

第2回
ビジュアルプログラミング
を学ぼう

あすたむらんど
& 四国大学

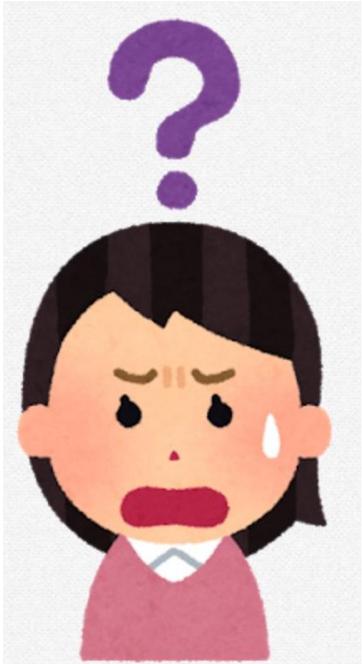
2021 / 05 / 29

第2回 フログラミングの基礎を学ぼう

ビジュアルフログラミングでジャンケン

- ・今までのフログラミングのイメージ
- ・ビジュアルフログラミングって何？
- ・基本の作法
- ・デバイス(マイコンユニット)のこと
- ・パネルにお絵かき
- ・「最初はグー」、ジャンケンゲームに挑戦！
- ・ドレミに挑戦！ ♪

・今までのプログラミングのイメージ



```
Start Page TopDesign.cysch main.c CapSense.cydwr
17 // INT1
18 CY_ISR(isr_1_isr) {
19     i ++;
20     PWM_1_WriteCompare(i);
21 }
22
23
24 // INT2
25 CY_ISR(isr_2_isr) {
26     j ++;
27     PWM_2_WriteCompare(j);
28 }
29
30
31 int main()
32 {
33     /* Start all the components */
34     PWM_1_Start();
35     PWM_2_Start();
36     UART_1_Start();
37     //UART_2_Start();
38
39     /* Enable Global Interrupts */
40     isr_1_StartEx(isr_1_isr);
41     isr_2_StartEx(isr_2_isr);
42     CyGlobalIntEnable;
43
44     for (;;)
45     {
46     }
47 }
48 /* [] END OF FILE */
49
```

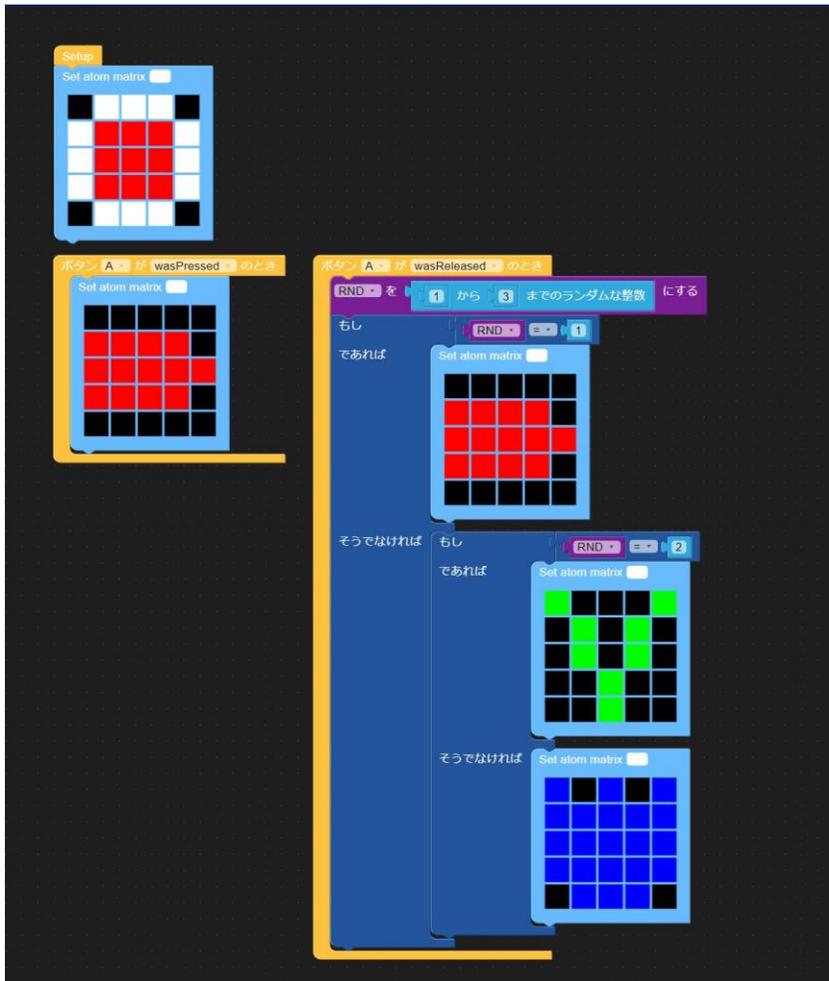
アルファベットばかりで分かりにくい

デバイスを学ばないと使えない

同時処理がむずかしい



・ビジュアルプログラミングの世界



やりたいことがすぐできる

デバイスの知識はいらない

同時処理が簡単

簡単



でもわからん

・基本の作法 とにかくやってみよう

やいたい「ブロック」をさがし、やいたい順番に並べるだけ
まずは使ってみよう



・デバイス(マイコンユニット)のこと

パネルには25個のLEDがついている

パネルにはスイッチもついている

おもて



うら

うらにはピンをさしこむ端子が出ている



・パネルにお絵かき

スタート画面を作ろう



最初の画面



好きな画面

・デバイスをセット

PCとデバイス(M5atom)をUSBケーブルで接続し、「UiFlow」を起動

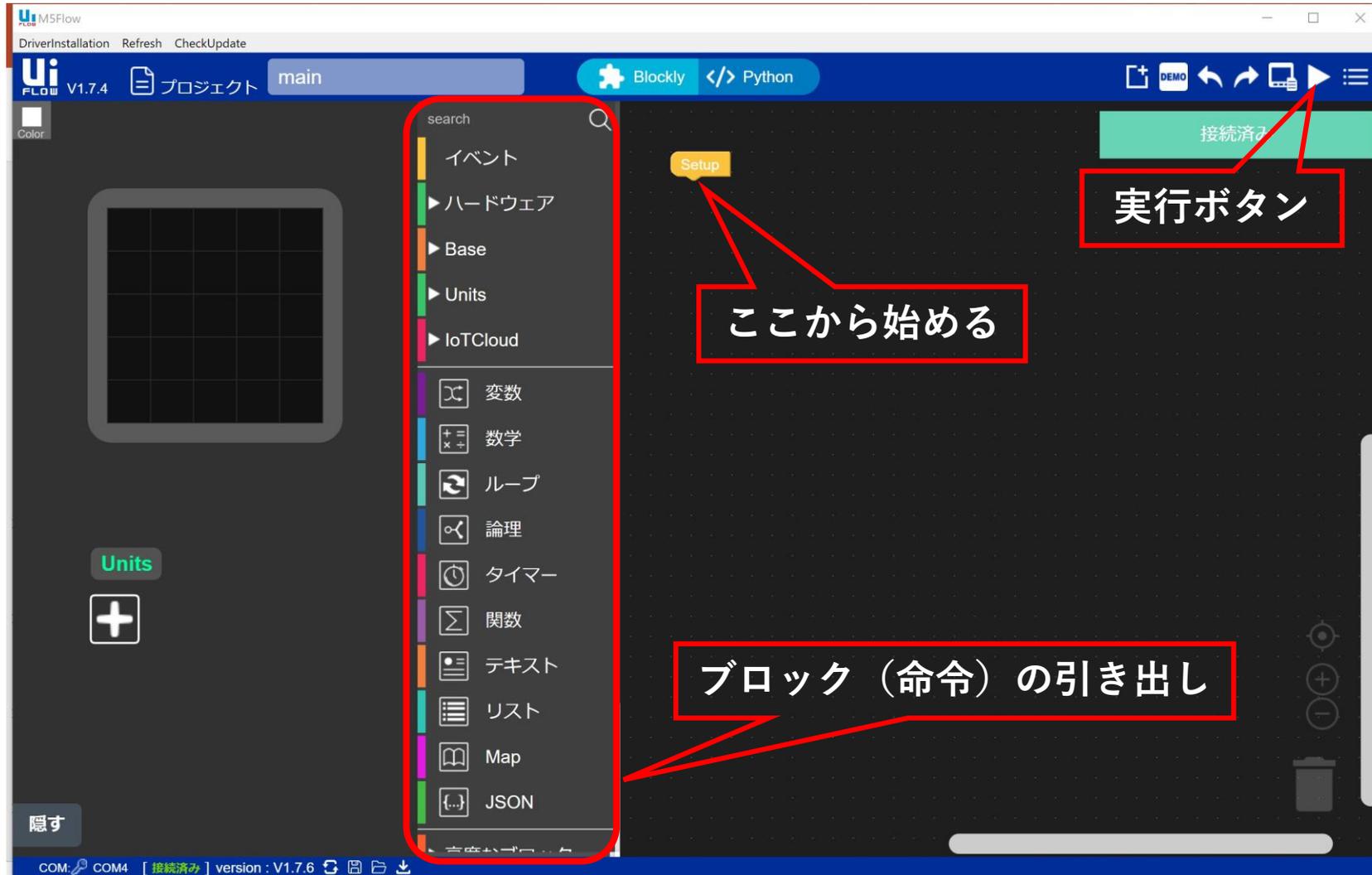


デスクトップのこのアイコン
をダブルクリックして
「UiFlow」を立ち上げると、
右の画面ができる

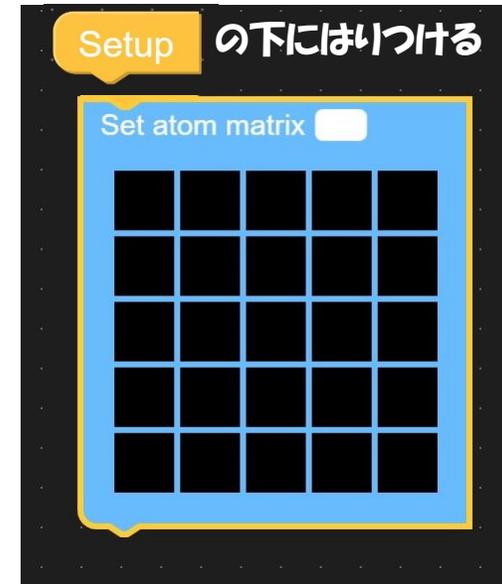


①にCOM番号が出ていることを確認して
②を押して、③を押す。

• UiFlow画面の説明



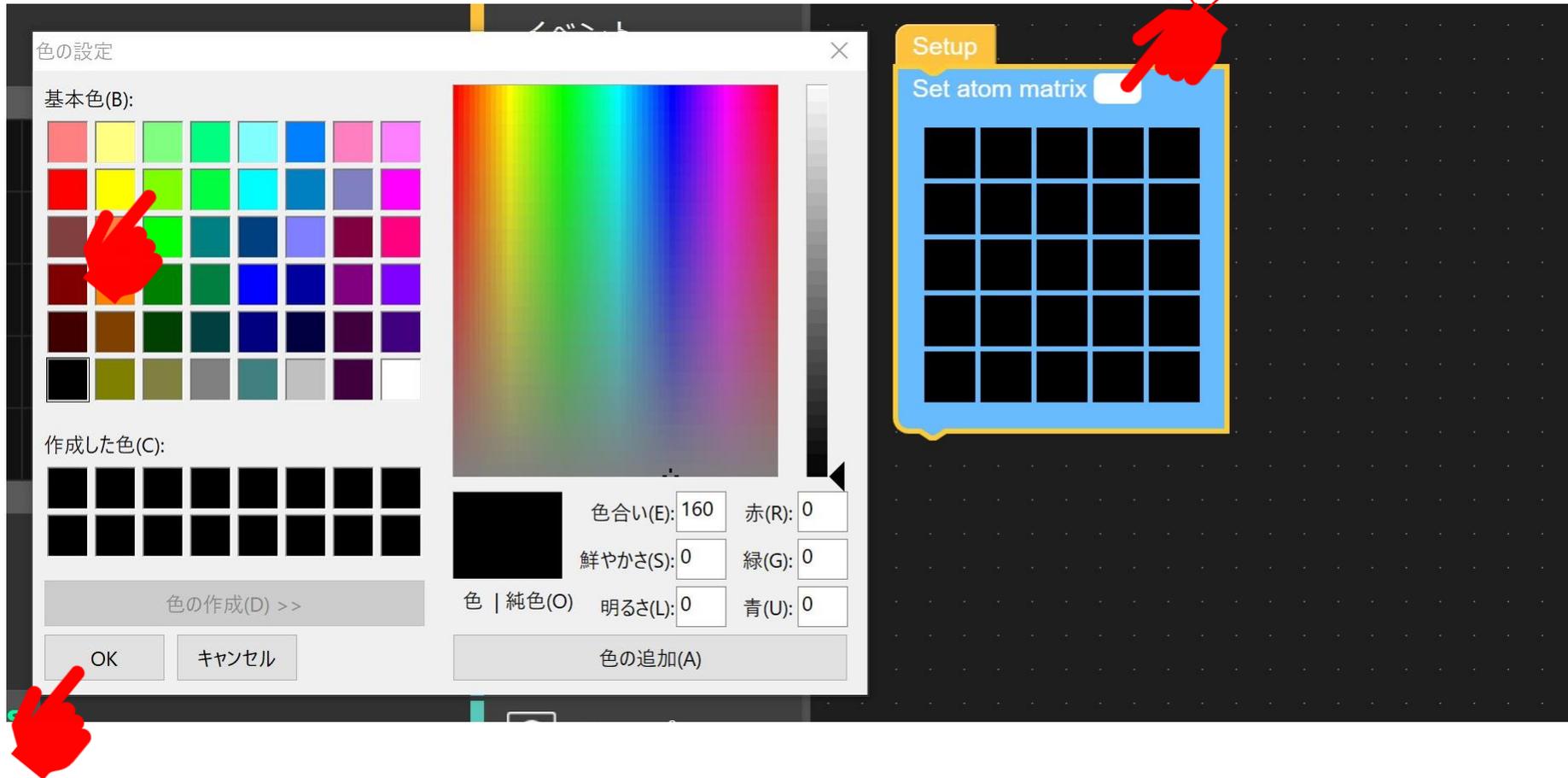
・スタート画面を作ろう①



ドラッグアンドドロップして

・スタート画面を作ろう②

①ここを押すと色パレットが開くので好きな色を選ぶ



・スタート画面を作ろう④



• LED と順番に表示するメッセージを作ろう



L と表示して

1秒止まり、

E と表示して

1秒止まり、

D と表示する

• 🖐️👉👋 ゲームに挑戦！①

考え方

1、2、3の目しかないサイコロを振って

1なら🖐️

2なら👉

3なら👋

と決める

1

?

2

3

• 🖐️🖐️🖐️ ゲームに挑戦！ ②

「**1**か**2**か**3**」ができるサイコロはどうすればつくれるんだ？

サイコロの出た目を入れておく入れ物(変数)を準備する



saikoroは

1?

2?

3?

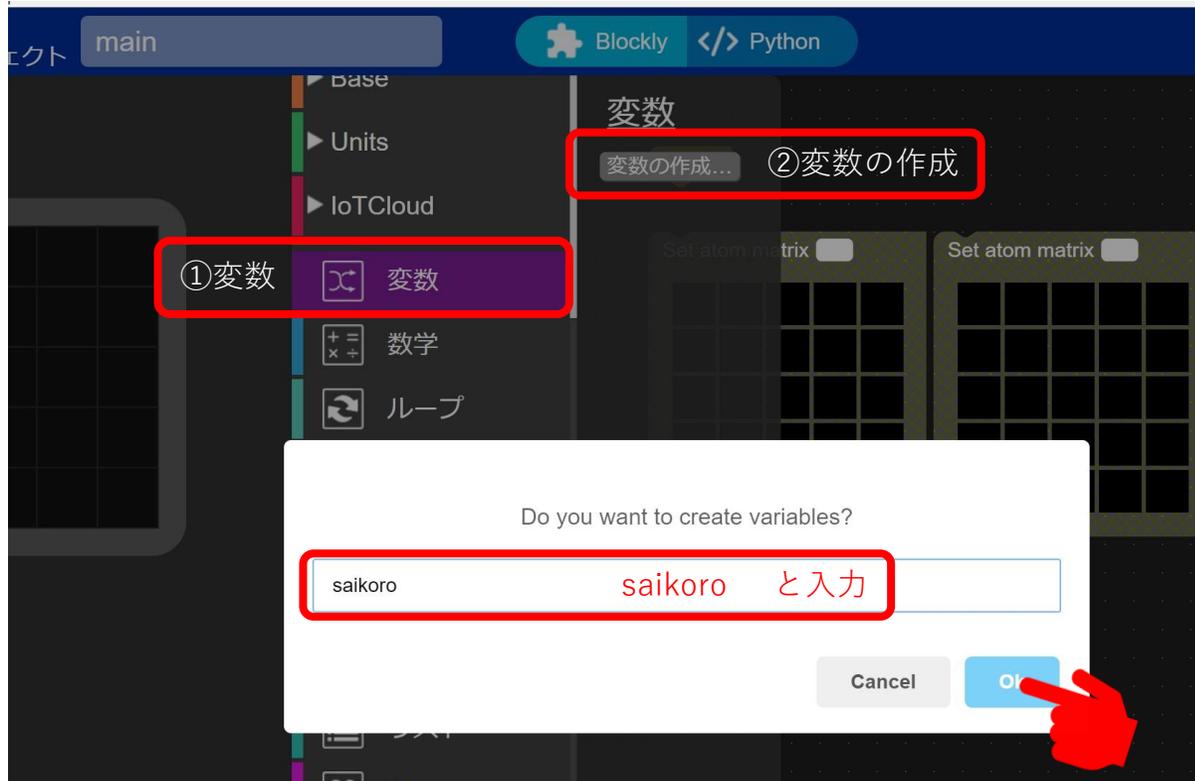
サイコロを振って(ボタンAを押す)、出た目(「**1**」または「**2**」または「**3**」)のことを
1から**3**までの乱数(らんすう)という



ボタンAが押されると、saikoro という入れ物に、**1**から**3**までの乱数
をつくり、入れる

• 🖐️🖐️🖐️ ゲームに挑戦！ ③

サイコロという名前の変数を作る



サイコロという名前の変数ができた

• 🖐️🖐️🖐️ ゲームに挑戦！ ④

ボタンAが押されると、saikoro という入れ物に、1から3までの乱数をつくり、入れる



「ボタンAが押されると」のブロック



saikoro という入れ物に、1から3までの乱数をつくり、入れる

• 🖐️🖐️🖐️ ゲームに挑戦！ ⑤

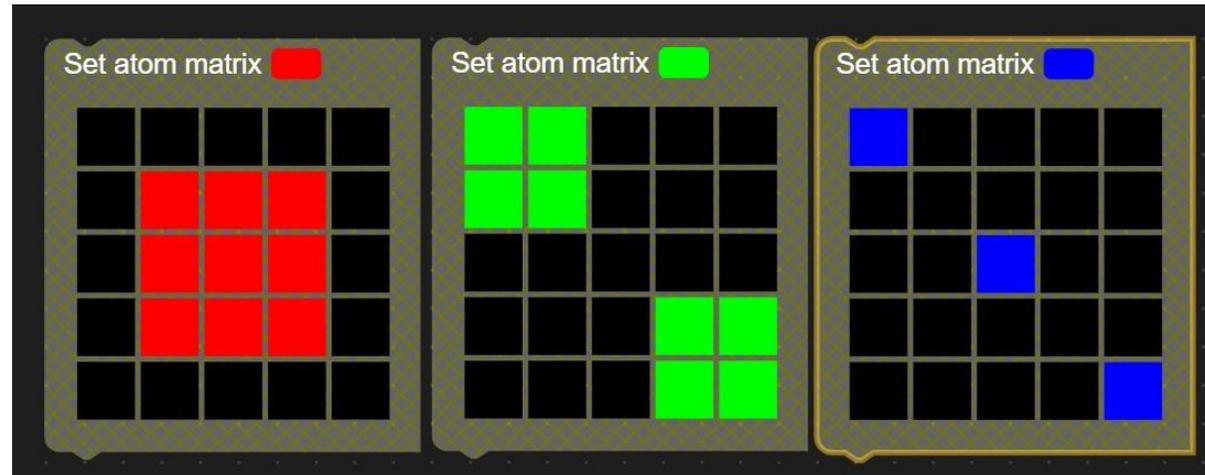
ボタンAが押されると、saikoro という入れ物に、1から3までの乱数をつくり、入れる



ちよつと休憩.....

• 🖐️✌️🖐️ ゲームに挑戦！ ⑥

サイコロのキャラクターを作ろう



1



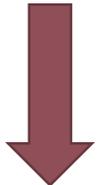
2



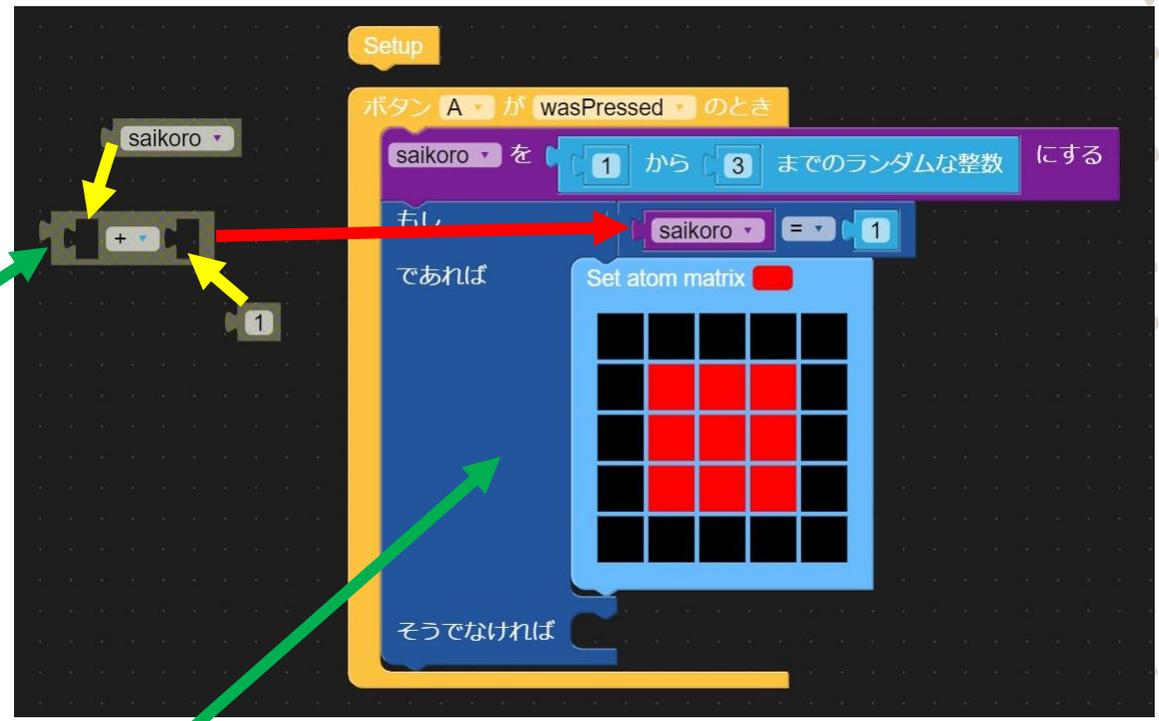
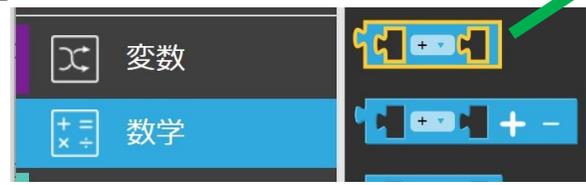
3

• 🖐️👌👋 ゲームに挑戦！ ⑦

もし、サイコロが
1なら 🖐️
2なら 👌
3なら 👋
を書き変える



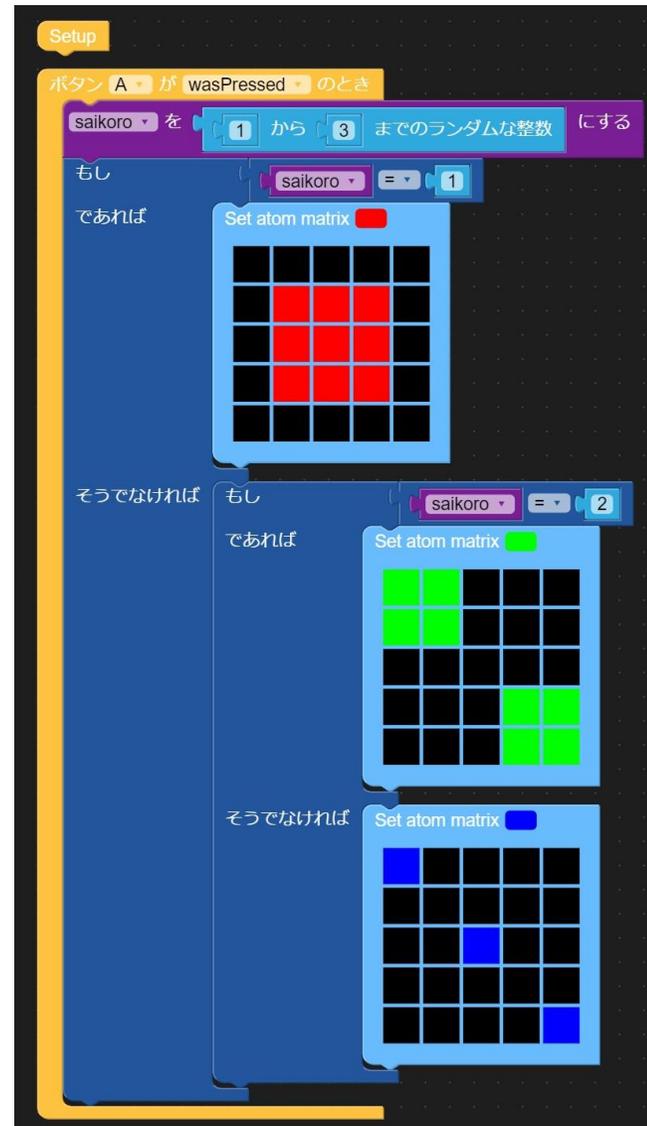
もし、サイコロが **1**
であれば 🖐️
そうでなければ、
もし、サイコロが **2**
であれば 👌
そうでなければ(3) 👋



• 🎲👋👋 ゲームに挑戦！ ⑧

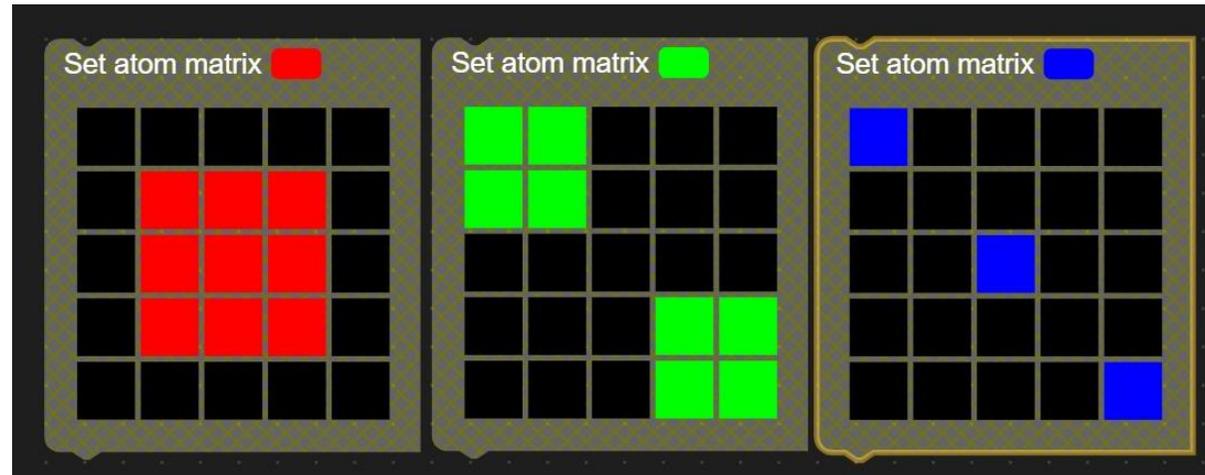
もし、サイコロが1
であれば 1
そうでなければ、
もし、サイコロが2
であれば 2
そうでなければ 3

サイコロ大会



• 🖐️ ✌️ 🖐️ ゲームに挑戦！ ⑨

🖐️ ✌️ 🖐️ のキャラクターを作ろう



1



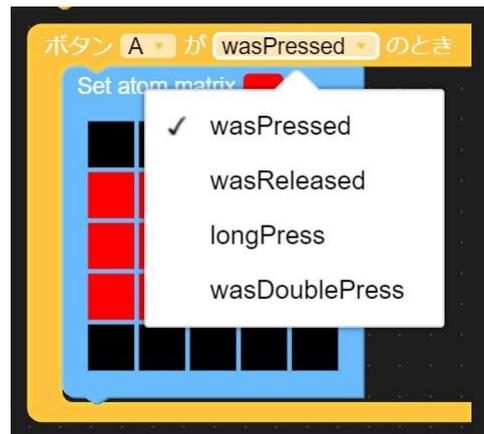
2



3

• 🖐️👉👈 ゲームに挑戦！ ⑩

「A」ボタンを押すと 最初は🖐️



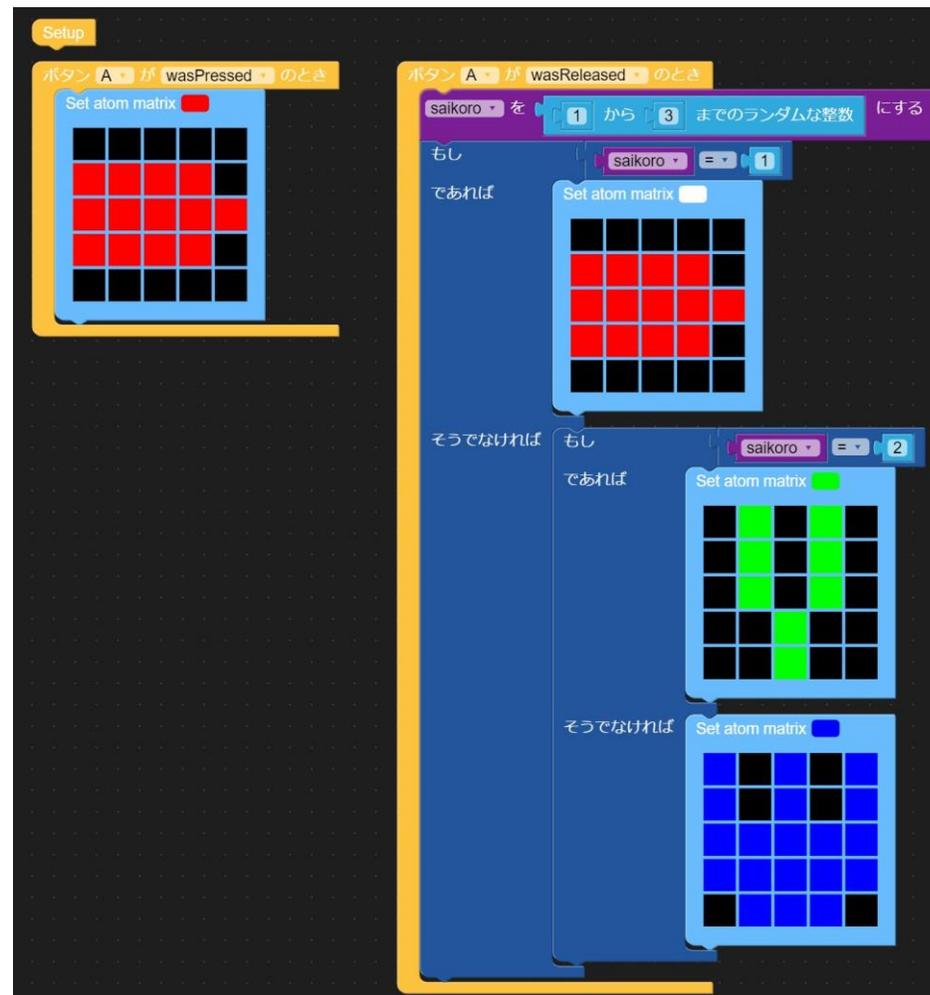
「A」ボタンを離すと じゃんけんぽん



• 🖐️👉👋 ゲームに挑戦！ ⑪

「A」ボタンを押すと 最初は🖐️

「A」ボタンを離すと じゃんけんぽん





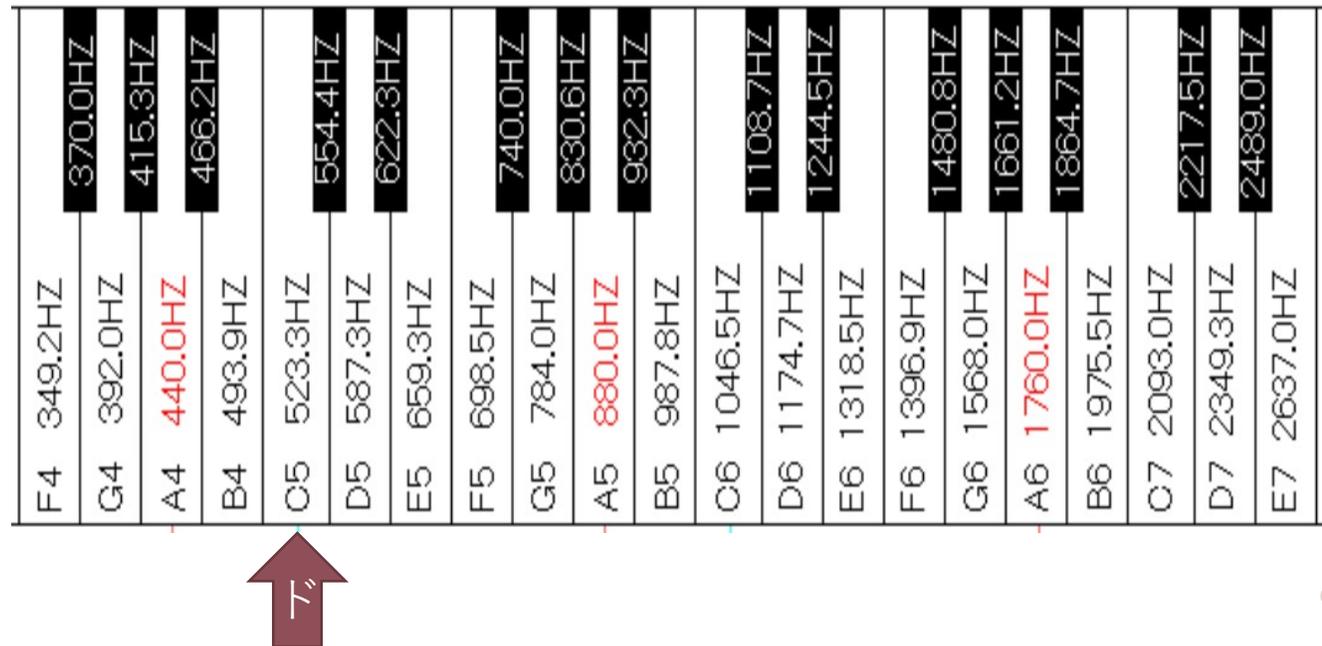
・ドレミに挑戦！ ①

スタートすると、「スタート画面」を
表示して、ドレミ・・・と音を鳴らす

音の基本は周波数(Hz)ヘルツ

鍵盤と周波数の関係

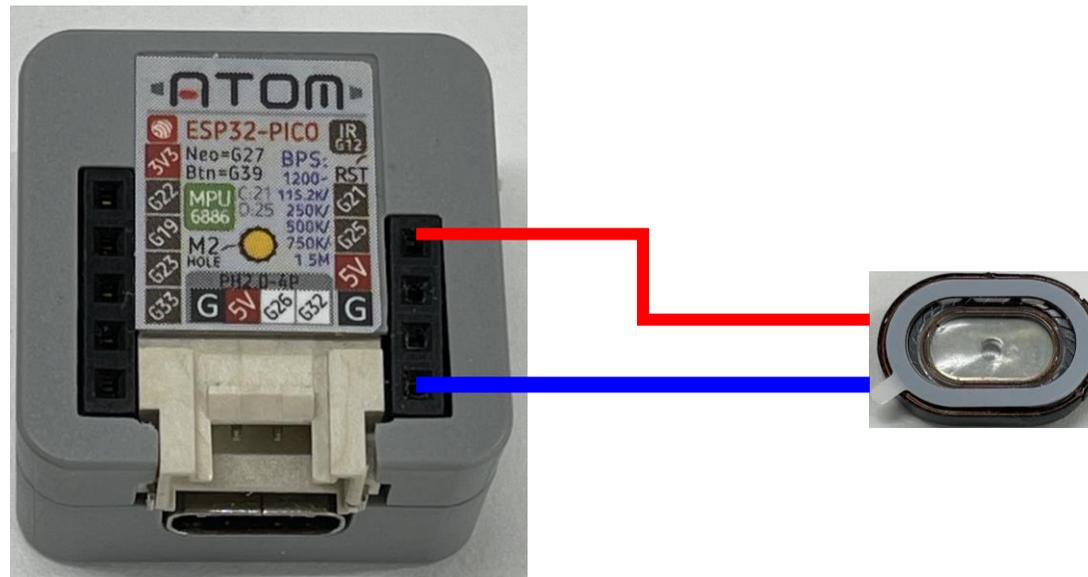
1秒間に何回振動するか



・ドレミに挑戦！ ②

ドの音を鳴らす

デバイスとスピーカーをつなぐ



・ドレミに挑戦！ 🎵 ③

音を鳴らす準備

The screenshot shows a programming environment with a left sidebar and a main workspace. The sidebar has a menu with items: ① 高度なブロック, Easy I/O, デジタル入出力, ② PWM出力, アナログ入力, DAC出力, UART, I2C, コード実行, and Network. The main workspace contains several blocks: a top block with two servo control instructions, a large pink block for PWM output configuration (circled ③), and three smaller pink blocks for setting frequency and duty cycle. At the bottom, there are two blocks for stopping and restarting the PWM output (circled ④).

① 高度なブロック

- ▶ Control the rotating of servo to 0°
- ▶ Control the rotating of servo to 180°

② PWM出力

PWM出力ピン PWM0 をピン番号 26 周波数 10000 [Hz] デューティ 50 [%] 使用タイマー 0 にする ③

PWM0 の周波数を 1 [Hz]に設定

PWM0 のデューティを 0 [%]に設定

PWM0 の出力を停止する

PWM0 の出力を再開する ④

・ドレミに挑戦！ ④

前ページの③④をスタート画面の下に貼り付け、図のように設定する

Setup

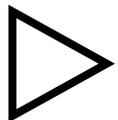
Set atom matrix

ド 523Hz

PWM出力ピン PWM0 をピン番号 21 周波数 523 [Hz] デューティ 50 [%] 使用タイマー 3 にする

PWM0 の出力を再開する

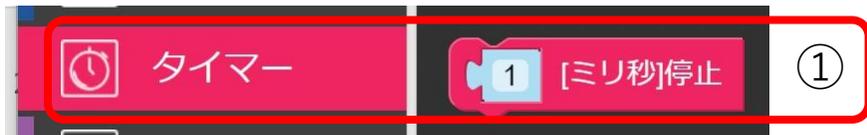
スピーカーの音をド
にセット



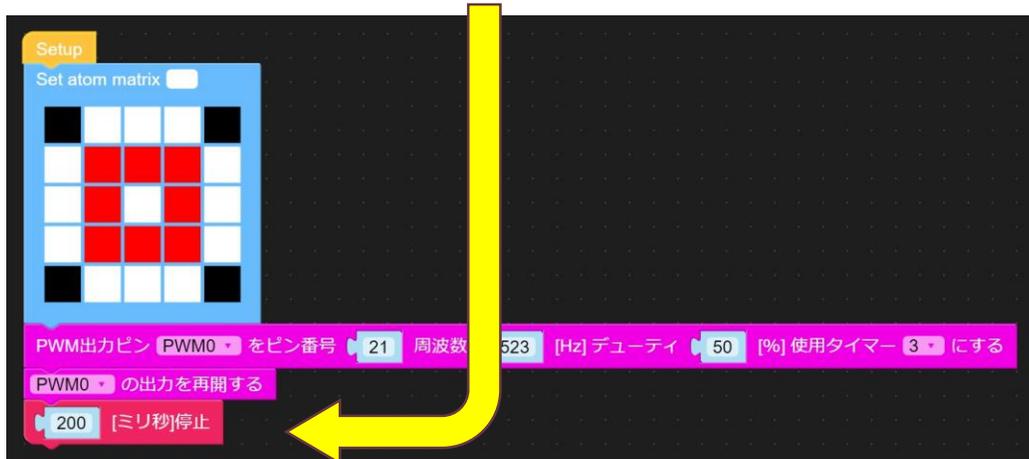
実行！

・ドレミに挑戦！ ⑤

音を鳴らす長さを決める



①をドのブロックの下に貼り付け、時間を決める



200にすると0.2秒だけ動きが止まる

・ドレミに挑戦！ ⑥

音を変える



音を止める



もう一度 音を出す



・ドレミに挑戦！ 🎵 ⑦

ドレミと鳴らしてみよう

The screenshot shows a Scratch-like programming environment with a dark background. At the top left, there is a 'Setup' block with a 'Set atom matrix' button and a 5x5 grid of colored squares (black, white, red, black). Below this, a series of code blocks are connected to create a sequence of notes:

- ド 523Hz**: A block 'PWM出力ピン PWM0 をピン番号 21 周波数 523 [Hz] デューティ 50 [%] 使用タイマー 3 にする' is followed by 'PWM0 の出力を再開する', a '200 [ミリ秒]停止' block, and 'PWM0 の周波数を 587 [Hz]に設定'.
- レ 587Hz**: A '200 [ミリ秒]停止' block is followed by 'PWM0 の周波数を 659 [Hz]に設定'.
- ミ 659Hz**: A '200 [ミリ秒]停止' block is followed by 'PWM0 の出力を停止する'.

・ドレミに挑戦！ 🎵 ⑧

明星チャルメラの音に挑戦！ ドレミーレド、ドレミレドレー

Setup

Set atom matrix

PWM出力ピン PWM0 をピン番号 21 周波数 523 [Hz] デューティ 50 [%] 使用タイマー 3 にする

PWM0 の出力を再開する

200 [ミリ秒]停止

PWM0 の周波数を 587 [Hz]に設定

200 [ミリ秒]停止

PWM0 の周波数を 659 [Hz]に設定

700 [ミリ秒]停止

PWM0 の周波数を 587 [Hz]に設定

200 [ミリ秒]停止

PWM0 の周波数を 523 [Hz]に設定

200 [ミリ秒]停止

PWM0 の出力を停止する

200 [ミリ秒]停止

下につづける

PWM0 の出力を再開する

PWM0 の周波数を 523 [Hz]に設定

200 [ミリ秒]停止

PWM0 の周波数を 587 [Hz]に設定

200 [ミリ秒]停止

PWM0 の周波数を 659 [Hz]に設定

200 [ミリ秒]停止

PWM0 の周波数を 587 [Hz]に設定

200 [ミリ秒]停止

PWM0 の周波数を 523 [Hz]に設定

200 [ミリ秒]停止

PWM0 の周波数を 587 [Hz]に設定

800 [ミリ秒]停止

PWM0 の出力を停止する

おろかおろか
おろかおろか