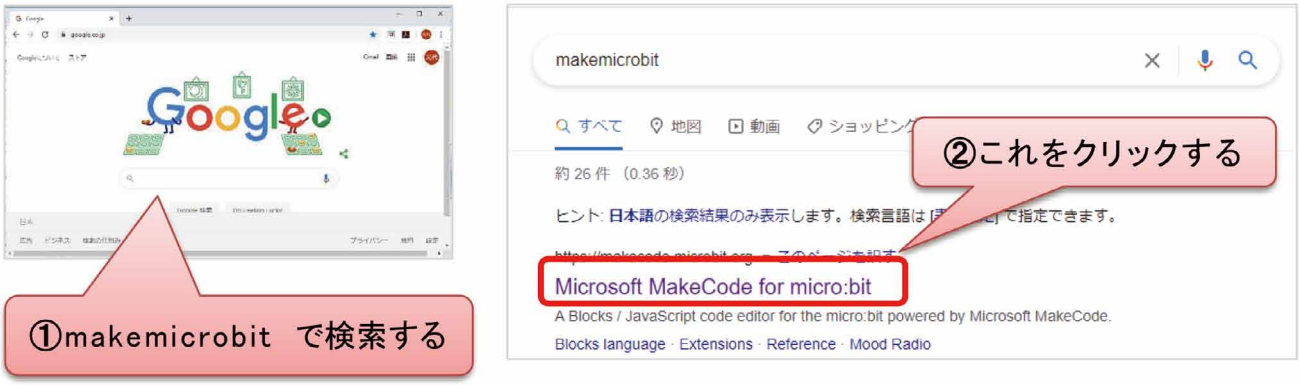
### 1．Web ページからブロックエディターを開こう

①パソコンの電源ボタンを押して、パソコンを起動し、インターネットに接続する

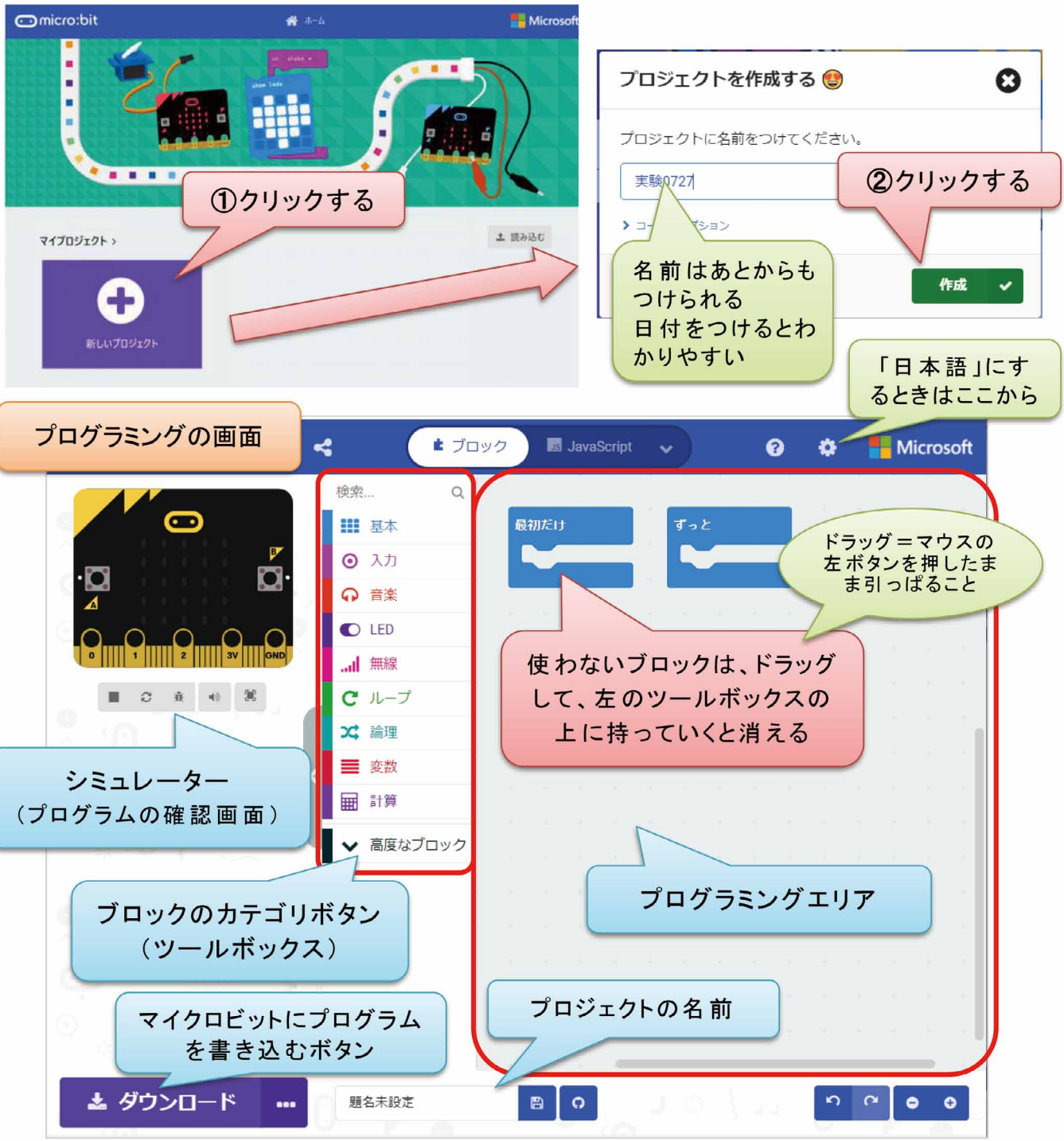
②ブラウザーをダブルクリックで開く（ここでは、Google Chrome で説明）

③「 makemicrobit 」と検索し、一番上に出てくる「Microsoft MakeCode for micro:bit」をクリックしてブロックエディターのページを開く

URL：　**https://makecode.microbit.org/**




④ 「新しいプロジェクト」をクリックし、次に出てきた画面の右下にある「作成」ボタンをクリックすると、プログラミングの画面が開く（プロジェクトの名前は、あとからつけられる）

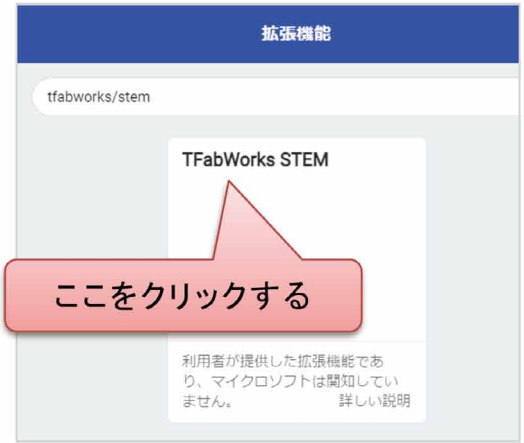


## 2．モーターを動かす「STEM」ブロックを追加しよう

①真ん中のツールボックスから、「高度なブロック」をクリックし、一番下に出てくる「拡張機能」をクリックする


②拡張機能の検索画面になるので、「tfabworks/stem」と入力して、虫めがねのマークをクリックして検索する


③検索結果をクリックすると、プログラミングの画面に切り替わり、「STEM」が見えるようになれば、準備完了！

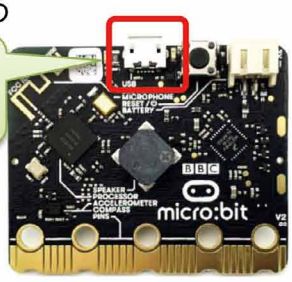



## 2．扇風機を動かすプログラミングをしてみよう

### 1．マイクロビットとプロペラ付きモーターを組み立てて扇風機を作ろう

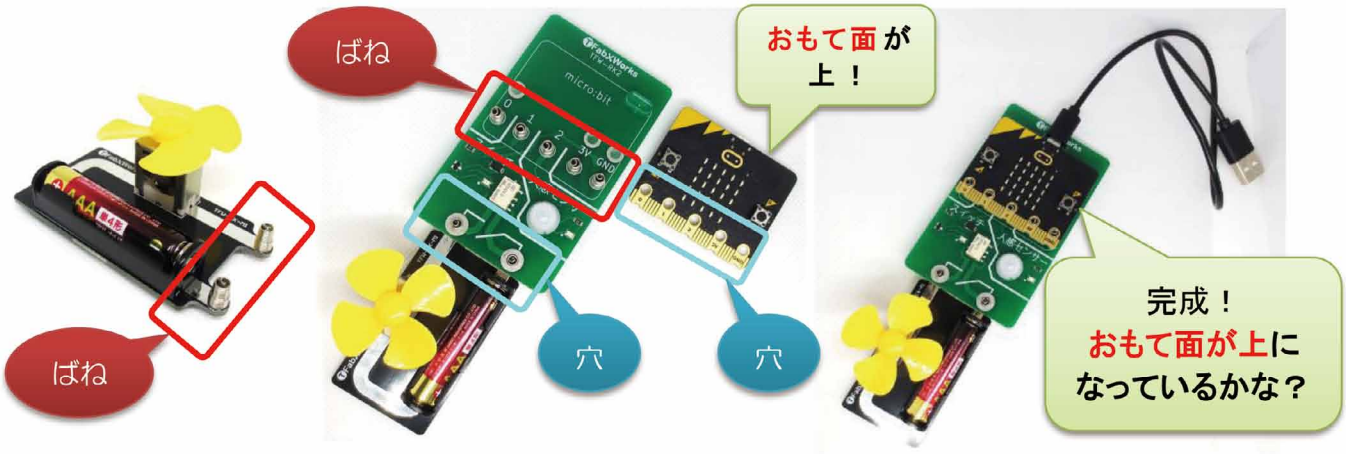
①USB ケーブルの小さい方をマイクロビットに接続する。

うら面を
見てみよう！




②プロペラ付きモーター回路の**ばね**と、理科ボードを下側にある**穴**に入れて合体する。

③**マイクロビットおもて面の下にある番号の穴**と、理科ボードの中央部分にある**ばねの番号を合わせて合体**して、扇風機の完成！　マイクロビットの LED（おもて面）が上になっているのを確認しよう！




### 2．マイクロビットとパソコンをつなごう

①USB ケーブルの大きな方をパソコンに接続する。USB ケーブルの形とパソコンの差込口（ポートの形）を確認して、かみ合うように差し込む。（初めて接続したときだけ音が鳴るが、それ以降は、前に書き込みしたプログラムが動いている）

②マイクロビットとパソコンを以下の手順で接続する。

## 3．扇風機を動かすプログラミングをしてみよう

5つのポイントとこまったときのチェックポイントを確認してから、プログラミングしてみましょう！

### プログラミングをするときのポイント

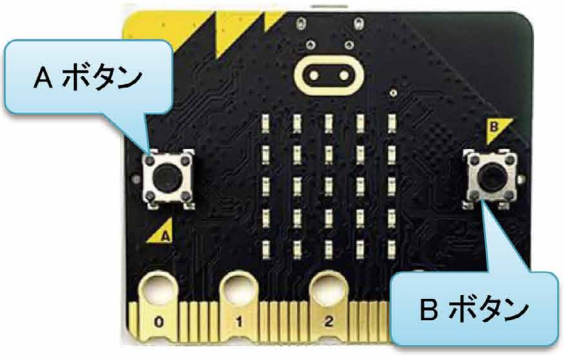
1．関係するブロックは、同じ色でカテゴリーボタンの中にまとまっている
2．ブロックは、マウスの左ボタンでクリックしたまま引っ張り（ドラッグ）、プログラミングエリアにもってくる　※いらないブロックはカテゴリボタンへドラッグして消しておく
3．動きのきっかけ（＝トリガー）となるブロックの中に、ブロックをつなげる
4．プログラムは上から下へ進む　（動きがどうなるか考える）
5．プログラムが完成したら、左下のダウンロードをクリックして、マイクロビットにプログラムを書きこむ　（ダウンロードすると、命令が動くようになる）

### こまったときのチェックポイント

1．マイクロビットの向き（おもて面が上）を確認する（４ページ真ん中の完成図）
2．パソコンとの接続を確認する　（４ページ右下の図）
3．簡単なプログラミングをダウンロードして動きを確認する
4．うまく接続できないときや動きを止めたいときは、USB ケーブルを一度抜く　（新しいプログラムをダウンロードするときに、もう一度、USB ケーブルをつなげばよい）

### 実験１　マイクロビットのボタンを押して扇風機を動かすプログラム

マイクロビットの A ボタンと B ボタンを使って、扇風機を動かしたり、とめたりしてみましょう。

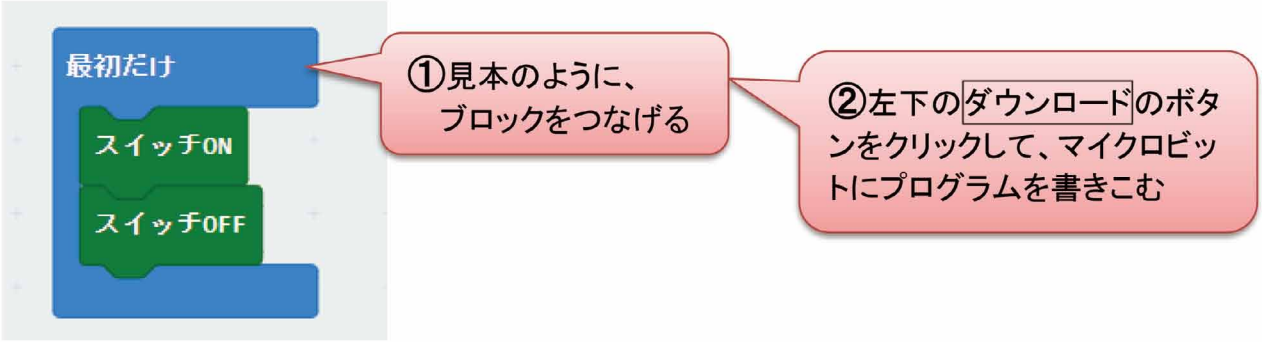

実験の結果をメモしておこう！　マイクロビット本体の A ボタンと B ボタンを押すとどうなったかな？

実験２　扇風機を動かすプログラム　（モーターのスイッチを ON にする）


実験の結果をメモしておこう！

★スイッチ ON のブロックだけだと、次の命令がないので、ずっと動いたままになりましたね。では、次の実験をしてみましょう。

実験３　扇風機を動かしたり、とめたりするプログラム



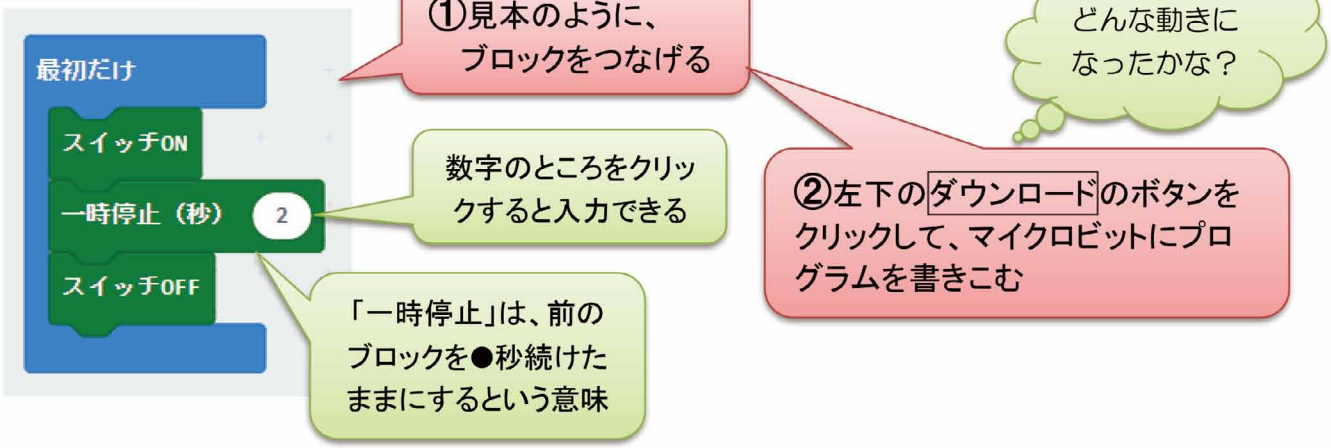

実験の結果をメモしておこう！

★プログラムは、上から下に、すごいスピードで命令が進みます。スイッチ ON とスイッチ OFF を並べるだけでは、「スイッチ ON！スイッチ OFF！」とすごいスピードで命令されるので、扇風機は動いていないように見えます。では、扇風機を動かして、とめるのは、どうしたらいいのでしょうか？

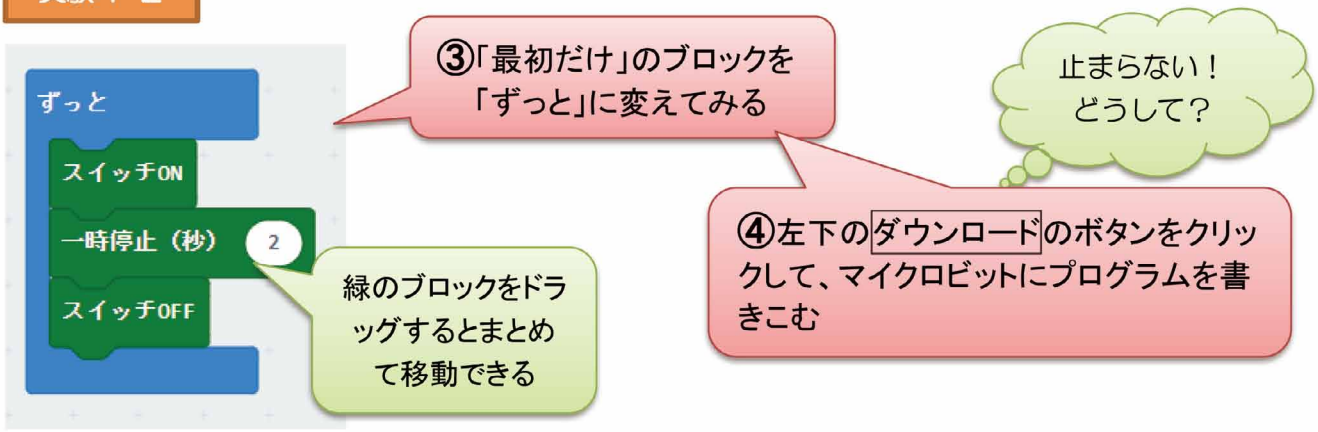
## 実験４　扇風機の ON と OFF の時間を設定するプログラム

「スイッチ ON」をどのくらいの時間していればいいのかを時間で設定することができます。「最初だけ」と「ずっと」のきっかけブロックの違いも実験してみましょう。

### 実験４-１

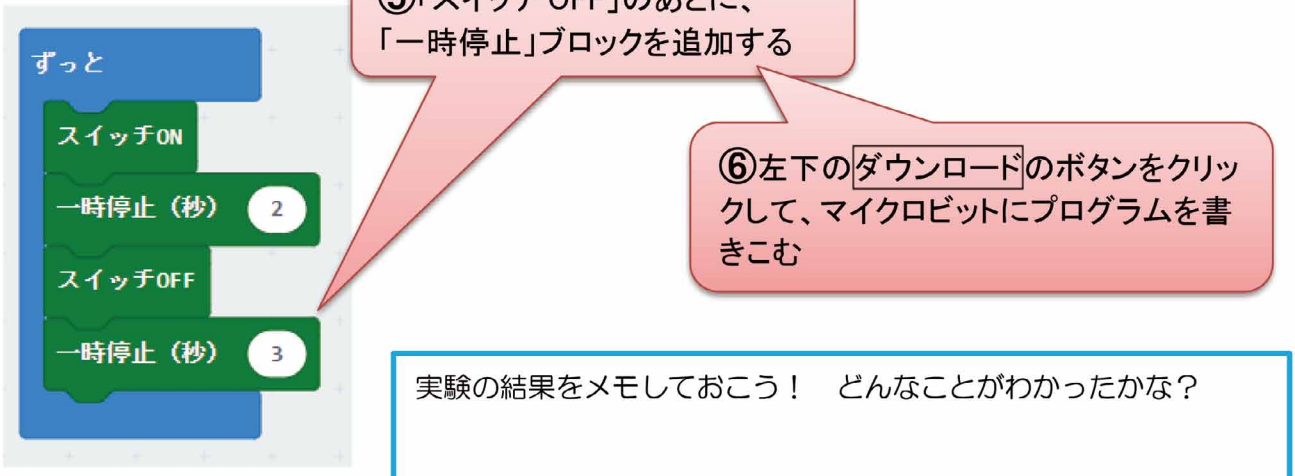



### 実験４-２




★ずっとプログラムを動かして、扇風機を動かしたり、とめたりするためには、スイッチ OFF にも時間を設定する必要がありそうですね。次のブロックを追加してみましょう！

### 実験４-３




実験の結果をメモしておこう！　どんなことがわかったかな？

## 4．人が近づいたら扇風機が動くようにプログラミングをしてみよう

自動ドアや手をかざすと水が出る水道のように、人を感知して扇風機が動くようにプログラミングしてみましょう。緑の理科ボードの人感センサーを使います。

実験５　人を感知して、扇風機の ON と OFF が切り替わるプログラム


実験の結果をメモしておこう！


人感センサーにチューブをつけて、感度を調整してみましょう。

ここまでできたら大成功！
拡張編で、もっといろんなことを実験してみよう！

## 5．プログラムの保存をしよう

プログラムは、同じパソコンでマイクロビットのプログラミングページを開くと、前に作ったプログラム（プロジェクト）が自動的に保存されています。自分のパソコンにダウンロードするときは、プロジェクト名の右の保存ボタンをクリックすると、パソコンの「ダウンロードフォルダ」の中に保存されます。パソコンに保存されたプログラムは、プログラミングエリアにドラッグすると、開きます。

プログラミングの保存ができて、マイクロビットの実験を終えるときは、次の人のために、白紙のプログラム（ブロックがなにもない状態）をダウンロードしておきましょう。

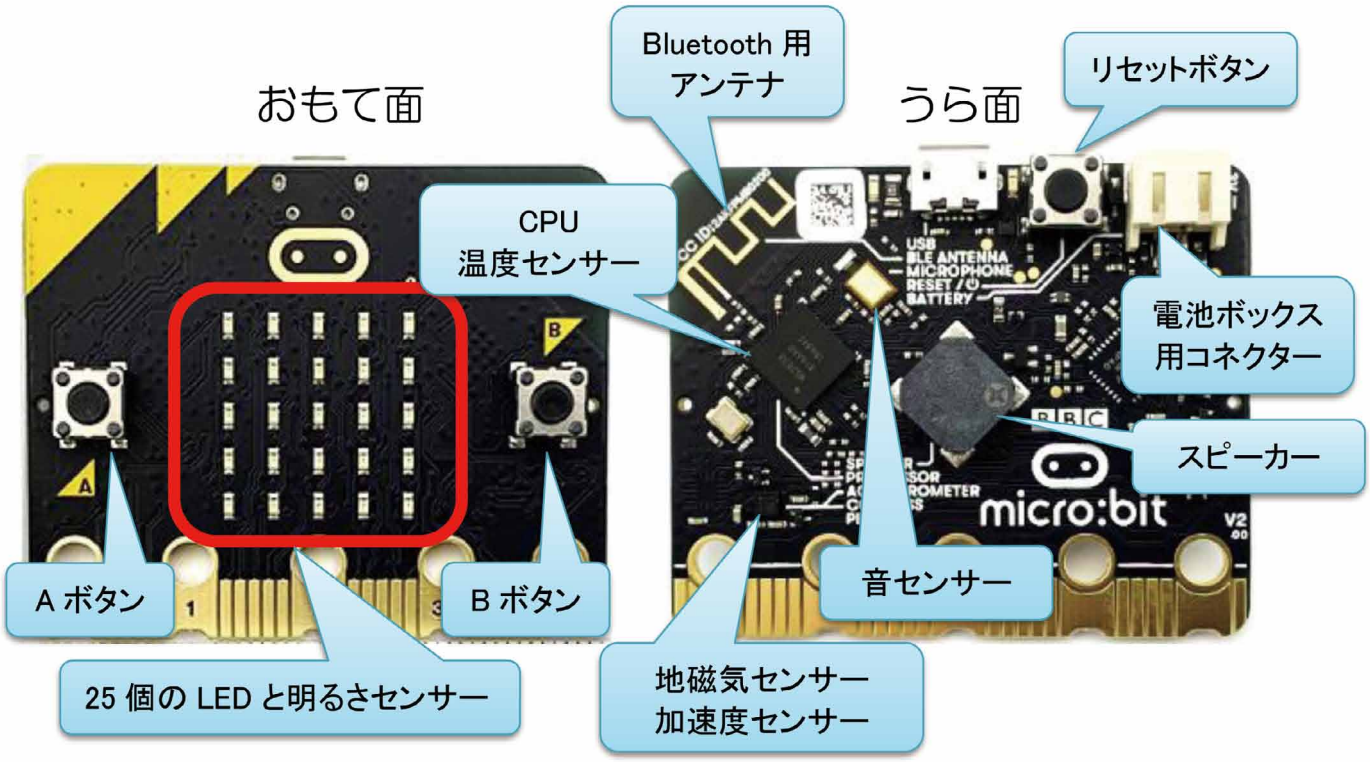
「マイクロビットプログラミングを子どもに教える教え方の考える」

# マイクロビットプログラミング　拡張編　Ver.3

昭和女子大学現代教育研究所　理科教育研究プロジェクトチーム主催　資料作成：星名由美

## 1．マイクロビットにはどんな機能があるかみてみよう

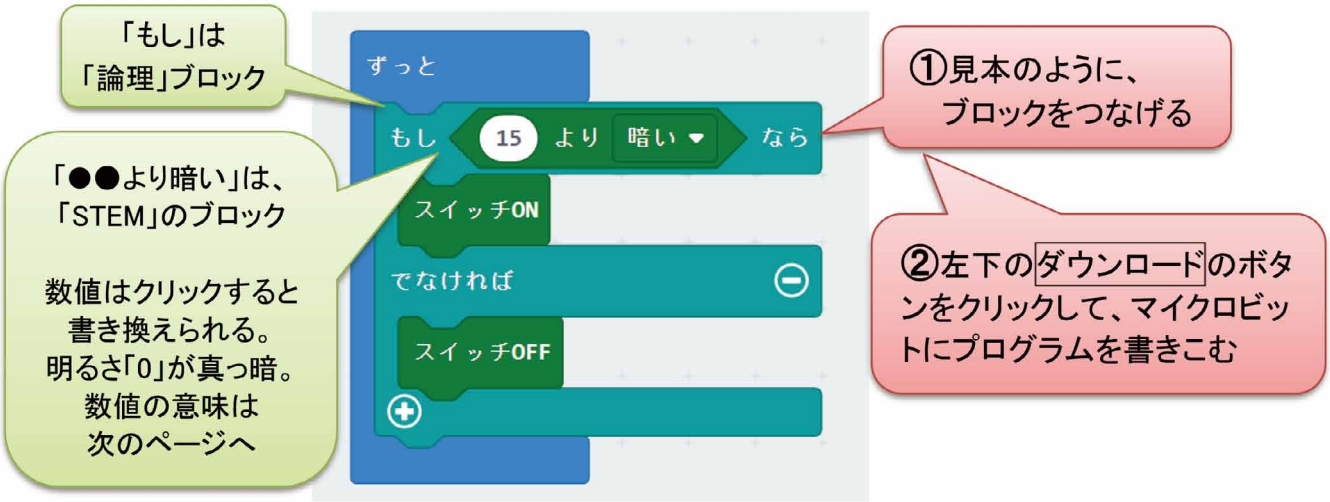
マイクロビットには、いろいろなセンサーや機能があります。マイクロビットを観察してみましょう。




## 2．いろいろな方法で、扇風機を動かすプログラミングをしてみよう

体験１　明るさセンサーを使って扇風機を動かすプログラム

マイクロビットの表面にある明るさセンサーを、手で隠して暗くしたり、手をはなして明るくしたりして、扇風機が動くようにプログラミングしてみましょう。



## 明るさの測定をしてみよう

明るさの値は、真っ暗が 0、一番明るい 255 までの数値になります。動きのきっかけとなる境目となる値のことを「しきい値」といいます。しきい値は、マイクロビットの表面にある明るさセンサーで測定して計算することができます。明るさの測定値は、LED で数字が表示されます。手をかざすと値が変化するので、確認してみましょう。

「数を表示」は、「基本」ブロック

「明るさ」は、「入力」ブロック


### しいき値の計算方法

（いちばん小さい値＋いちばん大きい値）÷２

## 体験２　音センサーを使って扇風機を動かすプログラム

マイクロビットの裏面にある音センサーを使って、音がしたら扇風機が動くようにプログラミングしてみましょう。無音 0～音が大きい 255 までの数値になります。実験①では、音で LED を光らせてみましょう。実験②の「でなければ」に、LED を光らせるとプログラムの動きが見えますね！


音の測定方法

「まわりの音の大きさ」は、「入力」ブロック

今回の使うブロックカテゴリー「基本」・「入力」・「論理」・「STEM」ブロック

他にも実験してみましょう

実験①


「まわりの音の大きさ」は「入力」ブロック

「アイコンを表示」・「表示を消す」は、「基本」ブロック

実験②


マイクロビットにプログラムをダウンロードした後、USB ケーブルを抜いて、マイクロビットに電池ボックスにつなぐと、パソコンから離しても動きます。ブラウザで、「マイクロビットキーワード」で検索するとたくさんのヒントが出てきますよ！みなさん、おつかれさまでした！

## 執筆者一覧

白數　哲久
　昭和女子大学准教授
久木田　寛直
　昭和女子大学現代教育研究所研究員
星名　由美
　埼玉大学 STEM 教育研究センター /
　昭和女子大学現代教育研究所 研究員
吉澤　弘
　NPO 法人ガリレオ工房

**理科教育研究グループ研究報告書 vol.4**
**-STEM 教育のものづくりとプログラミングの融合 -**

2022 年 2 月 22 日発行

発行：現代教育研究所
編集：パープクリエイト合同会社
　　市山　実奈