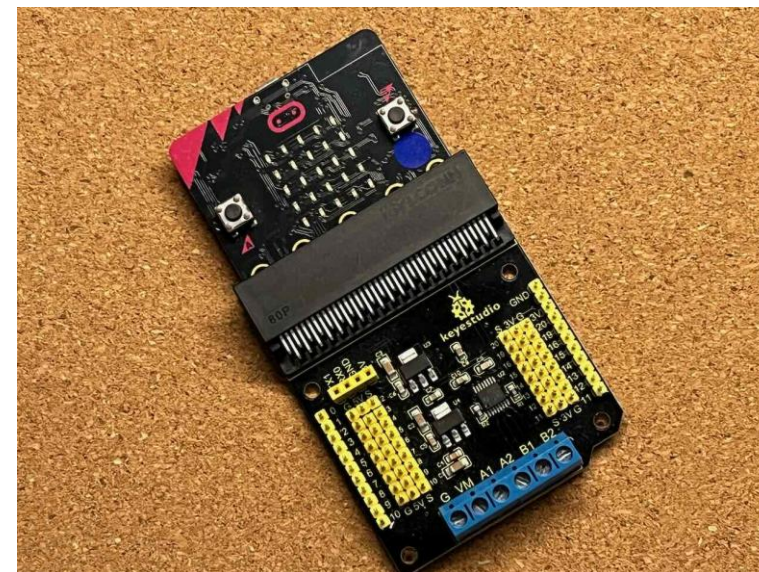


さとやまプログラミングクラブ2025

マイクロビットで「LEDグラス」をつくろう！

第3回

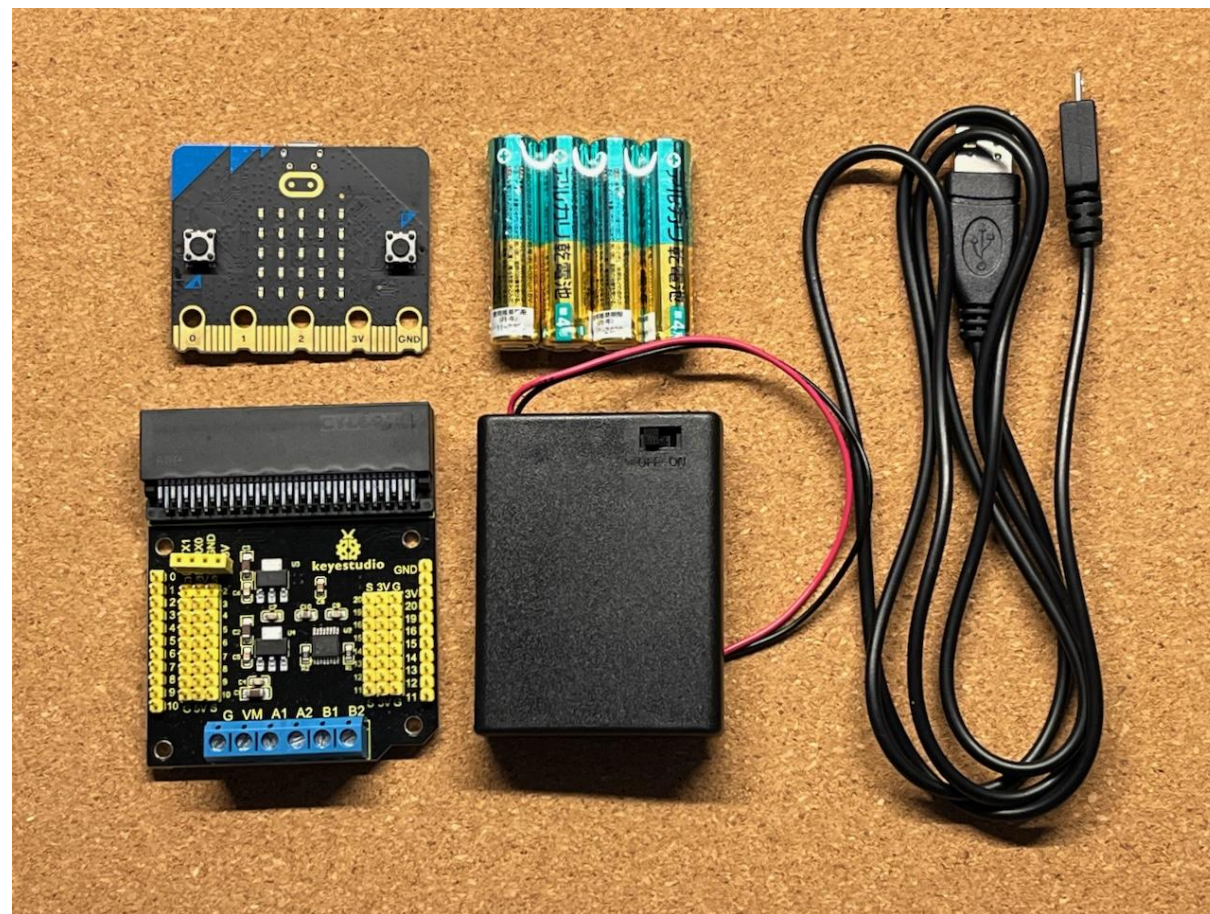


今日やること

- 「LEDグラス」のプログラムをしあげます。
- プログラムの仕様は以下のとおりです。
 - ふだんはフルカラーLEDは青色系のグラデーションで点灯します。
 - 人がちかづくとき赤色系のグラデーションにかわり、同時にクリスマスソングがながれます。
- 自分のすきなようにプログラムを改良します。

部品の確認 (マイクロビットセット)

部品	個数
マイクロビット	1
拡張ボード (KEYE STUDIO KS4033)	1
USBケーブル	1
単三x4電池ボックス	1
単三乾電池	4



部品の確認 (LEDグラス)

部品	個数
フルカラーLED	1
超音波センサ	1
インテリアストーン	2ふくら
ガラスコップ	1



※フルカラーLEDは、かたちがちがうものもあります

前回の復習

- 7個 または 8個のLEDにはグラデーションで色が設定されます。
- 超音波センサからの距離がちかいときは赤色系のグラデーション、とおいときには青色系のグラデーションになります。
- 0.1秒ごとに設定した色がとなりにずれていきます。

The code is organized into three sections: '最初だけ' (Initially), 'ずっと' (Forever), and 'でなければ' (If not).

- 最初だけ (Initially):**
 - 変数 超音波フラグ を 1 にする (Set ultrasonic flag to 1)
 - 変数 strip を 端子 P2 に接続しているLED 7 個のNeoPixel (モード RGB (GRB順)) にする (Set strip to 7 NeoPixels on terminal P2, mode RGB (GRB order))
 - 呼び出し 色設定1 (Call color setting 1)
- ずっと (Forever):**
 - もし [Grove - 超音波距離センサー] 距離(cm)を読み取る 端子 PB < 50 なら (If distance < 50 cm)
 - もし 超音波フラグ = 1 なら (If ultrasonic flag = 1)
 - 呼び出し 色設定2 (Call color setting 2)
 - 変数 超音波フラグ を 2 にする (Set ultrasonic flag to 2)
 - でなければ (If not)
 - もし 超音波フラグ = 2 なら (If ultrasonic flag = 2)
 - 呼び出し 色設定1 (Call color setting 1)
 - 変数 超音波フラグ を 1 にする (Set ultrasonic flag to 1)
 - strip を設定した色で点灯する (Turn on strip with set color)
 - strip に設定されている色をLED 1 個分ずらす (ひとまわり) (Shift color of 1 LED)
 - 一時停止 (ミリ秒) 100 (Pause 100 ms)

別のうごきを追加

- 今のうごきに、「超音波センサからの距離がちかくなったとき」に「アイコンを点滅」させるといううごきをくわえてみます。
- 「アイコンを点滅」は、「ハートを0.1秒表示→ハートを0.1秒消す」を3回くりかえすことにします。以下のように関数にしておきます。



この関数を実行するのに
0.6秒 かかります。

別のうごきを追加

- まずはふつうに、「超音波センサからの距離がちかくなったとき」に「アイコンを点滅」させる関数をよびだしてみます。

ずっと

もし [Grove - 超音波距離センサー] 距離(cm)を読み取る 端子 P8 < 50 なら

もし 超音波フラグ = 1 なら

呼び出し アイコンを点滅

呼び出し 色設定2

変数 超音波フラグ を 2 にする

でなければ

もし 超音波フラグ = 2 なら

呼び出し 色設定1

変数 超音波フラグ を 1 にする

strip を設定した色で点灯する

strip に設定されている色をLED 1 個分ずらす(ひとまわり)

一時停止(ミリ秒) 100

最初だけ

変数 超音波フラグ を 1 にする

変数 strip を 端子 P2 に接続しているLED 7 個のNeoPixel(モード RGB(GRB順)) にする

呼び出し 色設定1

関数 色設定1 strip の 0 番目のLEDを ...

関数 色設定2 strip の 0 番目のLEDを ...

関数 アイコンを点滅 くりかえし 3 回 アイコンを...

ここに追加

プログラムは上から順に実行していくので、「アイコンを点滅」がおわるまで次にすすみません。
このため、アイコンが点滅している間、フルカラーLEDの色は変わりません。

ふたつのうごきを同時に実行

- 「ずっと」ブロックを **ふたつ** つかいます。

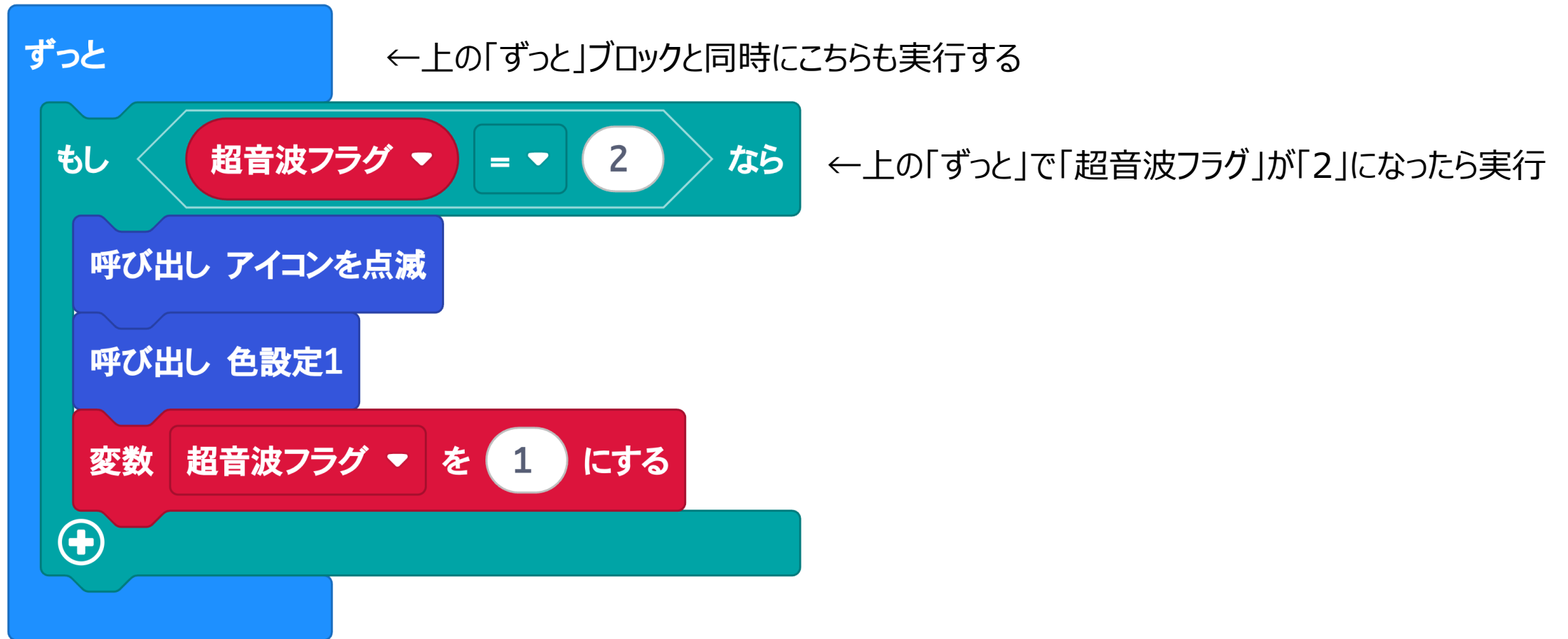
```
ずっと  
もし [Grove - 超音波距離センサー] 距離(cm)を読み取る 端子 P8 <> 50 なら  
もし 超音波フラグ = 1 なら  
  呼び出し 色設定2  
  変数 超音波フラグ を 2 にする  
  strip を設定した色で点灯する  
  strip に設定されている色をLED 1 個分ずらす(ひとまわり)  
一時停止(ミリ秒) 100
```

```
ずっと  
もし 超音波フラグ = 2 なら  
  呼び出し アイコンを点滅  
  呼び出し 色設定1  
  変数 超音波フラグ を 1 にする
```

```
最初だけ  
変数 超音波フラグ を 1 にする  
変数 strip を 端子 P2 に接続しているLED 7 個のNeoPixel(モード RGB(GRB順)) にする  
呼び出し 色設定1  
関数 色設定1 strip の 0 番目のLEDを ...  
関数 色設定2 strip の 0 番目のLEDを ...  
関数 アイコンを点滅 くりかえし 3 回 アイコンを...
```

上の「ずっと」がうごいているのと同時に、下の「ずっと」もうごきます。このため、下の「ずっと」で「アイコンを点滅」させているあいだも、上の「ずっと」の命令でフルカラーLEDの色が変わります。

下の「ずっと」ブロックの説明



距離がちかくなったら音楽をならす

- 「超音波センサからの距離がちかくなった」とき、フルカラーLEDの色をかえるのと同時にクリスマスソングをならします。
- 「ジングルベル」関数をつくり、中にメロディーをいれます。
- 「アイコンを点滅」関数のかわりに「ジングルベル」関数をよびだします。

The code is organized into three main sections:

- 最初だけ (Initially):** Sets the '超音波フラグ' (Ultrasonic Flag) variable to 1 and calls the '色設定1' (Color Setting 1) function.
- ずっと (Forever):** A loop that checks if the distance from the Grove Ultrasonic Sensor is less than 50 cm. If true, it checks if the '超音波フラグ' is 1. If so, it calls '色設定2' (Color Setting 2), sets the '超音波フラグ' to 2, and then calls the 'ジングルベル' (Jingle Bell) function. After the function call, it sets the LED color and then pauses for 100 milliseconds.
- ずっと (Forever):** A second loop that checks if the '超音波フラグ' is 2. If true, it calls the 'ジングルベル' function, then '色設定1' (Color Setting 1), and finally sets the '超音波フラグ' back to 1.

Two custom functions are defined:

- 関数 色設定1:** Sets the color of the 0th LED.
- 関数 色設定2:** Sets the color of the 0th LED.

The 'ジングルベル' (Jingle Bell) function contains a sequence of musical notes and rests:

- 鳴らす (Play sound): 真ん中のシ (Middle C), 高さ (Pitch): 真ん中のシ, 長さ (Duration): 1/4 拍 (Quarter note), 終わるまで (Until finished).
- 休符 (Rest): 1/4 拍 (Quarter rest).
- 鳴らす (Play sound): 真ん中のシ (Middle C), 高さ (Pitch): 真ん中のシ, 長さ (Duration): 1/4 拍 (Quarter note), 終わるまで (Until finished).
- 休符 (Rest): 1/4 拍 (Quarter rest).
- 鳴らす (Play sound): 真ん中のシ (Middle C), 高さ (Pitch): 真ん中のシ, 長さ (Duration): 1/4 拍 (Quarter note), 終わるまで (Until finished).
- 休符 (Rest): 1/2 拍 (Half rest).
- 鳴らす (Play sound): 真ん中のシ (Middle C), 高さ (Pitch): 真ん中のシ, 長さ (Duration): 1/4 拍 (Quarter note), 終わるまで (Until finished).
- 休符 (Rest): 1/4 拍 (Quarter rest).
- 鳴らす (Play sound): 真ん中のシ (Middle C), 高さ (Pitch): 真ん中のシ, 長さ (Duration): 1/4 拍 (Quarter note), 終わるまで (Until finished).
- 休符 (Rest): 1/4 拍 (Quarter rest).
- 鳴らす (Play sound): 真ん中のシ (Middle C), 高さ (Pitch): 真ん中のシ, 長さ (Duration): 1/4 拍 (Quarter note), 終わるまで (Until finished).
- 休符 (Rest): 1/4 拍 (Quarter rest).
- 鳴らす (Play sound): 真ん中のシ (Middle C), 高さ (Pitch): 真ん中のシ, 長さ (Duration): 1/4 拍 (Quarter note), 終わるまで (Until finished).
- 休符 (Rest): 1/4 拍 (Quarter rest).
- 鳴らす (Play sound): 上のレ (G4), 高さ (Pitch): 上のレ, 長さ (Duration): 1/4 拍 (Quarter note), 終わるまで (Until finished).
- 休符 (Rest): 1/4 拍 (Quarter rest).
- 鳴らす (Play sound): 真ん中のソ (F#4), 高さ (Pitch): 真ん中のソ, 長さ (Duration): 1/2 拍 (Half note), 終わるまで (Until finished).
- 休符 (Rest): 1/4 拍 (Quarter rest).
- 鳴らす (Play sound): 真ん中のラ (G4), 高さ (Pitch): 真ん中のラ, 長さ (Duration): 1/4 拍 (Quarter note), 終わるまで (Until finished).
- 休符 (Rest): 1/4 拍 (Quarter rest).
- 鳴らす (Play sound): 真ん中のシ (Middle C), 高さ (Pitch): 真ん中のシ, 長さ (Duration): 1 拍 (Whole note), 終わるまで (Until finished).
- 休符 (Rest): 1 拍 (Whole rest).

これでLEDグラス用プログラムは完成です！

プログラムの調整

- 自分のすきなように、プログラムを改良してみましょう。
- 「超音波センサ」のかわりに「光センサ」や「マイク」をつかって、「くらくなったときに色をかえる」「うるさくなったときに色をかえる」ことなどもできます。
- フルカラーLEDのひからせかたをかえることもできます。
- ほかのメロディーをならすこともできます。

【ノウハウ】マイクロビットにはいつているセンサ

- **【温度センサ】** 温度をしらべます。
 - 「入力」>「温度（℃）」
- **【光センサ】** あかるさをしらべます。
 - 「入力」>「明るさ」
- **【加速度センサ】** かたむきや速度の変化をしらべます。
 - 「入力」>「ゆさぶられたとき」など
- **【コンパス】** 方角や磁力をしらべます。
 - 「入力」>「方角（°）」など
- **【タッチセンサ】【タッチロゴ】** 端子、ロゴにタッチしているかしらべます。
 - 「入力」>「端子 P0 がタッチされている」など
- **【マイク】** 音量を調べます。
 - 「入力」>「まわりの音の大きさ」など

【ノウハウ】センサをかえた例

- 「超音波センサ」のかわりに「光センサ」をつかって、くらくなったときにLEDの色をかえます。

The image displays a Scratch script for controlling an LED strip. It is divided into three main sections: initialization, a primary loop, and a secondary loop.

- 最初だけ (Initially):** This section sets up the environment. It initializes the '超音波フラグ' (Ultrasonic Flag) variable to 1, configures the 'strip' variable to control 7 NeoPixel LEDs on terminal P2 in RGB mode, and calls the '色設定1' (Color Setting 1) function.
- ずっと (Forever):** The first loop contains a conditional check. A callout box labeled '光センサ' (Light Sensor) points to the '明るさ < 50' (Brightness < 50) condition. If this condition is met, it checks if '超音波フラグ' is 1. If true, it calls '色設定2' (Color Setting 2), sets the '超音波フラグ' to 2, and then uses 'strip' blocks to set the LED color and turn on the LEDs. A 100ms delay follows.
- ずっと (Forever):** The second loop checks if '超音波フラグ' is 2. If true, it calls 'ジングルベル' (Jingle Bell), '色設定1' (Color Setting 1), and sets '超音波フラグ' back to 1.

【ノウハウ】NeoPixel拡張機能のブロック

- **【strip をレインボーパターン (色相 1 から 360) に点灯する】** ストリップを虹色のグラデーションで点灯させます。
- **【strip を 赤 色に点灯する】** ストリップを指定した色で点灯させます (すべてのLEDが同じ色になります)。
- **【strip を設定した色で点灯する】** あらかじめ設定した色でストリップを点灯させます。
- **【strip の設定を解除する】** ストリップに対し、すべてのLEDをオフ (黒色) に設定します。
- **【strip に設定されている色をLED 1 個分ずらす】** ストリップに設定されている色情報をとなりのLEDにずらします。あいたLEDにはオフ (黒色) になります。
- **【strip に設定されている色をLED 1 個分ずらす (ひとまわり)】** ストリップに設定されている色情報をとなりのLEDにずらします。あいたLEDには先頭のLEDの色情報が入ります。
- **【strip の 0 番目のLEDを 赤 色に設定する】** ストリップの各LEDに、個別に色を設定します。
- **【strip の明るさを 255 に設定する】** ストリップの明るさを 0 ~ 255 の数値で設定します。
- **【RGB (赤 255 緑 255 青 255)】** 任意のRGB値でつくった色情報をつくります。

【ノウハウ】フルカラーLEDのひからせかたをかえた例

- Aボタンをおすと「シャキーン」と音になり、フルカラーLEDがひとつずつ順番に、一瞬だけ赤色にひかります。

The code consists of three main sections:

- 最初だけ (Initially):** A block to initialize the LED strip: "変数 strip を 端子 P2 に接続しているLED 7 個のNeoPixel(モード RGB(GRB順)) にする".
- ボタン A が押されたとき (When button A is pressed):** A sequence of blocks: "鳴らす (波形) ブロックを一度だけ" (with a red waveform icon), "呼び出し フルカラーLEDをひからせる" (calling a function).
- 関数 フルカラーLEDをひからせる (Function):** A loop of 7 iterations. Each iteration: "strip の 0 番目のLEDを 赤 色に設定する" (set the 0th LED to red), "strip を設定した色で点灯する" (turn on the LED with the set color), "くりかえし 7 回" (repeat 7 times), "strip に設定されている色をLED 1 個分ずらす" (move the color to the next LED), "strip を設定した色で点灯する" (turn on the LED with the set color), and "一時停止(ミリ秒) 50" (wait 50ms).

The screenshot shows a window titled "エディター" and "ギャラリー" with a close button. It contains a list of sound effects with their respective waveforms and play buttons:

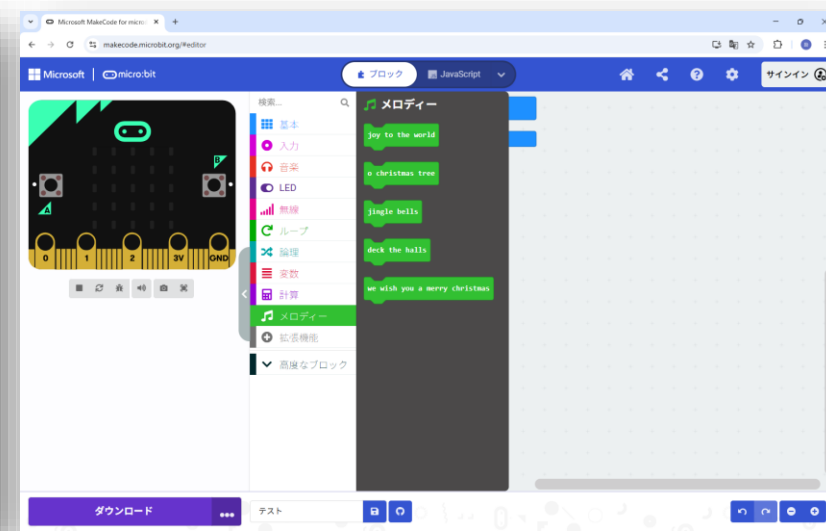
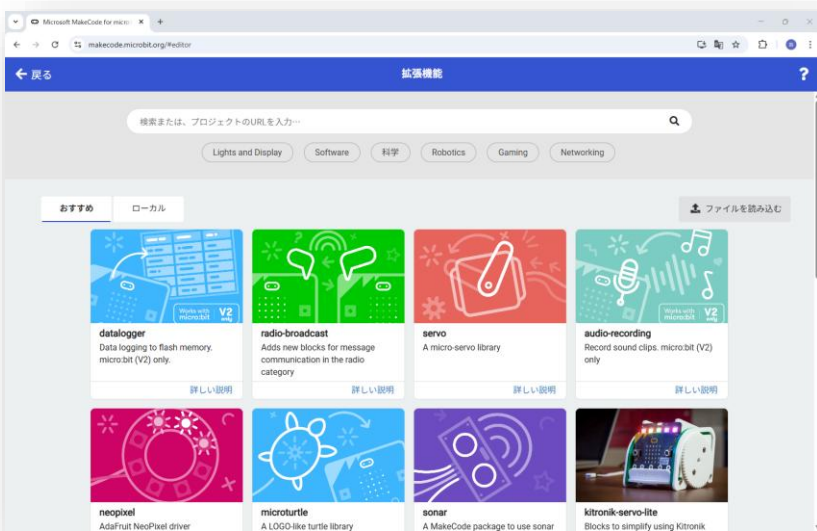
- レーザー光線 (Laser Light)
- 無線 (Wireless) - This is the selected effect, indicated by a blue arrow (4).
- ジャンプ (Jump)
- 水滴 (Water Drop)
- キックドフム (Kick Drum)
- タムタム (Tom Tom)
- スネア (Snare)
- ハイハット (Hi-Hat)

- ①「鳴らす(波形)」ブロックをつかいます。
- ②波形の部分をクリックするとウィンドウがひらきます。
- ③「ギャラリー」を選択します。
- ④「無線」をえらびます。

音のならしかた

【ノウハウ】入力済みのクリスマスソングのメロディーをつかう

- あらかじめ、クリスマスソングのメロディーを何曲か入力しておきました。
- オリジナルの拡張機能として公開していますので、よろしければつかってください。



ツールボックスで「拡張機能」をクリックします。

検索ウィンドウに

<https://github.com/yume-estc/pxt-melodies>

と入力します。

表示された「melodies」をクリックすると、
ツールボックスに「♪メロディー」という
拡張機能が表示されます。