

# コンピュータワールド(夏)

スクイークのプログラムでレスキュークローラーを動かそう



平成 xx 年 xx 月 xx 日

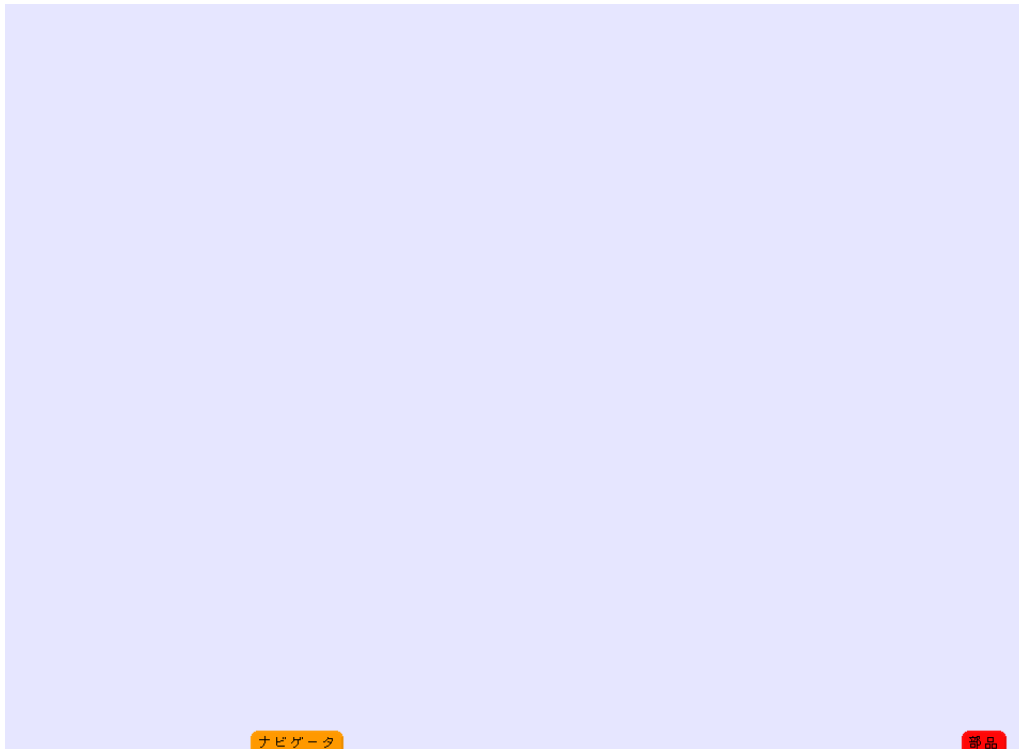
岐阜工業高等専門学校  
電気情報工学科

## 今回の目的

スクイークのプログラムでレスキュークローラーを動かしてみましよう。

### 1 スクイークの基本画面

スクイークを起動させると、下のような画面になります。




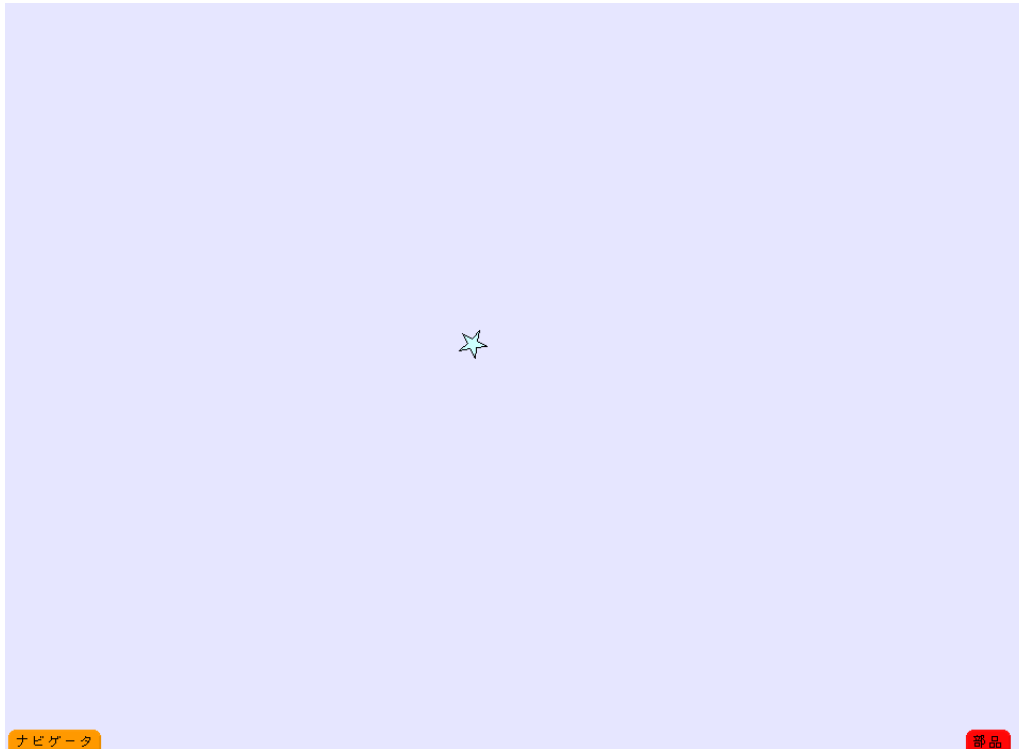
### 2 部品の出し方

まずは、スクイークを使ってみましよう。

画面右下の**部品**の部分をクリックすると、いろいろな部品が入ったバーがあらわれます。



その中から  (星) をマウスの左ボタンをクリックしたまま動かし、上部の無地のところでボタンを離して下さい。すると、星がその場所に残っています。



### 3 八口

星にマウスカースルを合わせて、中ボタン（ホイール）を押すと、周囲に があらわれます。これらのボタンのことを「八口」と呼びます。



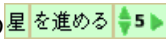
### 4 ビューワー

では、星の左側にある水色の八口 をクリックすると、右側に「ビューワー」があらわれます。

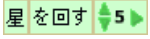


ビューワーでは、部品の状態を調べたり、部品を動かしたりできます。

## 5 やってみましょう

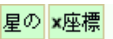

ビューワーの中の「星を進める」と書いてあるところの左側にある「!を何回かクリックしてみましょう。星はどうなりましたか？



星は（上に動いた・止まったまま・下に動いた）。


今度は、「星を回す」と書いてあるところの左側にある「!を何回かクリックしてみましょう。星はどうなりましたか？

星は（左に回った・止まったまま・右に回った）。

こんどは、直接をクリックして動かしてみましょう。もう一度クリックするまで、マウスと一緒にが動きます。このとき、「星の x座標」と「星の y座標」は、どのように変化しますか？

左に移動させるほど、「星の x座標」の値が（増える・減る）、  
上に移動させるほど、「星の y座標」の値が（増える・減る）。

八口を出し、左下の回転マークをクリックしたままで動かすと、星を回転させることができます。「星の 向き」の値がどのように変化するか見てみましょう。

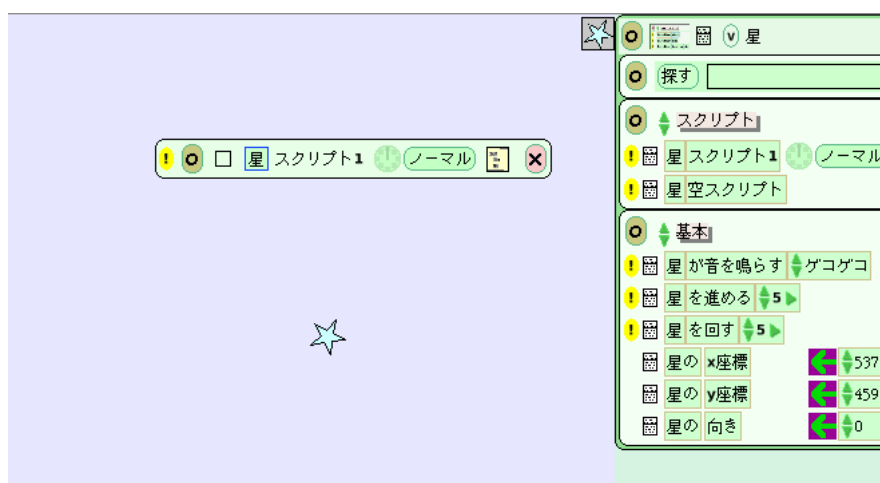
逆に、「星の x座標」に数値を入れ、リターンキーあるいはエンターキーを押すと、星が指定された位置へ瞬時に移動します。

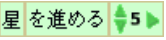

八口を出し、「x」のマークをクリックすると部品を捨てることができます。まだ捨てないで下さい。

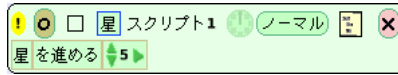
## 6 スクリプト

ビューワーを使えば、星を動かすことができました。つぎは、自動的に動かしてみましょう。これには「スクリプト」を作る必要があります。

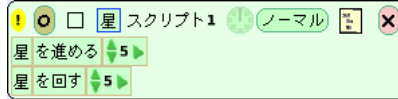
では、「星空スクリプト」をクリックして下さい。すると、マウスカーソルに「星空スクリプト」がついてきます。このままビューワーの外まで持って行って、クリックして下さい。すると、「星空スクリプト」がマウスカーソルからはなれて、下のよう新しい「スクリプト」が用意されます。



このスクリプトに星を動かすタイルを追加してみましょう。ビューワーの中の「星を進める」をクリックし、スクリプトのところまでもって行って「緑色のスペース」が現れたのを確かめて、もう一度クリックしてタイルを離して下さい。すると、「星を進める」が先ほど緑色だったスペースに収まります。



さらに、ビューワーの中の星を回す 5 を先ほどと同様に、スクリプトの星を進める 5 の下へ入れて下さい。



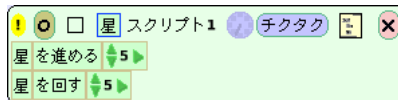
このスクリプトは「星を5進め、5回す」という意味になります。

スクリプトを動かすには、ビューワーのときと同じように、スクリプトの左側の黄色い雷マークをクリックします。

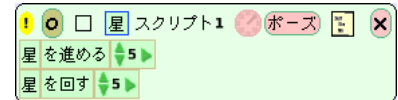
このスクリプトを連続して動かすと、星はどのように動くでしょうか？

あなたの予想： 星は 。

スクリプトの緑色の部分をクリックしてみてください。緑色の右の文字が「ノーマル」から「チクタク」に変わり、星が動きはじめます。

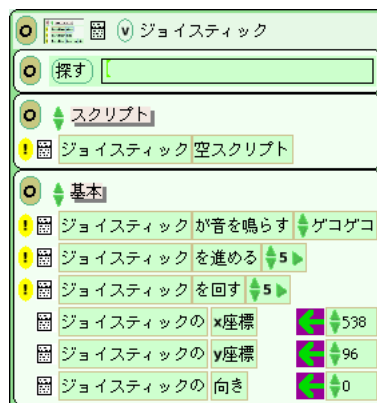


青い部分をもう一度クリックすると「チクタク」が「ポーズ」に変わり、動きが止まります。星は予想通りのうごきをしましたか？

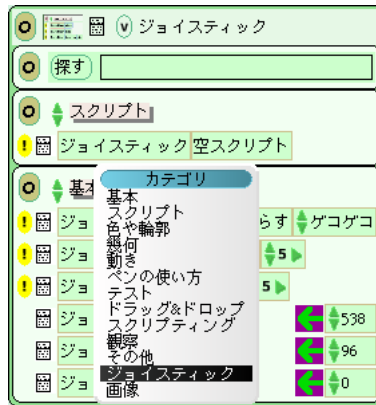


## 7 ジョイスティックを使う

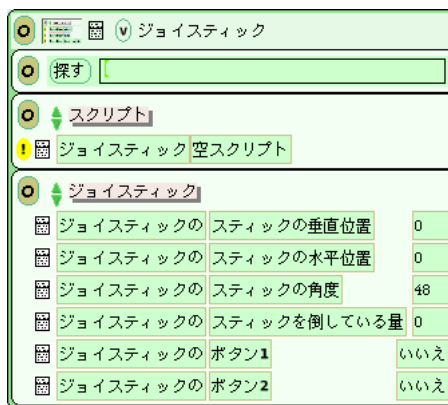
先ほどのスクリプトは、決まった動きをするものでした。次は、星を自由に動かしてみましょ。部品をクリックし部品の中から、ジョイスティックをマウスの左ボタンをクリックしたまま動かす、上部の無地のところでボタンを離して下さい。八口を出し、ジョイスティックをクリックして、ビューワーを開きましょう。




ビューワーの「基本」をクリックすると、下のようなメニューがでてきますので「ジョイスティック」を選びクリックします。




ビューワーにジョイスティック特有の状態が表示されるようになります。

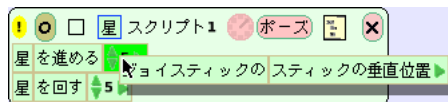



マウスで  の中心の赤丸をクリックしたまま動かして、**ジョイスティックのスティックの垂直位置** 0 と **ジョイスティックのスティックの水平位置** 0 の数値がどのように変わるか見てみましょう。

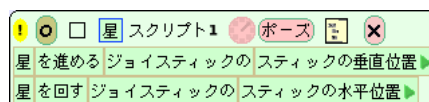
- ・赤丸を四角の上端まで動かすと、**ジョイスティックのスティックの垂直位置** 0 の数値は
- ・赤丸を四角の下端まで動かすと、**ジョイスティックのスティックの垂直位置** 0 の数値は
- ・赤丸を四角の右端まで動かすと、**ジョイスティックのスティックの水平位置** 0 の数値は
- ・赤丸を四角の左端まで動かすと、**ジョイスティックのスティックの水平位置** 0 の数値は



この数値を利用して、星を進める量と回す量をきめましょう。

まず、ビューワーの中の **ジョイスティックのスティックの垂直位置** 0 をクリックし、スクリプトの **星を進める**  5 の数字のところへ持って行って下さい。すると、数字の部分が緑色になります。これを確かめて、タイルを離します。



同じように、ビューワーの中の **ジョイスティックのスティックの水平位置** 0 をクリックし、スクリプトの **星を回す**  5 の数字のところへ入れて下さい。






それでは、このスクリプトを再度動かしてみましょ。スクリプトのをクリックすると、の右の文字が「ポーズ」から「チクタク」に変わり、スクリプトが連続して動きます。

このままでは、星は動きません。星の動く量はジョイスティックの位置によって決まり、ジョイスティックの赤丸が中央にあるからです。

さあ、マウスでジョイスティックを動かし、星を操縦してみましょ。

\* \* \* \* \*

思ったように動かせましたか？ これで、スクイークのお話は終わりです。最後に、とのハ口を出し、のマークをクリックして捨てて下さい。

## 8 レスキュークローラー

レスキュークローラーに取り付けられているマイコンは通信線 (RS-232C) から送られてくる命令に従って動くようになっています。

つまり、スクイークから通信線を使ってマイコンに命令を送ることにより、スクイークでレスキュークローラーを操縦するスクリプトを作ることができます。

まず、どのような命令があるか、また命令がどのような仕組みになっているかを理解しましょ。

## 9 モータの制御

レスキュークローラーには、左のキャタピラを動かすモーター、右のキャタピラを動かすモーター、アームを動かすモーターの3つのモーターがあります。このモーターを動かすには、次のような命令を用います。

命令： 

0	1	1	c1	c2	n1	n2	n3
---	---	---	----	----	----	----	----

8つのマス目があり、各マス目には0か1が入ります。つまり、「8桁の2進数」になります。

命令の中の c1 と c2 は、モータをどのように動かすか、すなわちモータの動作により下表のように決まっています。例えば、正転 (キャタピラなら前へ進む) の場合は、c1 のところに1を、c2 のところに0を当てはめます。

モータの動作

c1	c2	モータの動作
0	0	停止
1	0	正転
0	1	逆転
1	1	ブレーキ

命令の中の n1, n2, n3 により、どのモータを制御するか、すなわちモータの指定を行ないます。モータの指定は下表のように決まっています。例えば、左キャタピラの場合は、n1 のところに0を、n2 のところに0を、n3 のところに1を当てはめます。

モータの指定

n1	n2	n3	モータ
0	0	1	左キャタピラ
0	1	0	右キャタピラ
0	1	1	アーム

左キャタピラを正転（前進）させる命令は、次のようになります。

0	1	1	1	0	0	0	1
---	---	---	---	---	---	---	---

それでは、次のような動作をさせる命令は、どのようになるでしょうか？

- ・右キャタピラを正転させるには？ 

--	--	--	--	--	--	--	--
- ・左キャタピラにブレーキをかけるには？ 

--	--	--	--	--	--	--	--
- ・右キャタピラを逆転させるには？ 

--	--	--	--	--	--	--	--

## 10 光センサーの計測

レスキュークローラの底には、光センサーが3つついています。これにより地面の明るさを測ることができます。この光センサーで明るさを測るには、次のような命令を用います。この命令を送ると、マイコンは指定された光センサーで測り、スクイークに測った数値を送り返してくれます。

命令： 

0	1	0	0	0	s1	s2	s3
---	---	---	---	---	----	----	----

命令の中の s1, s2, s3 により、どのセンサーを使うか、すなわち光センサーの指定を行ないます。光センサーの指定は下表のようになっています。例えば、左の光センサーで測りたい場合は、s1 のところに 0 を、s2 のところに 0 を、s3 のところに 0 を当てはめます。

s1	s2	s3	光センサー
0	0	0	左
0	0	1	中央
0	1	0	右

それでは、次のような動作をさせる命令は、どのようになるでしょうか？

- ・左の光センサーで測るには？ 

--	--	--	--	--	--	--	--
- ・右の光センサーで測るには？ 

--	--	--	--	--	--	--	--

## 11 スクイークから命令を送るには

スクイークから命令を送るには、命令を数値に直す必要があります。つまり、命令は 2 進数でできていますので、2 進数を普通我々が使っている 10 進数に直せば良いのです。

10 進数の 1357 は、1000 の位に 1、100 の位に 3、10 の位に 5、1 の位に 7 がありますから、

$$\begin{aligned}
 1357 &= 1 \times 1000 + 3 \times 100 + 5 \times 10 + 7 \times 1 \\
 &= 1 \times 10^3 + 3 \times 10^2 + 5 \times 10^1 + 7 \times 10^0
 \end{aligned}$$

という意味があります。2 進数ではこの 10 のところが 2 になります。

つまり、キャタピラを正転させる命令 01110001 は、 $2^7 (= 128)$  の位に 0、 $2^6 (= 64)$  の位に 1 のようになり、下図のようになります。

0	1	1	1	0	0	0	1
---	---	---	---	---	---	---	---

128 64 32 16 8 4 2 1 の位





つまり,

$$\begin{aligned} 01110001 &= 0 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 64 + 32 + 16 + 1 \\ &= 113 \end{aligned}$$

となるのです。

この計算をするのは、大変ですね。

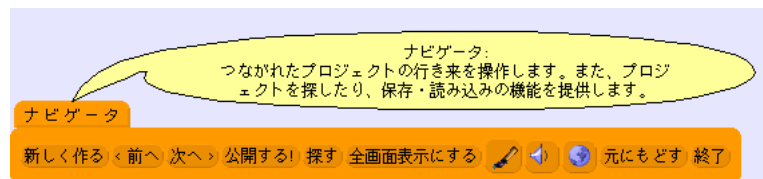
でも大丈夫。

計算はスクイクにまかせましょう。数値を入力するところで、のように、数値の最初に2進数であること表す2rをつけて入力して、リターンキーを押しましょう。すると、のように自動的に10進数に変わります。

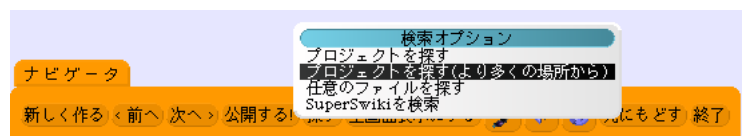
## 12 レスキュークローラーを動かそう

それでは、いよいよ、スクイクを使ってレスキュークローラーを動かしましょう。レスキュークローラーに命令を送るためには通信線 (RS-232C) を利用することになります。これを利用し、レスキュークローラーに命令を送る部品が用意してあります。

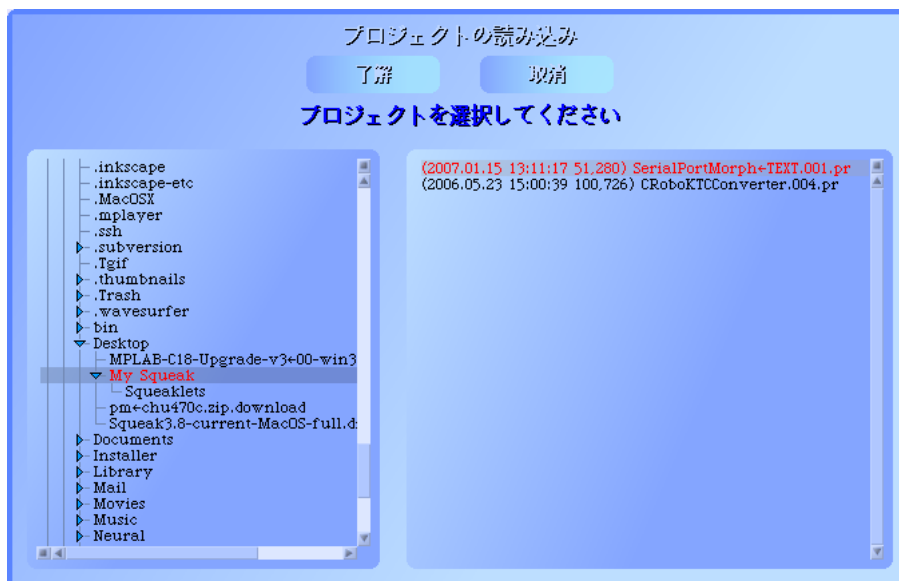
画面右下の **ナビゲータ** の部分を左クリックすると、いろいろなメニューが入ったバーがあらわれます。



「探す」を『長押し』して、検索オプションを出し、「プロジェクトを探す (より多くの場所から)」をクリックして下さい。



ファイルの一覧の中から、目的のプロジェクトのファイルをクリックして赤色に変え、「了解」をクリックします。

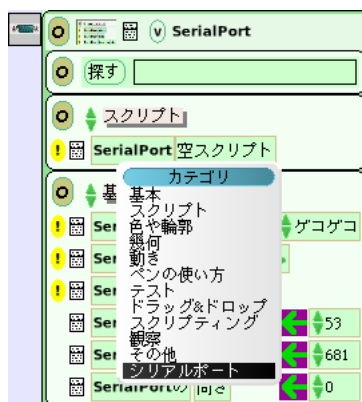


プロジェクトが開き、左上に SerialPort が用意されています。

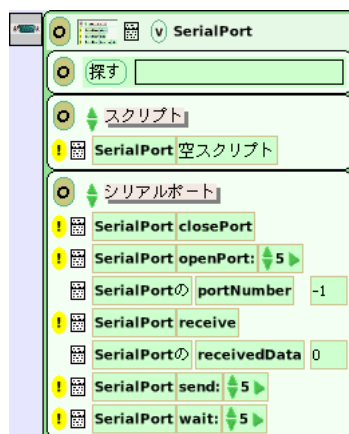


ハロを出し、ビューワを表示させましょう。

ビューワの「基本」をクリックし、「シリアルポート」を選びクリックします。



ビューワにシリアルポート特有の操作を行なうタイルや状態を表すタイルが表示されるようになります。



この状態では、レスキュークローラーに命令を送るための通信ポートが閉じています。これは、ビューワーの中の、**SerialPort**の **portNumber** **-1** をみると数値が **-1** になっていることからわかります。

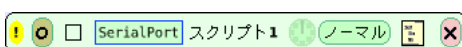
通信の準備をするには、ビューワーの中の **SerialPort openPort: 1** の数値を、通信に使用するポートの番号に変えてから、左側にある **!** を 1 回だけクリックして下さい。

すると、**SerialPort**の **portNumber** **1** となり、**-1** は通信に使用するポートの番号に変わり、通信の準備ができたことがわかります。

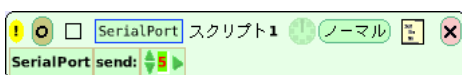
### 13 まずは前進

まず最初に、レスキュークローラーを前進させるスクリプトを作りましょう。

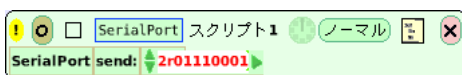
では、**SerialPort** 空スクリプト をクリックしたのち、ビューワーの外まで持って行って、クリックして下さい。**SerialPort** 空スクリプト がマウスカーソルからはなれて、新しいスクリプトが用意されます。



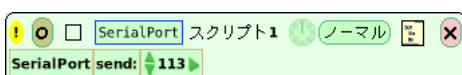
レスキュークローラーに命令を送るには、**SerialPort send: 5** を使います。ビューワーの中の **SerialPort send: 5** をクリックし、スクリプトの中に入れて下さい。



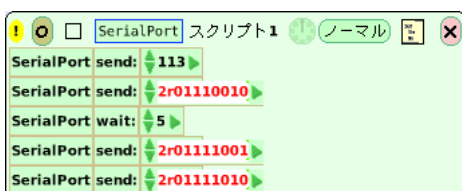
レスキュークローラーを前進させるため、まず、左キャタピラを正転させましょう。左キャタピラ正転の命令は「01110001」でしたね。スクリプトの中に入れた **SerialPort send: 5** の数値部分をクリックしてください。上の図のように、数値が赤色に変わります。左キャタピラ正転の命令を入力しましょう。



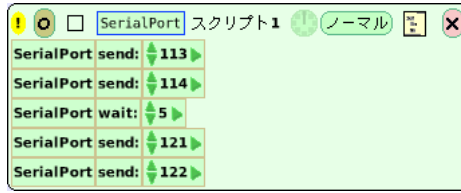
最後にリターンキーを押すと、入力された 2 進数は 10 進数に変換され、下の図のようになります。



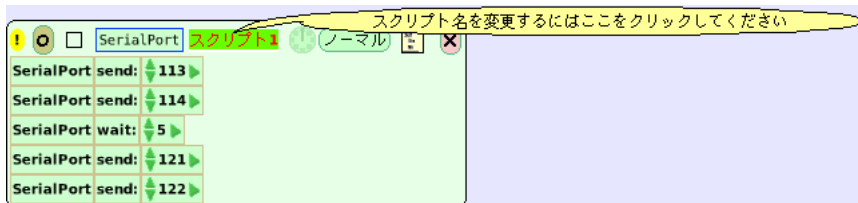
引き続き、右キャタピラを正転させる命令を作りましょう。レスキュークローラーが止まらなくなるとは困りますので、しばらく時間をおいた後、左キャタピラと右キャタピラにブレーキをかけましょう。しばらく時間をおくには **SerialPort wait: 5** を使います。ビューワーから、**SerialPort send: 5** を 3 つ、**SerialPort wait: 5** を 1 つ取り出して、下図のように数値を入力しましょう。



**SerialPort wait: 5** は 数値 × 0.1 秒の時間をとりますので、このスクリプトは、0.5 秒間前進するスクリプトになります。上で入力した 2 進数は、10 進数に変換され、下図のようになっています。レスキュークローラーの電源を入れ、スクリプトの左側の **!** をクリックしてみましょう。レスキュークローラーは少し前進しましたか？



作成したスクリプトに名前をつけておきましょう。スクリプト上部の「スクリプト 1」のところをクリックし、赤色にし、




キーボードから名前を入力しましょう。

レスキューローラーを前進させるスクリプトなので、「前進」と名付けることにします。




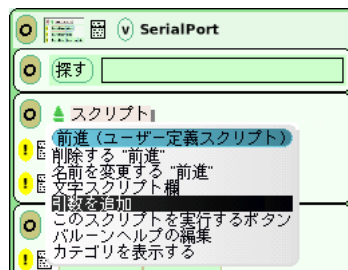
#### 14 前進する時間の指定

先ほど作った「前進」スクリプトで、**SerialPort wait: 5**の数値を変更すれば、レスキューローラーが前進する時間を変更することができます。一方、の星を進める **5** は後ろの数値で進む距離を変えることができました。

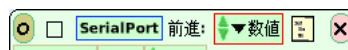
「前進」スクリプトを他のスクリプト内で使おうと思った時、このように後ろに数値をつけて、前進する時間の指定できると使い勝手が良くなります。

この数値のことを「引数」といいます。それでは、引数を「前進」スクリプトに引数を付けてみましょう。

ビューワーの中の **SerialPort 前進** のすぐ左にある  をクリックし、メニューを出します。



メニューの中から「引数を追加」をクリックします。すると、ビューワーの中の **SerialPort 前進** は **SerialPort 前進: 5** のように変わり、後ろに数値が付き、同時に、「前進」スクリプトの上部に「数値」というタイトルが追加されます。



このタイルを使って、待ち時間を引数で指定できるようにしましょう。まず、「前進」スクリプトの上部の「数値」をクリックすると、マウスに「数値」がついてきます。





「前進」スクリプトの「SerialPort wait: 5」の数字のところへ持っていくと、数字の部分が緑色になります。これを確かめて、クリックします。



これで、「前進」スクリプトに引数をつけることができました。



スクリプト左上にある  をクリックすると、一旦スクリプトをしまっておくことができます。画面からは消えますが、なくなってしまうわけではありません。

ビューワーの中の「SerialPort 前進: 5」の数値を変え、左にある  をクリックしてみましょう。レスキューローラーが前進する時間が変わりましたか？

## 15 前進だけでなく後進も

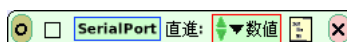
前進する時間が指定できるようになると、今度は、マイナスの数値を入れた時には、後進してほしいと思うようになることでしょう。


数値が0以上かそうでないかを判断し、それぞれの場合において送る命令を変えましょう。スクイークでは「テスト」というタイルを利用して、それぞれの場合を分けることができます。

では、「前進」スクリプトを作った時と同様に、「SerialPort 空スクリプト」を取り出して下さい。

スクリプトの名前「スクリプト1」から「直進」に変えましょう。

ビューワーで、「直進」スクリプトに引数をつけましょう。



「テスト」というタイルを出すためには、スクリプトの右上にある  をクリックして下さい。すると、下のようなタイルがあらわれます。



これをスクリプトの中に入れて下さい。

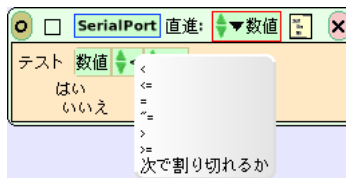


「テスト」の右には、場合を表す条件のタイルが入ります。今回は、引数がプラスかマイナスかで場合が分かれますので、「テスト」の右の条件は、数値  $\geq 0$  としましょう。これは、数値が0以上という条件です。「テスト」タイルの「はい」の右には、条件に当てはまる場合に行ないたいことを入れます。「いいえ」の右には、条件に当てはまらなかった場合に行ないたいことを入れます。

では、「前進」スクリプトの上部の「数値」をクリックし、「テスト」の右に入れて下さい。下図のようになったと思います。



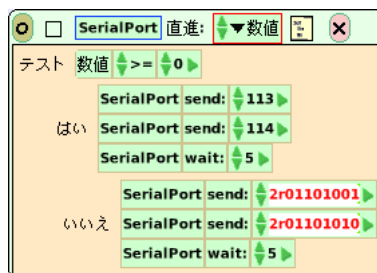
条件を設定しましょう。「数値 < 5」の < をクリックすると比較する記号があらわれます。



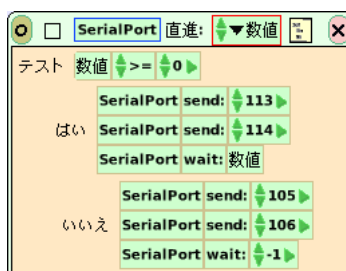
>= を選んで下さい。また、その右の数値を 0 に変えましょう。




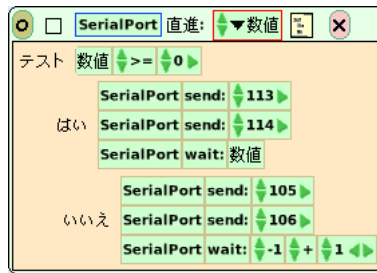
上のような条件を設定すると、「はい」のときは、引数がプラス(0を含む)の場合です。左右のキャタピラを正転させる命令を送り、指定された時間待ちましょう。「いいえ」のときは、引数がマイナスになります。左右のキャタピラを逆転させる命令を送り、指定された時間待ちましょう。





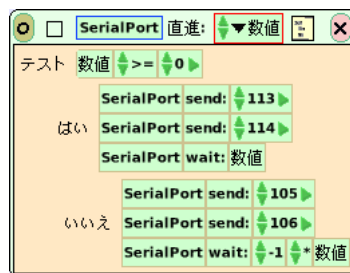
時間の指定を引数の数値にしましょう。このとき、「いいえ」のときは、引数はマイナスの数値ですが、待ち時間はプラスの数にしなければいけません。引数に -1 を掛けることにします。「いいえ」の中の「SerialPort wait: 5」の数値を -1 に変えます。



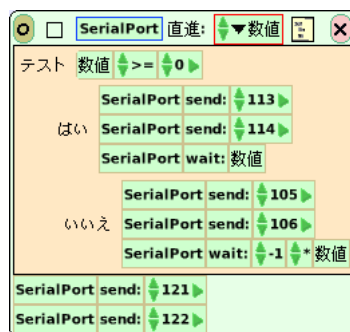
今変えた数値 -1 の右の  をクリックして下さい。次のようになります。



 をクリックして、掛け算をあらわす「\*」を選んで下さい。次に、「直進」スクリプトの上部の数値をクリックし、\*右の  の部分に持っていき、数字の部分が緑色になったことを確かめて、クリックして下さい。



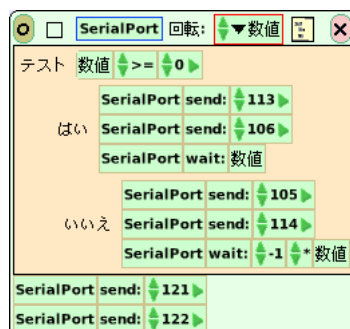
最後に、左右のキャタピラにブレーキをかけます。



## 16 回転


レスキューローラをその場で右回転させるには、左キャタピラを正転させ、右キャタピラを逆転させます。左回転はその逆です。

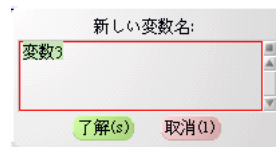
直進の時と同じように、プラスの引数ならその時間だけ右回転、マイナスの引数なら、左回転するようなスクリプトを作りましょう。もう、簡単ですね。




## 17 光センサーの入力

光センサーから数値を読み込むまえに、この数値をしまっておく場所が必要になります。これを変数といいます。ここでは、3つあるセンサーのうち、左右の2つを使うことにして、これらのセンサーから読みとった数値を、それぞれ「光センサー左」「光センサー右」という変数にしましょう。

まず、変数を追加しましょう。ビューワーの上の方にある  をクリックして下さい。新しい変数名を入力するウィンドウが開きます。



「光センサー左」と入力して  をクリックして下さい。

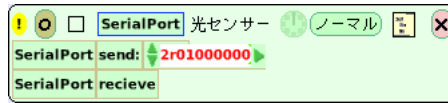


続いて同じ手順で、「光センサー右」を追加してください。







では、スクリプトを作りましょう。**SerialPort** 空スクリプトを取り出して下さい。  
 ビューワーから、**SerialPort send: 5** と **SerialPort recieve** をスクリプトに入れ、左の光センサーで測る命令を送るようにします。



センサーで測る命令に対して、レスキューローラーのマイコンは測った数値を送り返すようになっています。**SerialPort recieve** を行なうと、レスキューローラーからの数値を読み込み、あらかじめ用意された変数 **SerialPort** の **recievedData** にこの数値がしまわれます。この数値を「光センサー左」にしまい直しましょう。

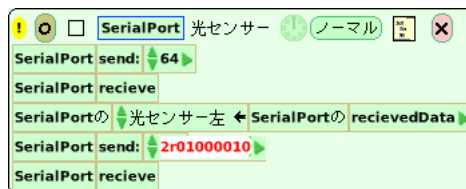
ビューワーの中の **SerialPort** の **光センサー左**  **1.0** の  の部分をクリックして、スクリプトの中に入れます。



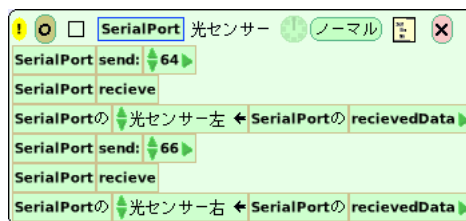
ビューワーの中の **SerialPort** の **recievedData** をクリックし、スクリプト内の **SerialPort** の **光センサー左 ← 1.0** の数値の上に持っていき、数字の部分が緑色になったことを確かめて、クリックして下さい。




左の光センサーの時と同じようにして、ビューワーから、**SerialPort send: 5** と **SerialPort recieve** をスクリプトに入れ、右の光センサーの数値を読み込むようにします。



読み込んだ数値をこの数値を「光センサー右」にしまい直しましょう。



スクリプトの左側の  をクリックしてみましょう。ビューワーの中の「光センサー右」、「光センサー左」の数値が変わりましたか？

レスキューローラーの光センサーを黒っぽい部分の上や白っぽい部分の上におき、光センサーの数値がどのように変わるか、調べましょう。

地面が黒いほど数値は（大きい・小さい）。

## 18 この後は...

さあ、ここまでやってきたあなたは、もう自由にスクリプトが作れるでしょう。  
後は、あなた次第です。

ここでは、1cm 幅の黒いラインを使った課題を紹介しておきましょう。

- ラインを踏んだら停止
- ライントレース
- 2本のラインの間を行ったり来たり