

はじめよう/ ジブン専用パソコン

監修・原案／青山学院大学客員教授 阿部和広
構成・文／塩野祐樹

カチッ



第14回

ジブン専用楽器を改造して、 録音再生機能をつけよう

ゲーム

工作

プログラ
ミング

スタディーノを使ったジブン専用楽器づくりの2回目だ。みんな、ガンガン演奏してくれているかな？今回は演奏に加えて、自分が演奏した曲を録音して再生できるようにするぞ！



購入ページ
QRコード

スタディーノはラズパイがなくても、普通のパソコンでも遊べるよ。

Scratchベースで動かそう！
Studuinoでラクラク電子工作

<http://prog.kodomonokagaku.com/studuino.html>

←キットの情報はKoKa Shop!へ。定期購読者割引あり

ジブン専用楽器の準備

まずは前回つくったジブン専用楽器を手元に用意しよう。まだつくっていない人は、バックナンバーの記事を見ながらつくってね。分解や改造をした人は元に戻しておこう。

ジブン専用楽器の準備ができれば、ラズパイとスタディーノをUSBケーブルでつないで、ブロックプログラミング環境(BPE)を起動しよう。前回つくったプログラムを保存していた人は、「ファイル」メニューの「開く...」から開いてね。

消しちゃった人は、バックナンバーを見ながらがんばってもう一度入力だ。

プログラムを読み込んだ場合は「入出力設定」も自動的にできているけど、入力し直した場合はこちらもやり直し。赤外線フォトリフレクタを「A7」、ブザーを「A0」に設定しよう。

前回の記事はココで読めるよ。

prog.kodomonokagaku.com/jibun/

プログラムの入力が終わったら、ちゃんと動かすか確認しよう。「実行」メニューの「テストモード開始」でテストモードに切り替えたら、緑の旗をクリック。つまみのブロックを動かしながら「A1」ボタンを押して、音が出るか確認する。うまく音が出ない人は、配線とプログラムを確認しよう。



録音のしくみとリスト

前回のプログラムは、「A1」ボタンを押しているときにつまみの位置(赤外線フォトリフレクタの値)を調べ、それに応じた音程(MIDIノート番号)の音が出るようになっている。ということは、鳴らした音のノート番号を順番に記録していけば、録音できるんじゃないかな。

そのためには、記録するための場所をつくっておかないといけないね。そこで使うのが「リスト」だ。

プログラミングをやっている人なら、「変数」を使えばいいんじゃないの? って思ったかもしれないね。でも変数には1つの値しか記録できない。今回のようにいくつもの値を記録したいときには、リストを使うのが便利なんだ。リストは、一般的なプログラミング言語で「配列」と呼ばれているものと同じだよ。

これが 録音再生機能付き ジブン専用楽器だ

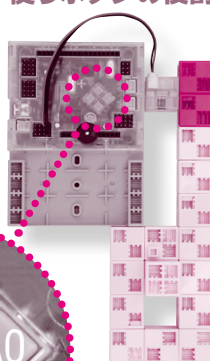
A0 今回は使わない

A1 演奏/
記録消去ボタン

A2 録音開始ボタン

A3 録音停止/再生ボタン

今回のプログラムで
使うボタンの役割



さっそくリストを
つくってみよう。

「**変数**」カテゴリーの「**リストを作る**」ボタンをクリックする(①)。リストの名前を聞かれるので、「**score**」と入力して「OK」ボタンをクリックだ(②)。そうすると、リスト「**score**」を操作するためのブロックが自動的に表示されるよ(③)。

このブロックの中にある「**(0)を(score▼)に追加する**」を使うと、リストに値を追加することができる。「0」を「60」に変えて、「**ブザー(A0▼)から(60▼)を出力する**」の下に入れば、音を出すのと同時にリストに記録できるぞ。



変数

リスト



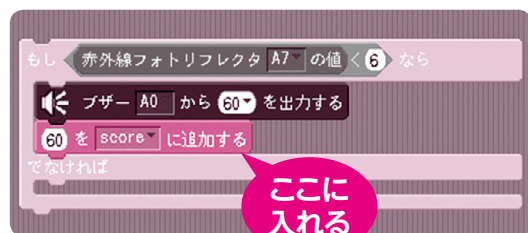
変数名 : **note**
内容 : 60

リスト名 : **score**
内容 : 60、62、64……

変数はいろいろな値を入れられる箱のようなものだ。1つの変数には1つの値しか入れられない。

リストはたくさんの変数がつながったようなものだ。たくさん値に番号をつけて記録することができる。

scoreには「得点」という意味もあるけど、ここでは「楽譜」という意味だよ。



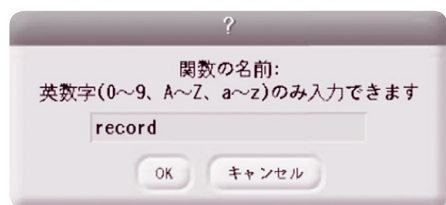
同じようにして、他の「ブザー (A0▼) から (～▼) を出力する」の下にも「(～) を (score▼) に追加する」を入れていこう。音が出ていないことも記録しないといけないから、「ブザー (A0▼) の出力を停止する」の下には「(0) を (score▼) に追加する」を入れよう。

ずっと記録し続けるわけにはいけないので、記録を止める命令も必要だ。「制御スタート」のすぐ下にある「ずっと」を、「～まで繰り返す」に変えよう。そして「(ボタン (A3▼) の値) = (0)」まで繰り返すようにしておく。

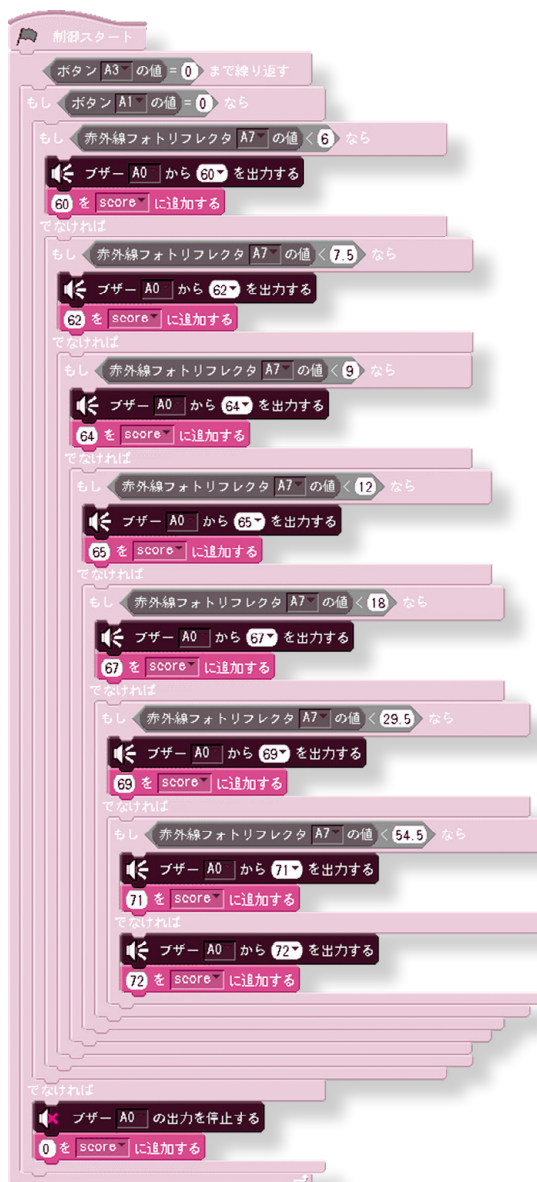
これで、「A3」ボタンが押されると記録が止まるというわけだ。ここまでのプログラムは右のような感じだよ。

今回は、録音以外にも再生と削除の機能もつくるので、それぞれを関数にしておこう。

「制御」カテゴリーにある「▼関数」の「▼」をクリックして、「新規...」で名前をつけよう。記録する関数だから、名前は「record」でいいかな。



「OK」ボタンをクリックすると「record▼関数」ができるので、「制御スタート」の代わりに取り付けよう。



再生のしくみをつくる

録音 (記録) はできたから、次は再生だね。これはリスト「score」に入っているノート番号を順番に鳴らせばよさそうだ。リストの番号順に鳴らすために、変数「index」をつくっておく。

「変数」カテゴリーの「新しい変数を作る」をクリックして、変数名に「index」と入力しよう。再生のプログラムは、関数「play」としてつくるよ。

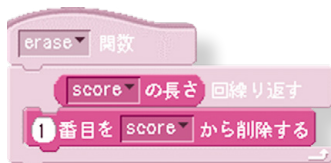
リスト「score」に記録されているノート番号の音を、順番に鳴らしているのがわかるかな。変数「index」は0から始まって、繰り返すた



びに1ずつ増えていくので、リスト「score」の内容を順番に指していくことができるんだ。ノート番号が「0」のときは音を止めているね。

楽譜を空にする

記録したものを消さないと無限にリストが大きくなってしまいますので、リストを空にする（消去する）処理も必要だ。「消す」という意味の「erase」という名前関数をつくろう。



「(score▼)の長さ」の回数分、「(1)番目を(score▼)から削除する」を繰り返しているね。これだと1番目しか消えないような気がするけど、1番目が消えると2番目が1番目に繰り上がるから、結果として全部消えるんだ。

全体をまとめるプログラム

これで、録音する関数「record」、再生する関数「play」、消去する関数「erase」ができたので、これらの関数を制御するためのプログラムを「制御スタート」の下につくろう。ここでは、「A2」ボタンで録音、「A3」ボタンで再生、「A1」ボタンで消去するようにしたよ。

「A1」ボタンを、演奏と消去の両方で使っているから不思議に感じるかもしれないけれど、演奏と消去は同時に動作しないようになっているので大丈夫なんだ。よ〜く流れを見てみてね。

また、リスト「score」が大きくなると、「erase」関数を呼び出した後の消去に時間がかかり、処理中なのかわからなくなってしまう。

そこで、消去が終わったら変数「index」を「0」にするようにした。こうしておけば、ステージに表示されている「index」の値が「0」になったことで、処理が終わったことを確認できるよ。

完成したら、さっそく演奏を録音してみよう。完成したら、さっそく演奏を録音してみよう。緑の旗をクリックしてから、「A2」ボタンを押せば録音が始まる。ブロックのつまみを動かしながら、「A1」ボタンで演奏して、終わったら「A3」ボタンで録音をストップ。



もう一度「A3」ボタンを押せば再生されるぞ。どう、ちゃんと聞こえた？「A3」ボタンを押せば何度でも再生できるよ。いい曲ができれば、他の人にも聞かせてあげよう。名前をつけてプログラムを保存すれば、曲も保存されるよ。

録音し直す場合は、「A1」ボタンで消去してから、もう一度「A2」ボタンを押そう。

今回のプログラムでは、サイズを変えながら大きなリストを使っているの、残念ながらスタディーノに転送してラズパイから切り離すことはできない。テストモードでだけ使うことができるプログラムだよ。

ラズパイとスタディーノとの連携は今月号でひとまず終わり。次号からは、文書作成や表計算などができる LibreOffice の使い方を紹介していくよ。