

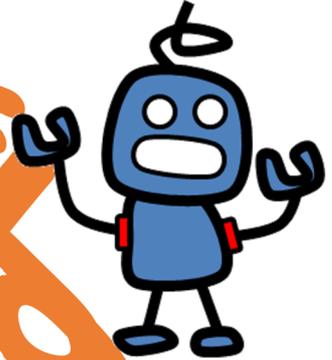


# 第1章 信号機を作ろう

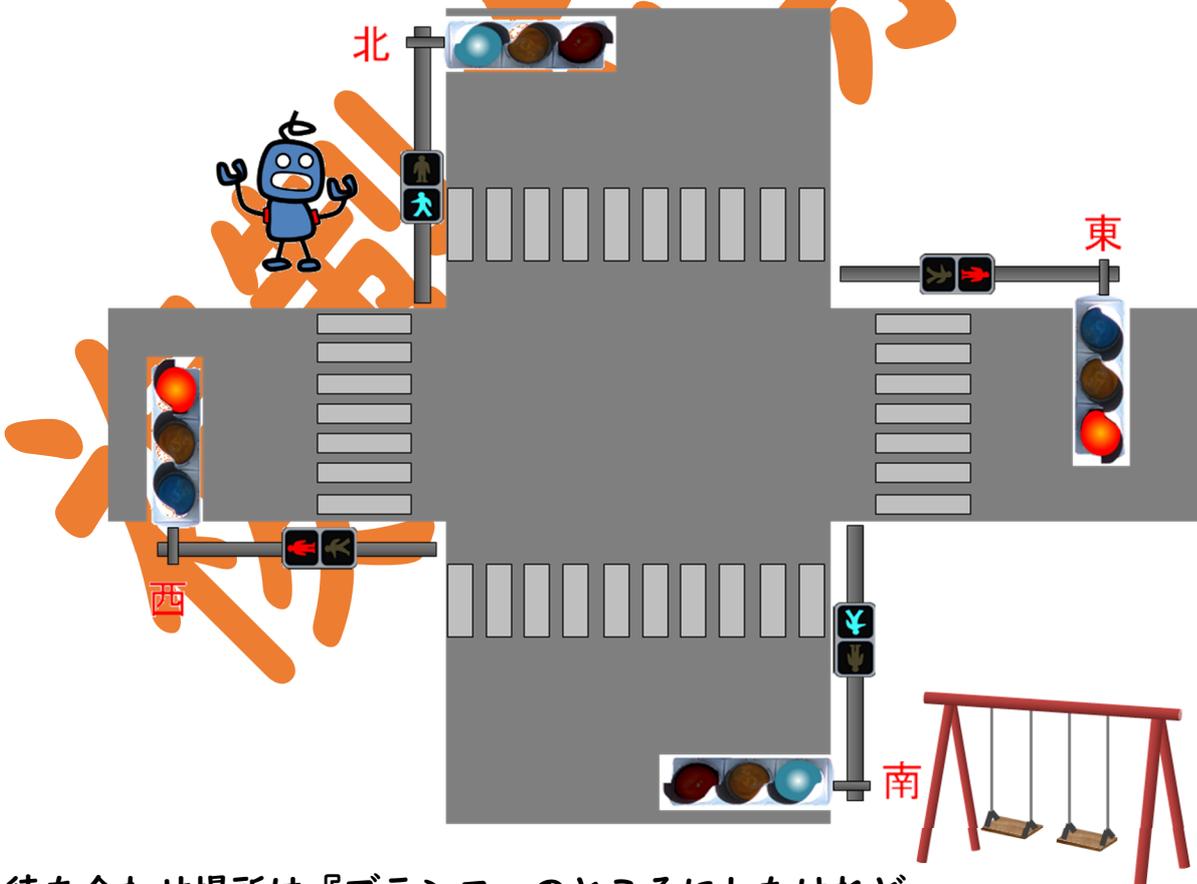
## 1. 身近な信号機を調べよう

ぼくはポンコツロボット。名前はまだない。  
今日は、友だちのノブ君と近くの公園で遊ぶことにした。

家を出て右に3分ほど歩いたら交差点があって、その向かい側に公園がある。公園に行くには横断歩道を2回通らなきゃいけないね。

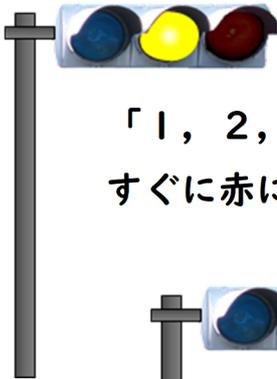
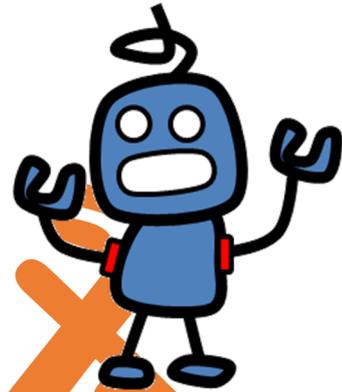
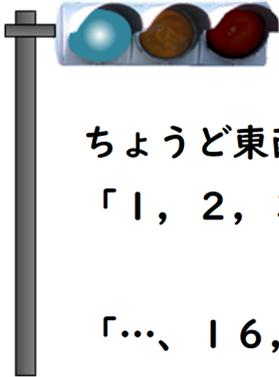


「ノブ君は、もう来てるかなあ？」



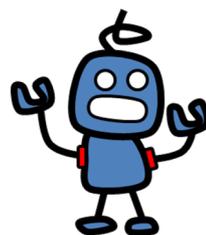
待ち合わせ場所は『ブランコ』のところにしたけれど、  
ノブ君はまだ来ていないようだ。

信号機の色を見て、数を数えて待つことにした。ぼくはポンコツロボットだけど、時計のように正確に数を数えられるんだよ。だって、ぼくは機械だからね。



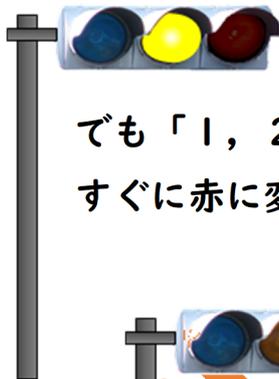
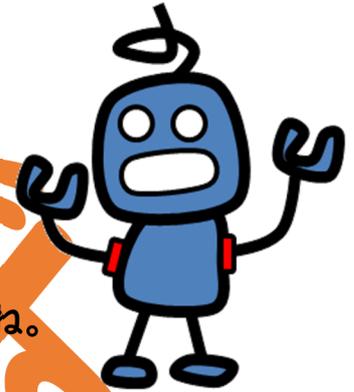
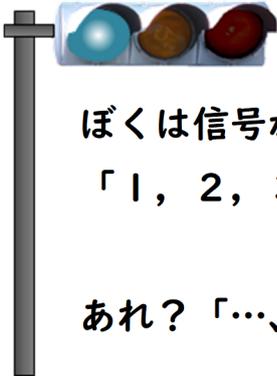
ここから先は、同じことのくり返しだよな。  
せっかくなので、紙に書いておこう。

それにしても、ノブ君遅いなあ。  
何をしているのだろうね。



あま 17 びょう  
き 3 びょう  
あか 18 びょう

じゃ今度は南北方向の信号機を数えてみよう。

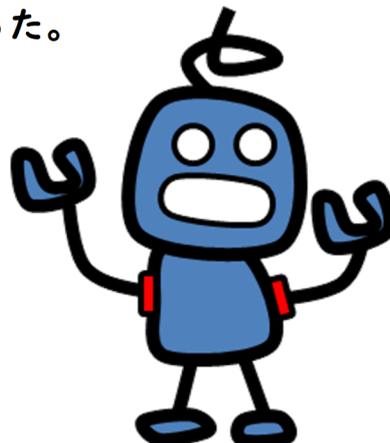


また「1, 2, 3, …」と数えはじめて  
今度は「…、23, 24」でやっと青に変わったね。

ここからは、  
やっぱり同じことのくり返しだった。



あっ! ノブ君がやってきた。



だれか、  
紙に書いてくれないかなあ。

今日、学校でみんなにボクが書いた紙を出したら、  
みんなが見に来た。



「何なの？」

「ひらがなばかりで、見にくいな～」

「表が別々で見にくいのかしら？」

「変だね～数字がバラバラだよ」

「数えまちがいじゃないかなあ」

### 課題1-1

表を一つにして、もっと見やすくしてみよう。



突然、ノブ君が叫んだ。

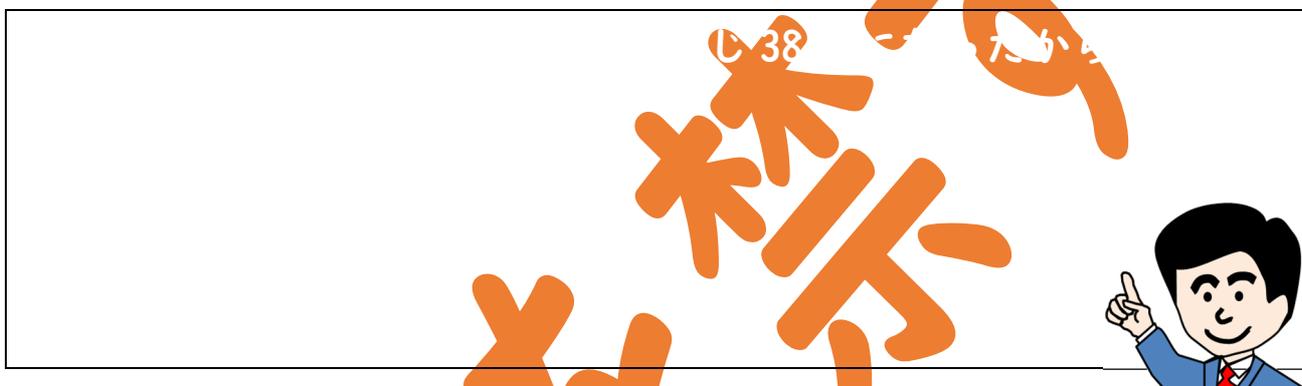
「正しい。正しいよ！」

「ポンロボ君は、きちんと正確に数えたと思うよ！」



ノブ君がいくら説明しても、ぼくには理解できなかった。

**課題1-2** ノブ君が、正しいといったのはなぜか、考えてみよう。

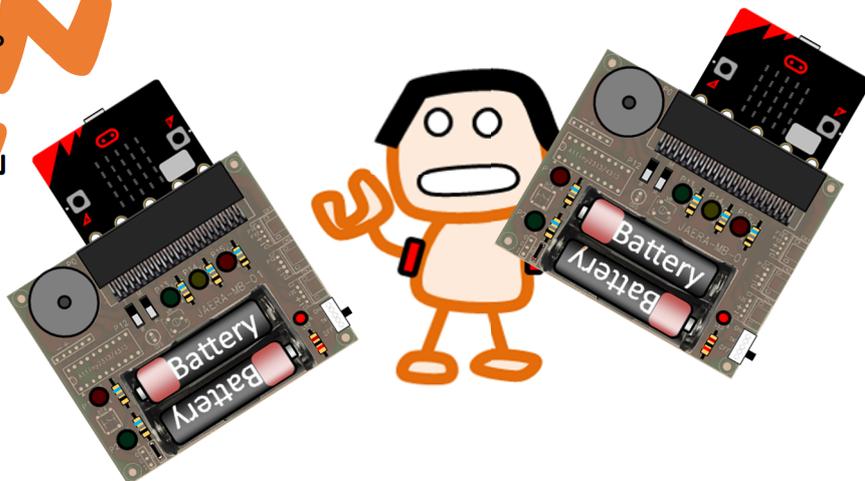


## 2. LEDを点灯し、消灯する

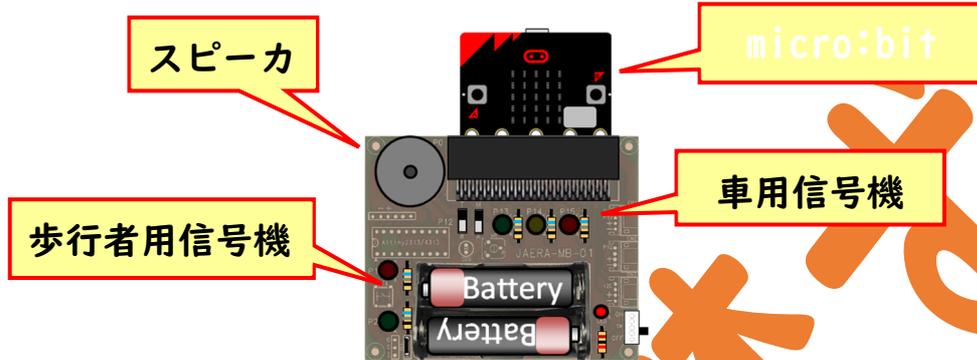
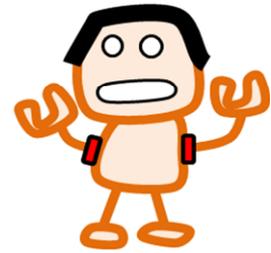
次の日、ノブ君に出会うと

「信号機を持ってきたよ。

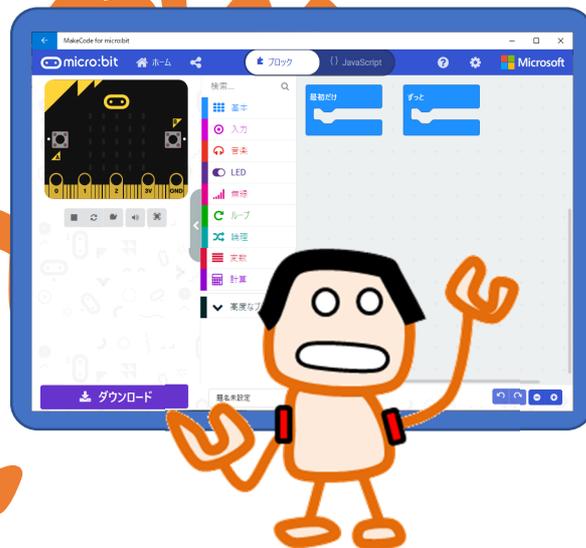
君が正しいことを、  
これで説明するためにね」



「これは、<sup>エンパット</sup>mbATという信号機ボードだよ。  
<sup>マイクロ ビット</sup>micro:bitというマイコンボードに  
プログラムを書き込んで、  
信号機を<sup>さいげん</sup>再現できるんだよ。」



「とにかく、やってみるね」  
ノブ君は、パソコンを取り出して、  
こんな画面をボクらに見せた。



「ここに、いろんなブロックを置いてプログラムを作るよ。  
できたら、パソコンと<sup>マイクロ ビット</sup>micro:bitをつなげて  
『ダウンロード』ボタンを押すだけだ」

<sup>さんこう</sup>  
《参考》

プログラムの呼び出し方や、パソコン（タブレット）と  
<sup>マイクロ ビット</sup>micro:bitの接続方法はいろいろあるから、先生に聞いてね。



「これから使うブロックは、

『高度なブロック』メニューの中にかくれた

『入出力端子』メニューの中の

『デジタルで出力する』ブロックを使うからね」

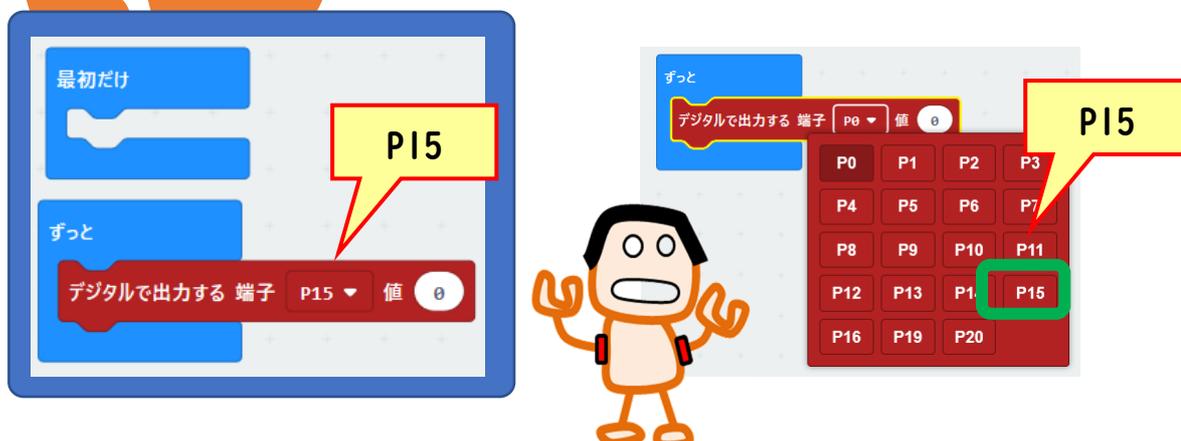


「じゃ。やってみるね。

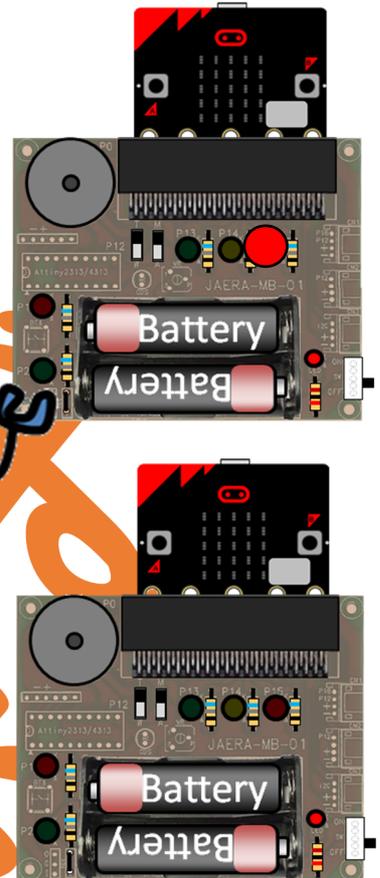
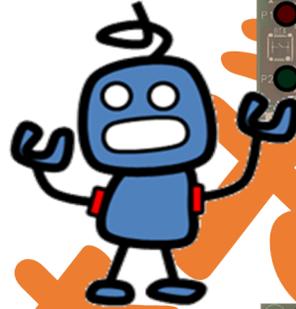
『ずっと』の中に『デジタルで出力する』ブロックを入れて

端子は「P15」にしてね。

プログラムができたなら『ダウンロード』ボタンを押してね」



「うわっ、赤のLEDが点いたよ！」  
ボクがおどろいていると  
ノブ君は  
「じゃあ、次は『値』を『1』にして  
『ダウンロード』ボタンを押してね」



「あれっ、赤のLEDが消えたよ！びっくりしたなあ」

《まとめ》赤のLEDを点灯する、消灯する命令

点灯する



消灯する



## 課題2

『端子』を、『P13』『P14』に変えてみよう。

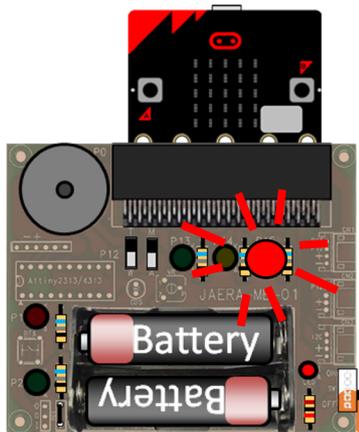
(結果)



### 3. LEDを点滅する

ボクがおどろいている間に、ノブ君の画面を見ると、ブロックが増えていた。  
何が始まるんだろう。

「じゃ『ダウンロード』ボタンを押すね」  
と言って、プログラムをmicro:bitに書き込んだ。

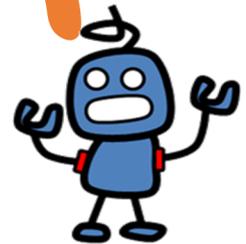


今度は、赤のLEDがピカピカと点滅した。

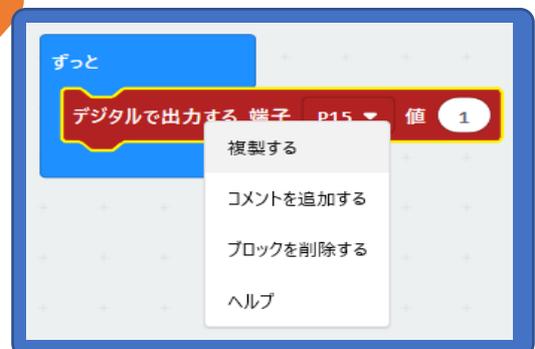
「すごいなあ」

「いったい、どうやったの」

ノブ君は、最初に戻して、ゆっくりと教えてくれた。



右クリックして、『複製する』を選ぶと、  
同じブロックが現われた。



《ヒント》  
パソコンは右クリック  
タブレットは長押し



「ブロックのコピーは右クリックするのか、なるほどね」

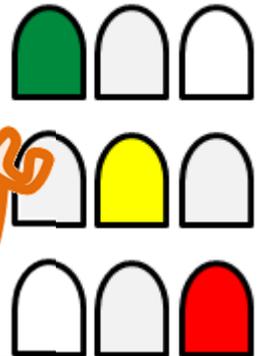


#### 4. LEDを順に点灯する

ノブ君がみんなにいつてきた。

「ここまでわかってきたら、青→黄→赤と順に点くのを  
繰り返すプログラムができるんじゃない」

P13 P14 P15



赤は P15 で、

青は【       】

黄は【       】だから....

しばらくして、「できた〜！」  
とボクは思わず叫んでいた。

「できたよ。できたよ」って  
みんなの周りを走り回って  
大きな声で何度も叫んだ。

とにかく、ボクが作ったプログラ  
ムを見せたかったんだ。

しかしこの後、ボクは『ダウンロード』ボタンを押して、恥ずかしい思いをす  
ることになるとは。

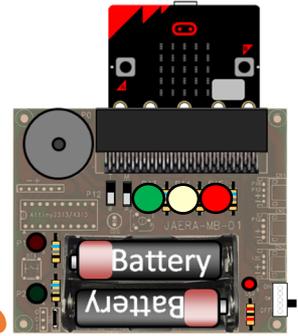
#### 課題4-1

なぜ、恥ずかしい思いをしたのだろう？



ボクがプログラミングした<sup>エンバット</sup>mbATを見ながら、ノブ君が言ってくれた。

「『失敗は成功のもと』ともいうじゃないか。失敗をしなけりゃ、成功もできないということだから、またがんばったらいんじゃない」



#### 課題4-2

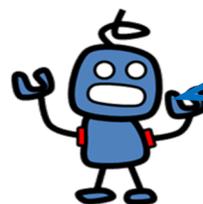
「青→黄→赤と順に点く<sup>つ</sup>」をくり返すプログラムを考えよう？



ようやく、できたぞ。

これで「青→黄→赤と順に点く<sup>つ</sup>の」をくり返すようになったよな。

あれっ、これって信号機に似ているなあ。



おーい、ノブ君！  
できたよ！

## 5. 信号機を作ろう

ノブ君が、ボクがかいた紙を出してきた。

「さっきできたプログラムを改造して  
信号機を再現してみよう  
まずは、おさらいから」



まあ17びょう  
さっき  
わが18びょう  
3びょう

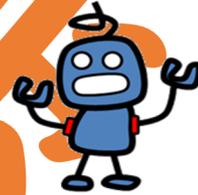
「一時停止は <sup>ま</sup>待つ という意味だったね。  
一時停止 100 ミリ秒 = 0.1 秒だけ待つ  
じゃあ、1 秒待つときは？」

一時停止 (ミリ秒) 100 ▼

「10 倍すればいいから、  
一時停止 【 1000 】 ミリ秒だよ。」



「じゃあ、17 秒待つなら？」



「さらに 17 倍すればいいから、  
一時停止 【 17000 】 ミリ秒だよ。」  
じゃあ、こうすればいいのか」



ずっと

デジタルで出力する 端子 P13 ▼ 値 0	
一時停止 (ミリ秒) 1000 ▼	17000
デジタルで出力する 端子 P13 ▼ 値 1	
デジタルで出力する 端子 P14 ▼ 値 0	
一時停止 (ミリ秒) 1000 ▼	3000
デジタルで出力する 端子 P14 ▼ 値 1	
デジタルで出力する 端子 P15 ▼ 値 0	
一時停止 (ミリ秒) 1000 ▼	18000
デジタルで出力する 端子 P15 ▼ 値 1	



《ヒント1》

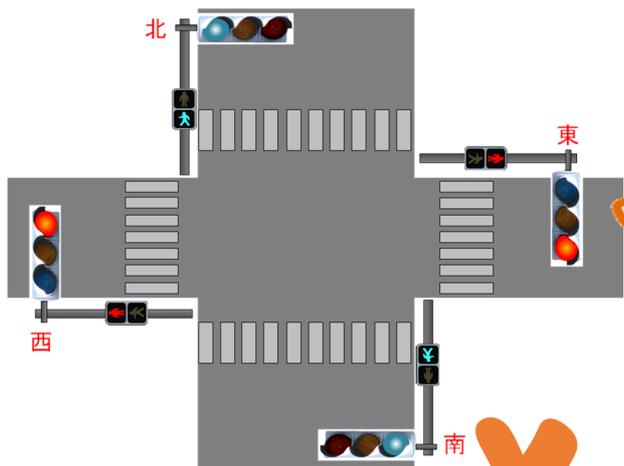
1メートル  
= 100センチメートル  
= 1000ミリメートル

《ヒント2》

プログラム作成中は、青  
と赤の点灯時間を短くし  
て試みましょう

ノブ君に一杯教えてもらって、何とかボクは東西方向の信号機を再現できた。

プログラムができたところで、ノブ君が言った。  
 「昨日、あの交差点に行って、信号機の点滅と時間をもっと詳しく調べてきたよ」と言って、1枚の紙を見せてくれた。



時間 (秒)	東西方向	南北方向
	車	車
17		
3		
2		
11		
3		
2		

「ほらね、ポンロボ君は、正確に調べていたんだね！」



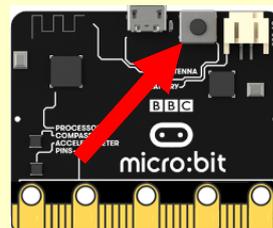
いや待てよ。ほんとかない  
 交差点の信号機は4台で一組だよな  
 「だれか、いっしょに作ろうよ〜」

**課題5** 2~4人グループになって、交差点を  
 実現しよう。

(感想)

《ヒント》

micro:bitの裏  
 にあるリセット  
 ボタンで、再ス  
 タートできるよ



## 第2章 プログラムのくふう

少しレベルが上がります。がんばってチャレンジしましょう！

### 1. 分かりやすいプログラム



ノブ君が教室に入ってきた。  
「この間はうまく行って、よかったね。」

「ちょっと、これを見てくれないかな」と言って、プログラムを見せてくれた。

「これで、前のプログラムと同じ動きをするんだよ」

前のプログラムは、何が起きるのは見てもわからなかった。でも、これは「青→黄→赤」となるのがよく分かるなあ。

「実は『ずっと』以外に、『関数 青』などのプログラムが別にあるんだよ」

「関数って、何だ？」ノブ君は図で説明し始めた。

「『関数』は小さなプログラムのかたまりで、呼び出すたびに同じことをしてくれるのさ」



「例えば『呼び出し 青』の命令で、『関数 青』にある命令をしてから、次の『一時停止』命令をするんだ」



さっそく、みんなで関数のあるプログラムをつくることにした。

『高度なブロック』の中の

- ① 『関数』メニュー
- ② 『関数を作成する』を選び
- ③ 関数名をつければ関数ができあがる。



### 課題1

『関数 青』『関数 黄』『関数 赤』を作ったためしてみよう。

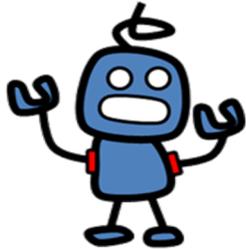
(プログラムと感想)



プログラムを分かりやすく作るとは大切なんだね。

## 2. 歩行者用信号機を作ろう

「実は・・・」とノブ君が言った。  
 「ついでに、歩行者用信号機も調べていたんだよ」



これもプログラムするんだね  
 関数はどうしたらいいの  
 かんすう  
 だろう？



時間 (秒)	東西方向	南北方向
	人	人
10		
7		
5		
6		
5		
5		

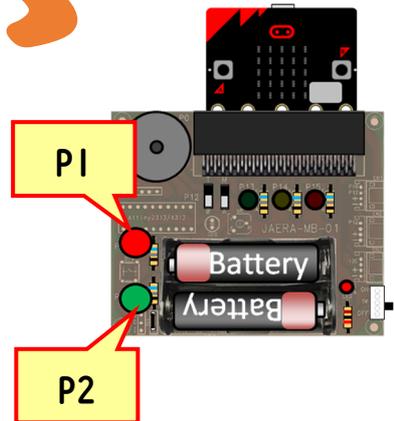
「こんなふうにしたらどうかな？」

The code is organized into two function blocks:

- 関数 歩行者 青 (Function: Pedestrian Green)**
  - 呼び出し 歩行者 青 (Call: Pedestrian Green)
  - 一時停止 (ミリ秒) 10000 (Pause: 10000 ms)
  - くりかえし 7 回 (Repeat: 7 times)
  - 呼び出し 歩行者 青点減 (Call: Pedestrian Green Point Decrease)
- 関数 歩行者 青点減 (Function: Pedestrian Green Point Decrease)**
  - デジタルで出力する 端子 P1 値 1 (Digital output: Pin P1, Value 1)
  - デジタルで出力する 端子 P2 値 0 (Digital output: Pin P2, Value 0)

Below these are two more function blocks:

- 関数 歩行者 赤 (Function: Pedestrian Red)**
  - 呼び出し 歩行者 赤 (Call: Pedestrian Red)
  - 一時停止 (ミリ秒) 21000 (Pause: 21000 ms)
- 関数 歩行者 赤点減 (Function: Pedestrian Red Point Decrease)**
  - デジタルで出力する 端子 P2 値 0 (Digital output: Pin P2, Value 0)
  - 一時停止 (ミリ秒) 500 (Pause: 500 ms)
  - デジタルで出力する 端子 P2 値 1 (Digital output: Pin P2, Value 1)



赤は【 】で、青は【 】だから...。「歩行者 青点減」と「歩行者 赤」の関数を作ればいいのかな？

### 課題2

歩行者信号機だけをプログラムしてみよう。



(感想)

### 3. カッコー音を鳴らそう

歩行者用信号機ができたけど、何かが物足りないなあ。そうだ、音が鳴っていたんだ。エンバットmbATでは音は出せないの？



『音楽』メニューの中のブロックでも音楽は鳴らせるけどもね。でも今回は『入出力端子』メニューの下

- ① 『その他』メニューの中の
- ② 『音を鳴らす』ブロックを使おう



**実習3-1** 次のプログラムを作成して、micro:bit にプログラムを書き込もう。

(結果)

急車のヒュー音がした

**実習3-2** カッコーのきおん擬音を鳴らしてみよう。

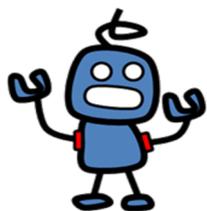


(感想)

**実習3-3** カッコー音も関数にしたプログラムを作ってみよう。



(感想)



あれっ？ もっと単純にできるよね  
「くりかえし」も関数「カッコー」の中に入れられるんじゃないの？

「ポンロボ君、すごいことに気付いたね！」



**課題3** くりかえしも、関数「カッコー」に入れてみよう。

(プログラムと感想)

**発展課題3-1** **課題2**の歩行者用信号機のプログラムで、青の時にカッコー音が鳴るようにしてみよう。



《ヒント》カッコー音は、歩行者信号が青のときの前半分ぐらいで鳴っているよ。

(感想)

**発展課題3-2** 車用と歩行者用信号機を合わせると、右のような表になった。プログラミングしてみよう。



《参考》カッコー音を入れると、もっと本物らしくなるよ！

(感想)

時間 (秒)	東西方向		南北方向	
	車	人	車	人
10				
7				
3				
2				
6				
5				
3				
2				

## 第3章 いろいろな信号機を作ろう

ここからは、かなりレベルが上がります。先生の指示に従ってチャレンジしましょう。



### 1. 短いプログラムのくふう



「この2つのプログラムは同じ  
はたらきをするプログラムだったね」



やっとボクもプログラムが分かるようになってきたよ。  
みんなはだいじょうぶかな？



「実は『ずっと』  
の中を単純にする  
方法があるんだよ」



ノブ君は、ボクにもう一つのプログラムを見せてくれた。

## 実習1

単純にした信号機のプログラムに改造してみよう。



《ヒント》関数に **num** を追加しよう。

①右クリック

②Edit Function

③Number

(感想)

## 発展課題1

右図のような右折信号機を持った交差点をプログラミングしよう。

《ヒント》

右折信号は、

『基本』メニュー

『LED画面に表示』

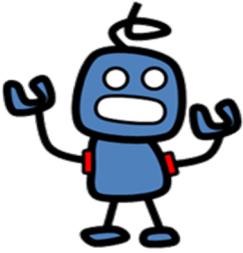
のブロックを使ってみよう。



	東西方向
	車
10秒	
3秒	
4秒	
3秒	
10秒	

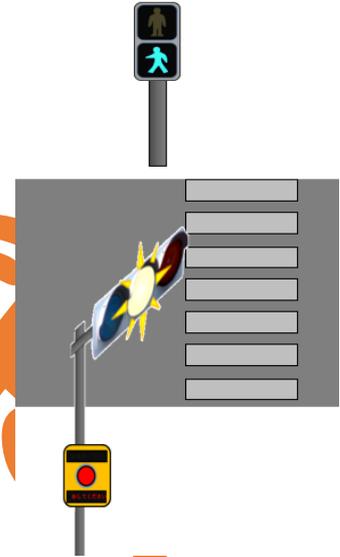
(感想)

## 2. 条件によって動作を変えよう

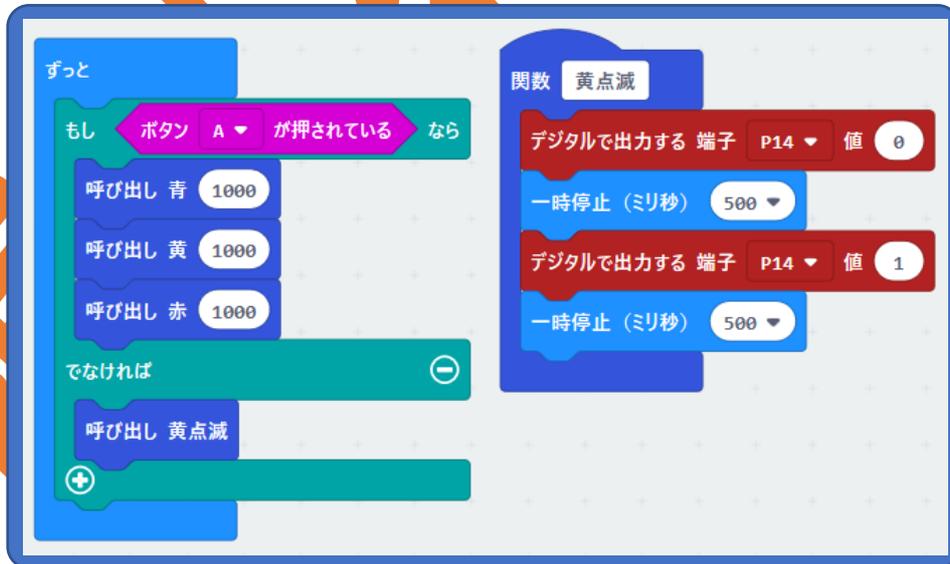


学校の前の横断歩道にある信号機は、黄色が点滅<sup>てんめつ</sup>しているね。でもボタンを押すと、青→黄→赤と変わっていくよ！

「こんなブロックを使ったら、条件に合うとき、合わないときの動作を変えることができるよ」

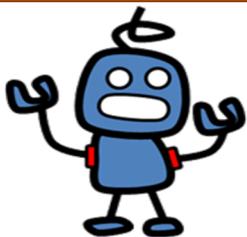


**実習2-1** 普段は黄色の点滅<sup>てんめつ</sup>信号が、Aボタンを押すと青→黄→赤と変わっていくプログラム（一部省略）を試してみよう。

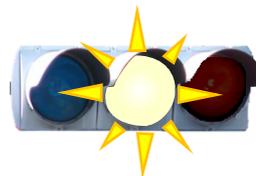


(結果と感想)

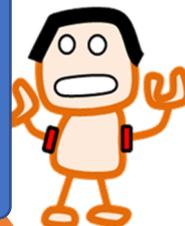
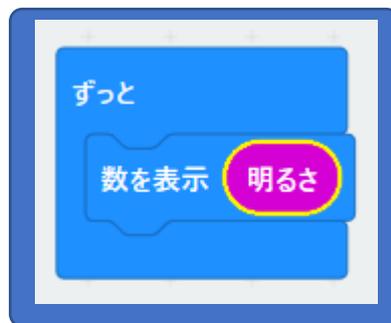
### 3. 明るさで変わるプログラム



『夜は黄色の点滅<sup>てんめつ</sup>信号なのに、昼間は普通の信号機があるよ』とお父さんが言っていたよ。



「ポンロボ君。そんな時は、明るさを感じるセンサを使えば、同じものが作れるよ」



**実習3-1** 右のプログラムを書き込み、明るさを計<sup>はか</sup>ってみよう。

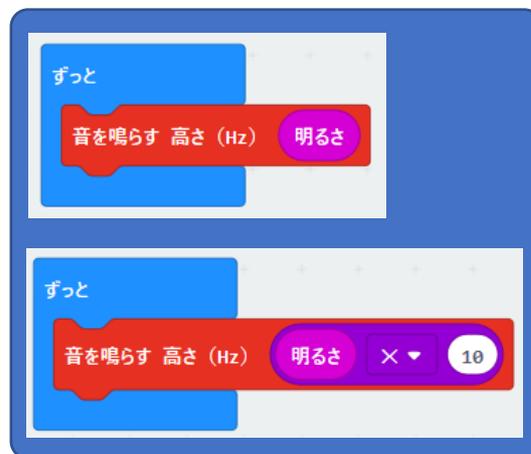
(結果と感想)

「『明るさ』を測定したとき、0～255の数値でわかるようになっていたんだ」



**発展課題3** つぎのプログラムを書き込み、動作をためしてみよう。

(結果と感想)



**実習3-2** 右のプログラムを micro:bit に書き込み、動作をためしてみよう。

(結果と感想)



### 課題3

夜は黄色の点滅信号<sup>てんめつ</sup>なのに、昼間は普通の信号機を作ってみよう。

(プログラム)

#### 4. 無線を使ったプログラム

「micro:bit って、お互いに無線を使って通信ができるんだよ。ちょっとやってみようか」



同じ数字（0～255）の人で通信ができます。



##### ① グループ分け

たくさん的人数だと分からなくなるので、2～4人ごとのグループに分かれよう。『最初だけ』の中の『無線のグループの設定』で、グループの中の出席番号の一番若い人の番号を入れることにしよう。

##### ② アルファベットや数字を送る

	送信できるもの	受信できるもの	備考
数字	数値を送信	received Number	半角数字のみ
文字	文字列を送信	received String	半角文字、数字のみ
変数	Name、数値を送信	Name、数値を受信	半角文字、数字のみ

#### 実習4

前のページのプログラムを micro:bit に書き込み、通信を試みよう。

(感想)



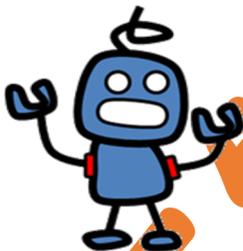
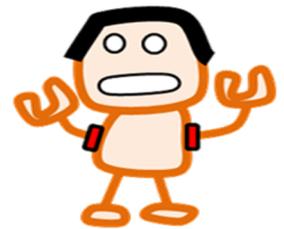
《ヒント》

receivedstring の値はドラッグ & ドロップでコピーできます。



#### 5. 信号機を同期する<sup>どうき</sup>

「無線を使えば、交差点の複数の信号機を確実に繋げることができるよ」とノブ君が言った。



?????????.....  
ボクにはさっぱりわからないよ～  
ノブ君、どうしたらいいんだい？

#### 実習5

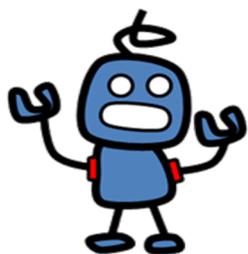
2台に、同じプログラムを入れて試してみよう。

(グループは変えましょう)

(感想)



「二つ以上の機械のタイミングをあわせることを同期<sup>どうき</sup>っていうんだよ。文字列の送信でタイミングを合わせているんだね」



そうか！

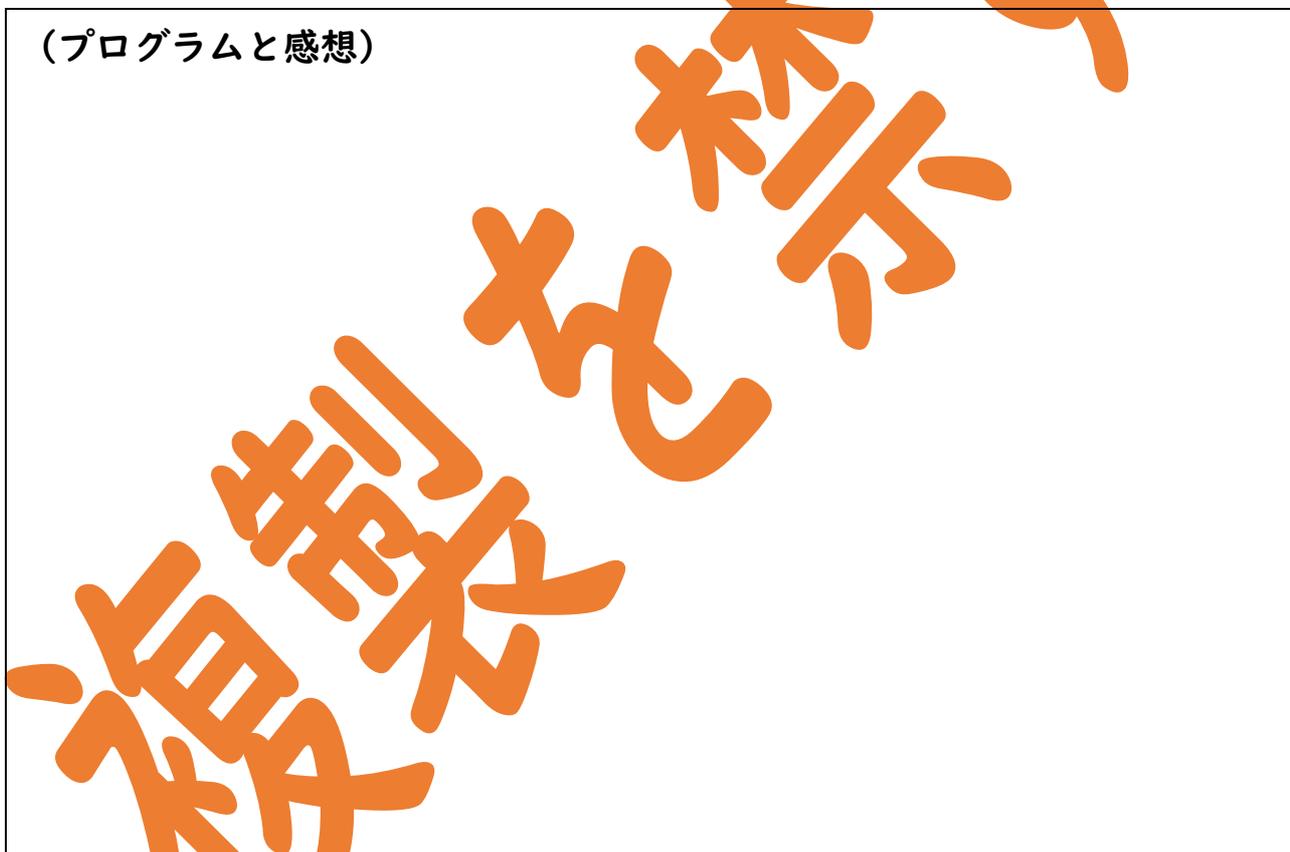
みんなで何かするとき、誰かが「セイノー」って声をかけてタイミングを同じにしているのによく似ているね。

つまり文字列の送信は、「セイノー」の役割<sup>やくわり</sup>りなんだ。

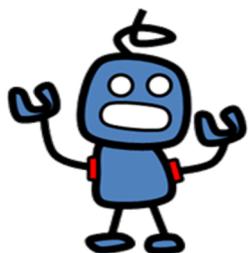
### 課題5

2～4台の mbAT を使って交差点の信号機を作ってみよう

(プログラムと感想)



「みんな、できたかい？」



ノブ君、ボクも少しずつプログラムが分かってきたよ。

ありがとうね！



## 《付録》 拡張機能

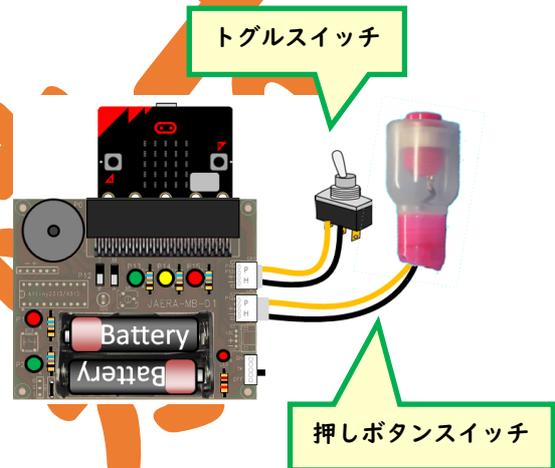
ここでは、mbAT に少しの部品を足すことでできることを紹介しましょう。



### 1. 外部スイッチを接続する

mbAT には様々なスイッチを接続することができます。

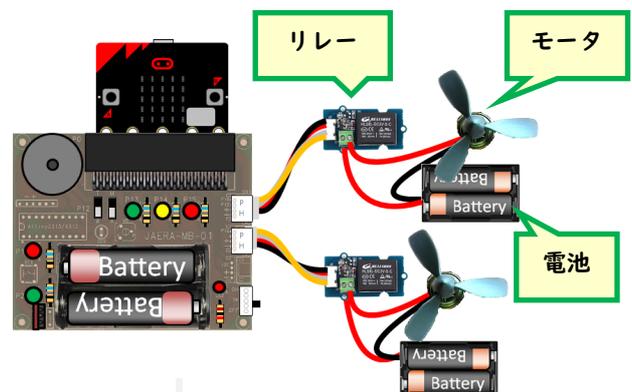
上につなぐと P16、下につなぐと P12 の値で条件分岐のプログラムを組みます。



### 2. リレーをつなげてモータなどを動かす

モータなどは micro:bit に直接つなげることはできません。リレーというスイッチのはたらきをする部品を使って動かします。

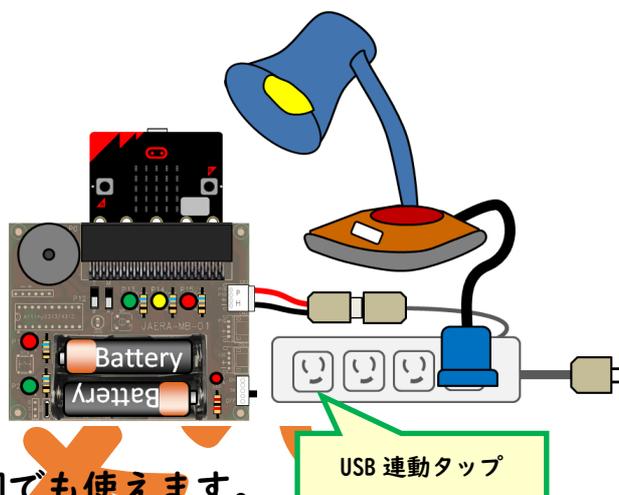
上につなぐと P16、下につなぐと P12 の値が「1」のとき動作します。



※外部スイッチ1個、リレー1個でも使えます。

### 3. 電化製品を動かす

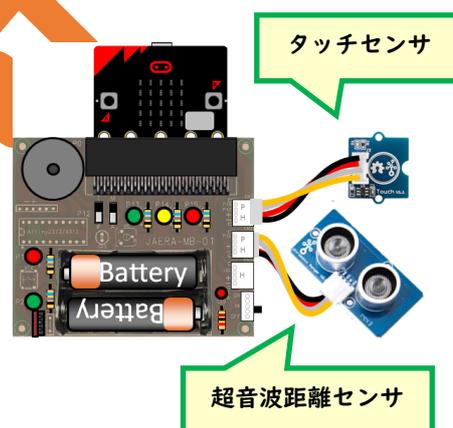
家庭の電化製品を動かすときは、USB 連動タップを使うと便利です。



※外部スイッチ1個、USB 連動タップ1個でも使えます。

### 4. タッチセンサや超音波距離センサを使う

センサによってはそのまま単純に使えるものと、少しくふうが必要なものがあります。



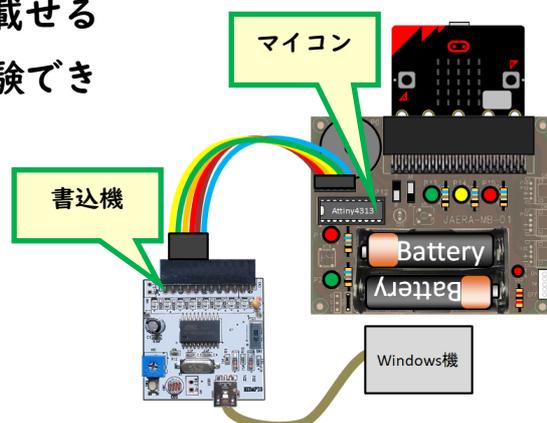
※詳しくはmbATのホームページをご覧ください)

### 5. 電子オルゴールを動かす

mbATには電子オルゴール用のマイコンを載せることができ、micro:bitで動かすことが体験できます。

P12の値が「1」のとき動作します。

※マイコンの曲も自分でプログラムできます。



# 目次

## 第1章 信号機を作ろう ..... 1

1. 身近な信号機を調べよう	1
2. LEDを点灯し、消灯する	5
3. LEDを点滅 <small>てんめつ</small> する	9
4. LEDを順に点灯する	11
5. 信号機を作ろう	13

## 第2章 プログラムのくふう ..... 15

1. 分かりやすいプログラム	15
2. 歩行者用信号機を作ろう	17
3. カッコ音を鳴らそう	18

## 第3章 いろんな信号機を作ろう ..... 21

1. 短いプログラムのくふう	21
2. 条件によって動作を変えよう	23
4. 無線を使ったプログラム	26

## 《付録》拡張機能 ..... 29

**JAERA**

ポンコツロボットと信号機を動かそう！  
micro:bit/mbAT 版

著作：浅田寿展

発行者：JA 教育研究会

【本書の無断転載を禁ず】

定価 330 円（税込）JAMBAT32R01A