動画③　「土の中の水分量が知りたい！」　ワークシート・説明資料

土の中の乾き具合をセンシングしよう！

|  |
| --- |
| センサで計測して問題解決 |

　このワークシートでは、「土の中の乾き具合をセンシングしよう！」を学習します。

　コンピュータなどで入力できるよう、Word形式のデータで保存してありますので、自分で文字を入力したり、画像を貼り付けたりしながら進めていきましょう。

　今回、皆さんは**「土の中の乾き具合が分からない」**という問題の解決に挑戦します。

それでは、Let’s プログラミング！

・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・

|  |
| --- |
| まずは準備しよう！  ・プログラミングに使うコンピュータ  ・マイクロビットV2※a　　・水分センサ　　・リード線3本　　・電池ボックス※b　　・電池※b  ※１　V1の場合でも方法を工夫することで問題解決ができます。動画を確認しましょう。  ※２　給電方法は電池ボックス以外にもあります。 |

【活動１】問題の発見と課題の設定

※「なるほど」「大事だな」と思ったところに、線を引いたり、ハイライトをしたりしてみましょう

皆さんの家では、なにか植物を育てていますか？　植物がよく育つためはいくつか必要なことがあります。そのうちの一つに「水」があります。

植物には、単純にたくさん水をあげればいいわけではありません。例えば、ミニトマトを甘くするには「土をいい具合に乾燥させて、根っこにストレスをかけるとよい」と言われています。

しかし、土がいい具合に乾燥しているかどうかは、どうしたら確認し、さらに「いい具合だ！」と判断できるのでしょう？指を土に入れてみる？それとも土を掘ってみましょうか？なんだか中学生の私達にはうまくできそうもなさそうです。

そこでこの動画では「土の中の乾き具合が分からない」という問題に対して「中学生の私たちでも分かるよう自動で知らせてくれる機械をつくる」という課題を設定し、その解決に挑戦することにします。

【活動２】解決策の構想

　そこで、「中学生の私たちでも分かるよう自動で知らせてくれる」方法を考えます。今回は「水分量を調べられるセンサの情報をもとにプログラムで判断し、土の乾き具合を私たちに知らせてくれる機械[[1]](#footnote-1)」、を考えてみましょう。どんな方法で、どんなことを知らせてくれたら、問題を解決できそうでしょうか？

|  |  |
| --- | --- |
| 考えた方法記入欄 | 〇　←ここに入力しましょう。  〇  〇 |

先生は、こう考えました。先生の考えを参考にしたい人は、下の記入欄の文字色を白色から黒色に変えてみてください。

|  |  |
| --- | --- |
| 先生の考えた方法 | 〇　水分が不足すると、ブザーでお知らせしてくれる。  〇　センサの値が大丈夫な時はLEDが青く光っていて、危険な値になったらLEDが赤く点滅するようになる。  〇　センサの値を通信で教えてくれる。 |

【活動３】問題の原因を解決しよう

　それでは、考えた方法を盛り込んだプログラムを作ってみましょう。

　みなさんが思い通りのプログラムが作れるようになるため、先生の考えた方法を、いくつかのレベルに分けて紹介します。動画を見ながら、手順にそって一緒にプログラミングしてみましょう。

【LEVEL１：ボタンを押したら計測しよう】

　このプログラムは、「マイクロビット本体のＡボタンが押されたら、水分センサの値を読み取り、その値をLEDで表示させる」ものです。基本的なブロックの種類と役割を知りたい人は，このワークシートの後半に載せたので、そちらを見てください。

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, チャットまたはテキスト メッセージ

自動的に生成された説明

（プログラムの説明）

〇　Ａボタンが押されたら、はさまれたブロックを順番に実行していきます。今回はさんだブロックは次のとおりです。

(1) 文字「wait...」を表示させます。センサの値を読み取ったマイクロビット本体が、人間のように考えているフリをしているようにしてみました。

(2) 文字の次に一呼吸おいて数字を表示させようと、「一時停止」を入れました。

(3) 水分センサの値を表示させます。センサの値は「高度なブロック」にある「入出力端子」から、「アナログ端子を読み取る」を選びます。端子は、センサをつないでいる端子にするため、今回は「端子P0」を選びます。

（水分センサと表示した数字の説明）

今回使用する水分センサは「土壌湿度センサ」として市販されています。

土の中の水分量について、２本の電極の間の抵抗値（電気抵抗）から測定し、アナログ値を返します。動画における「802」とは、これを示したものです。土の中が湿っていると抵抗値が小さくなるため電気が通りやすくなり、センサの値（アナログ値）は大きくなります。

このアナログ値は、何かの世界共通の値、というわけではありません。どんなセンサを使うのか、どんなコンピュータを使うのか、によっても変わってきます。そのため、今回使っているセンサとコンピュータの場合、どの値の時に、知りたいこと（今回は土の乾き具合）と言えるのか、を調べる必要があります。このことは、LEVEL２以降で考えていきます。

なお、動画内のセンサはシンプルな構造のため、比較的安く入手できますが、劣化や腐食が発生するため、注意が必要です。

【LEVEL２：リアルタイムで計測しよう】

　このプログラムは、「ボタンを押さなくても、ずっとセンサの値を表示させる」ものです。

　どんなことを繰り返したいのか考えて、プログラムを組んでみましょう。

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, チャットまたはテキスト メッセージ

自動的に生成された説明

②

①

（プログラムの説明）

①　最初に顔文字をLEDで表示させます。②では、ずっとセンサの値を表示させているため、プログラムが実行されたことを分かりやすく区別するため、最初だけ数字ではないものを表示させました。

②　センサの値をLEDで表示させます。乾き具合の変化がリアルタイムで分かるように、「ずっと」の命令を使っています。センサの値が表示し終わるまでやや時間がかかるため、「一時停止」の命令を入れることで分かりやすくしています。

【LEVEL３：条件によってブザーONしよう】

　このプログラムは、「もし土が乾いたらブザーを鳴らして教えてくれる」ものです。

グラフィカル ユーザー インターフェイス, アプリケーション

自動的に生成された説明

（変数とは）

　変数とは、プログラムで使用する「値をしまっておく箱のようなもの」です。箱には数字や文字列などの値を入れることができます。今回のプログラムでは、センサで読み取った数値を、「水分量」という箱に入れておき、箱に入っている数字を、分岐処理の判断に利用しています。

　LEVEL３において、変数を使ったときと使わないときで、命令がどう変わるか考えてみましょう。

（ア）：変数を使わないパターン

グラフィカル ユーザー インターフェイス

自動的に生成された説明

（イ）：変数を使うパターン

グラフィカル ユーザー インターフェイス, アプリケーション

自動的に生成された説明

上の（ア）と（イ）は、使っているブロックが少しちがいますが、同じ動作をします。

しかし、プログラムにおいて、命令を更に増やし、たくさんの処理をしたり機能を増やしたりする場合には、命令がどんどん複雑になっていきます。そのため、変数を使い、プログラムの構造を分かりやすくしたり、人がプログラムを読み取りやすくなったりすることが求められます。

（しきい値とは）

　動画のプログラムでは、センサの値が800を上回ったらブザーが止まりました。つまり、センサの値が800より低ければ「土の中が乾いている」と判断したことになります。

　このように、「何かが起こるために必要な、ちょうどの量やレベルの境目とする値」を「しきい値」と呼びます。世の中にある多くのシステムは、温度や明るさ、距離などのセンサから読み取った値に対してしきい値を決め、センサの値としきい値を比べて何かを制御するようなプログラムが組まれています。

【応用編】データロガーでセンサの値を記録しよう

　水分センサのより適切なしきい値を決めるために、今回は「データロガー」という拡張機能を使います。この機能を使うことで、一定時間センサの値を記録することができます。土の乾き具合を目で見たり、葉っぱの色やしおれ具合を観察したりすることと、その数値のデータロガーの数値を比較、確認することなどしながら、しきい値を決めてみましょう。データを表やグラフなどに直して、比較、確認してみると分かりやすいかもしれませんね。

　ここでは、データロガーを使うためのプログラムについて解説します。

グラフィカル ユーザー インターフェイス が含まれている画像

自動的に生成された説明

　動画では、このプログラムが組まれていました。

　まとまりごとに解説しますので、ワークシートの最後にあるブロックの紹介とあわせて、プログラムの理解を深めてください。

|  |  |
| --- | --- |
| グラフィカル ユーザー インターフェイス が含まれている画像  自動的に生成された説明 | ・プログラムが動作する一番はじめに、記録を止めることで誤動作を防ぐ命令です。  ・あらかじめ変数「logging」を作成しました。データを記録する、という意味をもつ変数にしました。  ・目で見ても分かるように、LEDで「×」を点灯させます。  ・コラム（行）を設定します。名前は、水分（wet）の頭文字で「w」としました。 |
| グラフィカル ユーザー インターフェイス が含まれている画像  自動的に生成された説明 | ・ボタンＡが押されたら記録を始める命令です。  ・変数「logging」を「真」にすることで、変数「logging」を使用可能な状態にする、という意味をもたせました。  ・記録の開始を表す「✓」を0.2秒間表示させます。 |
| グラフィカル ユーザー インターフェイス が含まれている画像  自動的に生成された説明 | ・ボタンBが押されたら記録を停止する命令です。  ・変数「logging」を「偽」にすることで、変数「logging」を使えない状態にする、という意味をもたせました。  ・記録の停止を表す「×」を0.2秒間表示させます。 |
| グラフィカル ユーザー インターフェイス が含まれている画像  自動的に生成された説明 | ・マイクロビット本体に記録できるデータは、量に限りがあるので、データが満杯になった場合を想定しておきます。  ・もしデータが満杯になったら、記録を停止させます。ボタンBを押したときと同じ命令です。  ・見た目で区別できるよう、LEDはすべて点灯させます。 |
| グラフィカル ユーザー インターフェイス が含まれている画像  自動的に生成された説明 | ・データやマイクロビット本体などに不具合が発生した万が一のときも想定してみました。  ・ボタンＡとBを同時押しした場合、「delete log」で記録を削除します。  ・記録を削除する様子は本体を見ても分からないため、LEDでドクロマークを表示させます。  ・記録が削除されたため、改めてコラム（行）を「w」として設定します。 |

グラフィカル ユーザー インターフェイス が含まれている画像

自動的に生成された説明

・１秒（＝1000ミリ秒）ごとに、はさまれた命令を実行します。

・「もしloggingなら」は、「もしlogging＝真なら」と同じ意味になります。つまり、変数「logging」が使用可能な状態なら、下にはさまれた命令を実行する、となります。

・二つ目の変数「wet」を設定します。代入するものは、アナログ端子P0の値としました。アナログ端子P0には水分センサが取り付けられているため、変数「wet」の値は水分センサの値となります。

・記録するデータの設定を決めます。ここでは、コラム（行）に対応する値（value）を、変数「wet」としました。これで、水分センサの値を次のような表の形で記録できるようになります。

|  |  |
| --- | --- |
| 時間（秒） | ｗ（水分センサの値） |
| １ | 771 |
| ２ | 792 |
| ３ | 853 |
| … | … |

（例）

データロガーで記録した値とトマトやあなたの決めた作物の育成状況を比較することを通して、ちょうど良い「土の乾き具合」を維持し、異常になったら私たちに知らせてくれるしきい値を決めてみましょう。

|  |  |  |
| --- | --- | --- |
|  | ｗ（水分センサの値） | どうしてそう判断したかと言うと |
| 土が乾きすぎと考えられる値 | （自分で入力しよう） | （自分で入力しよう） |
| 土が水分を含みすぎと考えられる値 | （自分で入力しよう） | （自分で入力しよう） |

【応用編】より便利なシステムに向けて

コンピュータ

アクチュ

エータ

センサ



|  |  |  |
| --- | --- | --- |
| 温度？　光？　　色？ | （マイクロビット本体） | モータ？　ブザー？ |

　あなたが設定した課題を見直し、ここまでの学びを生かそうとすると、更に便利なシステムが作れるかもしれません。

|  |
| --- |
| 例）  ・出力を「LED・ブザーで知らせる」から「モータやポンプを動かして水やりする」に変えてみる。  ・遠く離れた人の端末に無線通信機能で通知する機能を追加する。  ・データロガーで集めた情報を分析して、水を与える量やタイミングが何時なのか、分析する。  ・もし、1日以上データロガーの数値が変わらないようだったら、センサが腐食して壊れている可能性があるので、そのことを通知で知らせてくれる。 |

　マイクロビットに接続可能なものをWeb検索してかまわないので、新たなイメージをふくらませ、より便利なシステムにする改良案を考えてみましょう。

|  |
| --- |
|  |

【MakeCodeで使用するブロックについて（一部紹介）】

（基本ブロック）

|  |  |
| --- | --- |
| テキスト が含まれている画像  自動的に生成された説明 | 入力した数字をLEDで表示します。数字の代わりに、温度や明るさなど「センサの値」のブロックを入れ込むことで、センサの値を表示することもできます。 |
| テキスト が含まれている画像  自動的に生成された説明 | クリック（タップ）した部分が白くなり、その部分のLEDを点灯させます。 |
| テキスト が含まれている画像  自動的に生成された説明 | ハートや顔文字などのアイコンとして、LEDを点灯させることができます。 |
| テキスト が含まれている画像  自動的に生成された説明 | 入力した文字をLEDで表示します。半角の英数字は表示できますが、ひらがなや漢字などの全角文字は表示できません。 |
| テキスト が含まれている画像  自動的に生成された説明 | それまでに表示させていたLEDを消します。 |
| グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, チャットまたはテキスト メッセージ  自動的に生成された説明 | このブロックにはさまれている命令（ブロック）をずっと繰り返します。 |
| グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, チャットまたはテキスト メッセージ  自動的に生成された説明 | プログラムを実行すると、このブロックにはさまれた命令は最初だけ実行されます。一度実行されると、このブロックにはさまれた命令は、プログラムが終了するまで、それ以降実行されません。 |
| グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, チャットまたはテキスト メッセージ  自動的に生成された説明 | 命令を一時停止します。１秒＝1000ミリ秒なので、100ミリ秒は0.1秒となります。秒数を数字で直接入力することもできます。 |

（入力ブロック）

|  |  |
| --- | --- |
| グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, チャットまたはテキスト メッセージ  自動的に生成された説明 | マイクロビット本体の入力信号を受けたら、このブロックにはさまれた命令を実行します（分岐処理）。上はボタンが押されたとき、下は本体がゆれたときですが、「▼」を選択すると、ボタンの変更や傾き方の詳しいメニューを選ぶことができます。 |
| グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, チャットまたはテキスト メッセージ  自動的に生成された説明  グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, チャットまたはテキスト メッセージ  自動的に生成された説明 | 上の分岐処理ではなく、ただの入力信号としてのブロックです。ブロックが六角形の形をしているので、他のブロックの中で六角形の枠があれば、そこに入れ込むことができます。 |
| グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, チャットまたはテキスト メッセージ  自動的に生成された説明　　　　グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, チャットまたはテキスト メッセージ  自動的に生成された説明  グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, チャットまたはテキスト メッセージ  自動的に生成された説明　　グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, チャットまたはテキスト メッセージ  自動的に生成された説明 | マイクロビット本体のセンサから送られる値を示すブロックです。「入力」の「その他」のメニューには、さらに別のセンサを選ぶことができます。 |

（データロガー）

|  |  |
| --- | --- |
| モニター画面に映る文字のスクリーンショット  自動的に生成された説明 | データロガーで使用する内容（コラム：columns）に名前をつけます。「＋」を押すと、複数のカラムを設定できます。プログラムの最初に動作するようにするといいでしょう。  ※コラムは表における縦の「列」を指します。 |
| モニター画面に映る文字のスクリーンショット  自動的に生成された説明 | ログ（記録）を取りたいデータを設定するブロックです。「column」には、最初に設定したコラムを選択します。「value」には、データとして記録するセンサの値（ブロック）を入れることができます。コラムに対応したデータを選びましょう。 |
| モニター画面に映る文字のスクリーンショット  自動的に生成された説明 | ログを削除します。 |
| モニター画面に映る文字のスクリーンショット  自動的に生成された説明 | ログがいっぱいになった場合、ここにはさまれた命令を実行します。 |

1. このような、センサの情報を、コンピュータのプログラムに従って判断し、何かの動作をする（今回は「知らせる」）技術／システムのことを計測・制御の技術／システム、といいます。詳しくは皆さんが持っている教科書を確認してみましょう！ [↑](#footnote-ref-1)