

プログラミングして 反応する ミニロボットを作ろう！



手足はないけど
顔があるよ！

小型コンピュータ：M5 ATOMS3

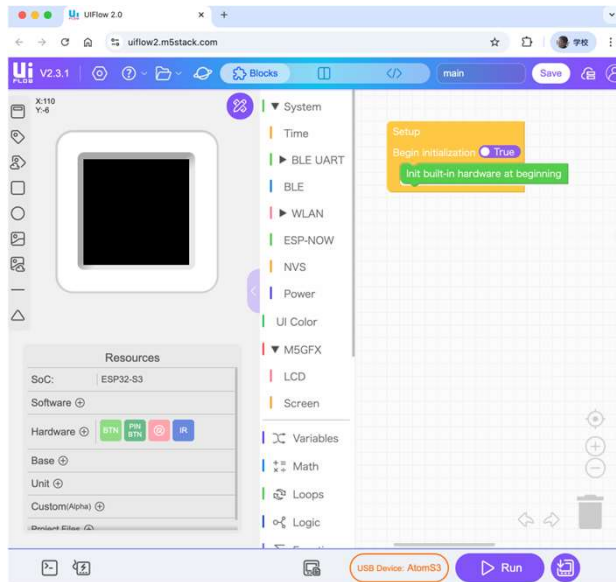


- 一昔前のスーパーコンピュータ並みの性能
- WiFi機能もついている
- 小型ディスプレイもついている
- 3000円くらい
- 中国深センのM5Stack社の製品

<https://www.switch-science.com/products/8670>

金沢のマルツでも売ってるよ！

ATOMS3のプログラミングのしくみ



Webブラウザ上で
プログラミング

WiFi



WiFi

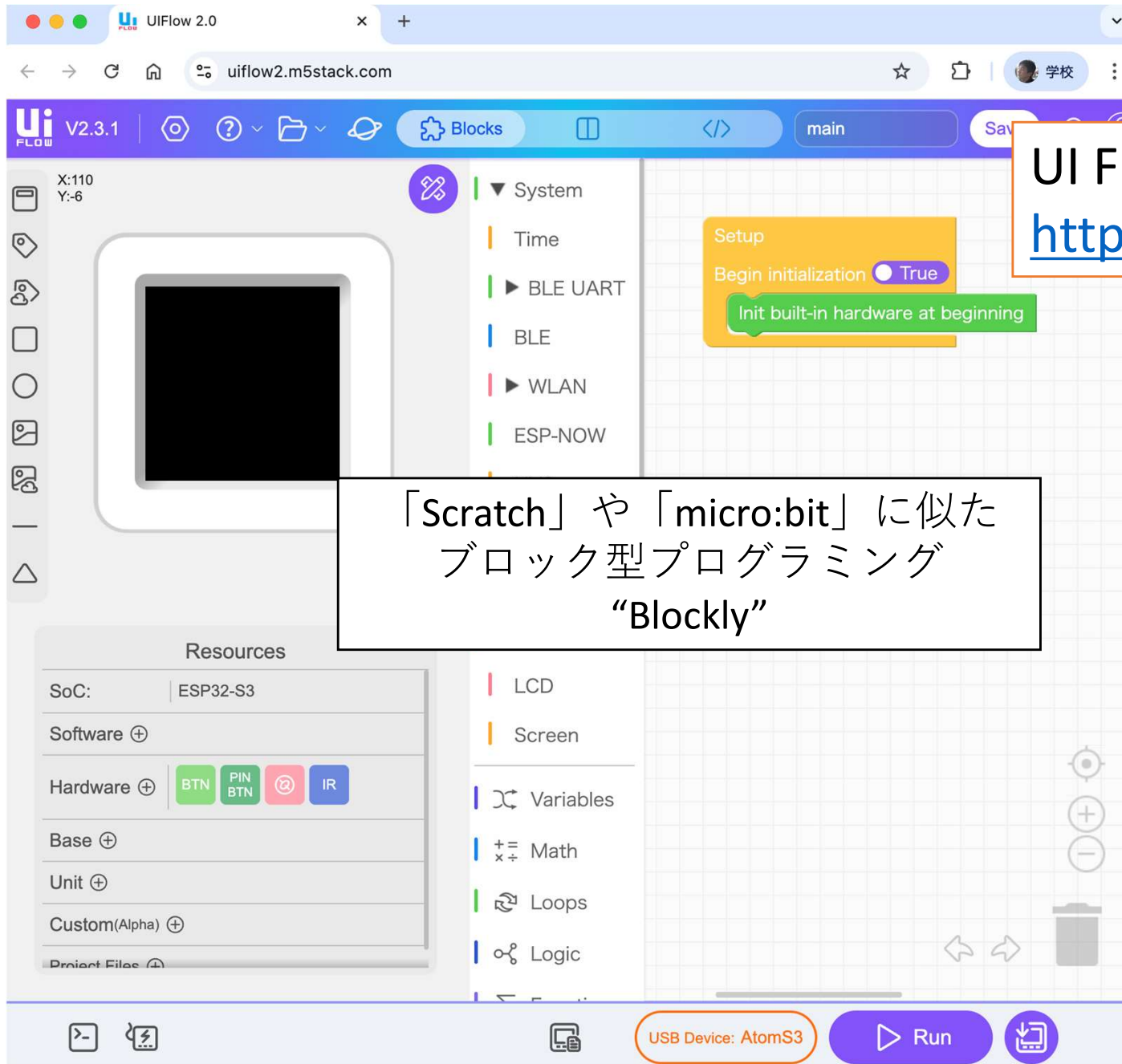


USBケーブルで直接
実行させることもできる

今回はこの方法を使います

実行！

(準備) UIFlowでプログラミング



UI Flow

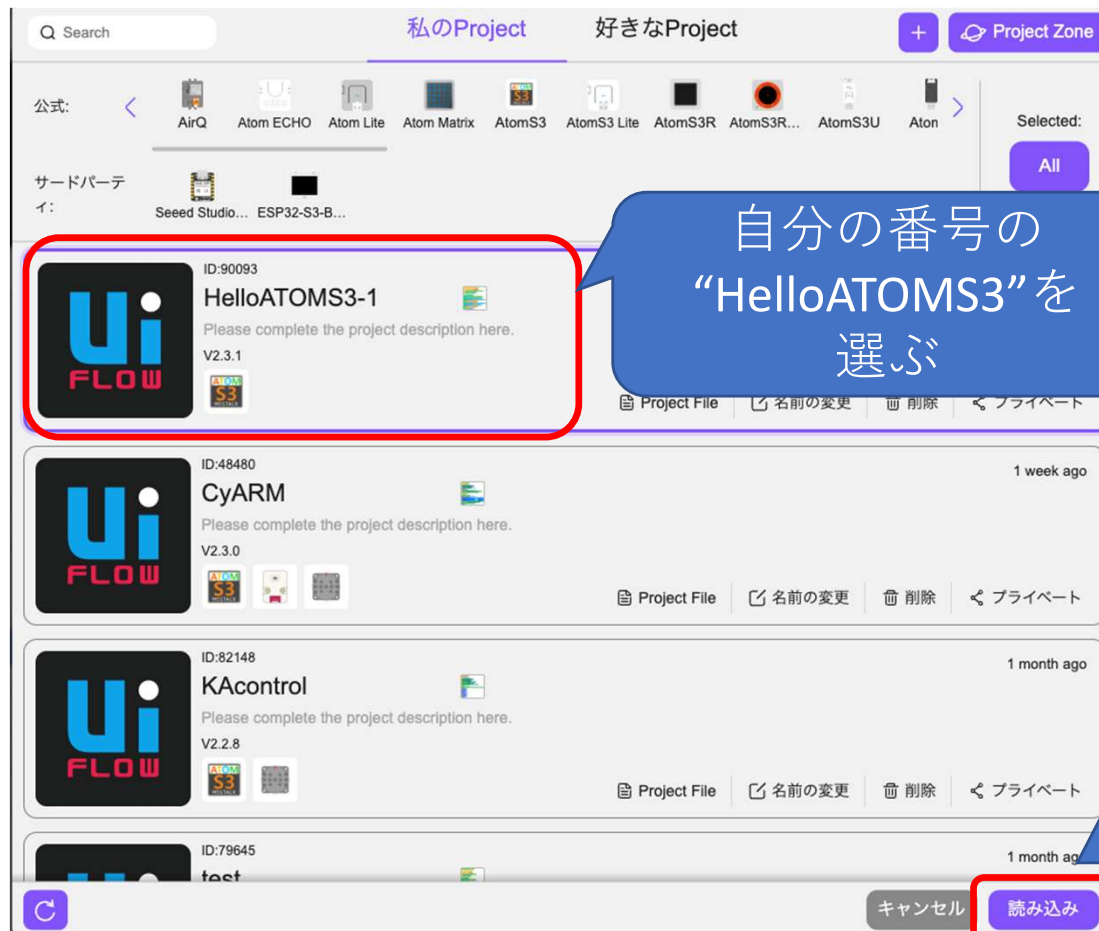
<https://flow.m5stack.com/>

「Scratch」や「micro:bit」に似た
ブロック型プログラミング
“Blockly”

(準備) UIFlowを開く

UI Flow2

<https://uiflow2.m5stack.com/>



(準備) UIFlowを開く

<https://uiflow2.m5stack.com/>



名前は適当に
(自分の名前など)

AtomS3を選ぶ

これら出てきたら、
右(2.0)を選んで
“Confirm (確認)”

Confirmで次へ

Customの「+」から
HelloATOMS3.m5b2を選んで
読み込む

「Custom」内に「HelloATOMS3」が出てくる
→一番上の”init”ブロックをSetupにはめる

(準備) UIFlowの画面が開く

UIFlow V2.3.1

Blocks

HelloATOMS3-1 保存

▼ M5GFX

- LCD
- 画面

変数

数学

ループ

論理

関数

テキスト

リスト

Map

{...} JSON

[...] Bytearray

[...] Bytes

▼ ハードウェア

- ボタン
- ピンボタン
- IMU
- IR

リソース

SoC: ESP32-S3

ソフトウェア ⊕

ハードウェア ⊕

Base ⊕

Unit ⊕

Custom(Alpha) ⊕ H

Project File ⊕ HelloATOMS3.py

Setup

Begin initialization ☒ True

内蔵ハードウェアの初期化

helloatoms3_0 init

Loop

ループ中でボタンやタッチパネルなどの状態を更新

☒ Update

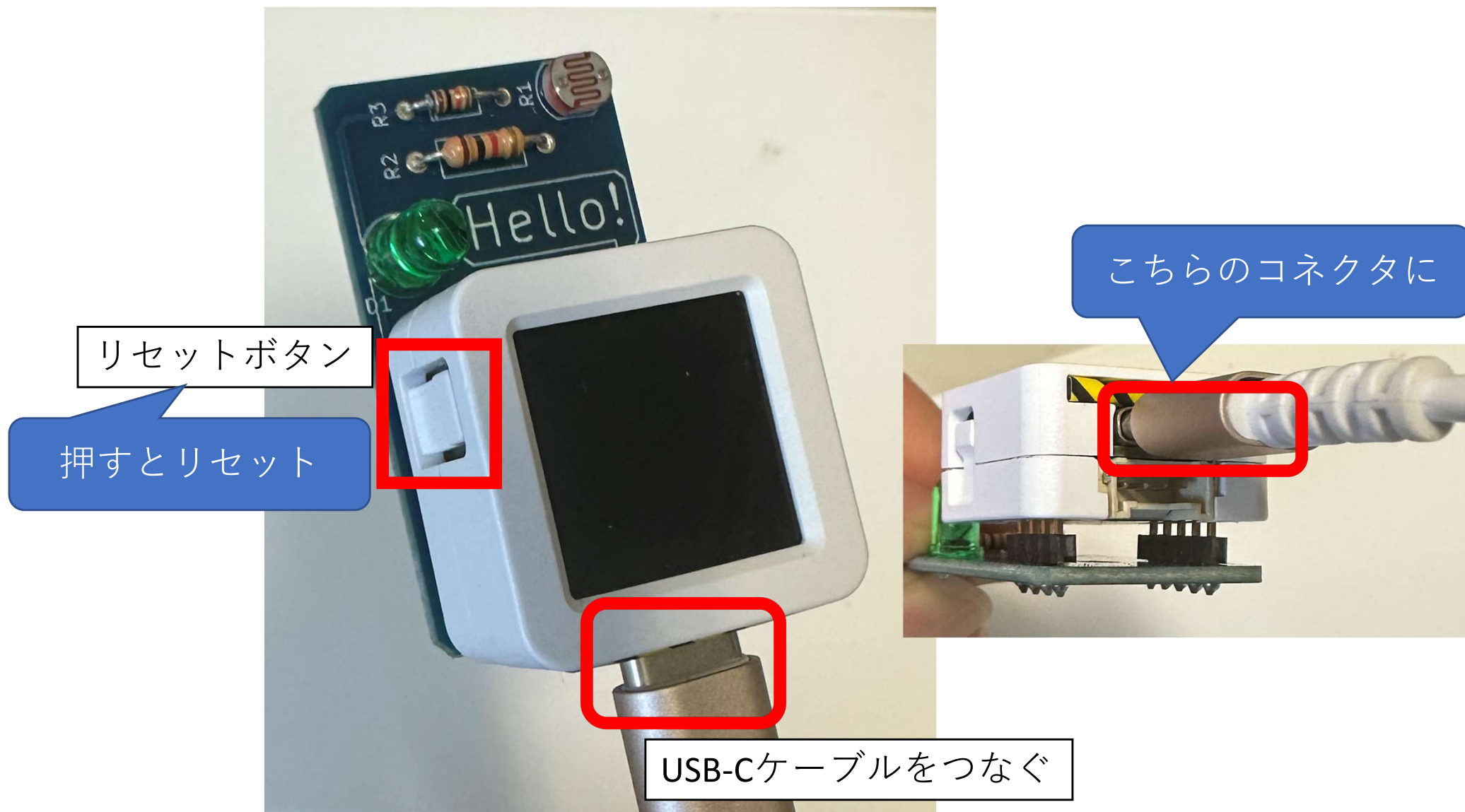
ここで
ブロックをはめて
プログラミング

USBデバイス: AtomS3

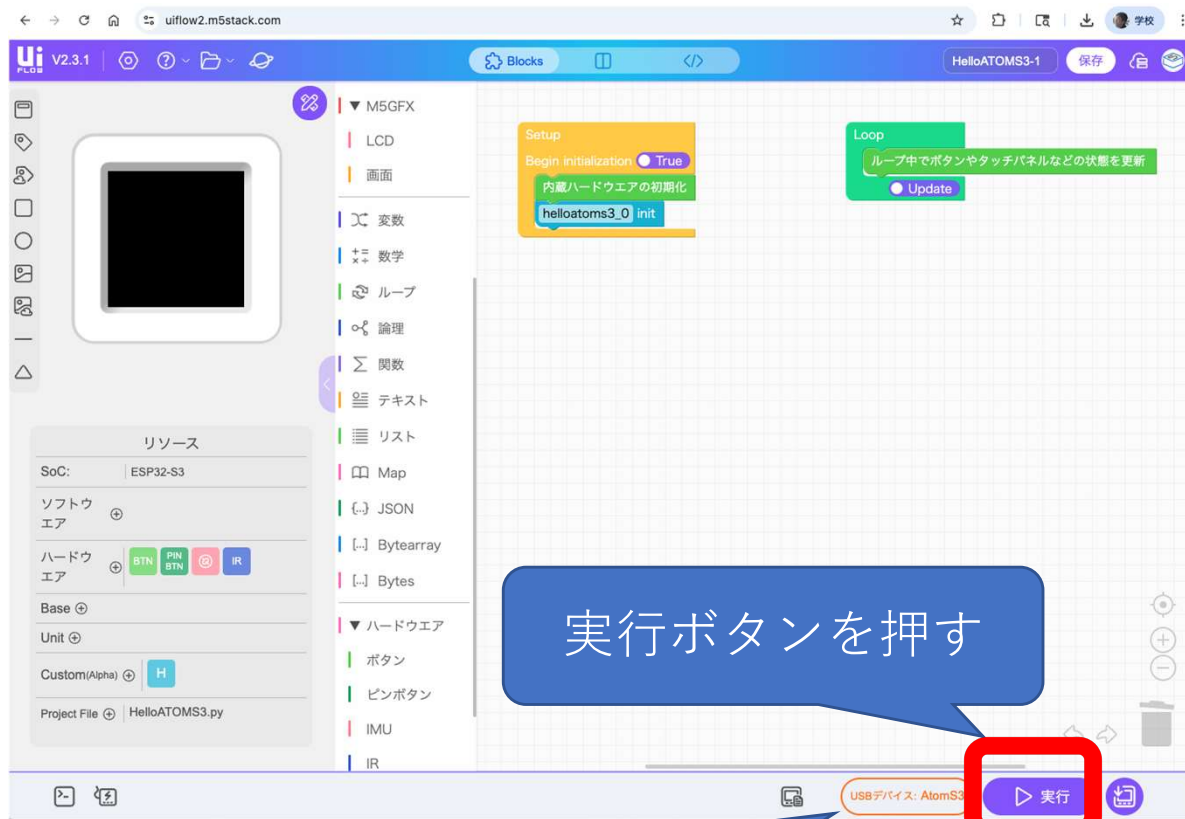
実行

(準備) ATOMS3の電源とリセット

- USB-CケーブルでPCに接続→電源ON



まずは実行してみよう



実行ボタンを押す

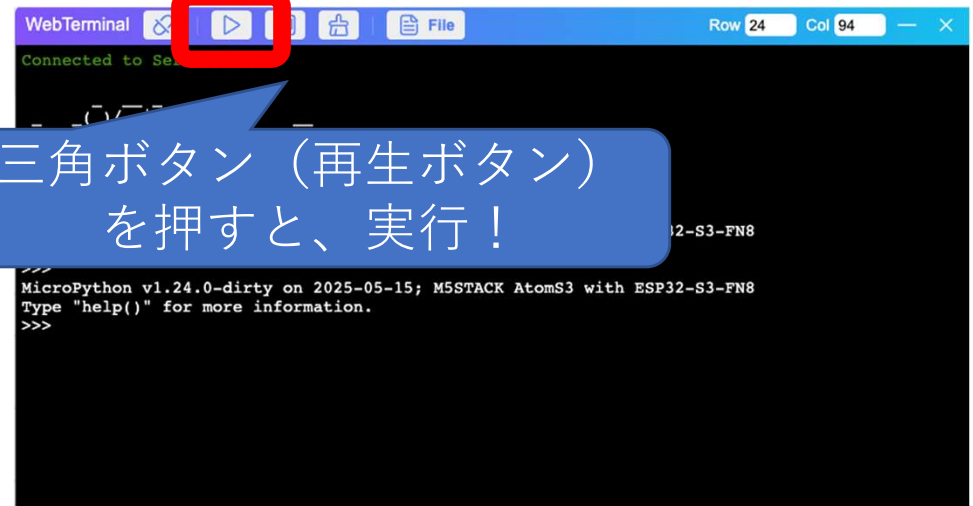
USBデバイス: ATOMS3

ちょっとめんどろだけど
毎回使うので、慣れよう



接続ボタンを押す

三角ボタン（再生ボタン）
を押すと、実行！



UIFlowでのプログラミング

画面に表示する文字などはここで設定する

“Setup”は最初に1回だけ実行

“Loop”は動作中、くりかえして実行

ここにプログラミングの部品がある

UIFlow V2.3.1 | Blocks | HelloATOMS3-1 | 保存

▼ M5GFX

- LCD
- 画面
- 変数
- 数学
- ループ
- 論理
- 関数
- テキスト
- リスト
- Map
- JSON
- Bytearray
- Bytes
- ▼ ハードウェア
- ボタン
- ピンボタン
- IMU
- IR

Setup

Begin initialization ☒ True

内蔵ハードウェアの初期化

helloatoms3_0 init

Loop

ループ中でボタンやタッチパネルなどの状態を更新

☒ Update

USBデバイス: AtomS3 | 実行

画面に文字を出してみよう

“Label”を画面内の上のはしあたりにドラッグ

“M5GFX”内の“ラベル”に、Labelに対するブロックが現れる

この時点では、まだATOMS3に転送・実行していないので、ATOMS3の画面には表示されていない

“設定”ブロックを、“Setup”のここに「はめる」（「はまる」と、グレーから色が変わる）

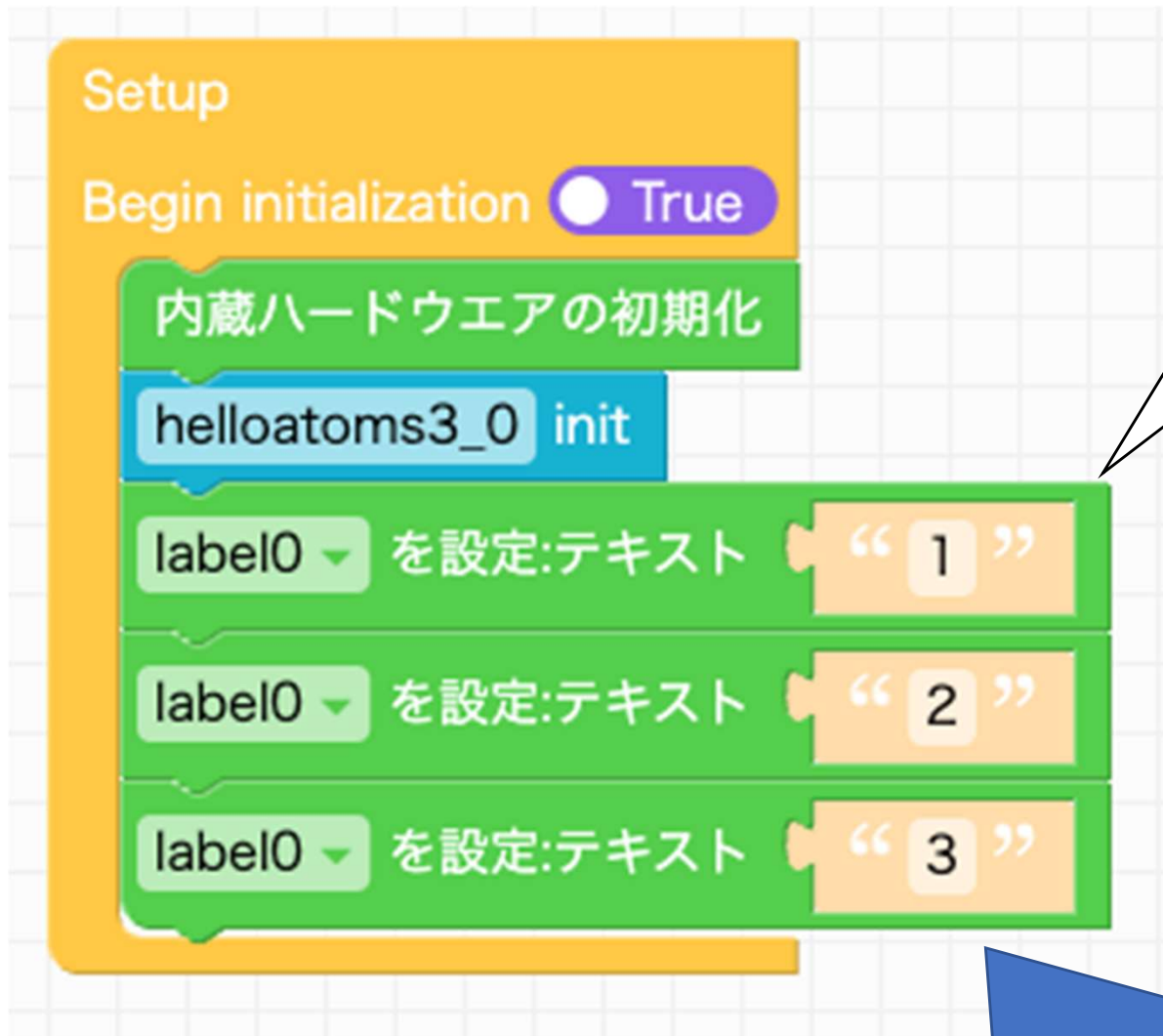
※“Setup”は、実行開始後の最初に実行される処理を「はめる」場所

Labelの文字の色やフォントを変える

置いたLabel（テキスト）をダブルクリック
→Labelの文字の色やフォントなどを変更できる

The screenshot shows the 'UI Editor' interface. On the left is a toolbar with icons for Title, Label, Label+, Rectangle, Circle, Image, Image+, Line, and Triangle. The main canvas displays a black square with the text 'label0' in white. On the right, the 'UI Components' panel lists 'screen' and 'label0'. The 'label0' component is selected, and its properties are shown: Name: label0, X: 34, Y: 0, Color: (white), Background Color: (black), Text: label0, Font: DejaVuSans 18. A font selection dropdown is open, showing a list of fonts: DejaVuSans 9, DejaVuSans 12, DejaVuSans 18, DejaVuSans 24, DejaVuSans 40, DejaVuSans 56, DejaVuSans 72, Efont CN24, **Efont JA24** (selected), and Efont KR24. A blue callout bubble at the bottom right contains the text: フォント(Font)で “Efont JA24”を選べば、日本語表示もOK (それ以外は英数字のみ).

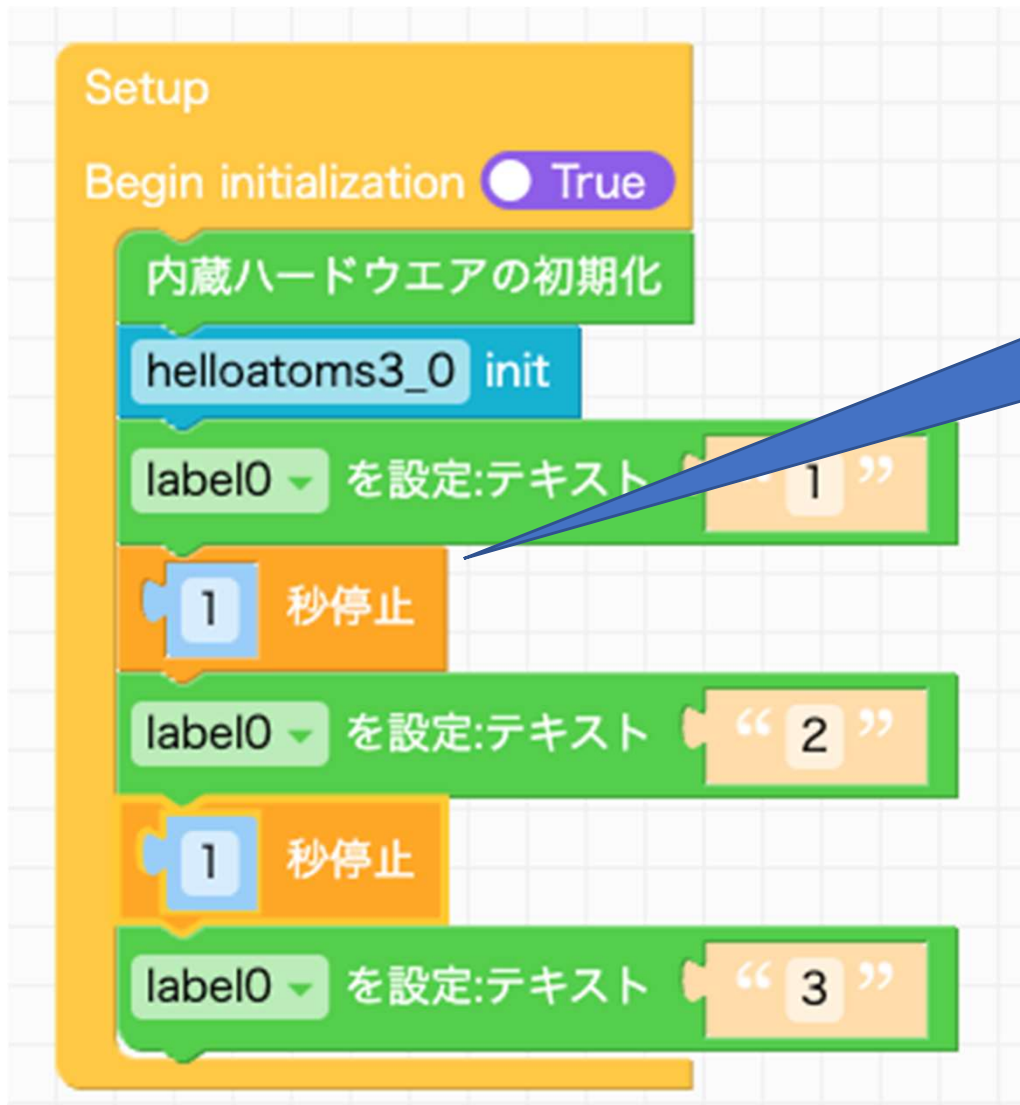
Labelの文字を順に変えてみよう



小技：
ダブルクリックで
ブロックをコピーできます

このプログラムを実行すると、どうなるだろう？
実際の動作はどうでしたか？
予想と違った場合、その理由はなんだろう？

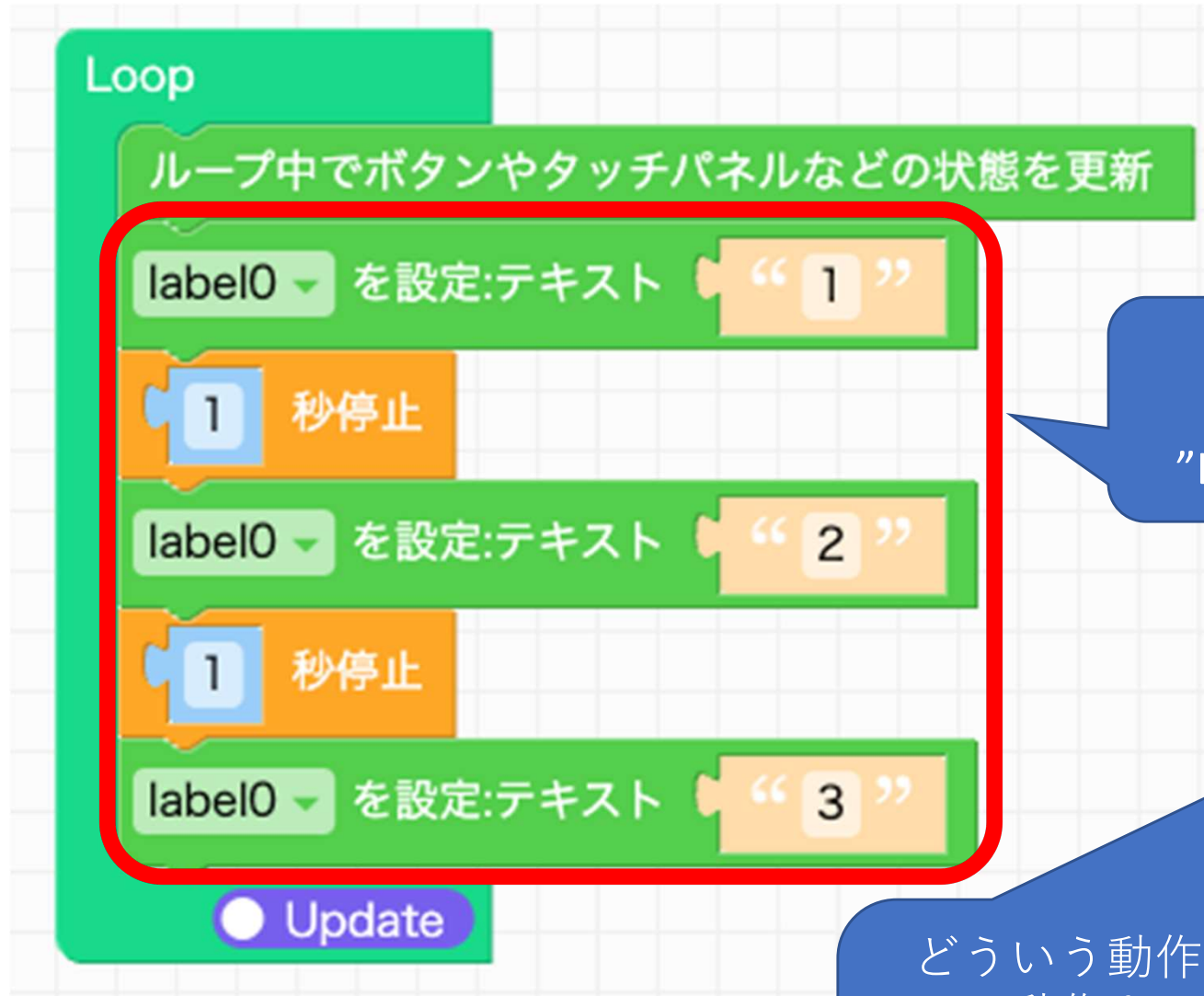
動作タイミングの制御：停止（Wait）



「システム」→「時間」の
"停止"で、
指定時間だけ待つ

指定する時間を変えて、
動作を確認してみよう

くりかえして動作させてみよう



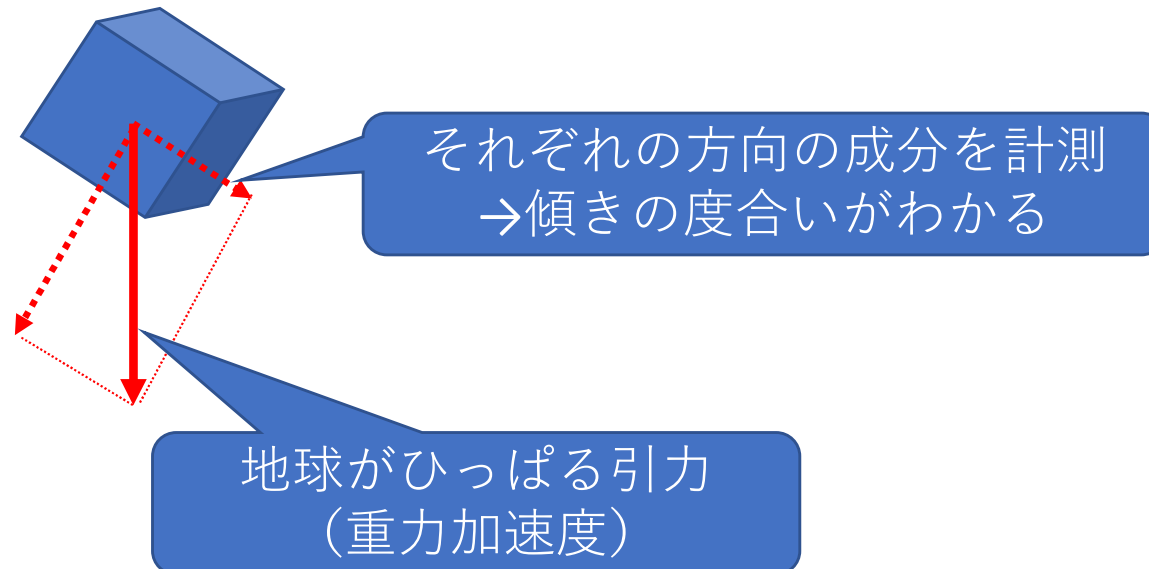
この部分を
まとめて
"Loop"へ移動

どういう動作をするか、予想して、
動作させて確認してみよう

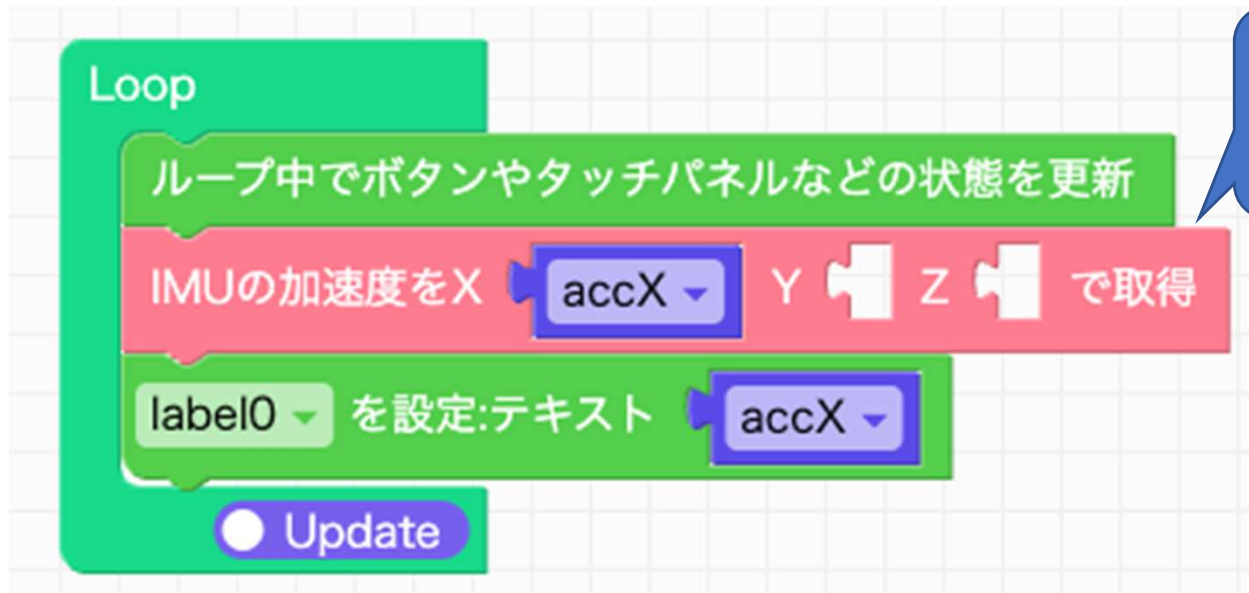
※最後の"3"は表示される？
表示させるにはどうすればいい？

センサを使ってみよう

- センサ = コンピュータの外の世界の情報をコンピュータにとりこむための装置
(物理量を数値に変換する)
 - 温度センサ → 温度をコンピュータで扱える数値に
 - 明るさセンサ → 明るさを数値に
 - 加速度センサ → 「加速度」という量を数値に
(傾きの計測などに使います)

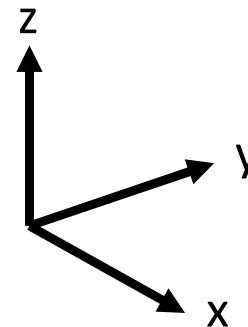


加速度センサを使ってみよう



下の方にある
「ハードウェア」→「IMU」
の中にある

ATOMS3を傾けると、
数字がどう変わるだろう？



センサの値によって動作を変えよう

The image shows a Scratch script for a loop that updates button/touch panel states and changes face expressions based on accelerometer data. The script is as follows:

```
Loop
  ループ中でボタンやタッチパネルなどの状態を更新
  accX Y Z で取得
  テキスト accX
  もし accX > 0.8
  do
    SleepyFace helloatoms3_0
  そうでなければ
    NormalFace helloatoms3_0
  Update
```

Annotations and Explanations:

- 「論理」から持ってくる**: Points to the 'もし' (if) block.
- 「数学」から持ってきて、数値を変える**: Points to the 'accX' variable block in the 'もし' condition.
- 大小比較は、「論理」のこのブロックを使う ("=" → ">" 変更)**: Points to the '>' operator in the 'もし' condition.
- 寝かせると (x軸加速度が0.8以上) 表情を「お眠」にする Sleepy=お眠 Normal=ふだんの顔**: Explains the logic of the 'do' and 'そうでなければ' blocks.

Diagram illustrating the change in the comparison operator:

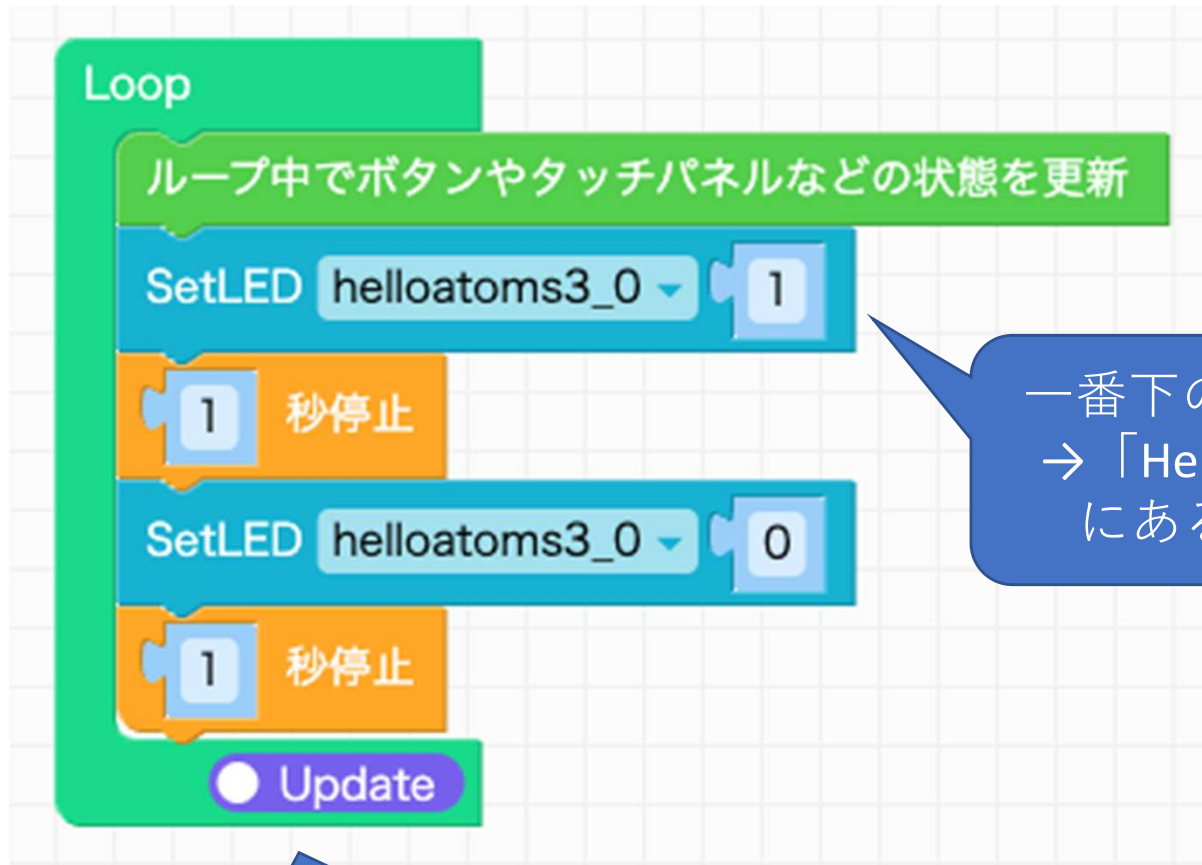
Before:

After:

ここからは後半戦

“Hello”ボードをはんだ付けで作ってから
続きを進めましょう

LEDをつけてみよう



一番下の「Custom」
→「HelloATOMS3」
にあるブロック

LEDを点滅させてみよう
(「Lチカ」といいます)

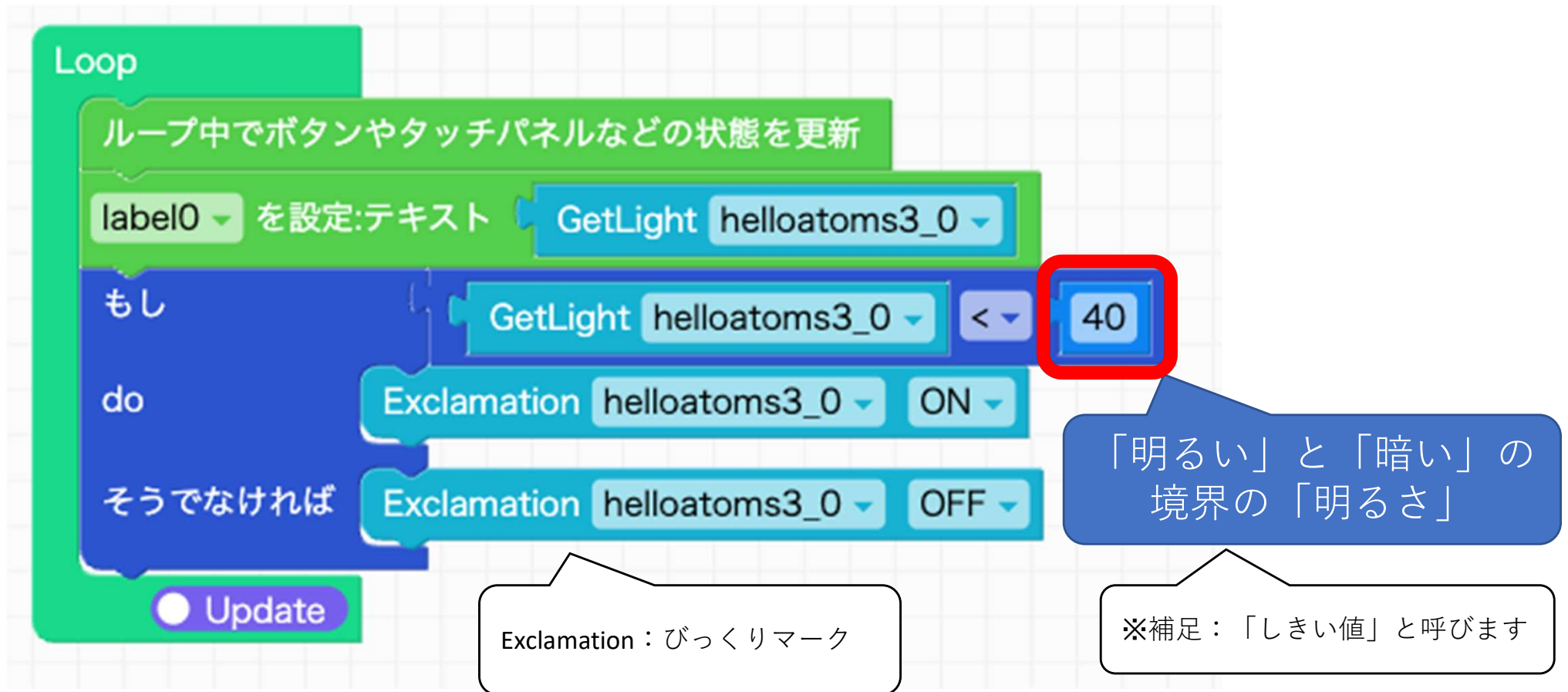
明るさセンサを読む



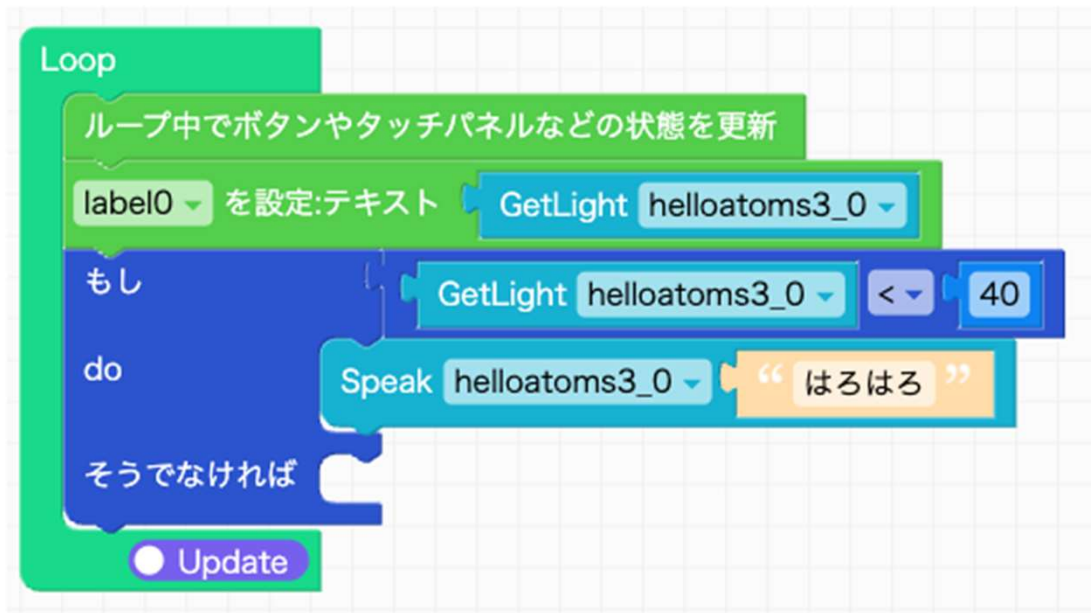
表示される値をメモしておこう（だいたいOK）

- (1) 部屋の電灯がついているとき : _____
- (2) 部屋の電灯を消したとき : _____
- (3) 光センサを手で隠したとき : _____
- (4) スマホ等のライトで照らしたとき : _____

暗くすると表情を変えよう



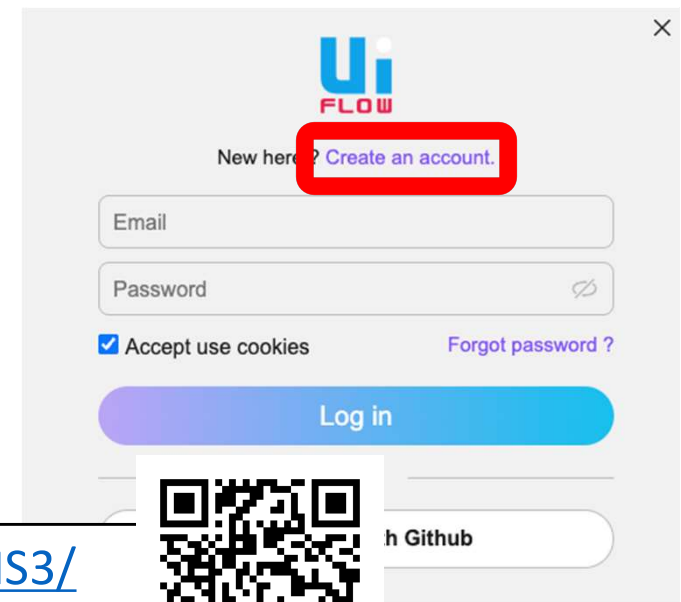
自由に遊んでみよう！



(おまけ) 続きはご自宅で . . .

- ちょっと初期設定が面倒ですが、以下の手順で使えます

- 1. アカウント作成
UIFlow2の右上のこのマーク
→ログイン画面が出るので
アカウント作成→ログイン
- 2. 顔表示関連を使うときは、
Customで以下にある
“HelloATOMS3.m5b2”を読み込む

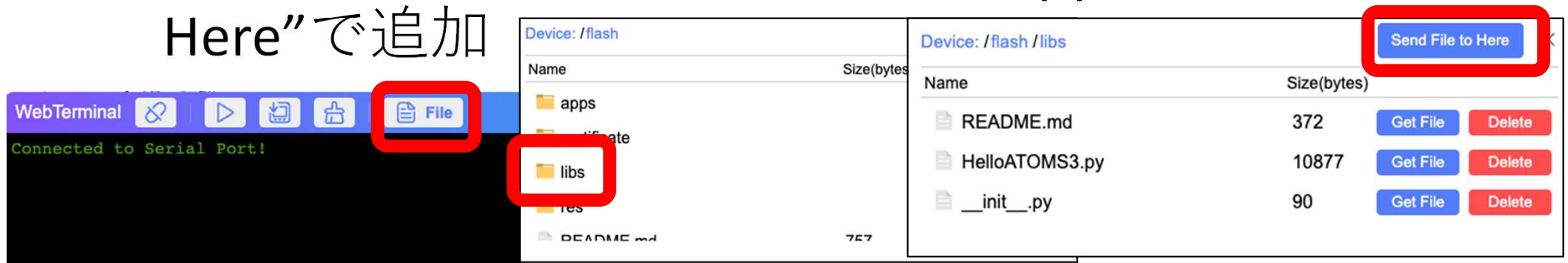


ファイル置き場： <https://github.com/akita11/HelloATOMS3/>

またはこちらから直接UIFlow2のサンプルプログラムを読み込んでもOK
<https://uiflow2.m5stack.com/?pkey=08524484bba14a5eada7d8ac0e1ab29b>

(おまけ) 新しいATOMS3を使うときは

- ちょっと面倒ですが、新しいATOMS3は、以下の初期設定を行えば、同じように使えます
- 1. 設定プログラムの書き込み
以下から”M5Burner”をダウンロードして実行
→左側の「ATOMS3」→「UIFlow2.0」を書き込み
(Download→Burn※UIFlow2と同じIDでログイン)
<https://docs.m5stack.com/en/download>
- 2. UIFlow2で実行時に出てくるTerminal (黒い画面) を開き、上のFileから/libs/に、前ページ「ファイル置き場」にあるHelloATOM3.pyを”Send File to Here”で追加



もっとやってみたい人は . . .

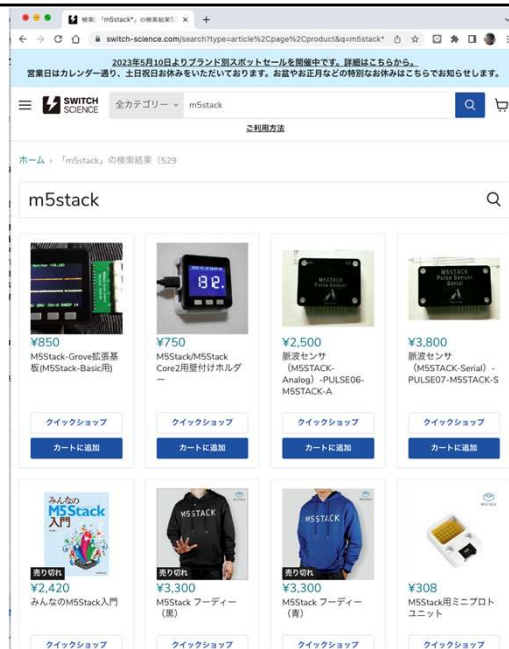
マルツ（電子部品などのお店）



金沢営業所（金沢西インター店）
〒921-8005 石川県金沢市間明町2-267



スイッチサイエンス（Webショップ）



“MakerFaire”などの展示会にいくと
いろんな作品を作っている人に会えます！
金沢だと「NT金沢」（毎年6月中旬）など
→“Maker イベント”で検索！

設定方法など、ご不明な点は、
お気軽にお問い合わせください
akita@ifdl.jp

<https://switch-science.com>