

マイコンのI/O制御からAI画像認識まで!好きなブロックを追加して、みんなにシェアして使ってもらおう!

# ビジュアル・プログラミング環境の拡張&カスタマイズの世界

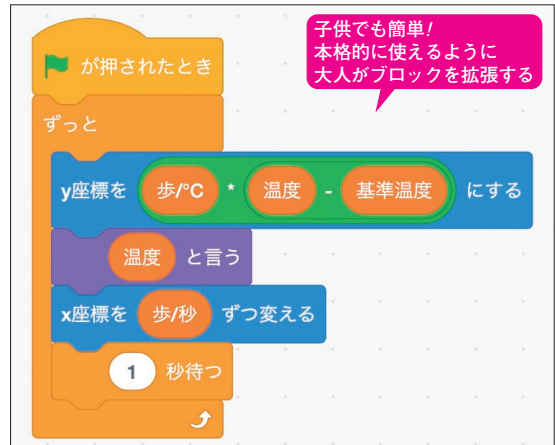
ご購入はこちら

横川 耕二

```

if (var1 == 0)
{
  out.pressure = 0;
}
else
{
  uint32_t p = (uint32_t)((1048576 - adc_P) - (var2 >> 12)) * 3125;
  if (p < 0x80000000)
  {
    p = (p << 1) / ((uint32_t)var1);
  }
  else
  {
    p = (p / (uint32_t)var1) * 2;
  }
  const int32_t
  dig_P7 = this->dig_P7,
  dig_P8 = this->dig_P8,
  dig_P9 = this->dig_P9;
  var1 = dig_P9 * (int32_t)((p >> 3) * (p >> 3) >> 13) >> 12;
  var2 = (int32_t)(p >> 2) * dig_P8 >> 13;
  out.pressure = (uint32_t)((int32_t)p + ((var1 + var2 + dig_P7) >> 4));
}
    
```

自由度は高いけれど一般の方や子供には敷居が高い



子供でも簡単!本格的に使えるように大人がブロックを拡張する

(a) テキストによるプログラム(C++)

(b) ビジュアル・プログラミング言語(Scratch)

図1 プログラムの表現にテキストを使わないのがビジュアル・プログラミング言語

テキストによるプログラミングは表現の自由度が高い反面、規則に則って正確に入力することが難しく、初学者や子どもには習得が難しい。ビジュアル・プログラミングは表現の自由度をある程度犠牲にする代わりに、ささいなミスで動かなくなることが少ない

## ビジュアル・プログラミングとは何か

### ● テキスト・プログラミングについて考えてみる

#### ▶ メリット…表現の自由度が高い

テキスト、特に高級プログラミング言語によるプログラミングは、文字の組み合わせが多様であるために図1(a)のように表現の自由度が高くなります。コンピュータ自体の開発から科学や金銭の計算、文書処理、インターネット・サービスの開発など、いろいろな特徴を持ったプログラミング言語がテキストによって開発され、便利に利用されています。

#### ▶ デメリット…習得が難しい

ところが、文字には莫大な数の組み合わせがあるので、言語仕様によって決められた単語や自分で定義した言葉などを簡単に打ち間違えます。言語によって異なる言葉の順番や、記号の役割などを覚るのが大変だったり、知っていてもうっかり間違えたりしてしまいます。

テキストによるプログラミングは、その表現の自由度が高いことが逆に作用して、初学者やキーボードの操作が難しい子どもなどにとっては、テキストを規則に則って正確に入力することが大きな壁となります。コンピュータにやらせたい計算やアルゴリズムなどを考える以前の問題です。

### ● ビジュアル・プログラミングの特徴

#### ▶ 文字を書かなくてもアルゴリズムを記述できる

テキストではなく、図1(b)のようにディスプレイ画面上に表示されたブロックやアイコンなどを組み合わせることでプログラムするプログラミング言語をビジュアル・プログラミング言語と呼びます。ボタンや入力フィールドをマウス・ポインタで貼り付けて画面を構成できるプログラミング環境はいろいろありますが、ビジュアル・プログラミング言語はそれとも異なります。計算やアルゴリズム記述も文字を書くことなく行えるようにしています。