

ScienceCUBEのセンサーを使えるようにするには、いくつかの手順や設定が必要です。

ステップ-1 「拡張機能」でセンサーの動作や測定の専用ブロックを準備

ステップ-2 各センサーの測定値を記憶する「変数」をつくる

ステップ-3 測定用関数ブロックをつくり、変数ブロックを入れる

ステップ-4 変数ブロックとセンサーのブロックを関連付ける

ステップ-5 プログラムをダウンロード

測定用関数ブロック



今回の体験教室では、**ステップ-4** と **ステップ-5** 行いプログラムを完成させます。

ステップ-1

「拡張機能」でセンサーの動作や測定の専用ブロックを準備

用意する拡張ブロック



ステップ-2

各センサーの測定値を記憶する「変数」をつくる



変数 = 「変わる数」
数値や文字を一時的に記憶するときに使います。
変数にはわかりやすい「名前」をつけることができる
[変数]ブロックは、micro:bitのメモリー内につくられる「引き出し」です。
micro:bitの電気が止まると内容は消えてしまいます。

ステップ-3

測定用関数ブロックをつくり、変数ブロックを入れる



関数ブロック
よく使う複数のブロックを、一組として名前（ラベル）を付けて使います。
プログラムの中で、名前を呼び出して何度でも利用できます。

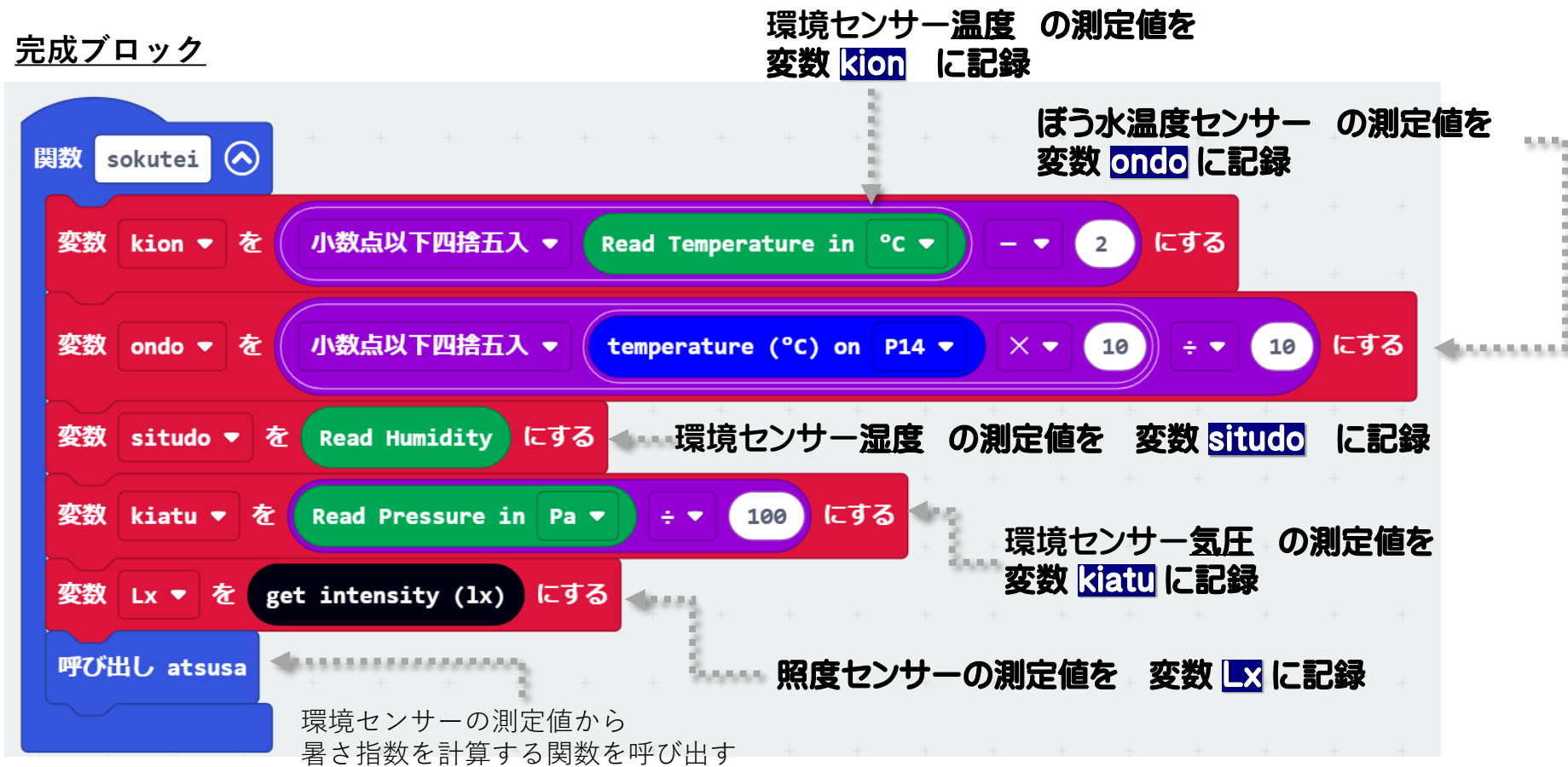
ステップ-4

変数ブロックとセンサーのブロックを関連付ける



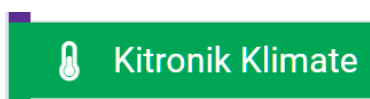
環境センサー(温度/湿度/気圧)、防水温度センサー、照度センサーの測定値を変数に記録する関数ブロック『sokutei』を完成させます。

完成ブロック



・使うセンサー用ブロックを準備します

ツールボックスから色を参考にして探そう！



気圧 Read Pressure in Pa ▼

気温 Read Temperature in °C ▼

湿度 Read Humidity



照度 get intensity (lx)



温度 temperature (°C) on P14 ▼

※接続端子は **P14** に変更

・測定用関数ブロック[sokutei]に、当てはまるのセンサブロックを入れて完成させましょう

関数 sokutei

気温⇒ 変数 kion ▼ を 小数点以下四捨五入 ▼ 気温 - ▼ 2 にする センサーの値が約2℃高いので2℃引いて補正

温度⇒ 変数 ondo ▼ を 小数点以下四捨五入 ▼ 温度 × ▼ 10 ÷ ▼ 10 にする 小数第3位まで測定されるので、小数第1位で表示されるよう計算

湿度⇒ 変数 situdo ▼ を 湿度 にする (((温度を10倍して)四捨五入)それを10倍)

気圧⇒ 変数 kiatu ▼ を 気圧 ÷ ▼ 100 にする 気圧の単位hPaに変化するため100で割る

照度⇒ 変数 Lx ▼ を 照度 にする

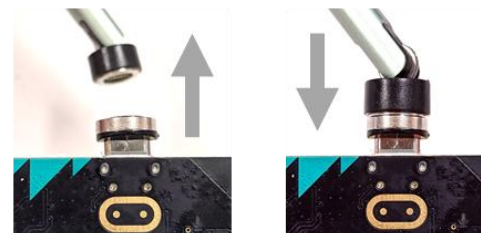
呼び出し atsusa

28.652	測定温度
286.52	(温度を10倍)
287	(四捨五入)
28.7	(10で割る)

① ロゴが  となっているか確認



ロゴが  **ダウンロード** の場合は、ケーブルをつなぎなおします



② ロゴの  点めつが終わってから、**ダウンロード**をクリック

③ ダウンロードが終わったことを確認



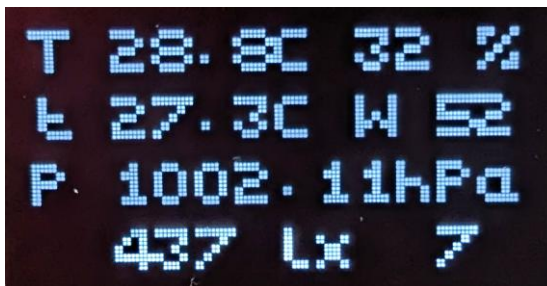
※なにもプログラムをつくらなかったの、micro:bitは動作しません。

micro:bitにプログラムを転送中はLEDが点滅します



モニターに表示される測定値

1. 気温 (°C)
2. 湿度 (%)
3. 気圧 (hPa)
4. 照度 (Lx)



5. 湿度 (%)
6. 暑さ指数 (°C)

※測定/送信回数
(5秒で1増える)

センサーの精度から、計算式で小数第1位までを表示する方法

◆環境センサー温度 測定ブロック

変数 kion を 小数点以下四捨五入 Read Temperature in °C × 10 ÷ 10 にする

気温の値は、25.78 など数第2位まで表示されます。

かけ算

Read Temperature in °C × 10

1. 気温を10倍する

四捨五入

小数点以下四捨五入 Read Temperature in °C × 10

2. 10倍した気温を四捨五入する

わり算

小数点以下四捨五入 Read Temperature in °C × 10 ÷ 10

3. 四捨五入した気温を10で割る

◆ぼう水温度センサー 測定ブロック

変数 ondo を temperature (°C) on P14 にする

気温の値は、25.785 など小数第2位や4位まで表示されます。

1. 温度を10倍する

0 × 0

2. 10倍した気温を四捨五入する

0 小数点以下四捨五入 0

3. 四捨五入した気温を10で割る

0 ÷ 0

測定温度

28.652

(温度を10倍)

286.52

(四捨五入)

287

(10で割る)

28.7

T 33.1C 46%
P 1005.1 hPa
L 27.8C 128
S 65 LX 59

T 32.9C 47%
P 1005.08 hPa
L 27.5425C 12
S 80 LX 1

```

最初だけ
  アイコンを表示
  init OLED with addr 60
  show string at x 1
  y 1
  text "Science CUBE"
  color 1
  set Address 35
  シリアル通信
  通信先を変更する
  送信端子 P15
  受信端子 P16
  通信速度 115200
  端子 P14 を ブルアップ する
  変数 tm を
    CLK P8
    DIO P9
    intensity 4
    LED count 4
  にする
  変数 moni を 5 にする
  変数 count を 0 にする
  くりかえし 100 回
    tm show number 0 から 9999 までの乱数
    一時停止 (ミリ秒) 100
  turn on
  音量を設定する 80
  呼び出し sokutei
  呼び出し atsusa
  呼び出し moniled

```

```

ずっと
  呼び出し sokutei
  変数 配列 を
    kion
    ondo
    situdo
    kiatsu
    Lx
    WBGT
  にする
  tm show number 配列 の moni の 1 番目の値

```

```

ボタン A が押されたとき
  もし moni < 配列の長さ 配列 なら
    変数 moni を 1 だけ増やす
  呼び出し moniled

```

```

ボタン B が押されたとき
  もし moni > 1 なら
    変数 moni を -1 だけ増やす
  呼び出し moniled

```

```

ボタン A+B が押されたとき
  変数 Display ON を 偽 にする
  clear
  表示を消す

```

```

ログが 短くタップされた とき
  呼び出し moniled

```

防水温度計の測定不能時、自動で画面クリアできないときにログをタップして強制クリア

```

関数 sokutei
  変数 kion を 小数点以下四捨五入 Read Temperature in °C - 2 にする
  変数 ondo を 小数点以下四捨五入 temperature (°C) on P14 × 10 ÷ 10 にする
  変数 situdo を Read Humidity にする
  変数 kiatsu を Read Pressure in Pa ÷ 100 にする
  変数 Lx を get intensity (lx) にする
  呼び出し atsusa

```

```

関数 moniled
  変数 Display ON を 真 にする
  数を表示 moni

```

```

関数 atsusa
  変数 WBGT を -3.246 にする
  変数 WBGT を 0.725 × kion だけ増やす
  変数 WBGT を 0.0368 × situdo だけ増やす
  変数 WBGT を 0.00364 × kion × situdo だけ増やす
  変数 WBGT を 小数点以下四捨五入 WBGT にする

```

環境測定ユニット

サイエンス Science CUBE プログラム

キューブ

環境測定ユニット

Sceinece CUBE

サイエンス
キューブ
プログラム

```
1 × 10 × 1000 ミリ秒ごとに
鳴らす 音を鳴らす 高さ (Hz) 上のシ 長さ 1 拍 バックグラウンドで一度だけ
もし Display ON なら
  clear
  呼び出し moniS
  呼び出し moniN
  シリアル通信 文字列を書き出す {"
  シリアル通信 文字列を書き出す 文字列をつなげる "temperature": kion " "
  シリアル通信 文字列を書き出す 文字列をつなげる "humidity": situdo " "
  シリアル通信 文字列を書き出す 文字列をつなげる "pressure": kiatu " "
  シリアル通信 文字列を書き出す 文字列をつなげる "KEY1": ondo " "
  シリアル通信 文字列を書き出す 文字列をつなげる "KEY2": Lx " "
  シリアル通信 文字列を書き出す {"}
  変数 count を 1 だけ増やす
```

```
関数 moniS
  show string at x 0
  y 0
  text "T"
  color 1
  show string at x 11
  y 0
  text "%"
  color 1
  show string at x 0
  y 2
  text "p"
  color 1
  show string at x 10
  y 2
  text "hPa"
  color 1
```

```
防水温度計の測定不能時の処理
エラーコード (1, 3, 4 等) が出たときは、
OLEDの表示をクリアー

temperature sensor errorMessage errorCode port
もし errorCode > 0 なら
  変数 ondo を -99 にする
```

```
関数 moniN
  show a Number at x 2
  y 0
  number kion
  color 1
  show a Number at x 8
  y 0
  number situdo
  color 1
  show a Number at x 2
  y 1
  number ondo
  color 1
  show string at x 8
  y 1
  text 文字列をつなげる "W" WBGT " "
  color 1
  show a Number at x 2
  y 2
  number kiatu
  color 1
  show string at x 2
  y 3
  text 文字列をつなげる Lx " Lx "
  color 1
  show a Number at x 10
  y 3
  number count
  color 1
```