

www.M5Stack.com



# はじめてのM5GO

## ESP32をビジュアルに プログラミング

この資料は中学生でも読みこなせるように作りました。  
60秒で、自分のIoTアプリケーションをつくりましょう



Shenzhen Mingzhan Information Technology Co.,Ltd  
RM106, Block A, MingLiang Technology Park, Zhuguangbei Road, Nanshan,  
Shenzhen, China.

E-mail: Support@m5stack.com

(秋田純一、協力: M5Stackユーザーグループ 林信夫・小南靖雄・上田拓也)





M5Stackは、老若男女を問わず使える開発キットです。M5GOは、LEGOブロックにはめることができ、ビジュアル・ブロックでプログラムできるUIFlow（オンライン／オフライン）を使います。M5Stackは、IoTアプリケーション、ロボット、科学・技術・工学・数学を学ぶのに最適です。

*M5Stack*



M5STACK

初めての人からエキスパートまで  
M5Stack

思いつきをプログラミング  
M5Stack

みんな使える  
M5Stack



For the future creators





# 目次

M5GO

## はじめに

### 始めてみよう

UIFlowについて .....	13
UIFlowを使ってみよう .....	15

### いろいろやってみよう

Hello World - 画面に文字や絵を出す .....	23
絵文字でアニメーションやゲーム .....	29
デジタル音楽 .....	35
光で遊ぶ .....	41
防犯アラーム .....	45
植物をモニタリング .....	49
気象観測 .....	53
時計をつくる .....	55
加速度センサ .....	61
ポテンシオメータ .....	67
信号機 .....	72
光センサと関数 .....	77
M5GO アドレス帳 .....	80
赤外線リモコン .....	85
ゲームのデザイナー .....	91
リモコン .....	99

## 付録

Wi-Fiの設定 .....	105
本体の更新 .....	109
さらに学びたい人は .....	113

BRING IN TO USE  
BEFORE WE START BEFORE WE  
UNIT INTRODUCTION  
CONSTRUCTION  
HARDWARE  
BEFORE WE START:

# 「始める前に」

アイデアを素早く形に

M5GO

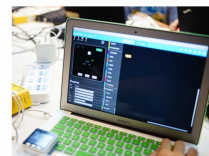
M5GO



始める前に



部品の構造



接続部品の紹介



さっそくやってみよう

M5GO

M5GO

M5GO  
M5GO

## はじめに

### 1. M5GOって何?

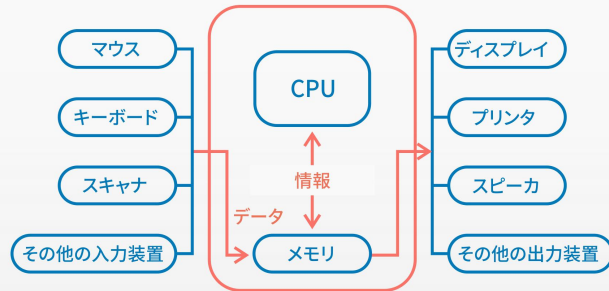
M5GOは、ESP32というマイコン(超小型コンピュータ)をつかったプログラミングできる小型コンピュータです。無線接続用のWi-FiとBluetoothが使えて、9軸加速度センサやスピーカー、フルカラーLEDバーなどもついています。M5GOはプラスチックのケースに入っていて、3つのボタンと3つの入出力端子がついています。LEDを点滅させたりするだけでなく、もっといろいろなことができるように設計されています。

### 2. M5GOは超小型コンピュータ



### 3. コンピュータとは?

コンピュータの4つの構成要素。



## コンピュータの構成要素

### CPU

コンピュータの中にあるいろいろな電子部品(チップ)の中で、中心的な役割を果たします。中央演算装置とも呼ばれ、計算や通信などのコンピュータが行う作業を全て行う、頭脳とも呼ぶべき存在です。

### メモリ(記憶装置)

メモリには2種類あります。

1. RAM: 電源を切ると内容が消えてしまうメモリ(揮発メモリ)。情報処理のためのデータを非常に高速に保存したり読み出したりできる。
2. フラッシュ(ROM): 電源を切っても内容が消えない。



### 入力

コンピュータが、私たちの実世界からデータを得るためのもの。マウス、キーボード、タッチパネルなど。

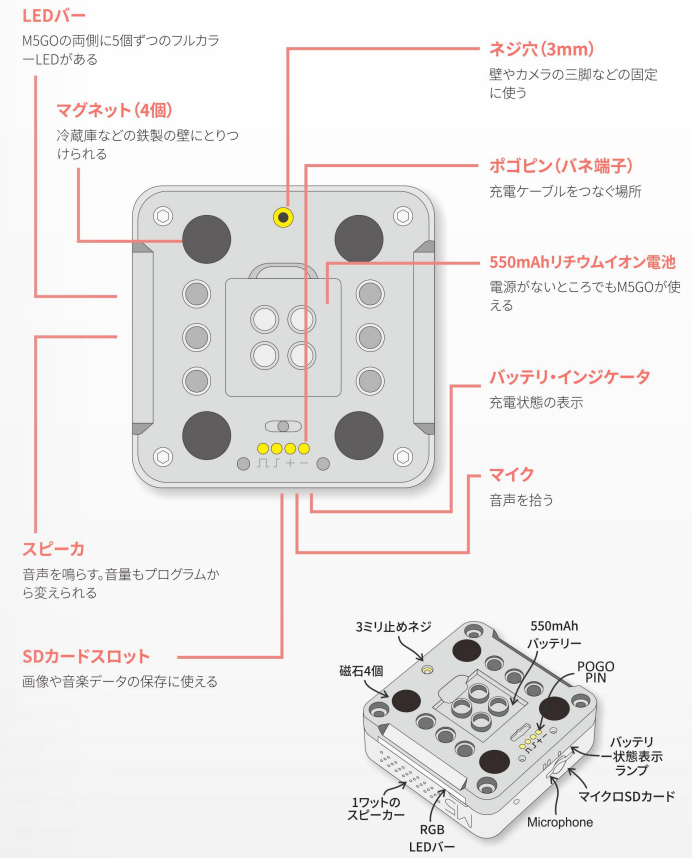
### 出力

コンピュータがデータを、私たちの実世界に出力するもの。スピーカー、ディスプレイ、プリンタなど。

### 4. M5GOとパソコンの比較

		
CPU	✓	✓
Memory	✓	✓
Inputs	キーボード、マウス	ボタン、加速度センサ、マイク
Outputs	ディスプレイ、スピーカー	フルカラーLEDバー、液晶ディスプレイ、スピーカー

## M5GOの各部の名称とはたらき



## はじめてみよう

- 1、横の赤いボタンを1回押すと電源がONになります。もう1回押すとリセットがかかります。素早く2回押すと電源がOFFになります。
- 2、電源をONにすると、メイン画面が表示されると、ボタンで3つの動作を選べます。



ボタンを1回押すとON

素早く2回押すとOFF

### プログラミングモード

UIFlowからMSGOをプログラムするためのQRコードとAPIキーが表示されます。

### アプリケーションのリスト (APP LIST)

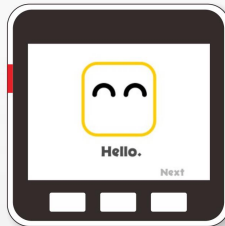
サンプルプログラムや、UIFlowからダウンロードしたプログラムを選んで動作できます。左右ボタンで動作するプログラムを選び、中央ボタンで動作します。右ボタンを長押しするとプログラムを消去できます。



### 設定 (SETUP)

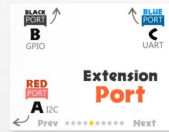
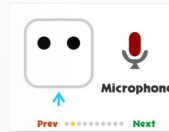
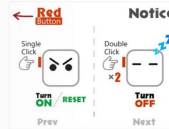
Wi-Fiの設定やUSBモードへの切り替え、起動時の音のON/OFF、接続したことがあるWi-Fiに接続、ができます。

### 3. デモモード



#### ちょっと補足

はじめて電源をONにしたときは、数秒後に、MSGOの機能の説明のデモモードが始まり、MSGOで何ができるかを一通り見られます。



### デモ1 ON/OFF

電源ON/OFFのやり方の解説。1回押すと電源ONまたはリセット、素早く2回押すと電源OFF。

### デモ2 スピーカ

MSGOのスピーカの機能の解説。中央ボタンを押すと音が鳴ります。

### デモ2 マイク

本体の横の小さな穴にマイクがあるので、何かしゃべったりテーブルをトントンして、表示される波形をみてみましょう。

### デモ4 ジャイロ

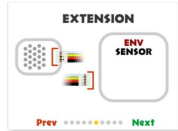
中に加速度センサとジャイロが入っています。本体を動かして、ボールを動かしてみましょう。

### デモ5 フルカラーLED

MSGOの横にあるLEDが光ったり、パトカーのライトのように点滅します。

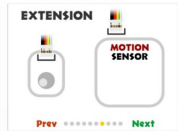
### デモ6:ポートの解説

ポートA~Cの色と機能の説明。



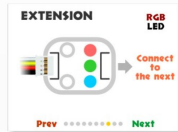
### デモ7 環境センサ

ENV (環境) センサは、温度、湿度、気圧を計測できます。



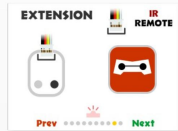
### デモ8 PIR (近接センサ)

PIR (近接センサ) をつないで、その前に手を置くと、ディスプレイ上の円の色が変わります。



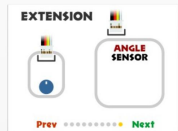
### デモ9 RGBユニット

本体の横にあるフルカラーLEDと同じようにLEDの色を自由に変えられます。ケーブルをつなぐコネクタには向き(「極性」といいます)があるので、間違えないようにしましょう。逆だと光りません。



### デモ10 赤外線リモコン

中央ボタンを押すとリモコンの赤外線信号が出ます。テレビのリモコンのように、いろいろな機器の制御に使えます。2台のM5GOを使ってお互いに通信させてみましょう。



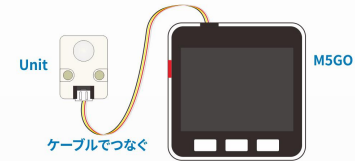
### デモ11 角度センサ

つまみを回すとLEDの明るさが変わります。

## ユニット (Unit) を使ってみよう

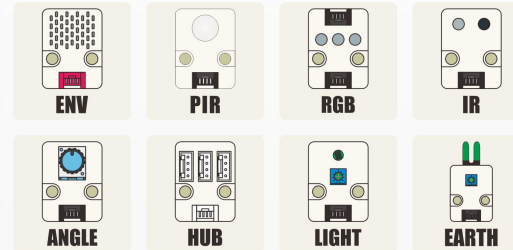
### 1. Unit (ユニット) とは?

Unitは、M5GOにつないで、いろいろな機能を拡張できるものです。あとから出てくるUIFlowで簡単に使うことができます。



### 2. Unitの種類

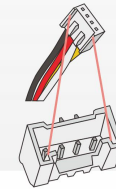
Unitは大きく分けて、入力と出力の2種類があります。入力は、光や音のような実世界の現象、情報を取得してデジタル信号に変換します。出力はM5GOが音を出したりモータを回転させたりという、実世界に何かを働きかけるものです。



### 3. Unitのつなぎ方

M5GOには色分けされているA, B, Cの3つのポート(ケーブルを繋ぐ端子)があります。Unitごとに、どこにつなぐかは決まっているものもあります。

コネクタの両端にある出っ張りがはまる向きに差し込みます。





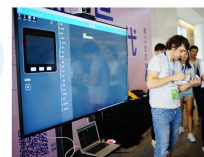
# 「準備体操」

アイデアを素早く形に

M5GO



UIFlowでプログラミング



UIFlow入門



保存して開く



プログラミングの仕組み



M5GO

M5GO  
M5GO

# プログラミング環境UIFlow

UIFlowは、blockly (ブロック型言語)とMicroPythonという言語で、M5GOのプログラミングを行うための環境です。

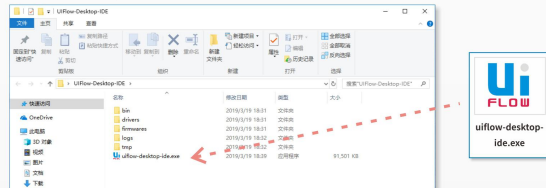
## 01 ダウンロード

https://m5stack.com へアクセスしてSoftware→Downloadsへ進み、使っているパソコンにあわせた最新のUIFlow-Desktop-IDEをダウンロードします。



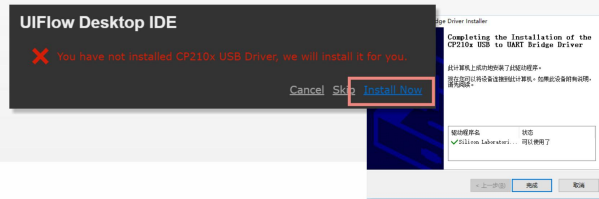
## 02 展開して実行

ダウンロードしたzipファイルを展開し、出てきたプログラム(インストーラ)を実行します。



## 03 CP210Xドライバのインストール

M5GOをパソコンにつなぐためのドライバのインストールが始まります。"Install Now"をクリックして手順に従ってインストールします。



## 04 M5GOをパソコンにつなぐ

インストールが終わるとUIFlowが動作し始めます。M5GOをパソコンにつなぎ、リストから"M5GO"を選びます。Windowsでは"COM"+数字、Macでは/dev/tty.SLAB\_USBtoUARTのような名前のはずです。



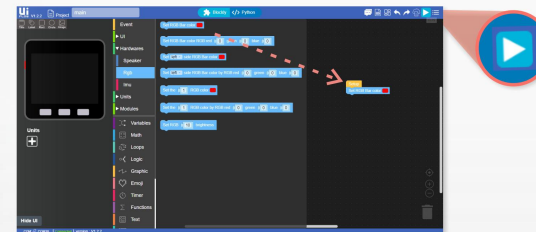
## 05 USBモードにする

電源/リセットボタンを押し、初期画面で右端のボタンを押してメニューに入り、"switch to USB mode"を選びます。



## 06 最初のプログラミング

ブロックリストの"Hardware"から、"RGB"の中にある"set RGB bar color"を選び、右のプログラミングエリアにある"Setup"バーの下につなぎます。できたら、右上の「再生」ボタンを押すと、M5GOにプログラムが転送されて実行がはじまり、本体のLEDが点灯するはずです。



## 07 アップデート

M5GOにはUIFlowにあわせたファームウェア(内部プログラム)が書き込まれています。ファームウェアはときどき機能追加やバグ修正のための更新が行われます。新しいファームウェアがあるときは忘れずに更新しましょう。

## UIFlowの構成

### プロジェクト名

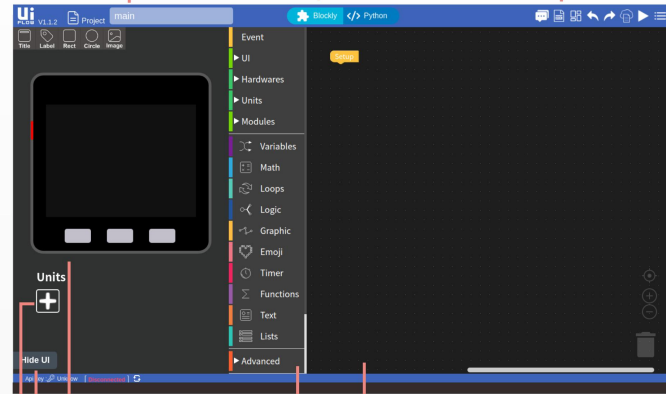
自分のプログラムに名前をつけてここに書けます。パソコンに保存したりM5GOで実行させるときには、その名前が使われます。

### Blockly-Python

Blockly(ブロック型言語)で書いたプログラムをPython言語のプログラムに切り替えて確認したり編集したりできます。

### メニュー

ここから、ユーザフォーラム、各種ドキュメント、例題、Undo(取り消し)やRedo(やり直し)、M5GOへの書き込みと実行、その他各種設定ができます。



### UIの確認

Text(文字列)や図形などをM5GOのディスプレイに表示させ、UI(User Interface、コンピュータの操作に関わる部分)をつくります。Python言語のプログラムが自動的に生成されます。置いたテキストなどは自由に位置を変えたり大きさを変えたりできます。

### Unit(ユニット)

各種センサなどのUnitをここから追加して、接続するポートを指定します。

### プログラミング画面

ここにブロックをドラッグで移動し、組み合わせてプログラミングします。

### プログラミング・ブロック

プログラミングに使うブロックのリストです。

### UIを隠す

UI編集画面を隠して、プログラミング画面を広くできます。

## プログラミング・ブロック

ここには、いろいろなプログラミング・ブロックがあって、プログラミングに使えます。M5GOに載っているLEDなどを直接制御するものから、数値演算、プログラムの流れの制御などのためのブロックがあります。まずは簡単なブロックから順に使っていきましょう。

### イベント

繰り返しや、ボタンを押したときの動作などを設定できます。

### ハードウェア

M5GOに入っているフルカラーLED、スピーカ、加速度センサなどの要素(ハードウェア)を制御します。

### Unit

Unitを追加すると、そのUnitに応じた動作のブロックが現れます。

### Math(数学)

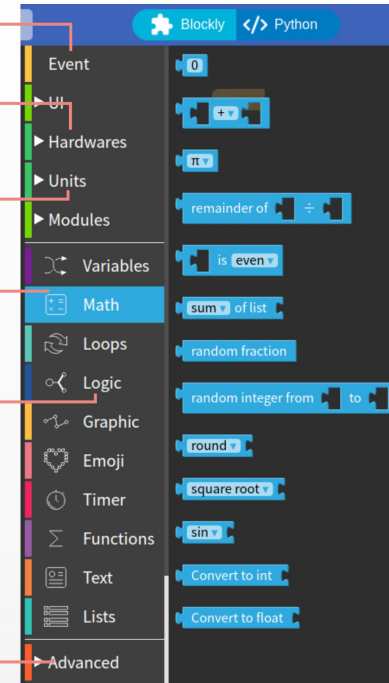
プログラミングでは数学は大切です。簡単な数値演算から複雑な計算までいろいろな数学処理のブロックがあります。

### Logic(論理)

プログラミングに必要な、条件に応じた動作の制御などの論理を設定できます。

### Advanced(上級者向け)

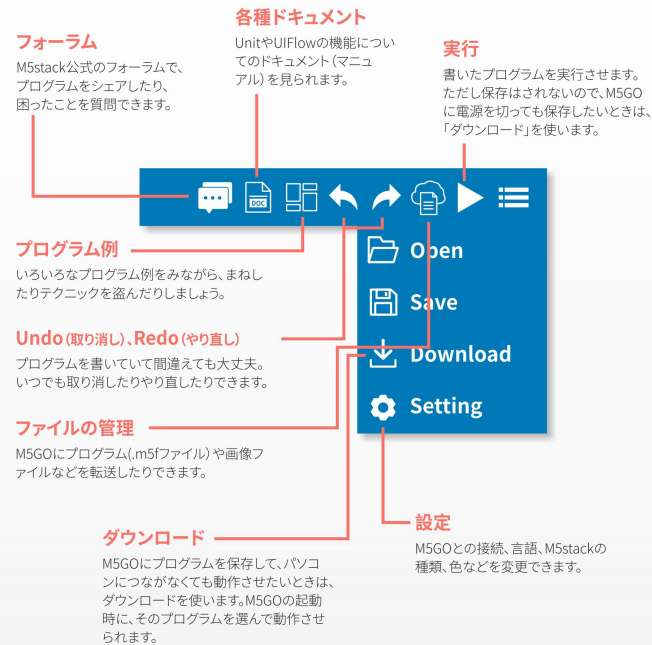
上級者向けのブロック。ネットワーク通信、デジタル・アナログ入出力などがあります。





### メニューの詳細

メインメニューは、フォーラム、各種ドキュメント、プログラム例、Undo (取り消し)、Redo (やり直し)、プログラムの書き込みや実行、各種設定ができます。



### Setup (セットアップ)

Setupブロックは、プログラムの実行が始まる時に最初に実行されます。



### Loop (ループ)

Loopブロックは、中に置いたブロックをずっと繰り返し、電源を切るかリセットするまで実行します。

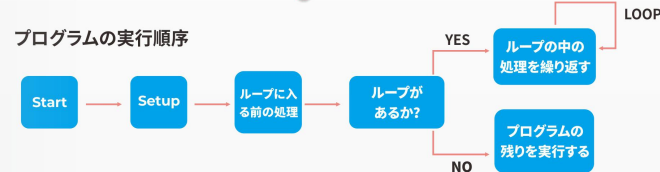


### Wait (待機)

Waitブロックは、指定した時間だけプログラムの実行を止めて待ちます。これがないと、表示されるデータがすぐに消えてしまて見えないといった事が起こってしまいます。

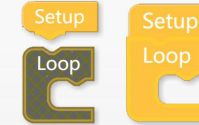


### プログラムの実行順序



### プログラミング・ブロックの接続

UIFlowではBlocklyというブロック型のプログラミング言語を使います。Blocklyはジグソーパズルに似ていて、はまるところに移動させると色が変わって、かちつとはまり、正しく動作するようになります。色が変わらなかったら、ちゃんとはまっていないか、その動作ができないかのどちらかです。

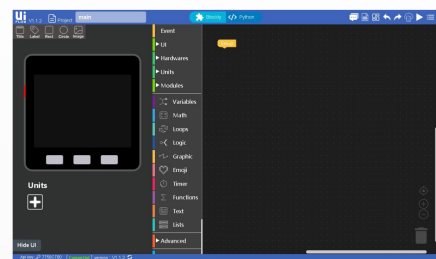


### UIFlowのちょっとしたテクニック

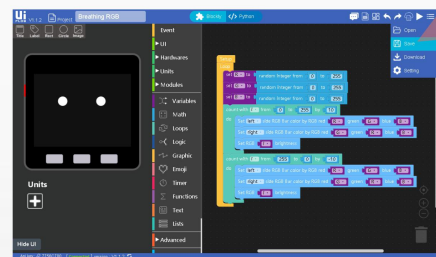
ブロックをコピーするには、マウスの左ボタンをダブルクリックするか、右クリックから複製(Duplicate)を選びます。ブロックを消すには、ブロックが並んでいるリストに戻るか、プログラミング画面の右下のゴミ箱に移動させます。

## プログラムの保存と転送

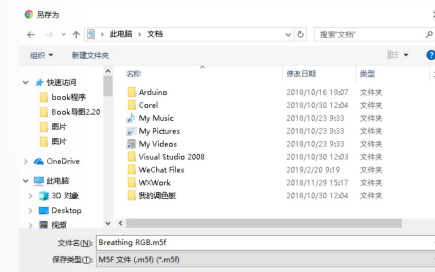
いよいよプログラミングのレッスンが始まりますが、これからいろいろなプログラムを作っていくことになります。複雑ですぐには終わらないものもあるでしょう。そういうときは、作ったプログラムをパソコンに保存しておいて、日を改めて続きをやりたり実行させたりできます。突然パソコンがフリーズしてしまっても困らないように、こまめに保存しましょう。メニューの保存(Save)から保存でき、開く(Open)から、保存されたプログラム(.m5fファイル)を読み込みます。



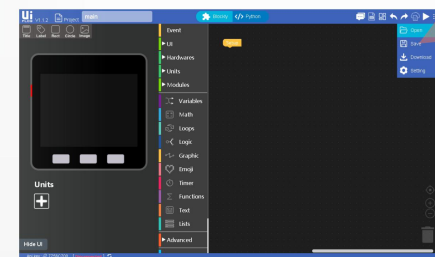
01 UIflowのWebページを開きます。  
([flow.m5stack.com](http://flow.m5stack.com))



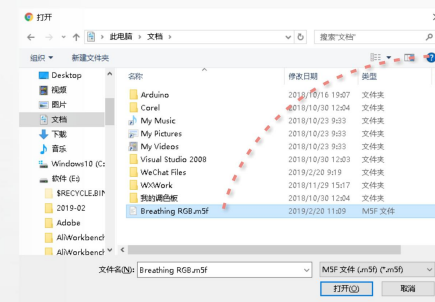
02 プログラムを書いたら、保存(Save)から保存しましょう。



03 m5fファイル(拡張子がm5f)として保存されます。



04 m5fファイルは、メニューの開く(Open)から読み込みます。



05 前に保存した.m5fファイルを指定します。

CURRICULUM:  
DIGITAL MUSIC  
BURGLAR ALARM  
M5GO TRAFFIC LIGHT  
PLANT STEWARD:  
GAME ENGINEER

# 「カリキュラム」

アイデアを素早く形に

M5GO

M5GO



デジタル音楽



防犯アラーム



信号機



ゲームのデザイン

M5GO

M5GO

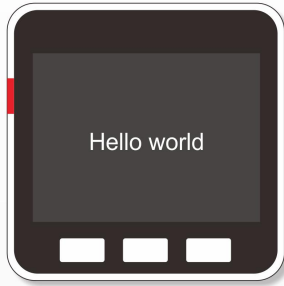
M5GO  
M5GO

# 1 Hello World - 画面に文字や絵を出す

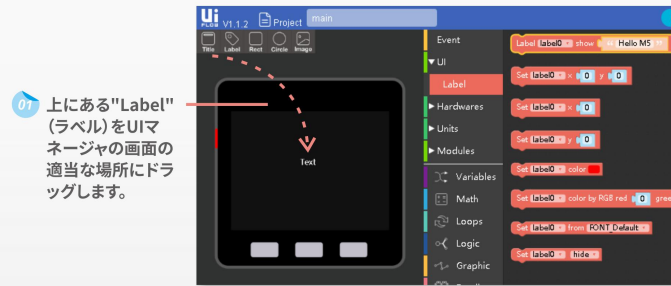
UIFlowの画面の左側には「UI Manager (UIマネージャ)」が表示されていて、ここに文字列やグラフィックが表示されます。  
このレッスンでは、M5GOのプログラムを書いて実行できるようになります。簡単な文字を表示して、オリジナル名刺をつくってみます。

## 1-1 超簡単プログラミング

"Hello World"は、昔からある「はじめてのプログラミング」の定番で、画面に「Hello World」と表示させるものです。簡単そうに見えますが、初めて使うプログラミング言語やコンピュータ(M5GO)を使うには最適です。これができれば、それを応用してさらに高度なプログラミングへ進めます。

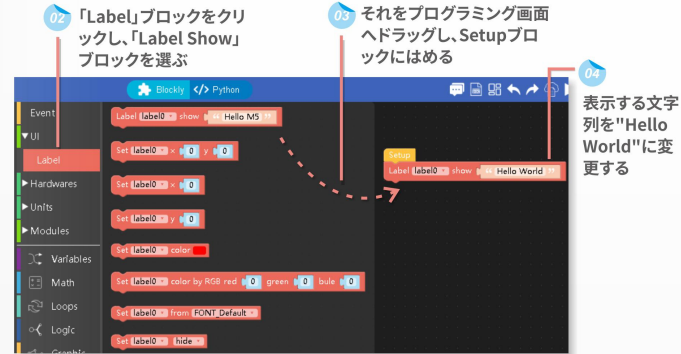


さあ、「Hello World」を始めてみましょう。



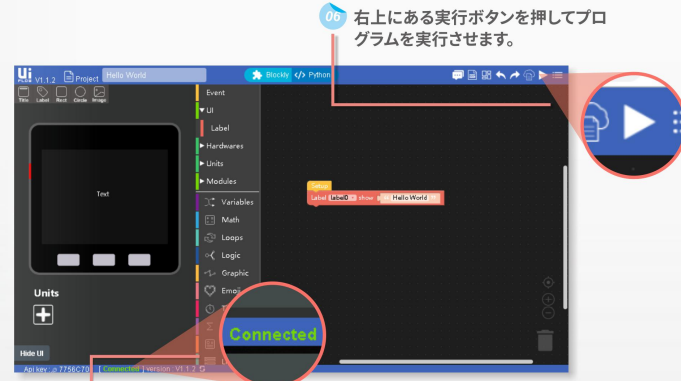
01 上にある「Label」(ラベル)をUIマネージャの画面の適当な場所にドラッグします。

ブロック一覧の「UI」をクリックすると、その中に「Label」というグループがあり、テキスト(文字列)に関するいろいろなブロックがあります。「Label [ラベル名] Show [表示する文字列]」というブロックをプログラミング画面へ移動させて、Setupブロックにはめます。これでできあがりです。



02 「Label」ブロックをクリックし、「Label Show」ブロックを選ぶ  
03 それをプログラミング画面へドラッグし、Setupブロックにはめる  
04 表示する文字列を「Hello World」に変更する

これで最初のプログラムが完成です。簡単でしたね。早速M5GOで実行させてみましょう。パソコンとM5GOをUSBケーブルで接続し、前の章で説明した手順で実行させてみましょう。

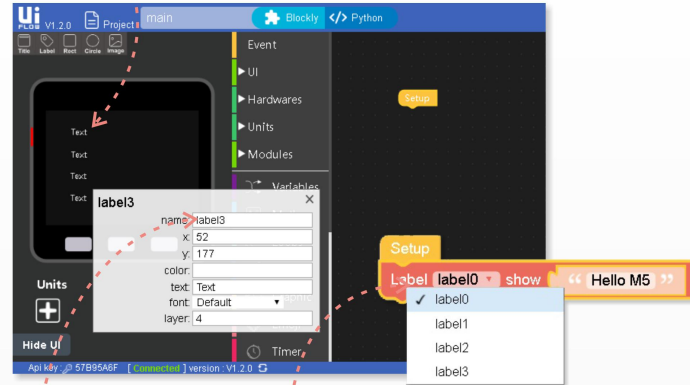


05 右上にある実行ボタンを押してプログラムを実行させます。

06 M5GOがちゃんと接続されていれば、ここの表示が緑の「connected」になります。そうならないときは、更新マークを押してみましょう。

## 1-2 デジタル名刺を作ってみよう

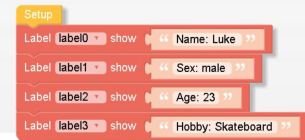
次に、表示させる文字を変えて、デジタル名刺を作ってみましょう。どんな内容を書きますか？名前、年齢、住所、趣味、などでしょうか。



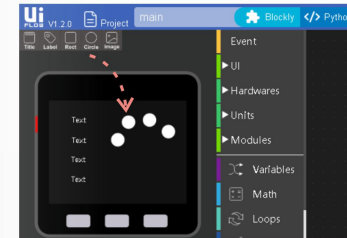
Labelを追加することによって、ラベルにつく「名前」は、「label」のあとに数字が増えていったものがつけられます。この名前は、プログラムの中で、どの文字列を対象とするのかを区別するのに使われ、ブロックのリストから選べます。置いたLabelの文字をクリックして、名前を変更できます。



2. ブロックをダブルクリックするか、右クリックして "Duplicate" を選ぶとコピーを作れます。

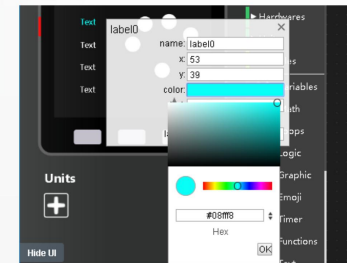


3. 変更したいLabelの「名前」を確認して選びましょう。



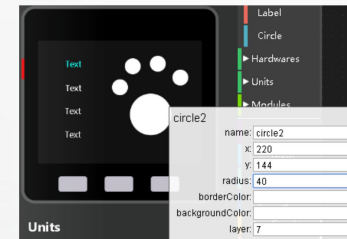
## 4. 絵を描く

文字だけだと地味なので、絵を追加してみましょう。Rectangular (四角) や Circle (丸) を UI マネージャにドラッグして描くことができます。クリックしてサイズや色を変えることもできます。



## 5. 色をつけてみる

文字にも絵にも色を付けることができます。文字や絵をクリックして出てくる画面の "Color (色)" をクリックし、色を選びます。好きな色を選べたら、OKを押します。



## 6. 文字や絵の属性

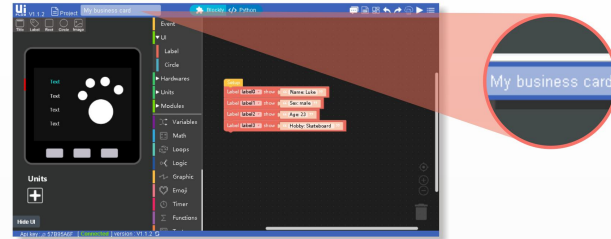
文字や絵をクリックして出てくる画面に現れる情報を「属性」と呼びます。大きさや位置、絵を塗りつぶすかどうか、枠や塗りつぶしの色を設定できます。文字のフォントや回転も設定できます。





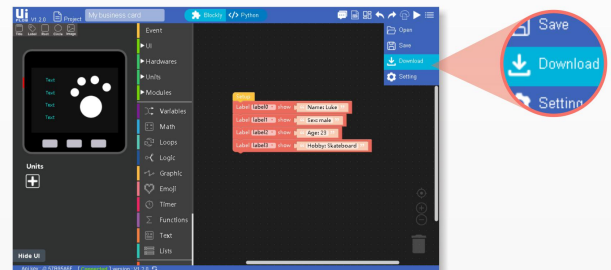
### 7. プロジェクトに名前をつける

自分で作ったプログラムには名前をつけましょう。とても大切なことです。UIマネージャの上にある文字列をクリックして名前を変更できます。今回は"my business card"などにしましょう(残念ながら日本語は使えませんので、英語かローマ字で)



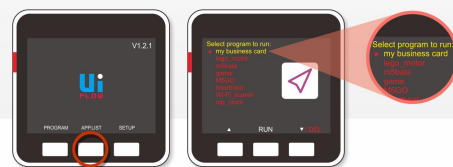
### 8. プログラムをM5GOで実行させる

右上の三角の「実行」ボタンを押すと、書いたプログラムがM5GOで実行されます。ただし電源をOFFにすると消えてしまいます。M5GO本体に保存するには、右上隅のメニューの中の"Download (ダウンロード)"を使います。M5GOの画面に書き込み中のメッセージが表示され、終わると保存されて自動でリセットされます。



### 9. 保存されたプログラムの実行

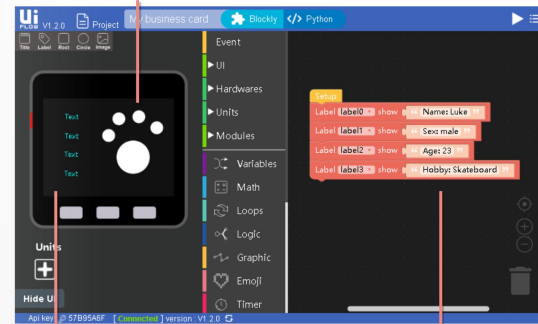
DownloadでM5GOに保存したプログラムは、リセット後のメニュー画面の"AppList"から選んで実行できます。M5GOをリセットして、中央ボタンを押すと、保存されているプログラムの一覧が表示されます。



### Python入門

ブロックを置いて書いたプログラムは、Pythonというプログラミング言語のプログラムと、内部では同じです。Pythonはよく使われるプログラミング言語です。中央上部の"Blockly | Python"切り替えをクリックして、Pythonのプログラムを見ることができます。ブロックとPythonとの対応を確認してみましょう。

これらの"circle"をUIマネージャに置くと、それぞれにPythonのプログラムが生成されます。



UIマネージャの画面の背景の色は、ここをクリックして変更されます。初期値は0x111111(16進数表示)です。16進数表記については、あとで詳しく説明しますが、ここではとりあえず"0xff0000"に変えて、どんな色になるかみてみましょう。

"Label"は画面に表示される文字です。Pythonでは、画面上の場所やフォント、色を確認したり変更したりできます。

ブロックとPythonのプログラムの切り替えはいつでもできます。ただしPythonで書いたり変更したプログラムは、ブロックに切り替えても反映されずに消えてしまいます。

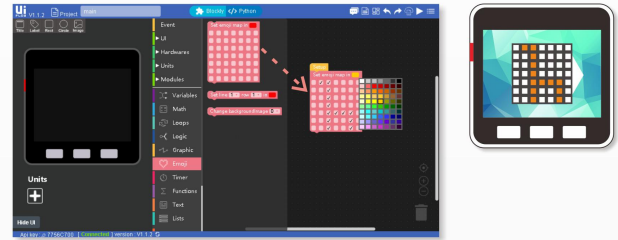


## 2 絵文字でアニメーションやゲーム

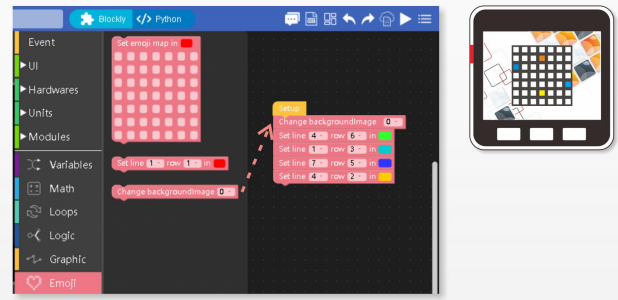
絵文字ブロックで、絵やアニメーション、さらにはゲームも作れます。  
絵文字の表示のさせ方、アニメーション(動き)の作り方や、ボタンでそれを動かしてみましょう。一定時間待つブロックも使えます。

### 2-1 絵文字でアニメーションをつくる

絵文字は、LINEなどで使うニコちゃんマークなどのマークです。絵文字は、マス目を塗って作ります。いろいろな絵文字をつくって画面に表示させたり、パラパラ漫画をつくってみましょう。絵文字("emoji")グループから絵文字のマス目を持ってきてSetupブロックにつなぎます。そしてマス目を塗って絵文字を描いてみましょう。右上のマークから色を変えることもできます。できたら実行ボタンを押してプログラムを実行させてみましょう。



絵文字の中に複数の色を使うには、絵文字グループの2つ目のブロックを使います。これはマス目の場所を指定して、その色を変えます。3つ目のブロックは、背景の色を選んで設定できます。

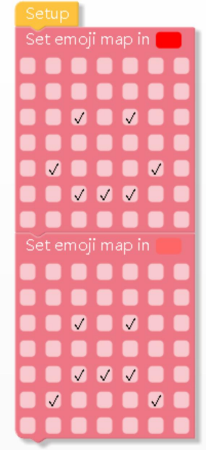
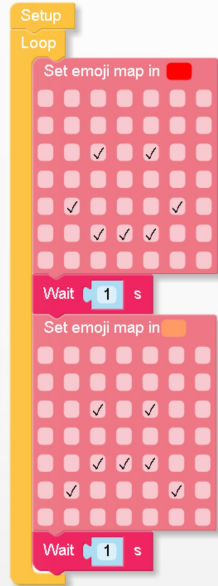


絵文字に慣れてきたところで、アニメーションをやってみましょう。何枚もの絵を順番にパラパラ切り替えていけば、残像で動いているように見え、これがアニメーションです。絵を切り替えるスピードが速いほど、なめらかなアニメーションになります。

まずは2枚の絵文字でやってみましょう。2枚目の絵文字を、さきほど作った絵の下にはめて実行させてみましょう。どうなりましたか?絵が切り替わるように見えましたが?おそらく2枚目の絵文字しか見えないかと思います。

### 1. 原因を探る

2枚目の絵文字しか見えませんが、実は1枚目の絵文字も表示されているのですが、すぐに2枚目に切り替わってしまうので、1枚目が見えないのです。M5GOのプログラムは、それぐらい速く実行が進んでいくのです。



### 2. 解決法

この問題を解決するには、2枚目の絵文字に切り替えるまでに少し待ち時間を入れれば大丈夫です。"Timer" (タイマー) グループの中に、"Wait" (待ち時間) というブロックがあります。これを使えば、秒[s]、またはミリ秒[ms] (千分の一秒) の単位で待ち時間を入れられます。

今回作った絵文字アニメーションはすごく短かったですね。実際のアニメでは、同じ絵文字を繰り返して使うことで手間を省けます。"Event (イベント)"グループの中に、"Loop" (ループ) というブロックがあります。このブロックは、アルファベットのCのような形をしていて、そのへこんだところに別のブロックをはめることができます。先ほどの絵文字アニメーションを、このLoopの中にはさんでみて実行させてみましょう。

こうすると、2枚の絵文字の表示が終わったら、また最初に戻って繰り返されます。そのため2枚目の表示がされたらすぐに1枚目の表示に戻ってしまうので、2枚目がすぐに切り替わってしまいます。そこで2枚目の表示のあとにもWaitを入れておきましょう。

## 2-2 花火

絵文字を使って、花火をつくってみましょう。

The image shows a Scratch script for creating a fireworks animation. The script is contained within a 'Loop' block and consists of the following steps:

- Set emoji map in:** A grid of 10x10 red squares with a yellow checkmark in the center.
- Wait:** 0.2 s
- Set emoji map in:** A grid of 10x10 red squares with a yellow checkmark in the center.
- Wait:** 0.2 s
- Set emoji map in:** A grid of 10x10 red squares with a green checkmark in the center.
- Wait:** 0.2 s
- Set emoji map in:** A grid of 10x10 red squares with a blue checkmark in the center.
- Wait:** 0.2 s

The background features a city skyline at the bottom and several colorful fireworks exploding in the sky. A large, detailed firework is shown in the upper right corner.

アニメーションをつくるには、よく少しだけ変えた絵文字、をつくるのがよくあります。そんなときは、絵文字ブロックを右クリックして複製 (コピー、duplicate) を選んでコピーをつくり、それを少し修正すれば楽です。花火が広がっていく様子の絵文字をつくってみましょう。

A context menu is shown over a 'Set emoji map in' block. The menu options are:

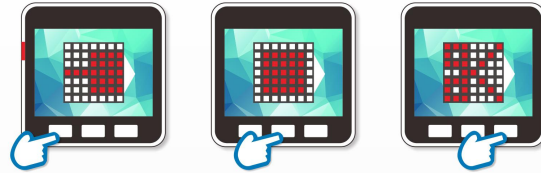
- Duplicate
- Inline Inputs
- Collapse Block
- Disable Block
- Delete Block
- Help

アニメーションが長くなっていくと、ブロックが画面からはみだしますよね。ブロックを右クリックして、"Collapse block" (ブロックをまとめる) を選ぶと、ブロックをまとめて、たたんでおくことができます。また中身をみたり編集したくなったら、"Expand block" (ブロックをひろげる) を選べば大丈夫です。

The image compares two context menus. On the left, the 'Collapse Block' option is highlighted. On the right, the 'Expand Block' option is highlighted. A double-headed red arrow indicates the relationship between the two options.

### 2-3 じゃんけん

もっと長いアニメーションを作るのもいいですが、ほかのこともやってみましょう。ゲームとか。M5GOのA、B、Cボタンで別の絵文字を表示して、じゃんけんゲームをつくっていきましょう。



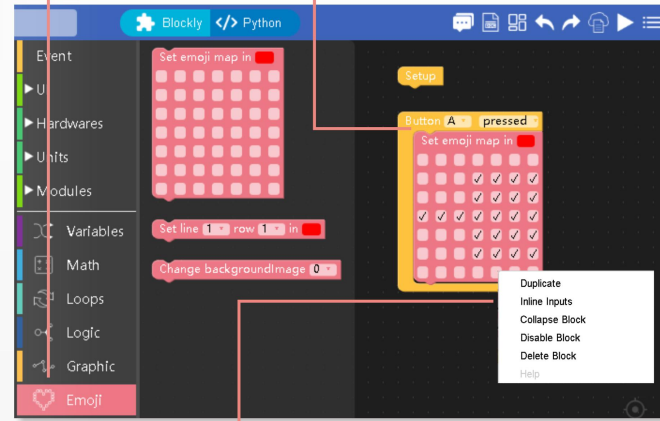
01 "Event"内の"button A pressed" (Aボタンが押されたら)を持ってきます。



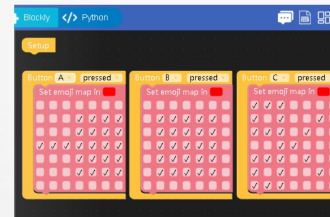
02 "A"のところをクリックするとリストからA、B、Cのボタンを選べます。そのボタンが押されると、このブロックの中の部分が実行されます。

01 絵文字を描くマス目を持ってきます

02 ここに、グー、チョキ、パーをそれぞれ描いてみましょう。



03 "Duplicate" (コピー)を使うと楽ですね。



A、B、Cボタンそれぞれに、グー、チョキ、パーの絵文字があることを確認しましょう。

二人で2台のM5GOを使って、同じプログラムを実行させて、掛け声に合わせてボタンを押してじゃんけんができます。



## 3 デジタル音楽

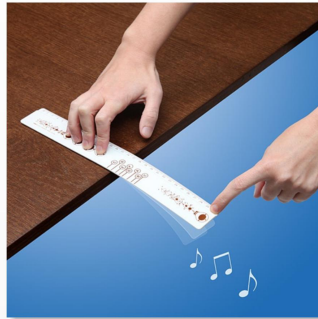
ここでは、M5GOで音をならせてみましょう。音楽のしくみと、繰り返し構造を使ってプログラムを簡単にするテクニックも学びます。

### 3-1 音楽をつくる

まずはM5GOのスピーカで音を鳴らしてみよう。ところで音って、なんでしょ？

音は、実は振動です。定規を机の端に置いて片方を固定し、もう片方の手で押し込んで話してみよう。どんな音が聞こえて、持っている手にどんな振動を感じますか？

定規の場所を変えてみたり抑える場所を変えてみたりしてみましょう。いろんな音が出るはずですよ。



この実験から、速く振動するほど音が高く、遅いほど低くなることがわかります。振動の速さは「周波数」と呼ばれて、「Hz（ヘルツ）」という単位で表します。人間の耳で聞こえるのは、およそ20Hzから20kHz（20,000Hz）の音です。それより低かったり高い音は聞こえません。

#### 1. 音程と周波数

もう少し進んで、音から音楽へいってみましょう。「Speaker（スピーカ）」ブロックを使うと、スピーカから音を鳴らすことができます。「Hardware」（ハードウェア）グループの中の「Speaker」ブロックを持ってきて、Setupにつなぎます。このブロックの1つ目は周波数で、2つ目はその音となる時間（ms=ミリ秒、1000分の1秒）です。これらの数値を変えてどんな音が聞こえるか試してみましょう。

Setup

Speaker.beep freq: 1600 duration: 200 ms

音楽を作るのに、いちいち周波数を入力するのは面倒です。そこで使われるのが音程です。C、D、E、F、G、A、Bの記号が、順にド、レ、…シを表し、それぞれの周波数が決まっています。ピアノなどの楽器では、更に高い音や低い音も、このC～Bを繰り返します（オクターブ）。それぞれの音程と周波数の関係は次の表のとおりです。

	C	D	E	F	G	A	B
Low	131	147	165	175	196	220	247
Middle	262	294	330	349	392	440	494
High	523	587	659	698	784	880	988

スピーカのブロックには、音程を直接指定できるブロックもあります。1つ目が音程、2つ目で、つまり音の長さを指定できます。

Setup

play tone Low A for 1 beat

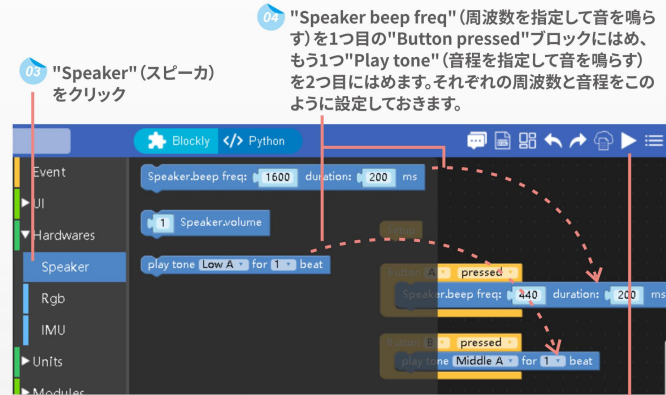
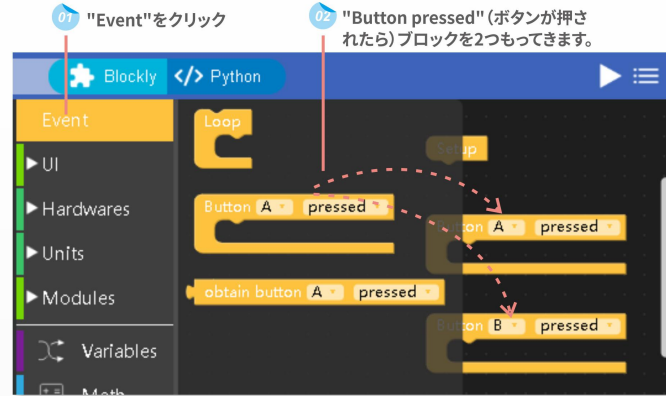
- Low A
- Low B
- Low C

Setup

play tone Low A for 1 beat

- 1
- 1/2
- 1/4

下の画面のように周波数と音程の関係を確認するプログラムを書いてみましょう。

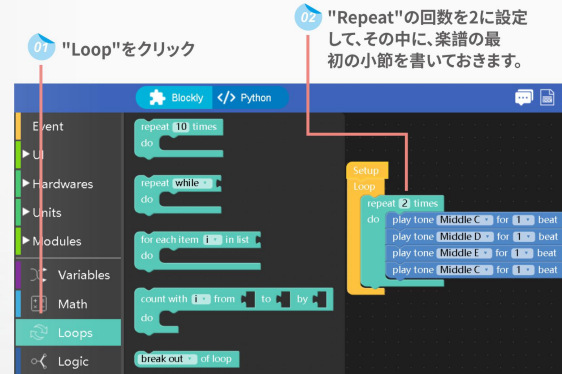


05 プログラムを実行し、AボタンとBボタンを順に押ししてみましょう。同じ音が鳴るのがわかります。

簡単な楽譜の音楽をつくってみましょう。



これは簡単な音楽の楽譜です。よく見ると、同じフレーズが繰り返されているところがあります。プログラミングでは、このような繰り返しを簡単に書くことができます。"Repeat" (繰り返し) ブロックは、"Loop" と似ていますが、指定した回数だけ繰り返すものです。



"Repeat"を使わないと、左のように長いプログラムになります。動作の結果は同じですが、Repeatブロックを使うと、ずいぶん短くて、すっきりしますね。

The image shows two code snippets side-by-side. The left snippet is a long, linear sequence of blocks: a 'Setup' block followed by a 'Loop' block containing 14 'play tone' blocks (Middle C, Middle D, Middle E, Middle C, Middle D, Middle E, Middle C, Middle D, Middle E, Middle C, Middle E, Middle F, Middle G, Middle C) and 14 'Wait 0.2 s' blocks. The right snippet is much shorter, using 'repeat 2 times' blocks to group the 'play tone' and 'Wait' blocks, reducing the total number of blocks significantly.

プログラミングでは、このように「楽をする」ことはけっこう大切です。

### 3-2 M5オーケストラ

みんなで合奏(合唱)してみましょう。それぞれが自分のパートを、ボタンを押して演奏します。誰か一人が指揮者になって、ボタンを押すタイミングを指示しましょう。



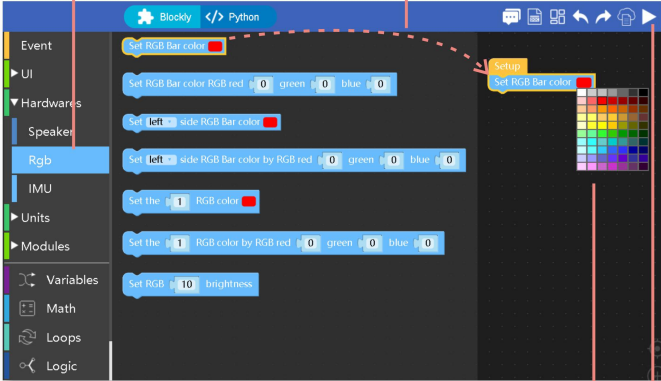
## 4 光で遊ぶ

ここでは、M5GOの横についているフルカラーLEDで、いろいろな色をつくってみましょう。まず光の理論を少し学び、それからループと待ち時間についても学びます。

### 4-1 フルカラーLEDをつけたり消したりする

01 "Hardware"をクリックして、"Rgb"を選ぶ

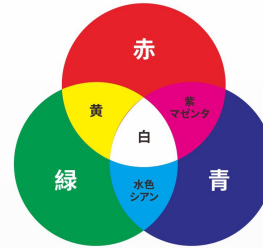
02 "set RGB bar" (RGBバーを設定)を持ってきてSetupにつなぎます。



03 色のリストから好きな色を選びます。

04 プログラムを実行させてます

色のリストから選ぶだけでなく、RGB値を指定することもできます。RGB値とは、R (red、赤)、G (green、緑)、B (blue、青)の強さを指定するものです。この赤、緑、青の3色を組み合わせると、どんな色もつくることができます。



フルカラーLEDは、中に小さい赤、緑、青の3つのLEDが入っていて、それぞれ好きな明るさで光らせられます。それぞれの明るさは0から255の値で指定します。0は光らず、255が最も明るく光ります。RGB値を(0,0,0)にすればLEDは消えます。

M5GOについている両側のLEDを同じ色にすることもできますが、左側と右側を別々の色にもできます。M5GOには両側にそれぞれ5個、全部で10個のLEDがあり、それぞれ別々の色にもできます。それは"set the [] RGB color"ブロックでできます。この[]の中の数字を1から10に変えると、それぞれのLEDの色を個別に設定できます。

Set RGB Bar color

Set RGB Bar color RGB red # 0 green # 0 blue # 0

Set left side RGB Bar color

Set left side RGB Bar color by RGB red # 0 green # 0 blue # 0

Set the 1 RGB color

Set the 1 RGB color by RGB red # 0 green # 0 blue # 0

Set RGB # 10 brightness



## 4-2 パトカー

次に、パトカーのライトのように点滅させて、サイレンの音を鳴らしてみよう。

```

Setup
Loop
  Set left side RGB Bar color red
  Set right side RGB Bar color black
  play tone High C# for 1 beat
  Set left side RGB Bar color black
  Set right side RGB Bar color blue
  play tone Middle G# for 1 beat
  
```

## 1. まずは簡単な光と音

これまでに学んだことを使って、別のRGB LEDライトを光らせ、カッコいい光と音のショーを演出してみましょう。

## その1.

まず全部のLEDを点灯させて、素早く別の色に変えていきます。"Repeat"ブロックを使って2回ずつ繰り返しています。

```

Setup
Loop
  repeat 2 times
  do
    Set RGB Bar color orange
    Wait 0.2 s
    Set RGB Bar color black
    Wait 0.2 s
  repeat 2 times
  do
    Set left side RGB Bar color yellow
    Set right side RGB Bar color black
    Wait 0.2 s
    Set left side RGB Bar color black
    Set right side RGB Bar color pink
    Wait 0.2 s
  
```

## その2.

後半では、左右のLEDを別の色で光らせています。



## 2. 機能の追加

ボタンでLEDの点滅を変えてみましょう。

```

Event
  UI
  Hardware
  Speaker
  Rgb
  IMU
  Units
  Modules
  Variables
  Math
  Loops
  Logic
  Graphic

Button A pressed
  repeat 10 times
  do
    Set RGB Bar color green
    Wait 0.2 s
    Set RGB Bar color black
    Wait 0.2 s

Button B pressed
  repeat 5 times
  do
    Set left side RGB Bar color red
    Set right side RGB Bar color black
    play tone High C# for 1 beat
    Set left side RGB Bar color black
    Set right side RGB Bar color blue
    play tone Middle G# for 1 beat

Button C pressed
  Set the 1 RGB color red
  Set the 2 RGB color orange
  Set the 3 RGB color yellow
  Set the 4 RGB color green
  Set the 5 RGB color cyan
  Set the 6 RGB color blue
  Set the 7 RGB color purple
  Set the 8 RGB color pink
  Set the 9 RGB color magenta
  Set the 10 RGB color black
  Wait 1 s
  
```

緑を10回点滅させます。

パトカーのライト(日本のパトカーのライトとは違いますが)を5回点滅させ、音も鳴らしてみます。

ここでは10個のLEDを全部違う色にしてみましょう。

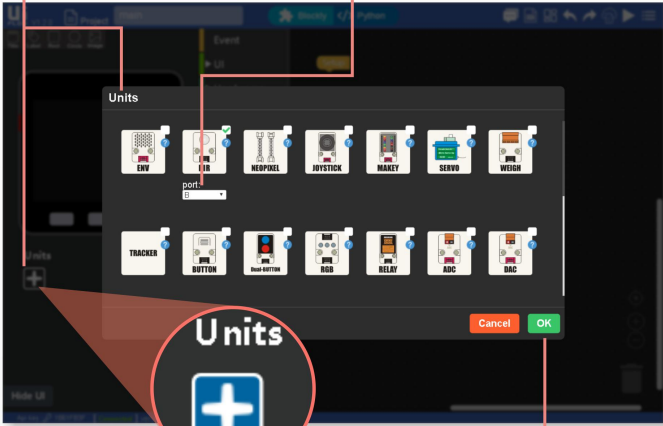
## 5 防犯アラーム

ここでは、PIR (人感センサ) を使って防犯アラームを作ってみましょう。ユニットの使い方と人感センサの機能、それから基本的な論理機能を学びます。

### 1. Unit (ユニット) を追加する

01 左のUIマネージャにある、"Unit" (ユニット) の下にあるプラスボタンを押します。Unitを選ぶ画面が出てきます。

02 PIRセンサ (人感センサ) をクリックして、右上に緑のチェックマークをつけます。その下に接続するポート名が出てきます。

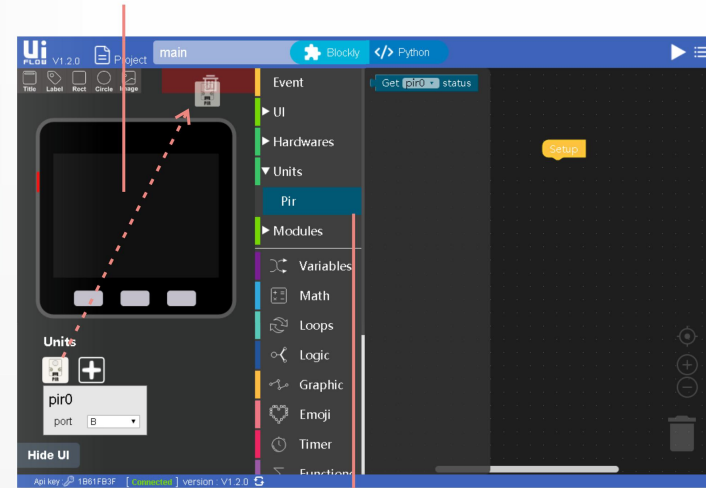


03 Okボタンを押すと、プログラムでこのUnitを使えます。

他にもいろいろなユニットがあるので、みてみてください。

### 2. Unitを消す

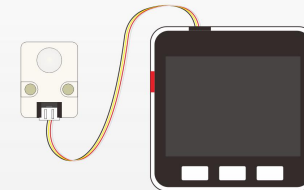
間違えてUnitを追加してしまったり、使わなくなったUnitがあったら、Unitのアイコンをドラッグして、出てくるゴミ箱へ入れれば消せます。



### 3. Unitを使う

Unitを追加したら、ブロックリストの"Unit"の中に、"PIR"が現れます。Unitに追加すれば、そのUnitに応じたブロックが使えるようになります。

Unitを使うときは、追加するときに表示されていた接続するポート名と同じ名前の、M5GOのポートのコネクタに、そのUnitを接続します。違うところに接続すると正しく動作しません。



#### 4. PIRセンサの設定

**01** まずUnitメニューからPIRユニットを追加します

**02** PIRをクリックします

**03** PIRユニットには1つのブロックしかありません。人がいたら1、いなかったら0の値になります。このブロックは、人感センサの値を取得します。

**04** UIマネージャで、Labelを追加しておきます

**05** "Get PIR status" (人感センサの値を取得) ブロックを、Labelの文字を設定するところにはめます。それを"Loop"の中に置きます。

**06** M5GOのポートBに人感センサを接続することにして、ポートBを選んでおきます。

**07** プログラムを実行します

#### 1. 人感センサのテスト

プログラムの実行をはじめたときに、人感センサの前に人がいなかったら0が表示されます。人感センサの前で手を動かすと1が表示されます。人感センサの反応は少し遅れがありますが、動きが止まると0に戻ります。

##### 人感センサのしくみ

家の玄関の防犯ライトも、同じ人感センサが使われていて、人やネコが歩くと反応します。これは生き物の体が赤外線を発していて、それを人感センサが検知するのです。

#### 5-1 防犯アラーム

人感センサの原理がわかったので、もうちょっとちゃんとした使い方として防犯アラームをつくってみましょう。プログラムとしては、人(泥棒)を検知したらアラームを鳴らすことになります。LEDやスピーカの使い方はわかるので、それを使ってみます。人感センサと光や音のアラームを、"if"ブロックでつなげます。"if"ブロックは、プログラムの動作を入力(この場合は人感センサ)に応じて変える働きがあります。

**if** - 条件を調べます

**do** - その条件が正しい(成立する)ならば、個々の処理を行います

**else** - そうでない(その条件が成立しない)場合は、この処理を行います

**01** "Logic" (論理) をクリックします

**02** "if/do/else"ブロックを、Labelの文字設定ブロックの下にはめます

**03** "Math"をクリックします

**04** "Logic"の中にある "[]=[]"ブロックをもってきます。"get PIR status"ブロックをコピー(duplicate)して、この最初の部分にはめます。その右側の部分には、"Math"にある "number" (数値) ブロックをはめて、その値を"1"にします。

**05** "do"のところには、スピーカブロックやLEDブロックをはめて、音を鳴らせたり光らせたりします。"else"のところには、LEDの色を黒(つまり消灯)にしてスピーカの音を止めるようにします。

人感センサUnitをM5GOにつないで、プログラムを実行させてみましょう。

## 6 植物をモニタリング

ここでは、土の水分量や水位を計測する"Earth" (地球) ユニットを使います。先ほどに続いて論理機能についてより詳しく、比較と論理値について学びます。

### 1. 論理値

前の章では"if"ブロックと"=" (イコール) ブロックを使いました。これは、人感センサの値が1に等しかったらなにかの動作をする、というものでした。論理値は、true (真) とfalse (偽) の2つの値をとるもので、このifの動作と同じことができます。さらに、どういときに"true"になるか、をプログラムで書くことができます。例えば、" $2+3=5$ "という式は正しいですが、イコールの右が5以外の値にした式は正しくない式になります。プログラムの動作の制御はこの論理値を使って行うことができます。"Logic"には、"true"と"false"の「値」のブロックがあり、"if"ブロックなどの中で使うことができます。

$$1+1=3 \quad \text{False}$$

$$2+3=5 \quad \text{True}$$

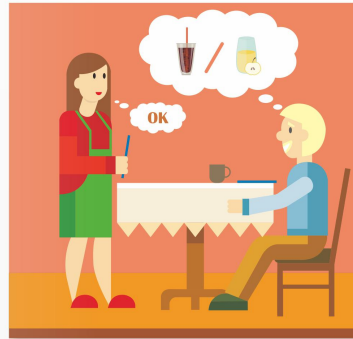
### 2. 論理式

論理値の考え方を使うと、複数の条件を組み合わせることでプログラムの動作を制御することができます。論理式には"and"、"or"、"not"があります。以下で順に見ていきましょう。



レストランで、氷入りのコーラを注文するとしましょう。つまり"coke"と"ice"の両方がそろったら、自分の注文通り、つまり"true"ということです。水や、氷なしコーラでは、"コーラ"と"氷"が両方そろっていませんから、"false"です。

次に、のどが潤いて、とりあえず何か飲みたいとしましょう。店員さんに、コーラか、なかったらジュースを注文します。つまり"coke"か"juice"が出てきたら"true"です。"coke"でも"juice"でもなかったら(例えば水)、"false"です。



最後に、コーラが飲めないアレルギー体質で、コーラ以外のものを注文するとします。つまり、"coke"以外のものが出てきたら"true"です。コーラ以外ならなんでも"true"で、コーラだけ"false"となります。



### 3. Earthセンサを使ってみよう

Earthセンサは土の水分量を計測できます。

01 Unitメニューから"Earth"ユニットを追加します

02 "Units"の中の"Earth"をクリックする

03 UIマネージャでLabelを2つ置き、それぞれにEarthセンサのアナログ値(水分量の値)とデジタル値(水分が多いか少ないか)を表示させることにします。

EarthセンサはポートBにつなぐように設定し、実際にポートBにつなぎます。

"Earth"にはブロックが2つあります。"get digital value"(デジタル値を取得)は水分が多いか少ないかの1または0の値を使います。"get analog value"ブロックは、水分量を0から1023の値で取得できます。

#### Earthユニットのしくみ

水は電気を通す性質があります。土の中に水分が多いほど電気が流れやすくなります。Earthユニットは、2本の端子の間の電気の流れやすさ(電気抵抗)を計測します。

### 6-1 植物をモニタリング

土の水分量を表示するのに、Earthユニットを使って、土が乾いているか、ちょうどよいか、水分が多すぎるか、を判別する簡単なプログラムで試してみましょう。そのためには、"else if"を使います。

3つの水分量の状態を次のように判別して表示してみます。

1. 水分量が20以下になったら、LEDを赤にして警告音を鳴らす
2. 水分量が20から40の間ならば、LEDを黄色にする
3. 水分量が40から60の間にあつたら、LEDを緑にする
4. 水分量が60よりも大きかったらLEDを青にする

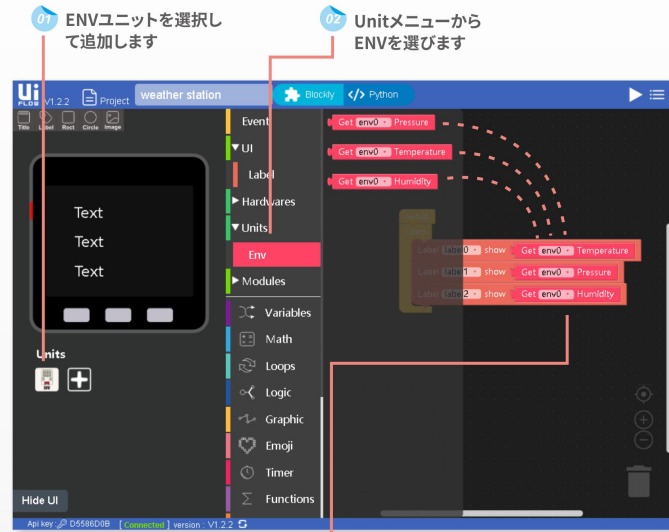
Earthセンサをつないで、このプログラムを実行し、Earthセンサの足を土にさして水分量をはかってみましょう。

# 7 気象観測

ここでは、ENVユニット（環境センサユニット）を使って、温度、湿度、気圧を計測して表示して、他にもいろいろな出力に使ってみます。変数の使い方や、より複雑な論理機能について学びます。

## 1. ENVユニットを使ってみる

ENVユニットは、温度、湿度、気圧を計測できます。



01 ENVユニットを選択して追加します

02 UnitメニューからENVを選びます

03 Labelを3個追加して、ENVの3個のブロックをすべて、Label表示のところにはめます。



## 2. 温度、湿度、気圧

このプログラムを実行すると、画面に3つの数値、順に温度、湿度、気圧が表示されます。

温度、湿度は聞いたことがあると思いますが、気圧って？

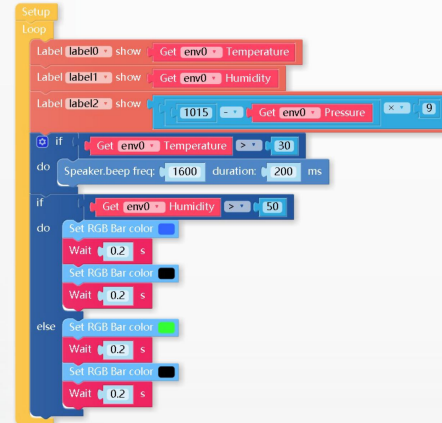
気圧とは、空気が表面を押す力です。単位はPa（パスカル）や、psi（pressure per square inch: 1平方インチ（約6.5平方センチ）あたりの圧力）です。

ENVユニットは、気圧はhPa（ヘクトパスカル、100hPa=1Pa）で得られます。高度が9m上がるごとに気圧は1hPaずつ下がる性質があるので、気圧から高度がわかります。

地表での圧力が1015hPaだとしましょう。ENVセンサで計測した圧力から1015hPaを引き、それを9倍した値が高度（単位：メートル）ということになります。

### 1-1 気象観測機

- (1) 温度が30度よりも高かったらスピーカから音を鳴らします
- (2) 湿度が50%よりも高かったらLEDを青にして、それ以下だったら緑にします。
- (3) 温度、湿度、気圧を画面に表示します
- (4) 計測された気圧から地表での気圧を引いて、それを9倍して高度を表示します



ENVユニットを接続してプログラムを実行したら、ENVセンサに行きを吹きかけて、値の変化をみてみましょう。高いところに登って気圧の変化をみてみましょう。

# 8 時計をつくる

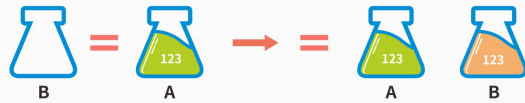
ここでは、これまでやってきたことを使って、時計をつくってみましょう。変数についてより深く学び、また条件の入れ子構造について学びます。

## 8-1 変数について

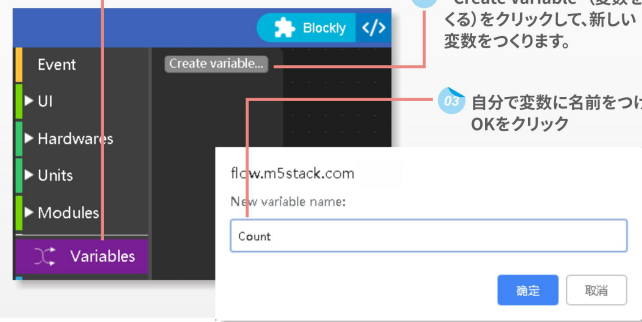
- (1) 変数は、ビンのようなものです。ビンに、例えば"A"という名前をつけて、そこに数値や文字を入れます。
- (2) ビンに別の値を入れると、もとの値は消えて、その値に書き換わります。



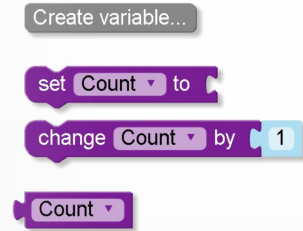
- (3) ある変数(ビン)の値を、別の変数に入れることもできます。この場合、元の変数の値が、入れた先の変数の値にコピーされます。



- 01. ブロックメニューの"Variable" (変数)をクリック
- 02. "Create Variable" (変数をつくる)をクリックして、新しい変数をつくれます。
- 03. 自分で変数に名前をつけてOKをクリック

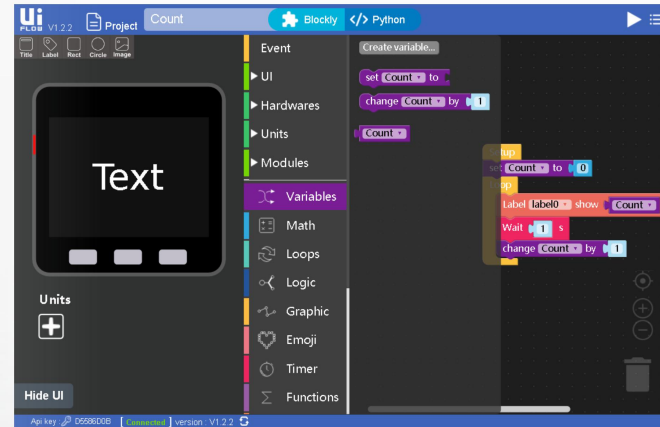


新しい変数をつくったら、その変数のブロックが現れます。



### 1. 数をカウントするプログラム

"count" (カウント) という名前の変数をつくって、その値を0にする (代入する) ブロックをSetupにはめます。Labelを追加して、そこに変数countの値を表示させて、これを1秒ごとに値を増やしていきます。

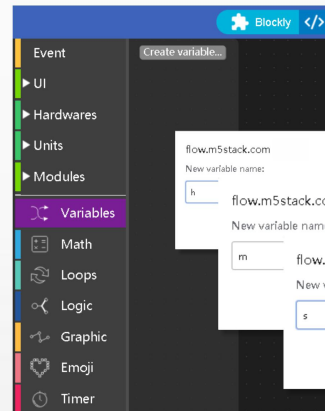


これでcountの値は1秒ごとに1ずつ増えて行くので、秒で測る時計(ストップウォッチ?)になります。60秒=1分、60分=1時間、24時間=1日、ですね。

## 8-2 時計をつくる

### 1. 画面の構成

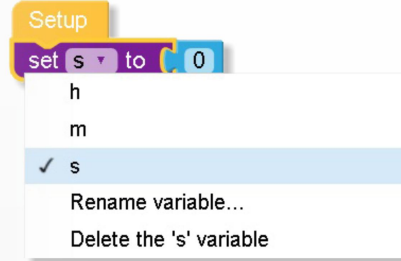
Labelを3個(秒、分、時)つくります。とりあえずすべてのLabelの表示は"0"にしておきましょう。好きなフォントや色を設定しておきましょう。分かりやすいように、それぞれのLabelの名前を、"h"(hour=時間)、“m”(minute=分)、“s”(second=秒)に変えておきます。それぞれの数値の間には、“.”(コロン)を表示するLabelを置いておきましょう。



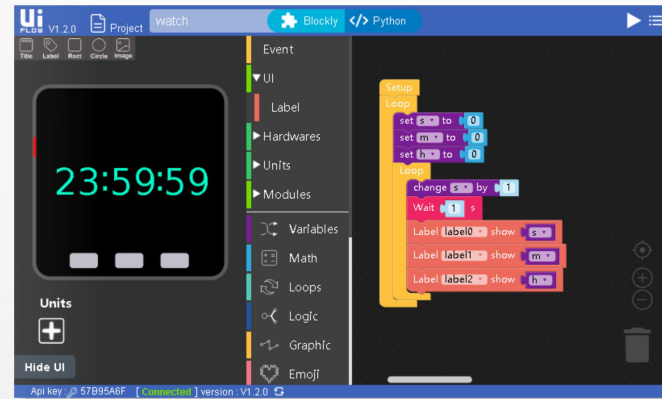
### 2. プログラムを書く

時、分、秒を入れておく3つの変数(名前はh, m, s)をつくります。

Setupの下にLoopブロックをはめ、その中に、変数の値を設定するブロックをはめて変数"h"を0に設定しておきます。同じように変数"m"と"s"も0を入れておきます。



- (1) 3つの変数の値を0にするブロックの下に、もう1つLoopブロックをはめます。
- (2) 時・分・秒を表示するLabelを追加しておきます。
- (3) 変数"s"の値を1増やすブロックをはめ、その後1秒間待つようにします。
- (4) Loopブロックや値を増やすブロック、Labelを変える部分は時間はほぼゼロなので、これではほぼ1秒毎に変数sの値が増えていきます。





```

Setup
set s to 0
set m to 0
set h to 0

Loop
Label label0 show s
Label label1 show m
Label label2 show h
Wait 1 s
change s by 1
if s ≥ 60
do
set s to 0
change m by 1
if m ≥ 60
do
set m to 0
change h by 1
if h ≥ 24
do
set h to 0
    
```

### 3. 論理の入れ子構造

秒(s)が増えていって59の次に、0に戻ると同時に分(m)が1増えます。

次に分(m)が59より大きくなったかをチェックして、そうなら"m"を0に戻して、時(h)を1増やします。

次に、時(h)が24になったかをチェックし、そうなら時、分、秒をすべて0に戻します。

### 4. 機能の追加

この時計に、指定した時刻になったらアラームが鳴る機能を追加してみましょう。アラーム音は、1000ミリ秒 (=1秒) を10回鳴らすことにします。もちろん時計が1秒毎に進む機能はすでに作ってありますが、アラームを鳴らせている間も、忘れずに1秒毎に1ずつ増やすようにしておきます。

```

Setup
set s to 0
set m to 0
set h to 0

Loop
Label label0 show s
Label label1 show m
Label label2 show h
Wait 1 s
change s by 1
if s ≥ 60
do
set s to 0
change m by 1
if m ≥ 60
do
set m to 0
change h by 1
if h ≥ 24
do
set h to 0

Button A wasPressed
change h by 1

Button B wasPressed
change m by 1

Button C wasPressed
change s by 1

if h == 0 and m == 0 and s == 5
do
repeat 10 times
do
Speaker.beep freq: 1600 duration: 1000 ms
Label label0 show s
Label label1 show m
Label label2 show h
change s by 1
    
```

- ボタンをつかって、時刻を設定できるようにしておきます。
- (1) Aボタンを押したらh(時)を1増やします。
  - (2) Bボタンを押したらm(分)を1増やします。
  - (3) Cボタンを押したらs(秒)を1増やします。

## 9 加速度センサで遊ぶ

ここでは、**加速度センサ**やジャイロの向き感知機能を使って、サイコロと歩数計をつくってみます。加速度センサの原理と、乱数と確率について少し学びます。

### 1. 加速度センサとは?

加速度センサは、ものが動くときの加速の度合いを計測します。加速とは数ミリ秒、数マイクロ秒の短い時間に速さがどれくらい変化するか、という量です。

例えば、まっすぐな道で、車が1秒間に20mph(時速32km)から30mph(時速48km)まで加速したとしましょう。このときの加速度は、1秒あたり10mph(時速16km)となります。

ちょっと問題:2台の車だスタートラインに止まっていて40mph(時速64km)まで加速したとします。その速度になるまでに、片方の車は6秒、もう片方は10秒かかったとします。2台の車の加速度は同じでしょうか?

もちろん答えは、違う、です。6秒かかった車のほうが、10秒かかった方よりも、加速が急、つまり加速度が大きいわけです。



加速度は、私達の身の回り、例えば手をちょっと動かすときでも現れます。動かす向きが変わるのも、加速度が生じているわけです。

加速度センサを使って、サイコロをつくってみましょう。



- (1) "dice" (サイコロ) という名前の変数をつくって、これにサイコロの目の数を入れておきます。
- (2) この変数のブロックをつくります。もちろん変数の名前を変えてもいいです。
- (3) Setupのところで、この変数の値を1にしておきます。

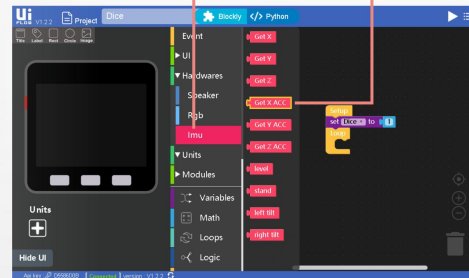
Setup

set Dice to 1

- ✓ Dice
- Rename variable...
- Delete the 'Dice' variable

01 "Hardware"の中の"IMU"(加速度センサ・ジャイロなどが入ったセンサの名称)をクリックします。

02 "Get X ACC"(X軸のACC(加速度)を取得)を選びます。これはMSGOのX方向(横方向)の加速度を計測します。

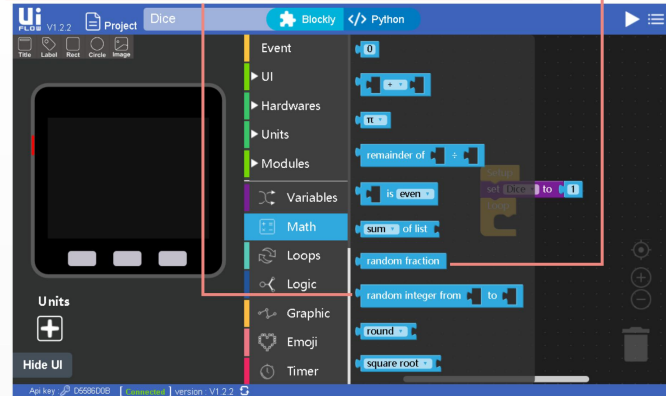


## 2. 乱数

"Math" (数学) グループには、2つの乱数をつくるブロックがあります。

"random fraction"は、小数の乱数をつくります。

"random integer from a to b"は、指定した範囲 (aとbの間) の整数の乱数をつくります。



"set variable to" (変数を設定) ブロックを持ってきて、"random integer"ブロックにはめ、その範囲を1から6にしておきます。これはサイコロの目の範囲です。



## 3. Whileループ

"Loop" (ループ) の中に、"While"というブロックがあります。これは、ここに書いた条件が"true" (真) ならばずっとここに書いてある動作をします。その条件が"true"でなくなったら (つまり"false")、このループを抜けて、プログラムの動作は次に進みます。



ここでは、MSGOを振っている間はずっと乱数でサイコロの目の値を得るようにしましょう。ここでは加速度センサのX軸 (横) とY軸 (縦) のどちらかが1.5以上または-1.5以下ならば、振られている、と判断することにして、そのために"or"を使っています。振っているかどうかをLEDでわかるようにもしましょう。サイコロを振っていなかったら、サイコロの目は最後の値のままにしておきます。得られた乱数のサイコロの目に応じてサイコロの目の形を表示するのにもifブロックを使っています。

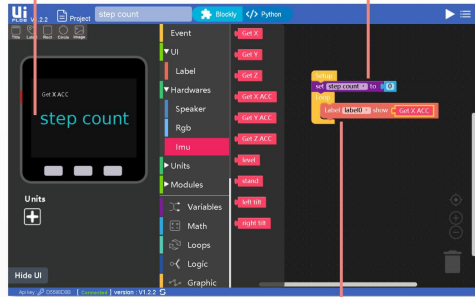


## 4. サイコロの目の表示

if-elseブロックを"Logic"から持ってきて、歯車マークから"elseif"を5個追加し、最後だけelseにしておきます。それぞれのifの条件には、サイコロの目の値が1から6と等しいかのチェックを順に行って、それぞれに応じた絵文字ブロックを表示しています。

### 歩数計

- 01 Labelを2つ作って、1つは加速度、もう1つは歩数を表示させます。
- 02 歩数の変数countをつくってSetupのところ値を0にしておきます。



- 03 加速度センサの値はずっと変化し続けるので、毎回Loopで値を取得するようにします。
- 04 "Label Show" (ラベルを設定) ブロックの中に、"Get X ACC" (X軸の加速度を取得) をはめておき、加速度の値を表示させます。

```

Setup
  set count to 0
Loop
  Label label0 show Get X ACC
  if Get X ACC > 1.5
  do
    change count by 1
    Label label1 show count
    Wait 0.5 s
  Button A pressed
  set count to 0
  Label label1 show count
  
```

- 05 加速度の値が1.5を超えたら、1歩歩いたこととして、歩数の変数countを1増やします。
- 06 歩数の値を変更したので、それを表示させておきます。
- 07 1歩で1回だけカウントするように、0.5秒待つようにしておきます。
- 08 Aボタンを押したら歩数をリセットするようにしておきます。

プログラムの実行はとても速いので、1歩で1カウントずつ増えるようにwait(待つ)を入れていました。うまく歩数をカウントできなかったら、加速度の判定値(1.5)や待ち時間を変えてみましょう。



## 10 ポテンショメータ(角度センサ)

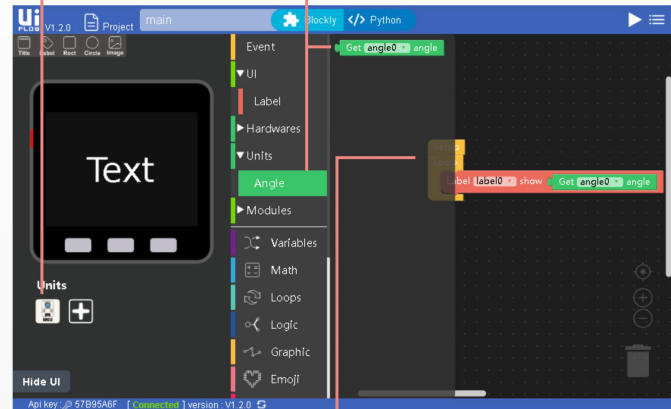
ここでは、角度センサ(ポテンショメータ)を使ってLEDの明るさを変えてみます。ポテンショメータの使い方と、変数の値の範囲を変換する方法について学びます。

### 1. 角度センサのテスト

角度センサは、ツマミの位置を取得できます。

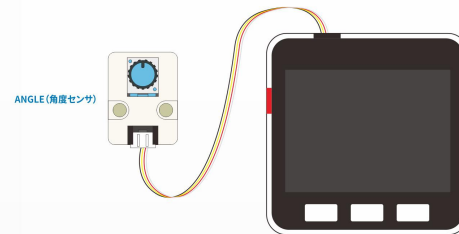
01 まずUnitメニューから"Angle Sensor"(角度センサ)を追加しておきます。

02 Angle Sensorの穴から"Get Angle"(角度を取得)ブロックをもってきます。



03 "Label Show"に、"Get Angle"ブロックをはめます。

角度センサをポートBに接続し、プログラムを実行させてみましょう。角度センサのツマミを回すと、表示される値は0から1023の範囲で変化するはずです。

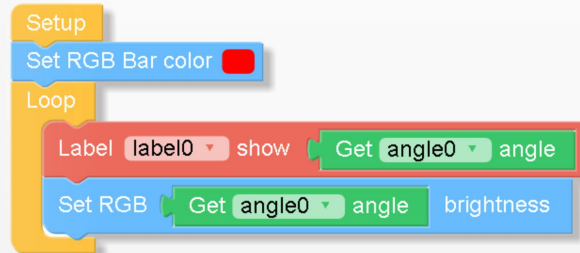


#### 角度センサのしくみ

角度センサは「ポテンショメータ」や「可変抵抗」とも呼ばれ、電気抵抗が変化する素子です。ツマミを回すと中にある抵抗素子の端子の位置が変わり、その位置に応じて電気抵抗が変わります。中の抵抗の両端には電源のプラス(5V)とマイナス(0V)がつながっていて、ツマミの位置に応じて、真ん中の端子の電圧が変わります。M5GOではその電圧を0から1023の値として計測できます。

### 2. 明るさを変える

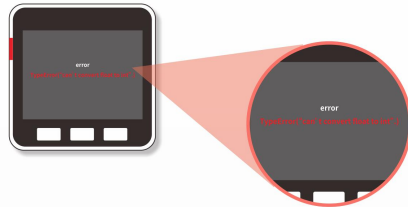
角度センサの仕組みと働きが少し分かったので、次のプログラムでLEDの明るさを変えてみましょう。



```

Setup
  Set RGB Bar color [Red]
Loop
  Label label0 show [Get angle0 angle]
  Set RGB [Get angle0 angle] brightness
  
```

このプログラムを実行してツマミを回すと、こんなエラーが出るはずですよ。



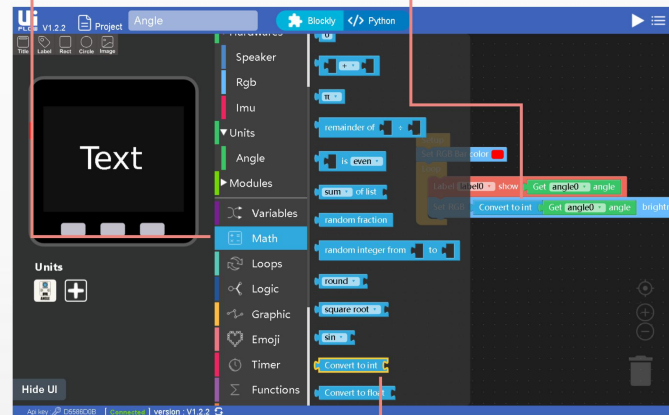
このエラーは、ブロックに与える値の「型」が違うのが原因です。LEDバーの明るさは整数で指定しますが、角度センサの値(ツマミの位置)はfloat型(小数を含む数=実数)です。

### 3. 整数と実数

整数(int)と実数(float)は、どちらも数字を表す形式です。整数は0, 2, -4のように小数を含まない数しか表せません。実数は、-1.1や2.5, 3.9のように小数を含む数を表せます。

01 "Math"を選びます

02 "Convert to Int" (整数に変換) ブロックを持ってきて、GetAngleブロックの前にはめます。これは実数(float)を整数(int)に形式を変換します。

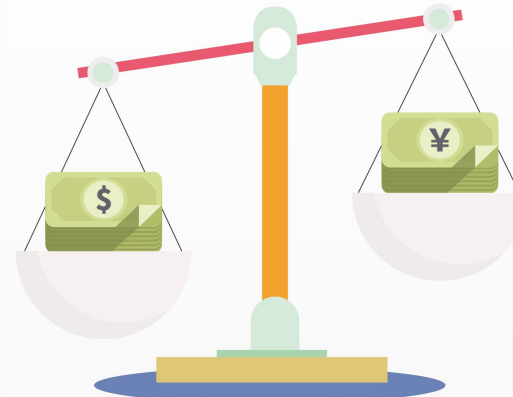


02 "Convert to Int" (整数に変換) ブロックを選びます。

これでエラーが出ずにプログラムが動作するはずですよ。でもLEDの明るさは255が最大ですが、角度センサの値は最大で1023です。つまりツマミを回しすぎるとLEDの明るさは最大から変わらないことになります。

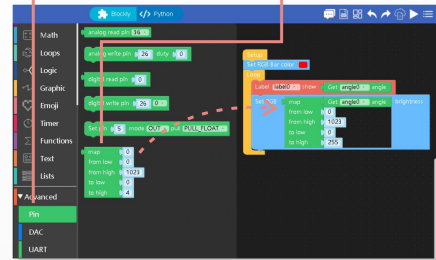
### 4. 値の変換

そんなときは、値の範囲を変換する"Map" (値の変換) ブロックを使えば大丈夫です。例えばドルからポンドに両替するときは、金額の数字の変換をします。その比率(為替レート)は変化しますが、お金がもっている価値は変わりません。



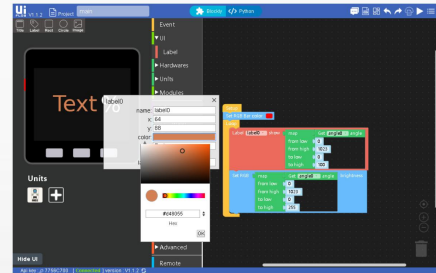
"EasyIO"グループにある"map"ブロックを使って、数値の範囲を変換してみましょう。ツマミの位置を取得する"Get angle block"ブロックを"map"ブロックにはめます。最初の2つの数値は、"from low"と"from high"と書いてあり、これは変換する元の数値の範囲(最小と最大)を表します。ここではもう角度センサの値の最小と最大である0と1023が入っているはずですが、その次の2つの数値には、0と255を入れておきましょう。これはLEDの明るさを指定する値の最小と最大です。ちなみに"map"ブロックを使うときには、さきほどの"convert to int"ブロックは使わなくても大丈夫です。

- 01 "Advanced"の中にある"EasyIO"を選びます。
- 02 "map"ブロックを、"RGB Brightness" (LEDの明るさ)ブロックにはめます。



### 6. 画面への明るさの表示

画面には、明るさを0から100%の範囲で表示させてみましょう。そのために、もう1つ"map"ブロックを使い、変換先の値の範囲を0から100にしておき、Labelに設定します。



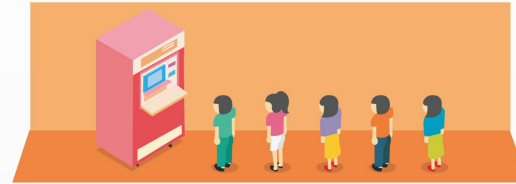
このプログラムで、ツマミをまわすとLEDの明るさを最小から最大まで調節できるはずですが、

## 11 信号機

ここでは、RGBユニットを使って信号機をつくってみましょう。変数と繰り返し処理について、より詳しく学びます。

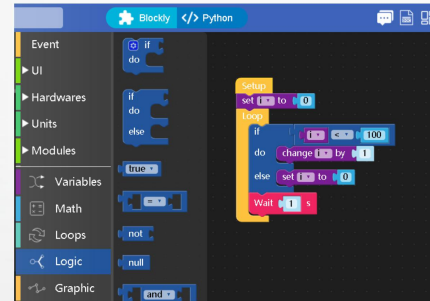
### 1. 繰り返し処理とは?

"count" (カウント) ブロックは、変数(なければ自動的に割り当てられます)を増やしながら、指定した値になるまで繰り返す処理を行います。値を増やす量は変えられます。例えばATM(銀行のお金を引き出す機械)に100ポンドが入っていて、5人が、20ポンドずつ引き出すとします。最後の一人がお金を引き出したら残金がなくなります。これと同じように、変数の値が指定した値(この場合は0ポンド)になるまで繰り返す、というものです。



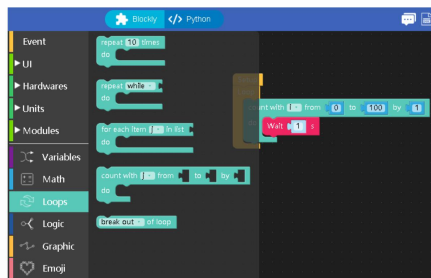
ちょっと考えてみよう: プログラムで繰り返し処理を使うと、どんないけないことがあるのでしょうか?

指定した回数を繰り返す処理を2つの方法でプログラムを書いて、どちらがよいか比べてみましょう。カウント値の変数を0から100まで1ずつ増やしていく場合を考えます。



#### 方法1

変数"i"が100より小さい間、"i"を1ずつ増やしていき、"i"が100になったら0に戻します。数が増えていく様子を確認できるように0.1~1秒のwaitを入れておきます。これらを1つのLoopブロックの中に入れます。

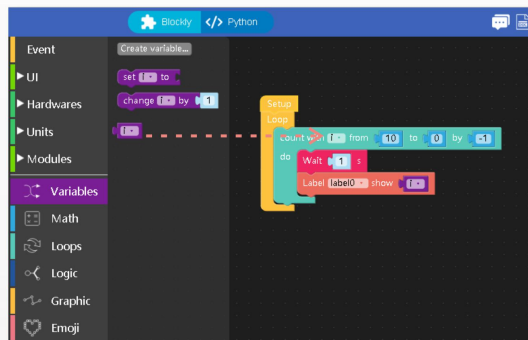


**方法2**  
 "count with i from"ブロックを使って、最初の値を0にして、max(最大値)を100、1回あたりの増える数を1にしてみます。確認のためにループの中に1秒の待ち時間を入れておきます。これでループの処理が終わるまで100秒間かかることになります。"i"の値をLabelで表示しておけば、0から1ずつ増えていって100まで行ったら0に戻るのが確認できるはずですよ。

方法2のほうが簡単ですね。このようにcountブロックを使ったほうが繰り返し処理のプログラムは簡単になることが多いです。

**補足**

UIFlowでは、Countブロックを置くと、自動的に変数"i"がつけられますが、別の変数を作成して、それをドロップダウンリストから選ぶこともできます。



**2. 信号機のプログラム**



- 01 Unitメニューから"RGB"ユニットを追加します。
  - 02 "Unit"の中の"RGB"を選択します。
  - 03 "RGB index" (RGBの色を指定)ブロックを3個、Setupにはめます。
  - 04 それぞれの"RGB index"ブロックの番号を1~3に変更します。
  - 05 "RGB"ブロック1~3の色を、それぞれ黄、青(緑)、赤に設定します。
- 06 RGBユニットをポートBに接続し、Unitを追加したときに設定した接続ポートもポートBになっていることを確認します。プログラムを実行すると、RGBユニットのLEDが点灯します。

**3. 信号機の表示の準備**



UIマネージャでLabelを2個追加します。Labelのフォントを大きい文字にしておきます。2つ目のLabelは、1つ目のLabelの右に置いて、文字列を"s"("秒"の意味)にしておきます。



#### 4. 信号機のプログラム

プログラムは、青、黄、赤の制御の3つの部分に分けて書きます。

01 "Set rgb index" (RGBの番号を指定)で番号を"1"にして、色に青(緑)を選んでおきます。

02 RGBブロックの下に、Loop(ループ)の"count with i from"ブロックを置きます。変数iの値の範囲を6から0、値の変化を"-1"にします。これで変数iの値は1ずつ減っていきます。

03 変数iの値をLabel(label0)に表示させて、カウントダウンしていく様子を確認できるようにします。

04 1秒の待ち時間を入れます。

05 カウントダウンが終わったら、そのLED(ここでは青(緑))を消します。"Set rgb index"ブロックで色を「黒」にしておきます。



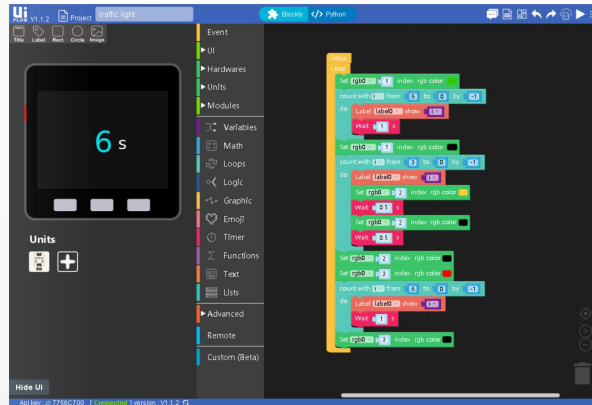
#### 黄色の制御

普通の信号では、黄色の時間は青(緑)や赤より短いですが、さきほどの青(緑)のプログラムをコピーして、カウントの値(ループする回数)を3、待ち時間を0.5秒にしておきます。RGB index(番号)を"2"にして、その色を黄色とし、0.5秒の待ち時間のあとにLEDを黒にして消しておきます。

#### 赤の制御

LEDの番号を3にして、色を赤に設定します。青(緑)のプログラムをコピーして、LEDの番号と色だけ変えておきます。

一番最後も、LEDを黒にして消しておいてから、最初に戻るようにします。



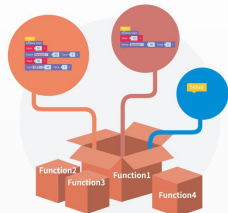
プログラムを実行すると、信号機がプログラム通りに動くはずですが、

## 12 光センサと関数

ここでは、光センサの使い方と、「関数」について学びます。関数の基礎について理解して、論理についてより深く学ぶとともに、光センサでLEDを制御してみます。

### 1. 関数の基礎

「関数」は、箱のようなものです。箱に名前をつけて、中にプログラムを入れておいて、使いたいときに、その名前で、箱の中に入っているプログラムを実行させられます。



### 2. 関数の作り方

01 ブロック一覧の中の "Function" (関数) をクリックします。

02 "to do something" (何かをする) ブロックを持ってきます。

03 関数の名前のところをクリックして、関数に名前をつけられます。

04 その関数のブロックの中に、その関数を使う時に実行したいプログラムを書きます。

### 3. 関数を定義する

関数をつけて名前をつけたら、「Function」一覧の中に、つけた関数の名前のブロックが現れます。このブロックを、他のブロックと同じように使えば、その関数の中に書いたプログラムを実行させられます。

### 4. 光センサを試す

光センサは、明るさに応じた数値を計測できます。

01 Unitメニューから "Light" (光センサ) Unitを追加します。

02 "Unit"一覧の中の "Light" から、"get light analog value" (明るさの数値を取得) ブロックをもってきて、Loopの中に置きます。

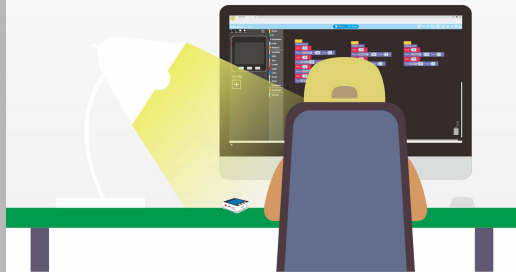
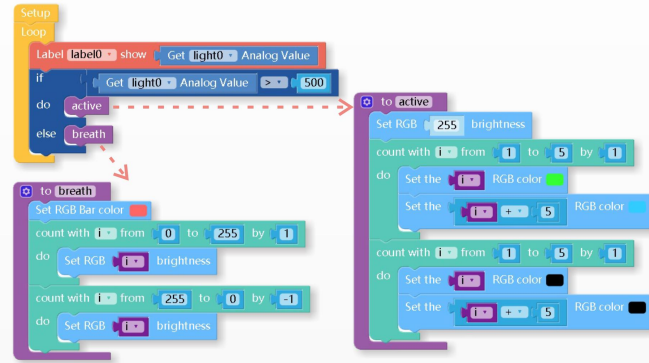
03 Labelをつかって、その表示内容のところに、さきほどの "get light analog value" をはめます。

LightユニットをMSG01につないで、このプログラムを実行させてみましょう。光センサの前に手をかざしたり、近づけたり遠ざけたりすると、表示される値が0から1023の範囲で変化するはずですよ。

ちょっと考えてみよう:光センサはどうやって明るさを数値に変えているのだろう?  
光センサは、光可変抵抗とも呼ばれる素子です。ポテンショメータ(可変抵抗)と同じように電気抵抗の値が変わりますが、ツマミはなく、明るさに応じて変わる性質があります。それを使って、電圧の変化を発生させて、計測しています。

## 5. 光センサの関数

ここでは、光センサを使って、明るさ制御のための別々の動作をする2つの関数をつくってみましょう。昼間は光センサの値は500以上になりますが、そのときは、関数"active"(活動)を使って、ライトをON/OFFさせます。夕方になって光センサの値が500以下になったら、関数"breath"(休憩)を使って、ゆっくりLEDがほわほわ光るようにします。



## 13 M5GO アドレス帳

ここでは、リストについて学び、簡単なアドレス帳をつくってみます。リストの考え方と使い方について学びます。

### 1. リストとは?

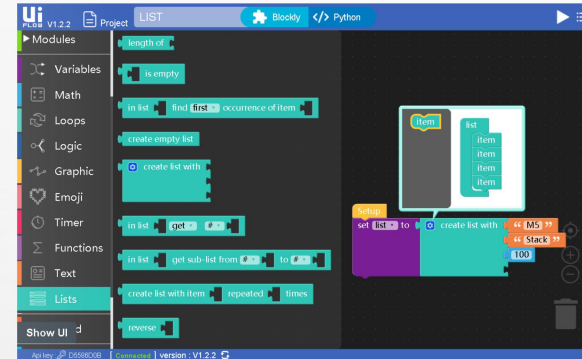
リストとは、いくつかの変数を一つにまとめて扱えるようにしたものです。変数は箱(またはピン)のようなものですが、リストは箱(ピン)を入れておく箱(コンテナ)のようなものです。コンテナの中のピンには番号がついていて、その番号で指定するピン(変数)の中身を読み出したり書き換えたりできます。



リストをつくるには、変数にリストを割当てて、そのリストの中の中の番号で、使いたい変数を指定します。

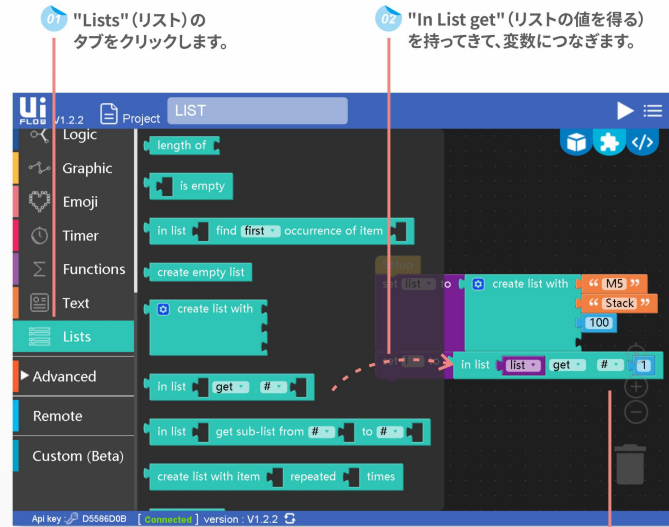
### 2. リストをつくる

まずリストに入れる変数をつくれます。それから、"Lists"の中にある"set list to"ブロックをSetupにはめます。その中で、



### 3. リストの使い方

リストをつくって中に値を入れたら、それをどう使うのか、次の例でみてみましょう。



01 "Lists" (リスト)のタブをクリックします。

02 "In List get" (リストの値を得る)を持ってきて、変数につなぎます。

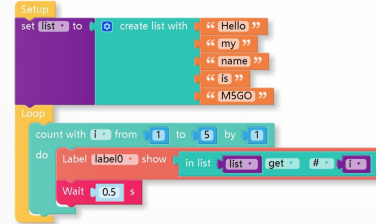
03 ドロップダウンメニューから、リストの中の番号を指定して、変数の値を得たり消したりできます。

UiFlowではリストの変数の番号は1からはじまりますが、Pythonでは0からはじまります。この例で"M5"という文字を取得するには番号(index)に1を指定します。

ちょっと考えてみよう: リストの中の番号を巡回して(最後の番号の次は最初に戻る)使うことはできますか? まさにそれがリストが便利な理由の一つです。

### ちょっとためてみよう

カウント(count)ブロックを使って、リストの番号を巡回できます。次のプログラムを実行してみてください。

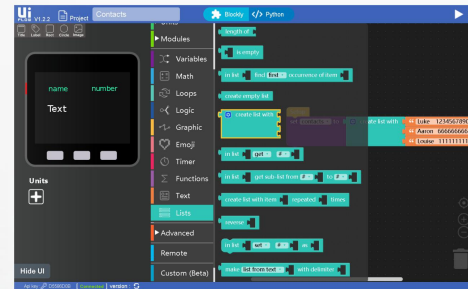


### M5GOアドレス帳

ここで作るプログラムは、リストに保存されている名前とその(リストの)番号を表示し、AボタンとCボタンで順番に切り替える(巡回させて)ものです。

#### 1. UIの設定

まずLabelを3つおきます。名前、番号の表示に1つずつ、最後の1つは空白(文字列なし)にしておいて、ここにはリストから読みだした情報(電話番号)を表示させることにします。



#### 2. リストにデータを加える

リストをつくって、"contacts" (連絡先) という名前をつけます。"Text"ブロックのタブのリストにある、中身が空の文字列(string(text))ブロックをもってきて、好きな数だけコピーして名前と電話番号をいれていきましょう。

### 3. リストを使う

01 ブロックのリストの"List"タブをクリックします。

02 "In List get" (リストの～を取得) を持ってきて、変数を割り当てます。

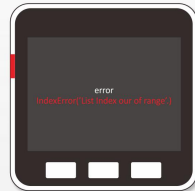
03 "ボタンが押されたら"イベントを2個つくります。その1つを"Cボタン"にして、"change by" (変数を変える) ブロックを置き、変数名に"i"を選んで値を1増やすようにします。もう1つは"A"ボタンにして、変数"i"を1減らすようにします。これでA/Cボタンを押したら変数"i"が1増える/減るようになりました。

The code blocks shown are:

- Setup:**
  - set `contacts` to create list with
    - “ Luke 1234567890 ”
    - “ Aaron 666666666 ”
    - “ Louise 111111111 ”
  - set `i` to 1
- Loop:**
  - Label `label2` show in list `contacts` get # `i`
- Button A pressed:**
  - change `i` by -1
- Button C pressed:**
  - change `i` by 1

### 4. 動作テスト

プログラムを実行して、A/Cボタンを押してみましょう。まずCボタンを押して正しく動作しているか(表示されるリストの項目が1ずつ増えていくか)を確認しましょう。ボタンを何回か押すとエラーがでるはずですが、何が起ったのでしょうか？



### 5. リストの番号のエラー

AやCボタンを何回か押していると、変数"i"が、リストの中にある変数の番号の範囲を超えてしまいます。今回の例ではリストには3つしか項目がないので、変数"i"が4になっても、リストに4番目の項目はなく、それでエラーになってしまいます。

### 6. エラーの修正

このエラーは、"if"ブロックを使えば起こらないようになります。A/Cボタンに対する動作を修正してみましょう。

Aボタンを押したあと、変数"i"の値が1より大きいかをチェックし、1より大きければ(つまり2以上)、1減らしても大丈夫なので1減らします。そうでなければ(つまり"i"が1のとき)、"i"はリストの最後の番号の3にします。

The code blocks shown are:

- Button A pressed:**
  - if `i > 1`
    - do: change `i` by -1
  - else: set `i` to 3
- Button C pressed:**
  - if `i < 3`
    - do: change `i` by 1
  - else: set `i` to 1

Cボタンのほうは、"i"が3より小さいかをチェックし、そうならば(つまり"i"が2以下)1増やします。そうでなければ(つまり"i"が3)リストの最初の番号の1にします。



## 14 赤外線リモコン

ここでは、M5GOから、別のM5GOを赤外線リモコンで制御してみましょう。  
ここでは画像の表示のさせかたと赤外線リモコンの使い方について学びます。

### 1. 赤外線とは?

赤外線(Infrared、IRとも呼ぶ)は、紫外線と同じく、目に見えない光の一種です。赤外線は1800年代にイギリスの物理学者William Herschel(ハーンセル)が発見しました。ハーンセルは太陽の光をプリズムで分け、別れた光の温度を測ってみました。そのとき、赤よりも外側が最も温度が高いことを見つけ、そこには目には見えない光があるはずだ、と結論付けました。これが赤外線です。目に見えない赤外線は、テレビやエアコンの制御リモコンに使われています。



ここでは、M5GOをテレビのリモコンのように赤外線リモコンとして使ってみましょう。M5GOを2台と、IRユニットを2個使い、それぞれに別のプログラム(送信側と受信側)を書きます。

### 2. 赤外線リモコン

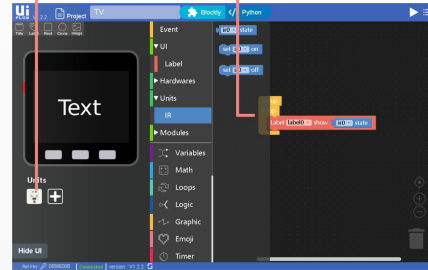
01 Unitメニューから、"IR"ユニットと、2つの"ボタンが押されたら"ブロックを置きます。

02 "Aボタンが押されたら"ブロックでは、"set IR on"(赤外線をON)ブロックをはめ、Labelに赤外線がONであることを表示させましょう。これでIRユニットから赤外線が発信されます。

03 2つ目の"Aボタンが離されたら(released)"ブロックでは、"set IR off"(赤外線をOFF)ブロックを置いて、赤外線をOFFにします。

### 3. 画面表示のプログラム(受信側のM5GO)

- 01 IRユニットを追加します。
- 02 "Loop"(繰り返し)ブロックを置いて、"Labelを表示"ブロックに"IR state"(IRの状態)を表示させます。"IR state"ブロックをLabel表示の中に置きます。



#### プログラムの詳細

2台のM5GOで、それぞれにIRユニットをつないで、それぞれのプログラムを実行させてみましょう。IRユニットを向かい合わせに置き、送信側(1台目)のAボタンを押してIR信号を送ってみます。

- (1) 送信側でAボタンを押すと、赤外線が発信されます
- (2) 受信側では赤外線信号が受信されているかを確認していて、受信されたら画面表示が"0"から"1"に変わります。

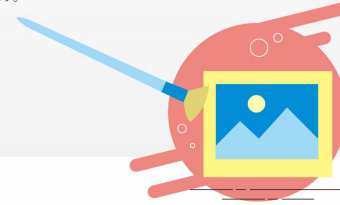
これで赤外線の送受信ができました。もっといろいろなことができそうですね。

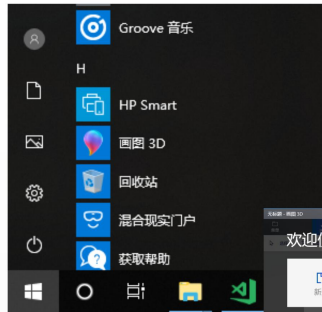
#### 画像の準備

赤外線信号の送信と受信ができるようになったので、もうちょっと凝ったことをやってみましょう。

#### 4. 使う画像について

- (1) 画像のサイズ(画素数)は320×240以下とします。これはM5GOの画面の画素数です。
- (2) 画像ファイルの形式は"bmp"か"jpg"にします。
- (3) ファイルのサイズは25KB(バイト)以下にします。

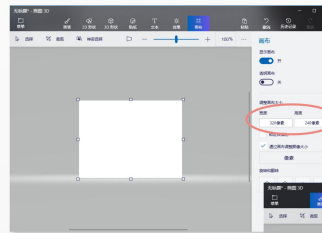




### 5. 画像編集ソフトで画像を用意する

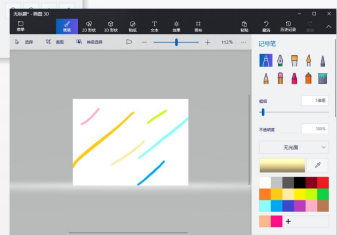
WindowsやMacなどには、標準で画像編集ソフトが入っているので、これを使ってM5GOに表示する画像を用意しましょう。パソコンで画像編集ソフトを開きます。ここでは、Windows10についての"Paint3D"を使った例を紹介します。

2. 新しいキャンバスを開きます



3. キャンバスのサイズを320×240に設定します。これは"イメージ(画像)"メニューなどにあります。

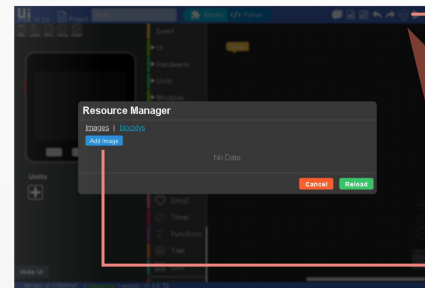
4. ペイントブラシや鉛筆などを使って、絵を描きます。



5. 4コマ漫画のようにストーリーをもたせて何枚かの絵を描くのもいいですね。絵を描いて保存するときには、必要ならば"圧縮"を指定してファイルサイズが25Kバイト以下になるようにします。ファイル名はimage1~image4のようにしておきます。



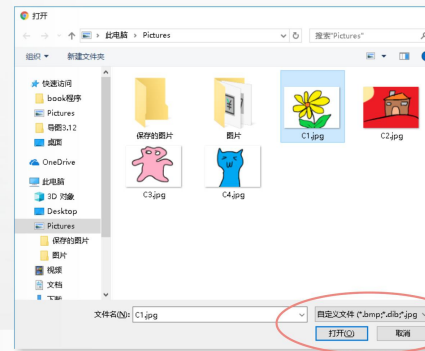
### 画像のアップロード



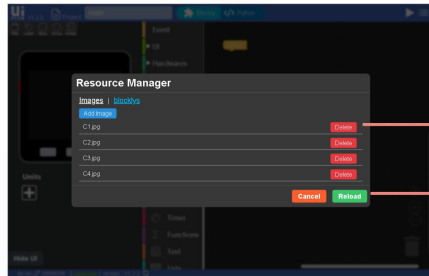
01 M5GOがWi-Fiにつながっていることを確認し、"Resource Manager" (リソース・マネージャ) をクリックします。



02 "Add Image" (画像を追加) をクリックします。



さきほど作成して保存した画像ファイルを選び、"Open" (開く) でアップロードします。



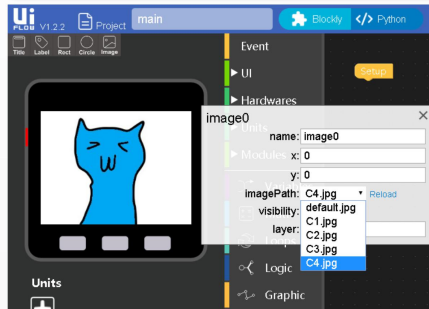
画像ファイルをすべてM5GOにアップロードしたら、このようにリストに現れ、プログラムから使うことができます。

#### Delete (削除)

使わない画像は、Delete (削除) で削除できます。

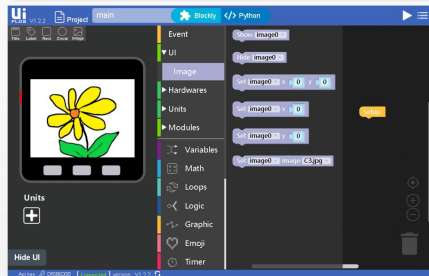
#### Reload (再読み込み)

画像フォルダにある画像ファイルの一覧を更新するには"Reload" (再読み込み) を押します。

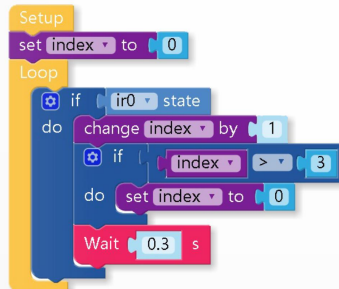


## 1. 絵をプログラムする

UIマネージャで画像を表示する方法は、円や長方形などの図形の表示と同じです。画像アイコンをUIマネージャにあるM5GOの画面のところに持ってきて、クリックしてプロパティを開き、"ImagePath"で、表示させたい画像を選びます。場所も指定できます。

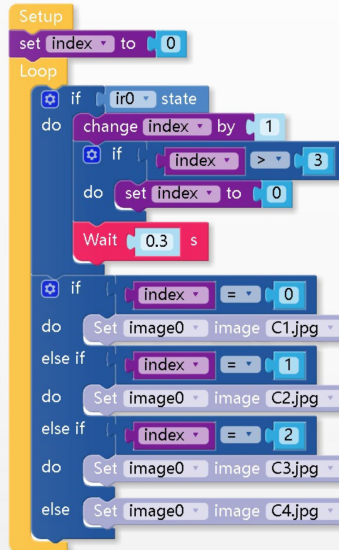


UIマネージャで画像を追加すると、ブロックリストの"UI"の中に、画像を扱うためのブロックが現れます。



## 2. 変数と論理の設定

表示させる画像の番号を指定するindexという変数をつくります。ifブロックの条件に、"IR state"ブロックをはめます。"change by"ブロックを使って、赤外線信号を受信したらindexを1ずつ増やすようにします。もう1つifブロックを追加して、indexが最後の3を超えたら0に戻すようにしておきます。



## 3. 画像を繰り返すプログラム

これで画像の番号を決める部分はできたので、次にそれぞれの画像の番号に画像を割り当てます。ifブロックをもう1つ使って、画像番号indexが0~3のそれぞれで、別々の画像を表示させます。

ちょっと考えてみよう:

私たちが普段使うテレビなどのリモコンにはボタンがたくさんあって、それぞれ別の動作をします。これはどのようにやっているのでしょうか? M5GOのIRリモコンでは、0と1のそれぞれで別のパルスを送っています。これを組み合わせれば、3つ以上のボタンの情報を送ることができます。



## 15 ゲームのデザイナー

ここでは、ボタンを使って画面上の文字や図形を移動させてみましょう。文字や図形の位置を表すための変数や、論理についてより深く学びます。

### チクタク爆弾

チクタク爆弾は、時間が進むととんどん音が高くなり、ゼロになるとLEDが赤になって画面に"Boom!"(ばーん!)という文字を出します。

時間を減らすのと、LEDをスムーズに点滅させるのに、処理の待ち時間をなるべく減らしたいところです。どうやりましょう?

爆弾が爆発するまでの時間を決めて、論理ブロックを使って、時間がタイマーがゼロになったかどうかを、ループの中で確認するようにします。

爆弾の状態を表示するLabelをおきます。



```

01 "time" (時間) という名前の変数
   をつくり、値を50にしておきます。

02 論理ブロックを使って、"time"が"0"より大きいかどうかを
   調べます。0より大きいなら、まだ爆弾が爆発していないので、
   少しずつLED点滅と音の速さを早くしていきます。変数"time"
   の値は1減らして、画面に表示しておきます。

Setup
set time to 50
Loop
if time > 0
do
Set RGB Bar color green
Speaker.beep freq: 1600 duration: 100 ms
Wait time ÷ 100 s
Set RGB Bar color black
Wait time ÷ 100 s
change time by -1
Label label0 show "Bomb running....."
else
Set RGB Bar color red
Label label0 show "Boom!"

03 "time"の値を100で割った時間だけ、
   waitブロックで待ちます。

04 "time"が0になったら、LEDを赤にして、
   "Boom!"と表示します。

Button A pressed
set time to 50
    
```

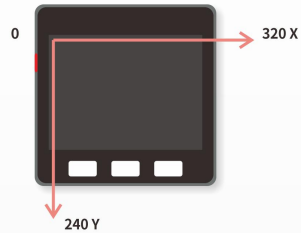
05 Aボタンを押したら"time"を50に戻して、またプログラムの動作が始まるようにします。



## ドッジボール・ゲーム

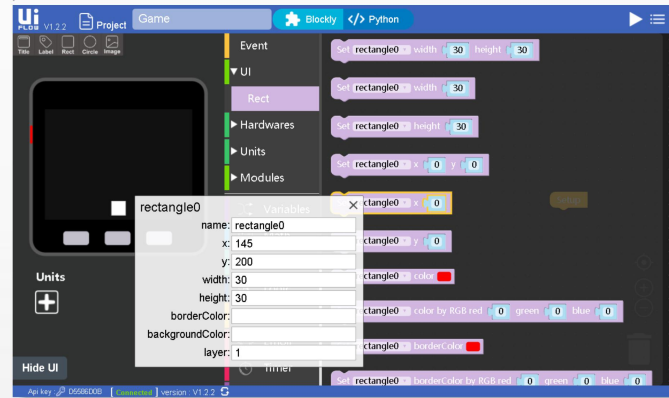
### 1. UIマネージャの「座標」について

M5stackの画面は320×240個の画素(ドット)があって、UIマネージャの画面と対応しています。画面上に置かれたLabelや図形の位置は、「座標」で指定されます。左上を基準として、右方向がx座標、下方向がy座標といえます。Labelや図形は、指定したx, y座標を基準に、画面に表示されます。



### 2. プレーヤをつくる

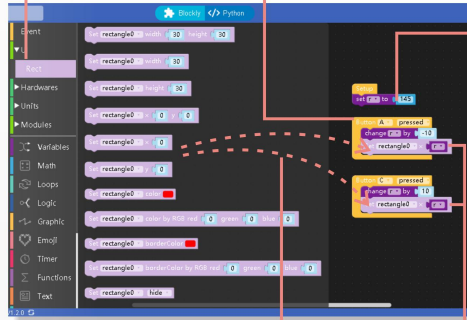
UIマネージャでRect(長方形)を適当な場所に置きます。その後、クリックしてプロパティを開き、xを145、yを200に設定します。



03 UIグループの"Rect"(長方形)タブをクリックします。

02 ボタンのブロックを2個もってきて、"change by"(ボタンの状態が変わったら)にして、ボタンは、1つを"C"、もう1つを"A"にします。それぞれで、変数"r"の値が、-10(左へ10画素)と+10(右へ10画素)変わるようにすることにします。

01 変数"r"をつくり、値を145にします。これをSetupブロックにはめておきます。



04 "set rect x"(RectのX座標を設定)ブロックをもってきて、代入する値のところに変数"r"をはめます。

05 ボタンの"change by"ブロックで、Aボタンを押したら"r"が10減り、cボタンを押したら"r"が10増えるようにしておきます。これでRectのX座標が10減るか増えるかすることで、左右に10画素分移動することになります。

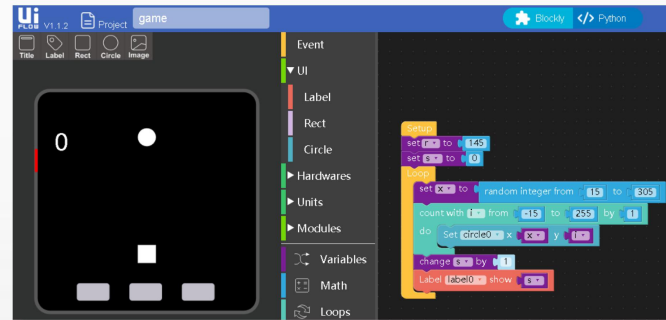


### 3. ランダムに落ちるボール

Circle (円) をUIマネージャの画面においておきます。その円をランダムな場所に現れて、そこから落ちていくボールにしてみましょう。  
 そのために、まず変数"x"をつくって、15~315の範囲のランダムな値を入れます。これは画面内に円が収まる座標の値です。それから、Countブロックで、変数"i"を-15から255まで1ずつ増えるようにします。そのCountブロックで行う内容に、UIのところにある"circle xy"ブロックで、x座標に変数"x"、y座標に変数"i"をはめておきます。

### 4. 得点計算のプログラム

このゲームの得点のための変数"s"を作って、値を0にしておきます。プレーヤがボールをキャッチしたら"s"は1ずつ増えるようにして、その特典は画面にLabelで表示しておきます。続いて、プレーヤとボールの当たり判定も必要です。ボールがあたったらゲームオーバーとします。

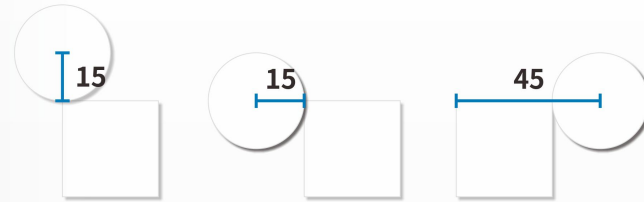


### 5. 当たり判定

ボールをプレーヤの当たり判定はどのようにやりましょう?

Logic (論理) ブロックを使えば、2つの図形がぶつかっているか (重なっているか) を判定できます。

2つの図形 (ボールとプレーヤ) が同じ平面にのっているとして、半径15のボール (円) が、高さ・幅の30のプレーヤ (四角) と重なっているかをチェックすればよいのです。次のような場合が考えられます。



まず1つ目の場合は、"i"が185より大きいかをチェックします。これはボールがプレーヤ (四角) の高さまで落ちてきたときに、プレーヤの四角にあたっているかのチェックです。

次に、xがr-15より大きいか ( $x > r - 15$ ) をチェックします。そうならば、ボールはプレーヤの四角の左に落ちてきているので、あたりにはなりません。

xがr+45より小さいか ( $x < r + 45$ ) ならば、ボールはプレーヤの四角の右側に落ちてしまっており、これもあたりにはなりません。

これらの3つの条件がすべて満たされたら、ボールとプレーヤの四角はあたっていることになります。この条件をプログラムで書いてみましょう。

あたり判定をCountブロックの中でやらなといけないのはなぜ?

あたり判定をプログラムの他のところでもできるのでしょうか?Countブロックのループの中で、ボールは上から下へと落ちていきます。あたり判定はボールが動くたびに行うべきです。そのため、プログラムの他のところであたり判定をすると、見落としがおこってしまいます。

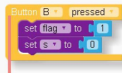
## 6. ゲームを動かしてみよう

このプログラムだと、プログラムの実行が始まるとそのままゲームも始まります。スタートボタンでゲームが始まるようにはしていません。またボールにあたらしたらゲームオーバーですが、次のゲームを始められません。ゲームオーバーになったか、またはゲーム中かの状態をチェックすべきです。これを次によってみましょう。

01 変数"flag"をつくって、Setupのところで値を0にしておきます。



02 変数"flag"の値によって、ゲームオーバー状態か、ゲーム中かで、Repeatで動作を分けます。



03 ボールがあたらしたら、変数"flag"を0にして、ゲームのループを抜けるようにします。

04 Bボタンを押したら変数"flag"を1に設定するようにして、ゲームがもう一度始まるようにします。このとき点数を0にしておきます。

プログラムの開始時には"flag"は0にしているので、ゲームは自動では始まりません。Bボタンを押すと"flag"が1になってゲームが始まります。つまりゲームをリセットしたいときには"flag"を1にすればよいわけです。

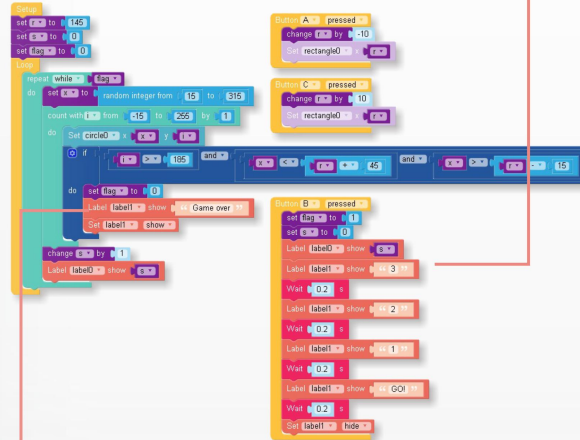
## 7. プログラムの最適化

ここまでで、プレイヤーの四角はボタンで左右に移動でき、またボールはランダムな位置から落ちてきて、ぶつかったらゲームオーバーになるようになりました。でもゲームオーバーか、ゲームを始められる状態かを示す表示がありません。またスタートボタン(Bボタン)を押すとすぐにゲームがはじまるので、ちょっと心の準備はできません。これらの点を修正してみましょう。

ちょっと考えてみよう:ゲームが始まるまでの状態を、どのように表示したらいいでしょう?

たぶん普通はカウントダウンでしょう。最初に"Bボタンを押してスタート"を表示して、Bボタンを押したらカウントダウンして0になったらゲーム開始としましょう。またゲームオーバーになったらそのメッセージを表示します。

01 "Bボタンを押したら"ブロックのところで、Labelブロックとwaitブロックでカウントダウンを表示します。カウントダウンが終わったら、ゲーム開始なので、カウントの表示を消します。



02 あたり判定とゲーム状態のチェックのところで、ゲームオーバーの表示をさせます。そのあとで、さきほど消していたLabelを再び表示されるよに"label show"を使います。

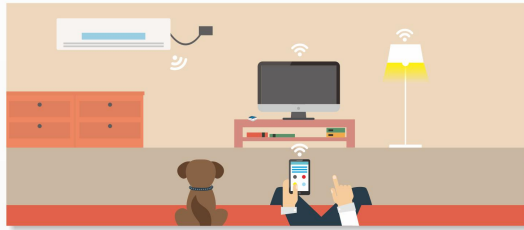
## 16 リモート制御

ここでは、Remote (リモート) ブロックを使って、離れたところにある照明を制御したり情報を表示したりします。"Remote"ブロックでON/OFFスイッチやスライダーを使う方法を学びます。

**注意:**"Remote"ブロックを使うには、M5GOをインターネットにつなぐ必要があります。p.105の手順でWi-Fiについてください。

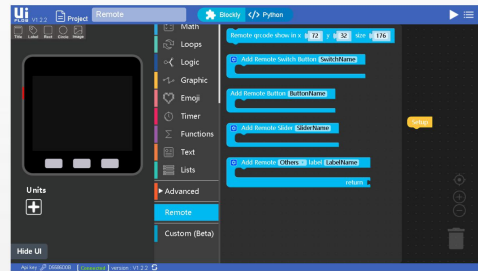
リモートコントロールとは何?なぜ学ぶのか?何に使えるのか?

前に赤外線リモコンを使ってみました。"リモート・コントロール"は、離れたところにあるものを、ケーブルなしで制御するもので、赤外線やWi-Fiなどが使われます。リモート制御で照明やテレビや電化製品を、ソファーに座ったままでON/OFFできます。スイッチをON/OFFする以外にも、エアコンの設定温度の確認などもできます。とても便利ですね。



### 1. "Remote"

ブロックのグループの下の方にある"Remote"グループを使えば、スマホにリモート制御用のパーツを表示させることができます。次のページで、リモコンをつくってみましょう。



Setup  
Remote qrcode show in x: 72 y: 32 size: 176

Add Remote Button ON  
Set RGB Bar color: [Red]

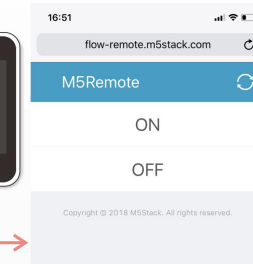
Add Remote Button OFF  
Set RGB Bar color: [Black]

QRコード  
"Remote"では、"Remote QR code show" (Remote QRコードの表示) を使います。これを使うとリモート制御のためのWebページへのアドレスが表示されます。またはUIFlowの画面の右の方に表示されているQRコードアイコンをクリックしてWebページのアドレスを表示させて使うこともできます。



### 2. Remoteボタン

"Remote Button"ブロックは、スマホやPCの画面上にボタンを表示して、ON/OFFをできるようにします。スマホのそのボタンを押すと、M5GO本体のボタンを押したときと同じように、何か動作をさせられます。



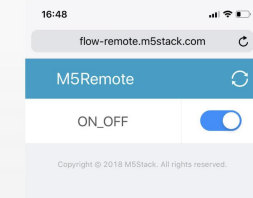
Setup  
Remote qrcode show in x: 72 y: 32 size: 176

input name: x  
inputs  
input name: x

Add Remote Switch Button ON OFF with: x  
if [x] == 1  
do Set RGB Bar color: [Green]  
else Set RGB Bar color: [Black]

### 3. Remoteスイッチ

"Remote Switch"ブロックは、スマホ上にスイッチを表示させて、ON/OFFを切り替えられるようにします。スイッチの状態を保存しておく変数"x"を作って、使う必要があります。スイッチがON状態ならば変数の値は1、OFFスイッチならば0になります。これを使えばライトやモータなどをON/OFFするのに使えます。



```

Setup
Remote qrcode show in x: 72 y: 32 size: 176
    
```

input name: X

inputs

input name: X

Add Remote Slider Bright

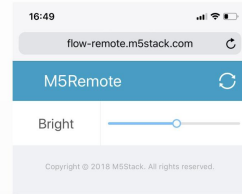
Set RGB Bar color

Set RGB map brightness

from low	0
from high	100
to low	0
to high	255

#### 4. Remoteスライダー

"Remote"スライダーは、左右に動かせるスライダー(つまみ)で、変数の値を設定できます。"Remote"スライダーを作る前に、その値を入れておく変数をつくっておきます。スライダーを動かすと、変数の値が0から100の範囲で変わります。



```

Setup
Remote qrcode show in x: 72 y: 32 size: 176
    
```

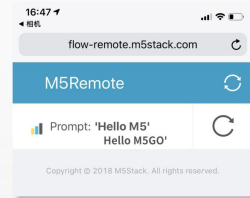
Add Remote Others label LabelName

- Others
- Temperature
- Humidity
- Pressure
- PIR Status
- Angle
- Weight

return: "Hello M5GO"

#### 5. "Remote"ラベル

"Remote"ラベルは、スマホ上にセンサのデータやメッセージなどを表示させるのに使います。たとえばENVセンサの温度の値をスマホに表示させることができます。



#### 概要

"Remote"ブロックを使って、LEDをバーON/OFFをしたり、明るさや色を変えたり、ENVセンサの値を表示させたりしてみましょう。

```

Setup
Remote qrcode show in x: 72 y: 32 size: 176

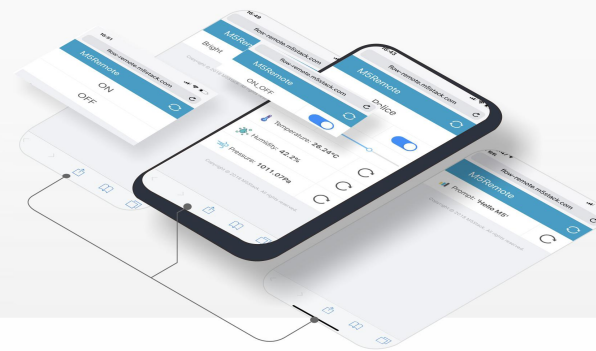
Add Remote Button Police
repeat 10 times
do
Set left: side RGB Bar color
Set right: side RGB Bar color
Wait 0.2 s
Set left: side RGB Bar color
Set right: side RGB Bar color
Wait 0.2 s
Set RGB Bar color

Add Remote Temperature label Temperature
return: Get env0: Temperature

Add Remote Humidity label Humidity
return: Get env0: Humidity

Add Remote Switch Button ON/OFF with: X
if X
do
Set RGB Bar color
else
Set RGB Bar color

Add Remote Slider Bright with: X
Set RGB map brightness
    
```



APPENDIX:  
FIRMWARE UPGRADE  
APPENDIX  
M5GO Wi-Fi CONFIGURATION  
LEARN MORE:  
FIRMWARE UPGRADE

# 「付録」

アイデアを素早く形に

M5GO

M5GO



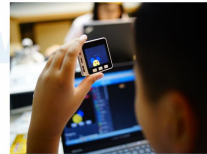
ファームウェアの更新



Wi-Fiの設定



付録



より深く学ぶ

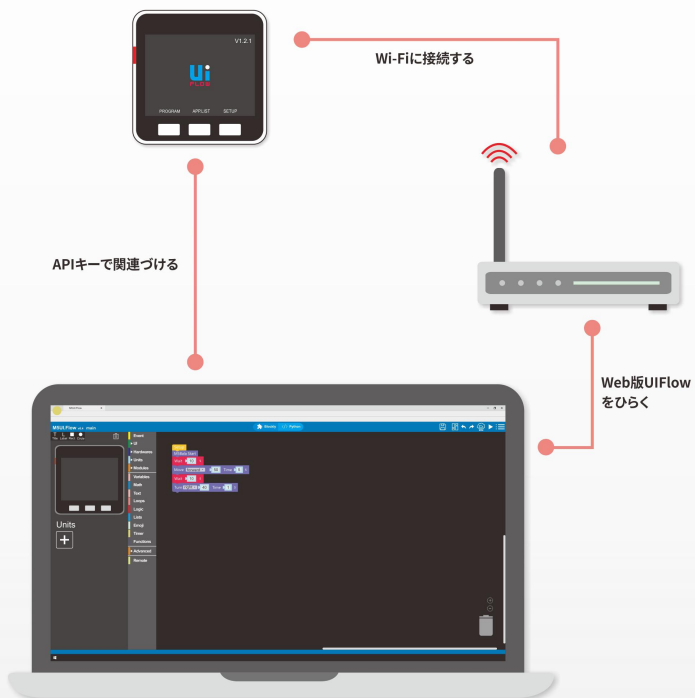
M5GO

M5GO

M5GO  
M5GO

## Wi-Fiの設定

USB経由でプログラムを実行させる以外に、M5GOはインターネット経由でプログラムを実行させることもできます。Web版のUIFlowでは、無線通信やIoT関連の機能も使えます。Web版とオフライン版の機能はほとんど同じですが、Web版ではM5GOをWi-Fiにつなぎ、"APIキー"で関連付ける必要があります。



## Wi-Fiの設定

01 横の赤ボタンを押してM5GOの電源を入れます。



02 メニューがでたところで、本体の右ボタンを押してWi-Fi設定に入ります。

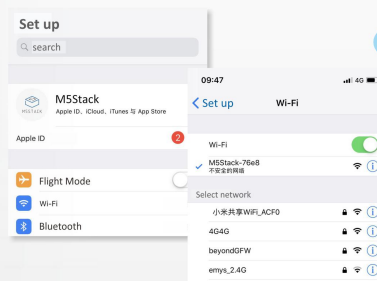


03 左・右のボタンを使ってメニュー項目の"Change Wi-Fi Connect" (Wi-Fi接続先の変更) を選びます。

04 "Change Wi-Fi Connect"うを選んだら、M5GOのWi-Fi名が表示されます(この例では"M5Stack-76e8")



05 パソコンやスマホのWi-Fi設定から、このM5GOのWi-Fiを見つけて接続します。



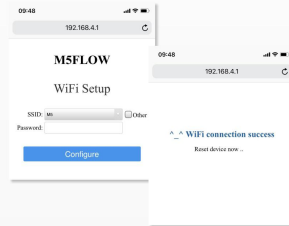


- 06 M5GOのWi-Fiにつながると、画面にQRコードとIPアドレスが表示されます。



IP:192.168.4.1

- 07 Webブラウザを開いて、アドレスに"192.168.4.1"と入力するか、スマホのQRコードリーダーでそのQRコードを読み取ると、Wi-Fi設定のページが表示されます。



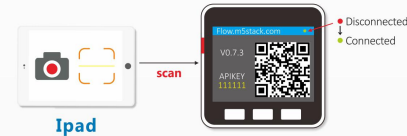
- 08 接続したいWi-Fiを選んで、そのWi-Fiパスワードを入力します。それからConnectボタンを押します。M5GOがWi-Fiに接続できたら、自動的にリセットします。

## APIキーの入力

- 01 M5GOの電源を入れるかリセットしたあと、メインメニューが表示されるので、Programの下ボタンを押して、UIFlowからプログラムを実行できる状態にします。



- 02 プログラミングモードに入ったら、画面の右上に小さな丸が表示されます。それが赤から緑に変わったら、無事準備完了です。



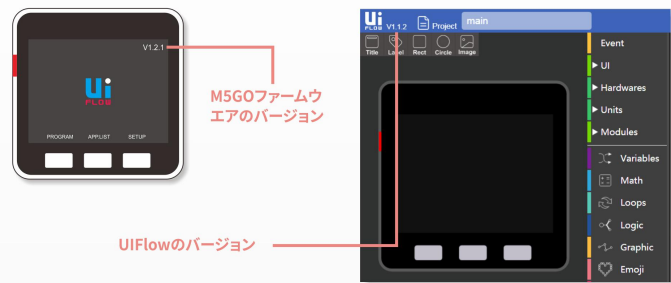
- 03 Webブラウザで、"flow.m5stack.com"を開くか、表示されているQRコードを読み取りると、Web版UIFlowが開きます。さいしょにAPIキーを入力するダイアログが出てくるはずですが、出てこない場合は、UIFlowの右上にある歯車マークをクリックします。



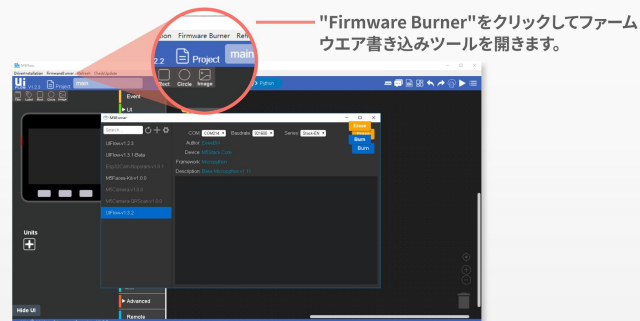
- 04 M5GOに表示されているAPIキー（6桁のアルファベットと数字）を入力します。これでプログラムの実行ができるようになりました。

## ファームウェアの更新

UIFlowの最新の機能を使うためには、定期的にM5GOのファームウェア (内部のプログラム) を更新しましょう。M5GOのファームウェアは、UIFlowのバージョンと同じバージョンのものを使います。M5GO起動時に、画面の右上にバージョン番号が表示されます。UIFlowのバージョンは、UIFlowの画面の左上に表示されています。UIFlowのバージョンのほうがM5GOよりも新しいならば、UIFlowのバージョンが更新されたので、M5GOのファームウェアも更新すべきということになります。UIFlow (オフライン版) にはファームウェア書き込みのツールもついています。



ファームウェアを更新するには、M5GOをパソコンにUSBケーブルで接続します。接続したら、画面の上の方にある"Firmware Burner"をクリックし、ファームウェア書き込みツールを開きます。



### 1. COMポートの番号

M5GOをパソコンにつなぐと、COMポートの番号が割り当てられます。ファームウェア書き込みのときには、M5GOに割り当てられたCOMポート番号をドロップダウンリストから選びます。Windowsパソコンでは、COMポート番号は、COMのあとに数字がついた名前 (COM32など) です。Macでは、"/dev/tty.SLAB\_USBtoUART"のような名前になります。

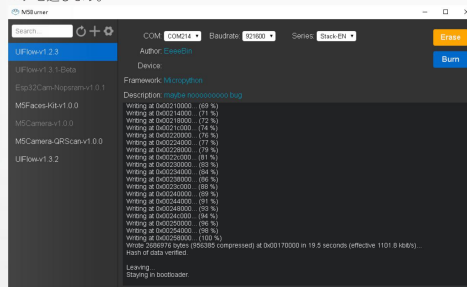


### 2. Baud (通信速度)

Baud rate (通信速度) は、ファームウェア書き込みをするときのM5GOとパソコンの通信速度です。通常は自動的に115200か921600になります。

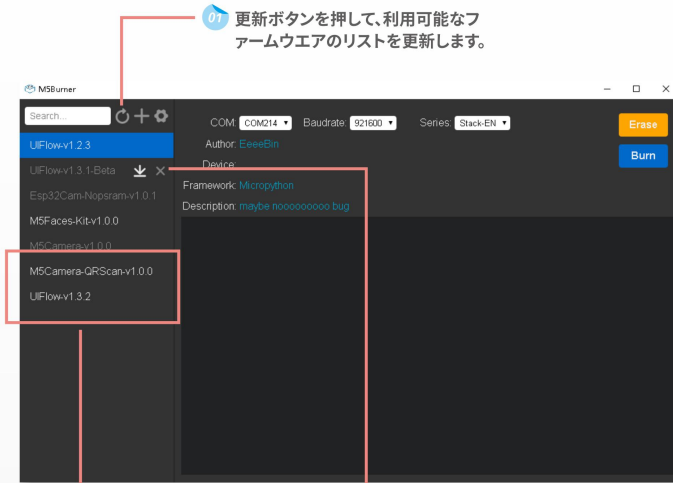
### 3. ファームウェア

左のリストから、書き込みをするバージョンのファームウェアを選びます。M5Stackデバイスの種類にあわせてファームウェアを選びます (M5GOはM5Stackを選ぶ)。基本的に、UIFlowと同じバージョンのファームウェアを選びます。



#### 4. ファームウェアのバージョン

ファームウェアのバージョンを選びます。



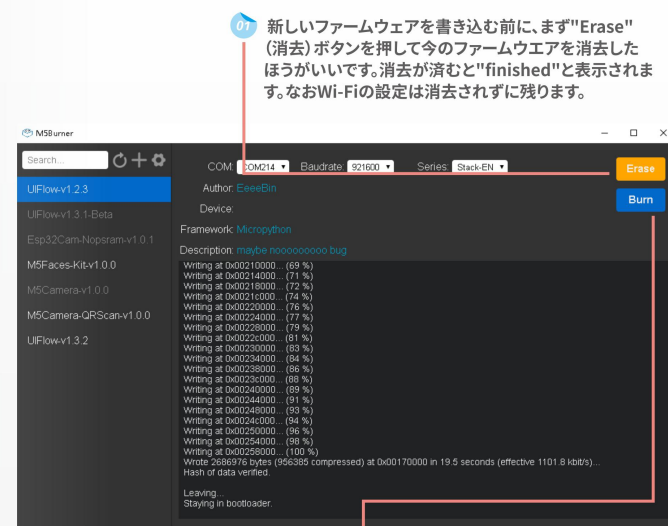
01 更新ボタンを押して、利用可能なファームウェアのリストを更新します。

02 色が白っぽいものはダウンロード済みで書き込みができます。

03 グレーのものはまだダウンロードされていないので、右側のダウンロードのアイコンをクリックしてダウンロードします。

#### 5. ファームウェアの書き込み

書き込みをしたいバージョンのファームウェアを選んで書き込みをします。



01 新しいファームウェアを書き込む前に、まず"Erase" (消去) ボタンを押して今のファームウェアを消去したほうがいいです。消去が済むと"finished"と表示されます。なおWi-Fiの設定は消去されずに残ります。

02 "Burn" (書き込み) ボタンを押して書き込みを開始します。最後に"Finished"が表示されたら、書き込み完了です。

# M5GO



M5stackについて、より詳しくは、<https://m5stack.com> をご覧ください。

## M5GO

この本で、M5GOをUIFlowでプログラミングする基本を知ってもらえ、また自分のプログラムをプログラミングする楽しさに気づいてもらえることを祈っています。これはまだ始まりにすぎません。もっと面白いことを学んでいてください。

M5GOで使えるUnitには、もっといろいろなものがあります。ぜひこちらから購入して使ってみてください。(日本ではスイッチサイエンス [www.switch-science.com](http://www.switch-science.com) から購入できます)



facebook



twitter



shopify

