お茶の水女子大学理系女性教育開発共同機構主催

於：お茶の水女子大学アカデミックプロダクション研究棟

2020

micro:bit によるプログラミング講習会

～マイコンとセンサで科学研究にチャレンジ～

2020／02／09

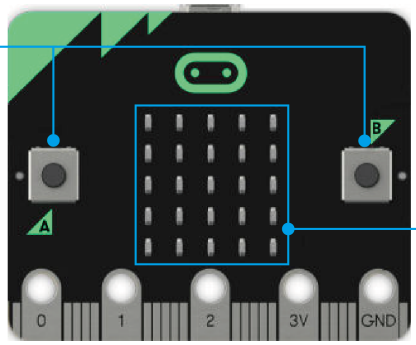
柏木 隆良

micro:bit について

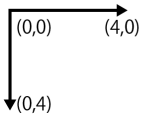
micro:bit は BBC（英国放送協会）が主体となって作っている教育向けマイコンボードで、ユーザーが動作をプログラミングできる 25 個の LED と 2 個のボタンスイッチのほか、加速度センサーと磁力センサー、ブルートゥース機能を搭載しています。イギリスでは 11～12 歳の子供に無償で配布されており、全ての子供がプログラミングを学ぶ環境が整い始めています。

ユーザーボタン A,B
携帯ゲームマシンのような 2 つのボタン入力。プルアップ (抵抗を介して電源に接続) しており、通常は 1 の信号、押すことで 0 の信号が出力され、マイコンに伝わります。

ディジタル／アナログの入出力端子
外部との様々な制御・通信に利用できます。0,1,2 の文字がある端子は、P0,P1,P2 信号としてマイコンに接続されています。大きな端子のためワニクリップやバナナプラグの接続が可能です。



LED マトリックス
縦 5x 横 5 計 25 個の LED が赤く光ります。プログラム上の位置座標は左上端が (0,0) になります。
明るさを替えることもでき、アルファベットと数字の表示も可能です。
複数の文字列は左から右へ流れるように連続的に表示が行われます。
赤色 LED の特性を利用して光センサーとしても利用しています。



電源端子
エッジコネクタの中の GND はグラウンド端子、3V は電源端子です。外部に対して電源を供給するときに使います。表示は 3V ですが、micro:bit の動作電圧約 3.2V が出力されます。逆に、USB 端子や電源ソケットが接続していないとき、3V 電源端子から micro:bit に対して電源を供給することもできます。

USB 端子
多くのスマートフォンと同じ Micro-B 規格の USB 端子。
PC と接続すると USB メモリと同じく接続ドライブのようにみえます。
作成したプログラムの転送に使用します。動作中は PC 間とのデータ通信や電源供給にも利用します。

USB 確認用 LED
USB ケーブルを接続すると黄色に点灯します。
プログラム転送時には点滅し、終了すると点灯に戻ります。
点灯状態になったことを確認してから、プログラムの動きをみます。

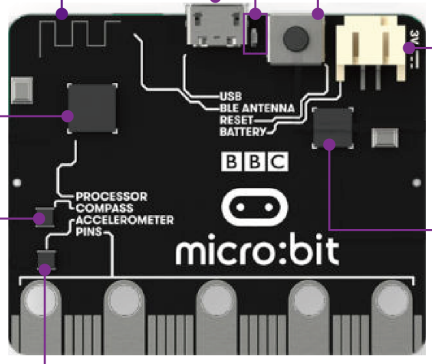
リセットボタン
メイン・マイコンを再起動するときに使います。
プログラム転送後は自動的にリセットがかかりますが、確実にプログラムを起動したいときに押すこともあります。

2.4GHz アンテナ
基板のパターンがアンテナになっています。
BLE(Bluetooth Low Energy) の通信を行うためのものです。

メイン・マイコン
micro:bit のメインとなるマイコンです。micro:bit の全体の機能を動作・制御します。BLE による通信機能や温度センサーも内蔵されています。

地磁気センサー
3 軸 (X,Y,Z) の磁気の強さをデータ化して、マイコンに送ります。micro:bit を水平に置くと平面内 (X,Y) データにより、磁北を 0 度としてロゴのある場所が向いている方向を求めることができます。

加速度センサー
3 軸 (X,Y,Z) の加速度センサー。micro:bit の動きや傾きデータを測定してマイコンに送ります。



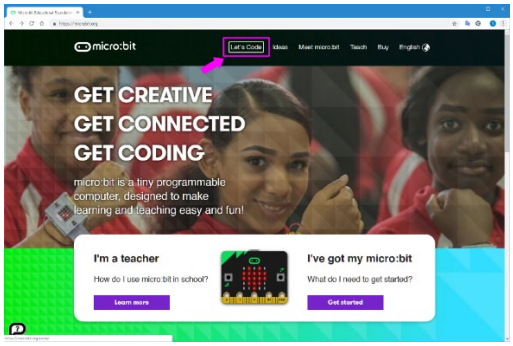
電源ソケット
USB を接続していないとき、外部から電源を供給するための端子です。
1.5Vx2 の乾電池ボックスなどをつなぎます。

インターフェイス・マイコン
メイン・マイコンと USB を繋げるためのマイコンです。性能的にはメイン・マイコンと同じレベルのものですが、基本的には通信のやり取りしか行いません。

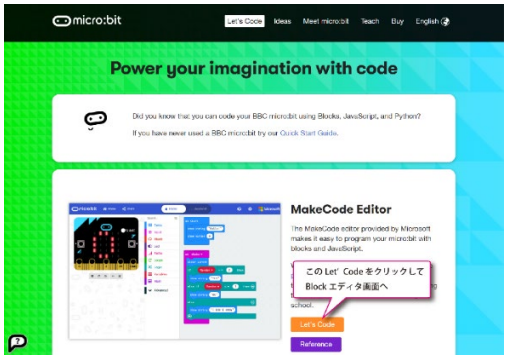
ACTION1 ： 図形をＬＥＤに表示しよう

① Web ブラウザのアイコンをクリックして起動し、micro:bit のページにアクセスします。

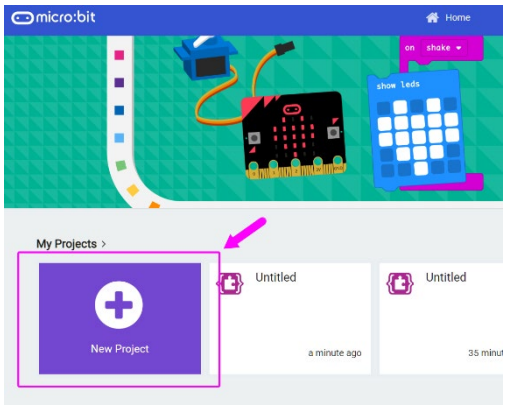
https://microbit.org/



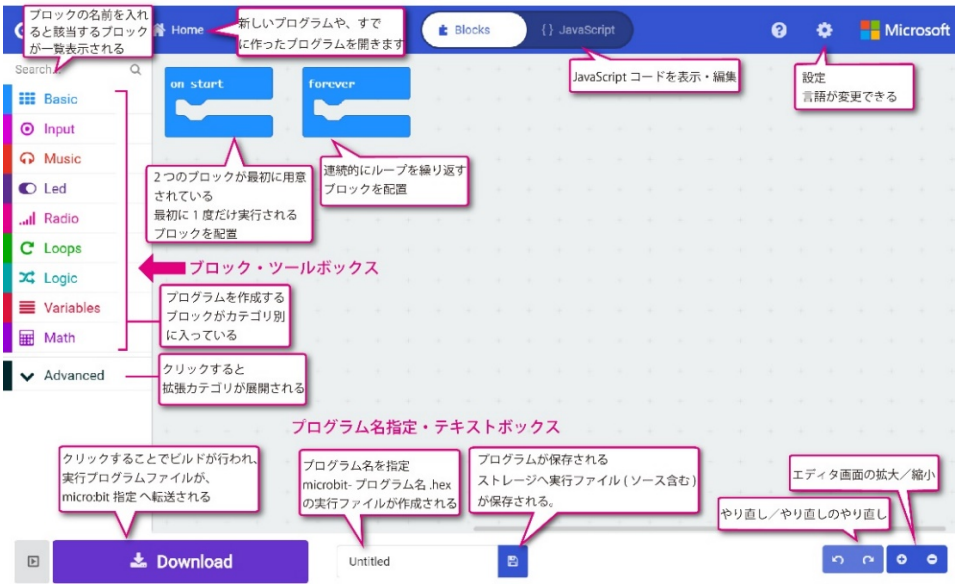
② Let's Code ボタンをクリックします。



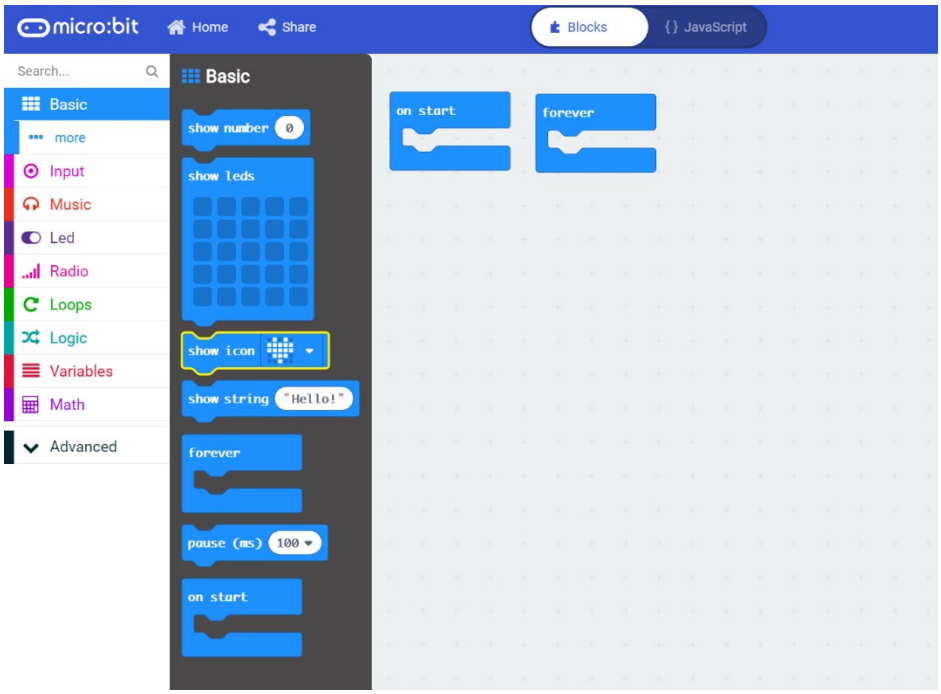
③ 右の画面が表示した場合、New Project をクリックする。



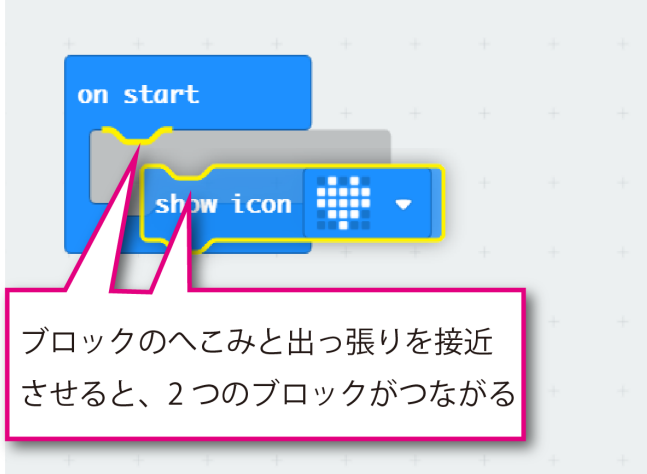
④ MakeCode の Block エディタ画面が表示されます。



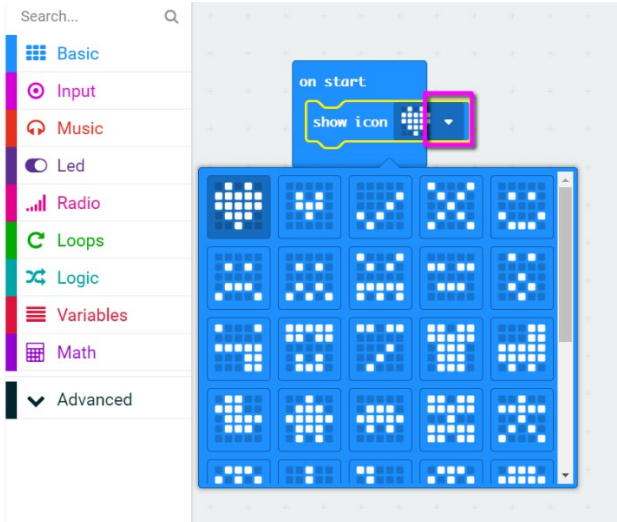
⑤　新規に作成する場合は、ステージに、on start ブロックと、forever ブロックが配置されています。Basic ブロックカテゴリーの show icon ブロックを使用します。



⑥　show icon ブロックを on star ブロックの中に配置します。

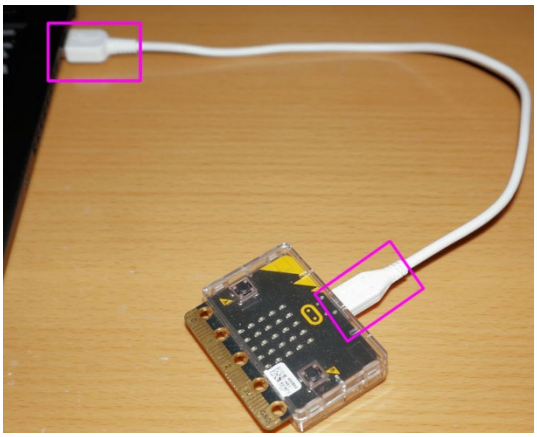


⑦　show icon ブロックの右下矢印をクリックして、表示したイメージ一覧から、表示したいイメージを選択します。

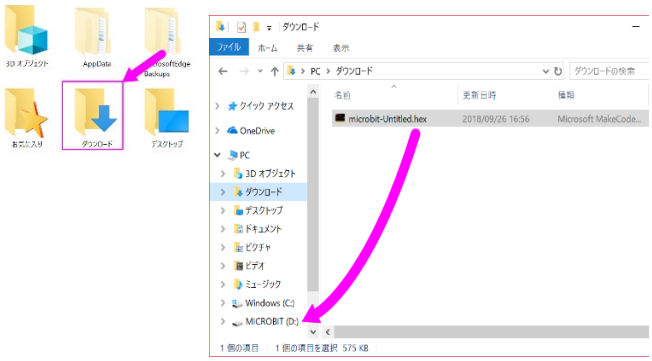


⑧　❶テキストボックスにプログラム名 act1 を入れた後、❷Download ボタンをクリックします。すると、micro:bit が実行できるプログラムへの変換（コンパイル）が行われ、ダウンロードフォルダに microbit-act1.hex ファイルが保存されます。

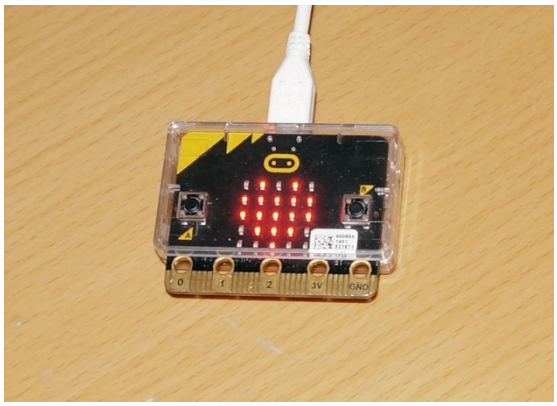
⑨　micro:bit をパソコンと USB ケーブルで接続します。



⑩　ダウンロードフォルダにある microbit-act1.hex ファイルを MICROBIT(D) にドラッグ＆ドロップします。



⑪　micro:bit へのプログラムの転送中は、黄色 LED が点滅します。点滅が終わるとリセットがかかり、プログラムが最初から実行されます。



ACTION 2 ： 文字列 ABC を表示させましょう。

① forever ブロック枠内に Basic カテゴリの中の show string ブロックを配置します。

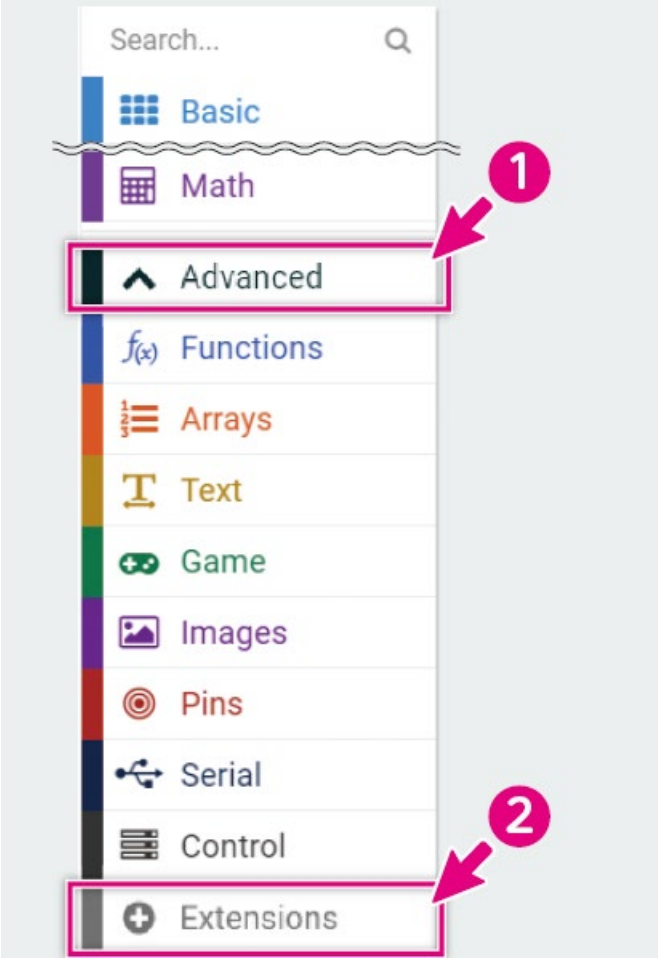
② show string ブロック内の Hello!を ABC(自分の名前でも可)に変更します

③ プログラムを実行すると ABC 文字列が横スクロールして表示され、それを繰り返します。

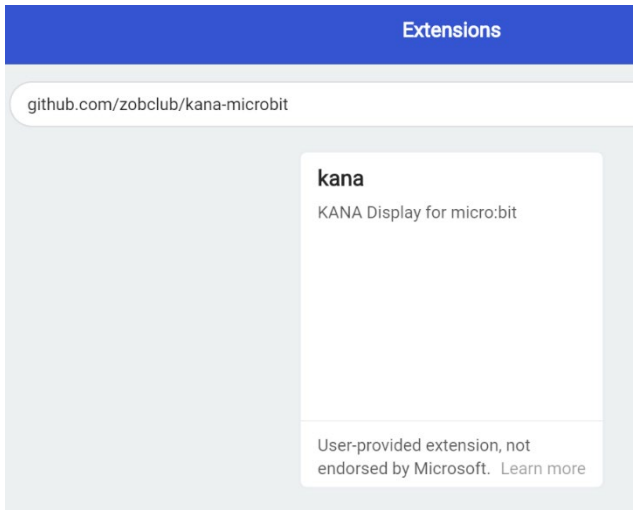
ACTION 3 ： 拡張機能を利用して、カナ／ひらがなを表示させましょう。

① MakeCode は、ブロックを拡張する機能を有しています。

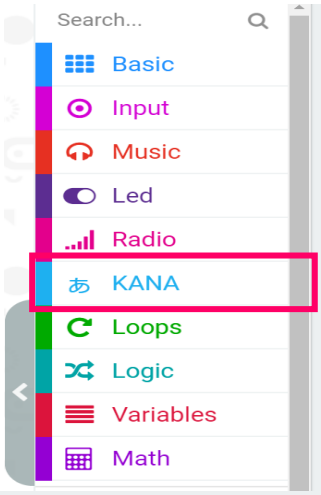
ブロック・ツールボックスの一番下に位置する Advanced カテゴリ❶をクリックすると Fig.2_32 のようにブロックカテゴリの一覧が広がります。そのまた一番下の Extensions❷をクリックします。



② ［Extensions］のウィンドウが開くので、ここでテキストボックスに「github.com/zobclub/kana-microbit」と入力して、右端の虫眼鏡アイコンをクリックして検索します。すると1 つのブロック・パッケージが見つかります。



③ 左のブロックツールボックスに **KANA** が表示されれば成功です。これでカナ／ひらがなを表示するプログラムを作成することができるようになります。



④ forever ブロック枠内に KANA カテゴリの中の show KANA ブロックを配置します。

⑤ カナカナを全角で入力します。キーボードの半角／全角キーを押してからローマ字入力でカタカナを入力します。ひらがなで入力された場合は、文字確定する前に F7(キーボードの上にある)を押すとカタカナになります。

⑤ この拡張 KANA ブロックは、ひらがなもなんとか表示できます。文字列のところにひらがなを入力してみましょう。

KANA カテゴリ内の set scroll ブロックを前に入れるとスクロールスピードとスクロールする方向を変更することができます。

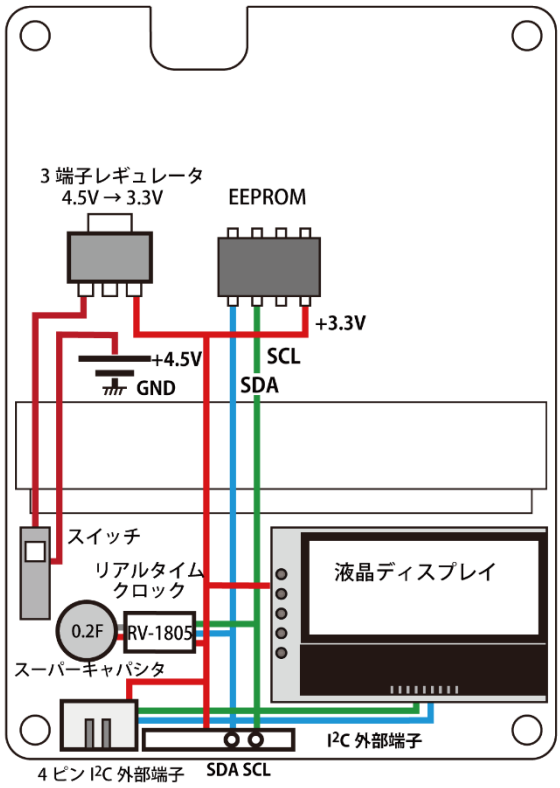
vertical を選ぶと縦にスクロールします。

STEM 教育の学習内容を考えてみよう

- Science（科学）
- Technology（技術）
- Engineering（工学）
- Mathematics（数学）
- それぞれの単語の頭文字をとったものとなっています。世界が今後も大きく成長していく中で重要な分野だと考えられています。STEM 教育は科学と数学を土台として展開する科学技術人材育成を行おうという戦略があり、社会に出る前の子どもたちが、将来そうした舞台でリーダーとして活躍することを目的とした教育です。
- IoT 機器を使って、どのような STEM 教育が考えられるでしょうか。

科学研究用拡張ボード(SCIENCE:bit)を使ってみよう。

科学研究用拡張ボード(SCIENCE:bit)は、実践的な科学研究用のデータ・ロガーとして活用できるように、電池による駆動と、不揮発性メモリ ERPROM へデータを記録する機能も有しています。



SCIENCE:bit には次のような機能があります。

1　8 文字行のバックライト付き液晶ディスプレイにより、リアルタイムに測定データを表示

2　リアルタイムクロックを搭載し、1/100 秒単位で時間を計測でき、スーパーキャパシティにより電源や電池を接続しなくても、時を刻み続けます。

3　6 ピン(Pmod 互換)の I2C 端子と 4 ピン基板コネクタ(Grove システム互換)を備え、様々な I2C センサデバイスの接続が可能

4　単 4 電池×3 により電源を供給して、micro:bit+SCIENCE:bit ボードのみで動かすことができます。

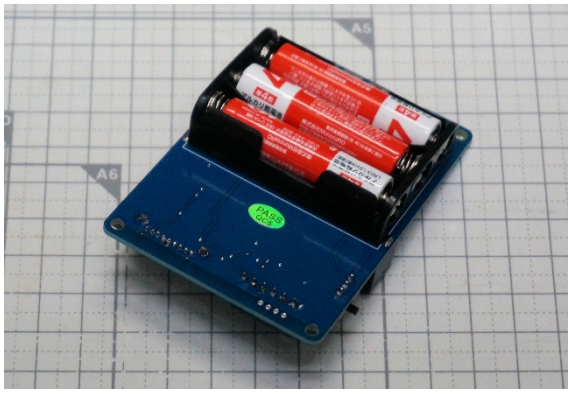
5　コンパクトで、楽に持ち運びができ、これだけで高機能なデータ・ロガーとなります。

センサ及びリアルタイムクロックは、MakeCode のブロック用のライブラリが用意されているので、簡単にプログラミングが可能です。また、データ保存用に EEPROM アクセス用のブロックもあるため、本格的な科学研究の測定用のプログラムを作成することができます。

SCIENCE:BIT の準備

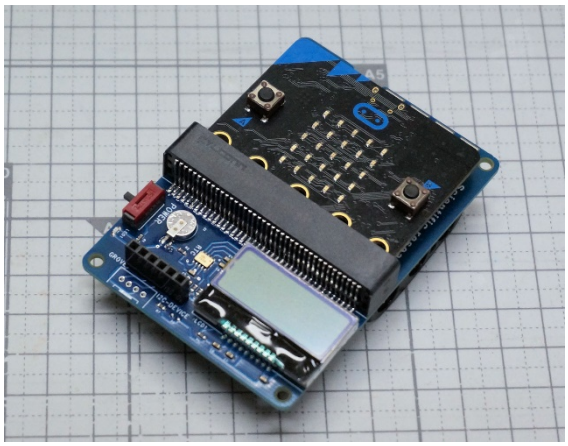
①電池セット

SCIENCE:bit の裏側に単四電池 3 本をセットします。電池の向きを交互に入れます。表側のスイッチを ON にして赤 LED が光ったらセット成功です。



②スイッチを OFF した後、micro:bit をセット

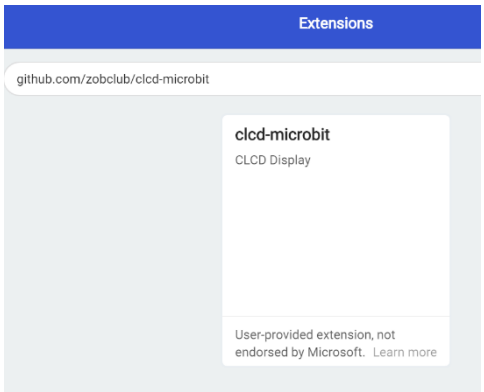
SCIENCE:bit の表側に micro:bit をセットします。USB ケーブルを PC に接続する前に、スイッチを ON します。



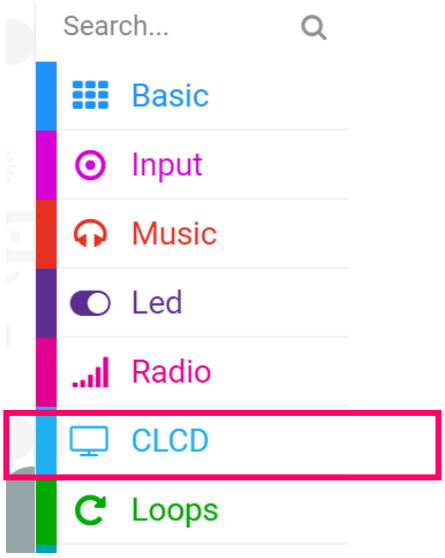
ACTION4 ： LCD(液晶ディスプレイ)に数／文字を表示させましょう。

SCIENCE:bit には、8 文字 2 行のアルファベットと数字表示が可能な液晶ディスプレイが備わっています。micro:bit では表示しずらい数値／文字列を見やすく表示できます。

① Action 3 と同様にブロック・ツールボックスの一番下に位置する Advanced カテゴリ、Extensions をクリックします。検索ボックスに「**github.com/zobclub/clcd-microbit**」と入力して、アイコンをクリックして検索します。clcd-microbit のブロック・パッケージが見つかります。



② これをクリックすると左のブロックツールボックスに CLCD カテゴリ表示され、これで CLCD を操作するプログラムを作成することができるようになります。



③ 図のようにブロックを配置します。

init clcd: 液晶ディスプレイの初期化（使用する前に 1 度だけ必ず実行）

set contrast:液晶ディスプレイの文字の濃さを設定(35 程度)

print string: 文字列を表示(数字とアルファベットのみ)

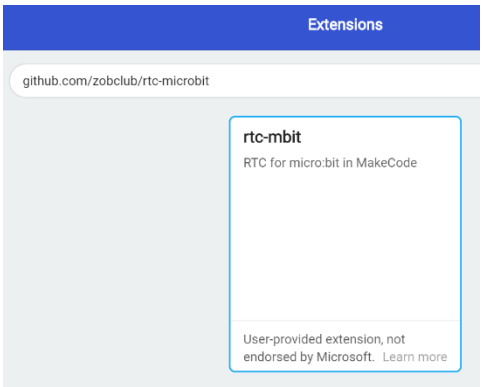
液晶ディスプレイに指定した表示が表示されます。

ACTION5 ： 時計プログラムを作ってみよう。

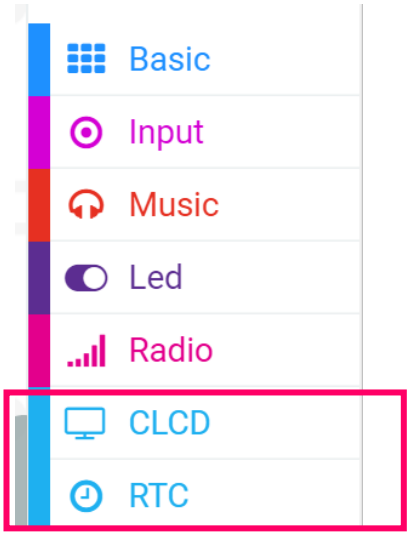
SCIENCE:bit には時を刻むリアルタイムクロックが搭載されています。1/100 秒の正確な動作を行い、超消費電力で動作します。

① ブロック・ツールボックスの一番下に位置する Advanced カテゴリ、Extensions をクリックします。検索ボックスに「**github.com/zobclub/rtc-microbit**」と入力して、アイコンをクリックして検索します。rtc-microbit のブロック・パッケージが見つかります。

clcd-microbit のブロック・パッケージと同時に使用するので Action 4 のプログラムを変更する形でプログラムを行います。



② これをクリックすると左のブロックツールボックスに **RTC** カテゴリが新たに表示され、**CLCD** カテゴリと同時に使用してプログラムを作成します。



③ 1 行目に年月日と 2 行目に時分秒を表示します。

init rtc:リアルタイムクロック初期化

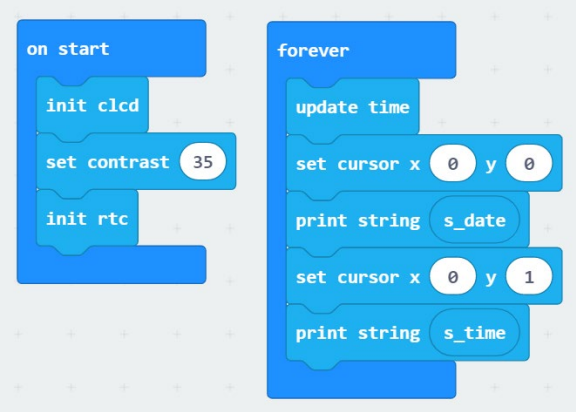
forever ブロック枠内に

update time: 時間データを更新

set cursor で文字表示位置を指定(x < 7,y=0 or 1)

print string に t_date をセットすることで”年/月/日”が表示

t_time をセットすることで”時：分：秒を表示



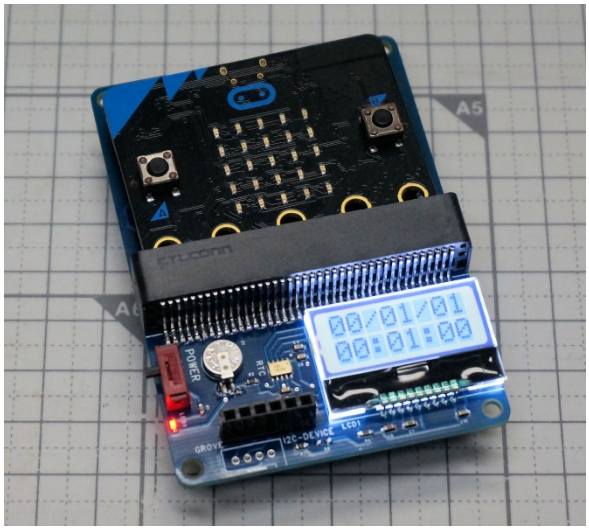
④ プログラム名を **rtc1** として保存した後、プログラムを実行すると右写真のような表示になります。

初期データの設定をしていないので、

2001 年 1 月 1 日(2001 の先頭の 20 は表示されません)

00 時 00 分 00 秒

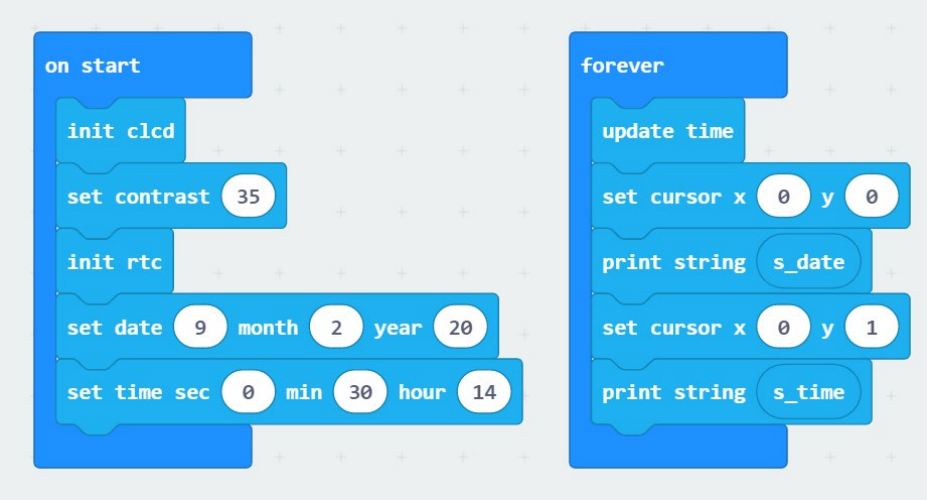
から表示されます。1 分以上表示を続けると、スーパーキャパシティに電気がたまり、スイッチを OFF にしても、以後 RTC は動き続けます。(35 時間以上)



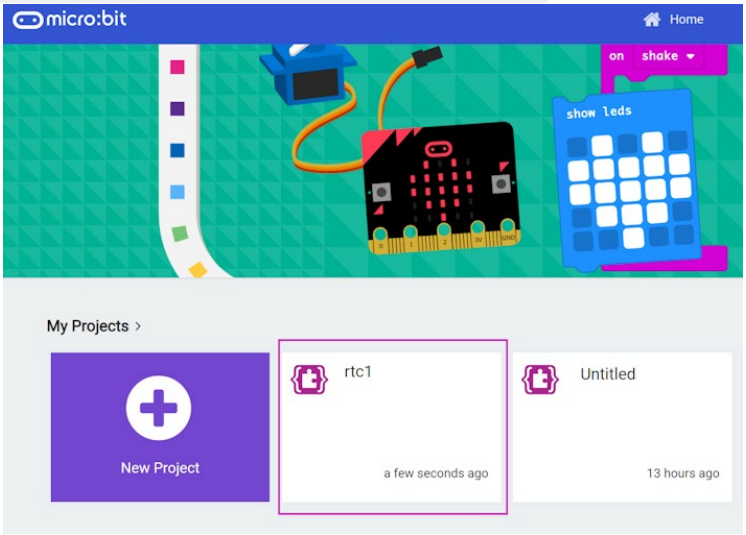
⑤ 初期データを設定します。

プログラム名を **rtc1** の on start ブロック内に **set date** ブロックと **set time** ブロックを配置し、今の日時と時間をセットします。このプログラムの実行時にデータは設定されるので、現在時刻より 30 秒程度後の時間を設定します。

プログラム名は **rtc2** として保存した後、1 度だけ実行します。

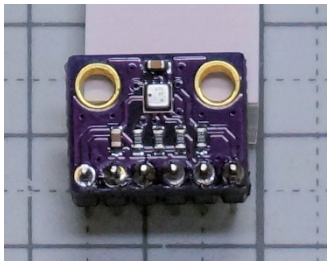


⑥　プログラム **rtc1** を読み込んで実行することで時計の完成です。初めから Action1 の操作をすると　**rtc1** プロジェクトが表示されてるはずです。そこをクリックしてロードします。



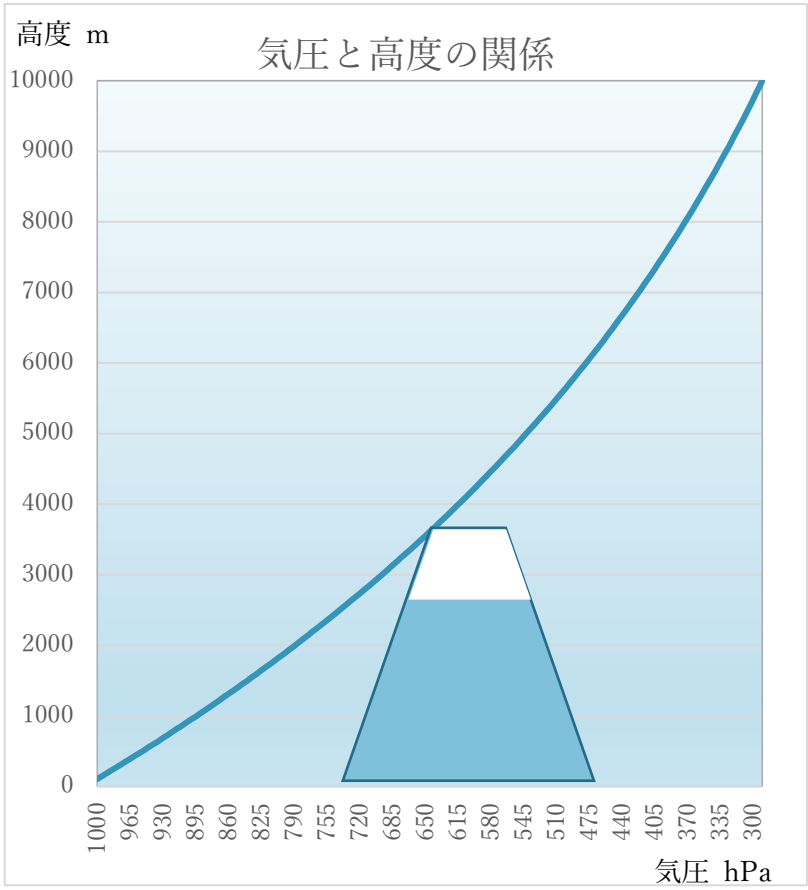
気圧を測定しよう

・環境センサ　BME280



環境の変化を取得するセンサとして、色々なセンサが存在します。その中で、気圧・温度・湿度を高精度にリアルタイムに測定できる環境センサを micro:bit に接続します。
ドイツの自動車部品メーカとしても有名な BOSCH 社から気圧の測定を主機能とした次のような複数の環境センサが提供されています。
2.5mm 平方と非常に小さなデバイス IC で、気圧を 0.2 Pa(パスカル）の精度で測定できます。これは、1.7cm の高度差を調べることができるレベルの精度であり、部屋の中での上下運動をデータの変化としてとらえることができます。
温度センサ部は、気圧を求めるために利用されていますが、micro:bit で利用されているマイコン内の温度センサより正確にデータを求めることができます。加えてに湿度の測定が可能です。

・気圧と高度について



地球には、重力により空気の層が宇宙に逃げずに存在します。空気には質量があり、水の質量により潜水艦が浮かぶことと同じように、空気の中でも気球は浮かびます。気球が浮かんで静止しているとき気球の質量は、その気球の体積の空気の重さと同じになっています。

地上には、高い高度からの空気の重さがかかっています。その重さを地上面の面積で割ったものが、地上の大気圧になります。海抜 0m の地上には、1 平方メートルあたり 10300kg(約 10 トン)もの空気が乗っているのです。この質量が地面を押す力を求めるために、重力加速度(9.8m/s²)をかけ算します。求まった力は 1 平方 m にかかる力なので、これが地表面の気圧になります。

$$10300 \times 9.8 = 101300\ N/m^2$$

力の単位は N(ニュートン)を m2 で割った形式が圧力の単位になりますが少し長いので、圧力の基本原理を発見したフランスの天才数学者パスカルの名前をかりて Pa が圧力の単位として使用されています。k(キロ)は 1000 倍を意味する接頭語ですが、100 倍を表す接頭語として h(ヘクト)もあります。すなわち海抜 0 の地表面を意味する標準気圧である 1 気圧は

1 気圧 ＝ 101300 Pa ＝ 1013 hPa(ヘクトパスカル)

となります。

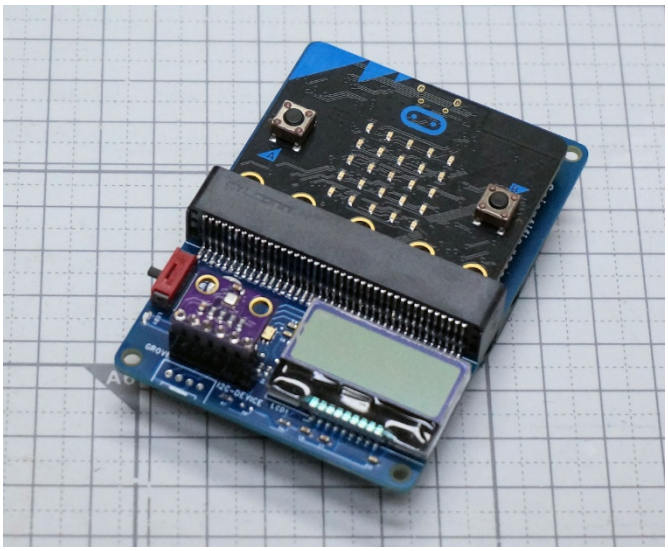
高度が高くなると、その上に乗っている空気の質量も小さくなるため、高いところでは気圧が低くなります。計算式で高さと気圧の関係を表すと次のようになります。

$$h = 18400 \times \log(p0/p)$$ ※底 10 の対数

実際は、気温の差による補正も必要とします。これをグラフで示すと図のようになり、高さ 3776m の富士山の頂上は地上に対して 70%程度の気圧であることが分かります。

SCIENCE:BIT に BME280 センサを取り付け

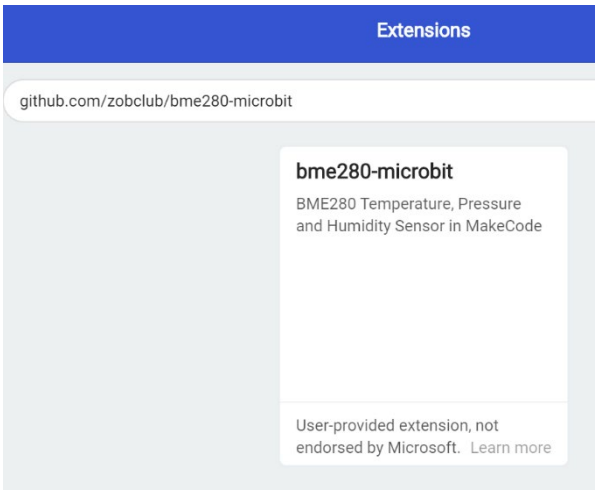
SCIENCE:bit の表側前方に BME280 センサを写真のように取り付けます。
スイッチは OFF の状態でセンサチップ付近を触らないようにセットしてください。



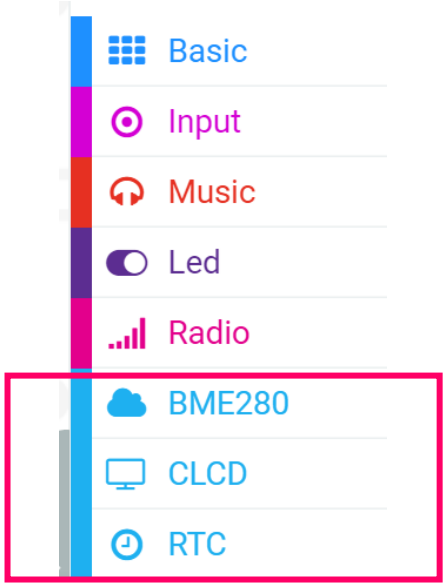
ACTION 5 ： 気圧を測定するプログラムをつくろう。

SCIENCE:bit に接続しセンサのデータを読み取り、液晶ディスプレイにリアルタイムの表示させます。研究用の測定データは、そのときの時間が重要です。そこで 1 行目に時間を 2 行目に気圧を表示させます。

① ブロック・ツールボックスの一番下に位置する Advanced カテゴリ、Extensions をクリックします。検索ボックスに「**github.com/zobclub/bme280-microbit**」と入力して、アイコンをクリックして検索します。
bme280-microbit のブロック・パッケージが見つかります。
clcd-microbit と rtc-microbit のブロック・パッケージと同時に使用するので Action 4 のプログラムを変更する形でプログラムを行います。



② これをクリックすると左のブロックツールボックスに **BME280** カテゴリが新たに表示され、**CLCD** カテゴリ **RTC** カテゴリと共に使用してプログラムを作成します。



③ 1 行目に時分秒を 2 行目に気圧を表示します。

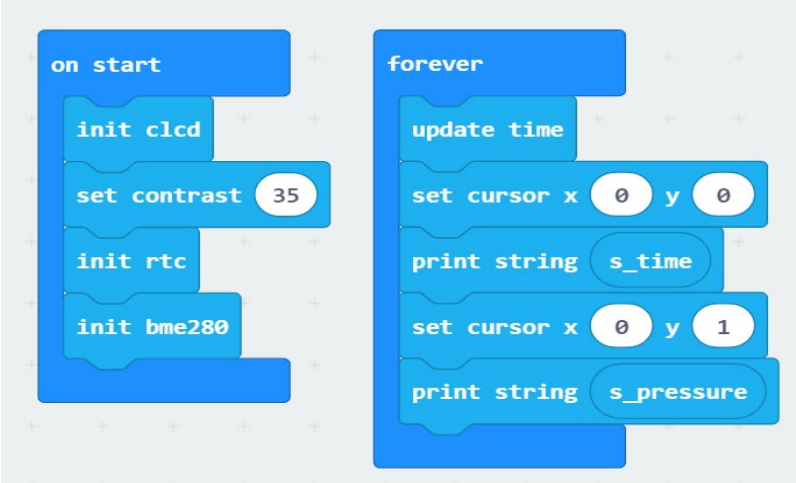
init bme280:BME-280 センサを初期化

forever ブロック枠内に

print string : s_time をパラメータとして時間を表示

print string : s_pressure パラメータとして気圧データを表示

プログラム名 **bme280** として保存して、実行します。



④ 現在の時間と気圧データ表示されます。

晴れていれば 1013hPa(ヘクトパスカル)程度の値が表示され、Action4 から経過した時間の時刻になっているはずです。

SCIENCE:bit を上下に動かすと室内のわずかな気圧の変化により、気圧データが少し変化します。

パラメータを s_temperature にすると温度を表示することも可能です。