

iCar - Scratch(互換) 入門編

株式会社 カーネル・ソフト・エンジニアリング

LED 1秒点灯プログラムを作成しよう (1/3)

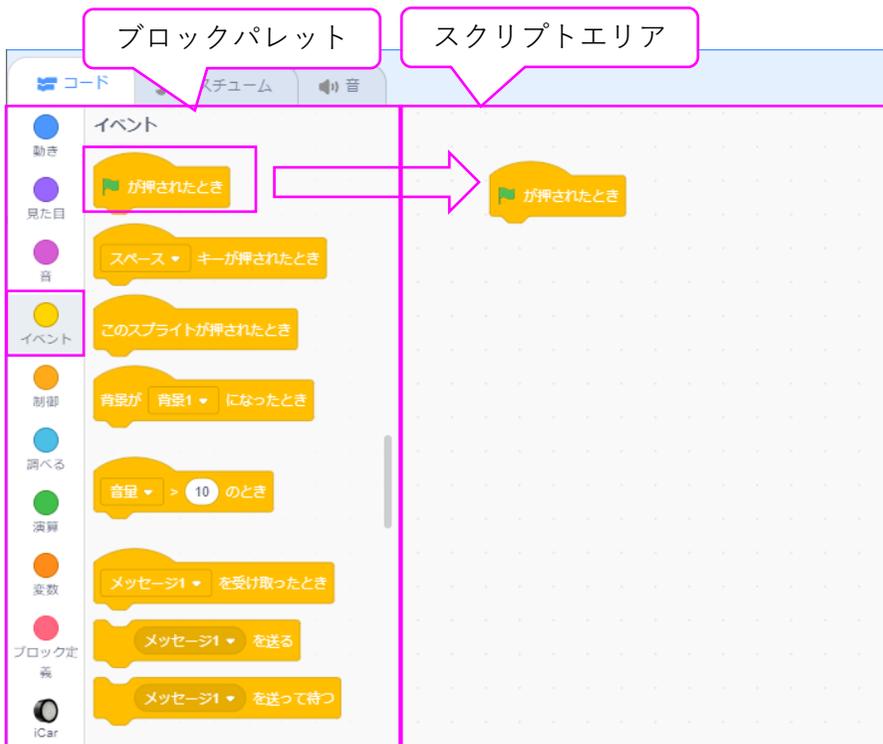
Scratchの導入とiCarとの接続が出来たら、プログラムを作成しましょう。

ここからは、LEDを1秒間点灯させるプログラムを作成します。

※このプログラムは、「1_LED1秒点灯」というファイル名でサンプルプログラムに含まれています。

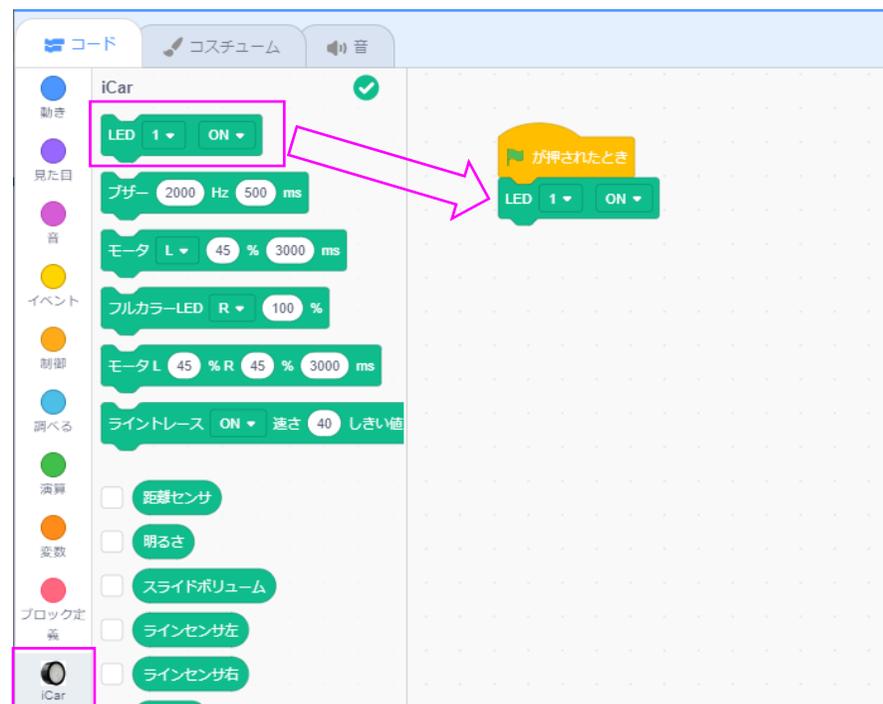
(1) デスクトップまたはスタートメニューより  のアイコンのScratch for iCarを起動します。

(2) Scratch for iCarが起動したら、ブロックパレットから「イベント」を選択し、 をドラッグしてスクリプトエリアに移動します。



(3) ブロックパレットから「iCar」を選択し、

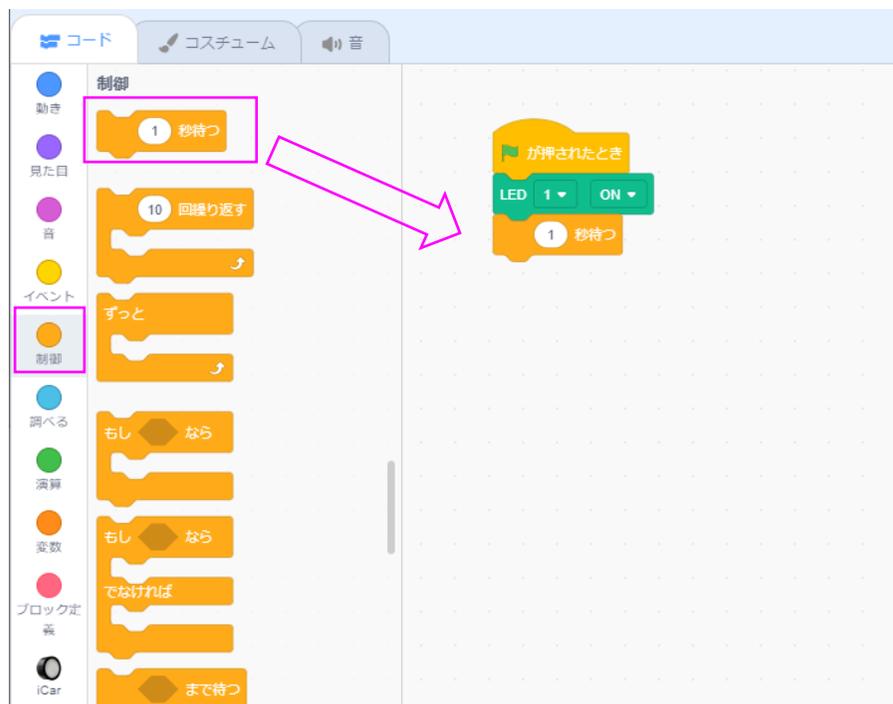
 をドラッグしてスクリプトエリアに移動して  ブロックの後に接続します。



LED 1秒点灯プログラムを作成しよう (2/3)

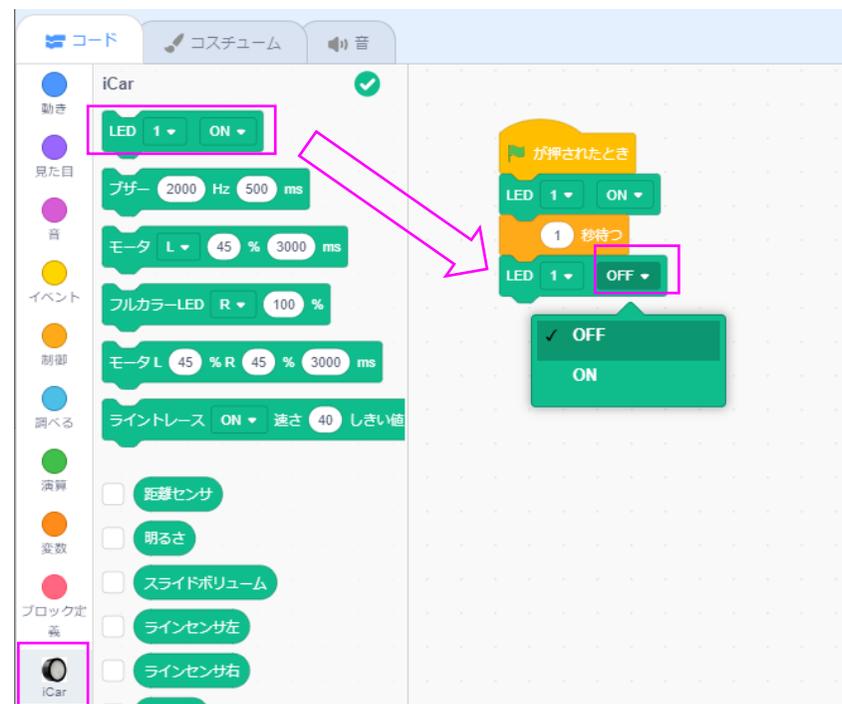
(4)ブロックパレットの「制御」から、

 をドラッグして、接続します。



(5)ブロックパレットの「iCar」から、

 をドラッグして、接続した後、「ON」の部分をクリックして「OFF」に変更してプログラム完成です。

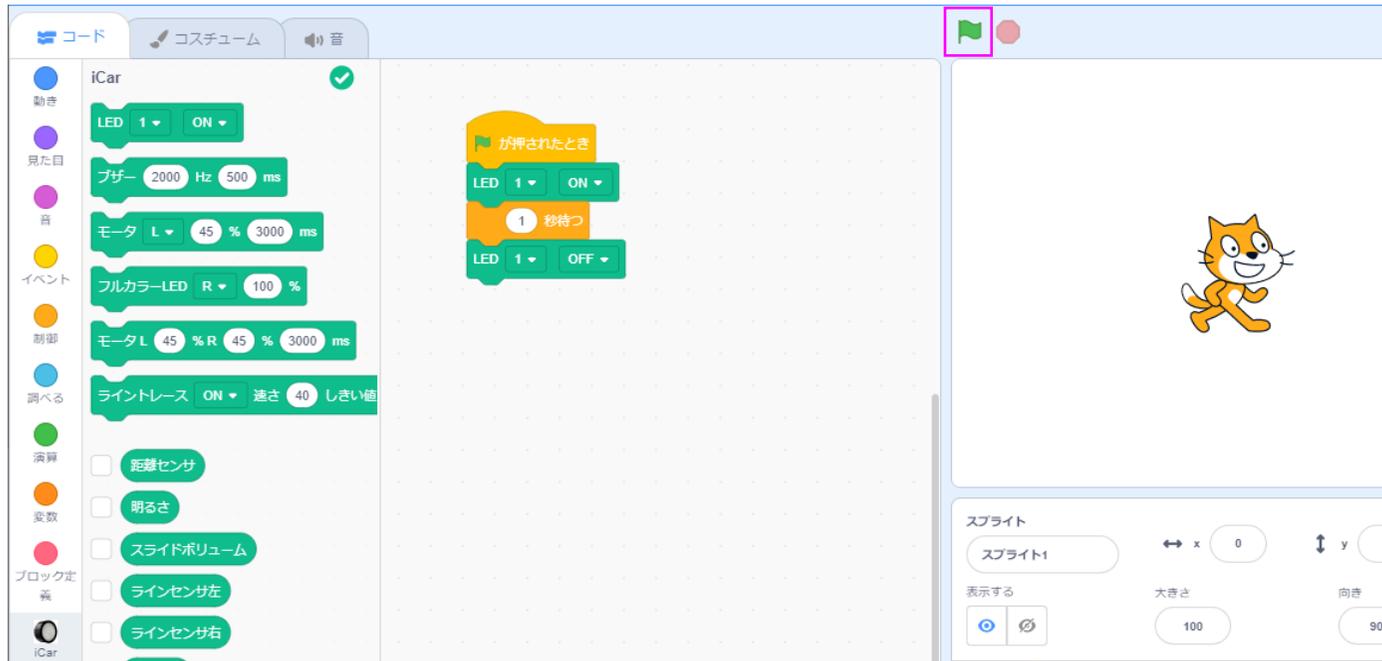


LED 1秒点灯プログラムを作成しよう (3/3)

プログラムが完成したので実行してみましょう。

作成したプログラムは、 に、LED1を1秒間ONして、OFFするプログラムなので、Scratchの画面上の  ボタンを押してプログラムを開始して、動作を確認してみましょう。

 ボタンを押すと、iCarのLED1が1秒点灯して、消灯する動作となるはずですが。



The screenshot shows the Scratch IDE interface. On the left, the 'iCar' sprite is selected, and its code blocks are visible. The script consists of the following blocks:

- When clicked (Event)
- LED 1 ON (LED Control)
- wait 1 second (Control)
- LED 1 OFF (LED Control)

The 'run' button (a green flag icon) in the top right corner of the IDE is highlighted with a pink box. The Scratch cat character is visible in the center of the stage.

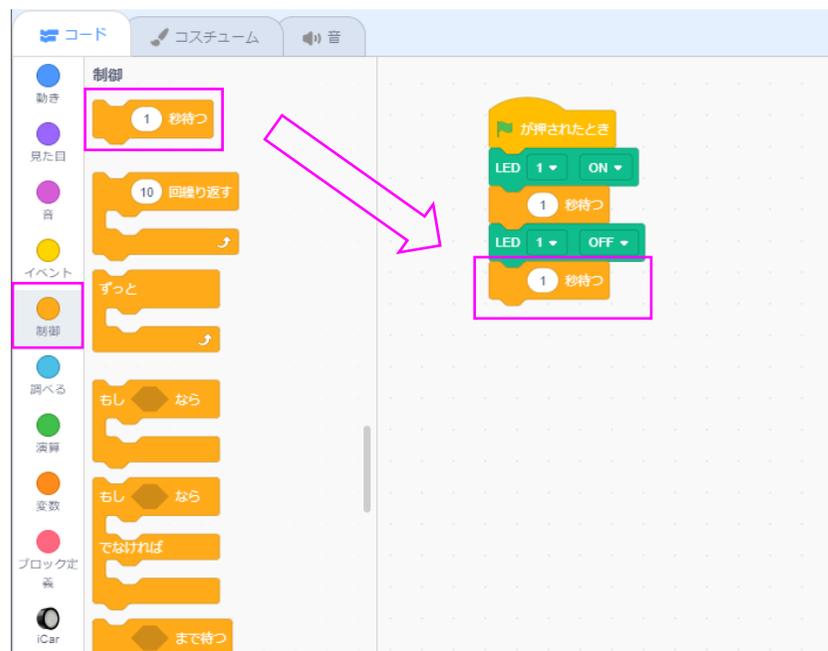
プログラムを変更しよう その1(ずっと(反復)を使う)

前のページで作成した  を押したらLED1秒点灯プログラムを、反復処理を使って、LEDがずっと点滅するプログラムに変更してみましょう。

※このプログラムは、「2_LED点滅」というファイル名でサンプルプログラムに含まれています。

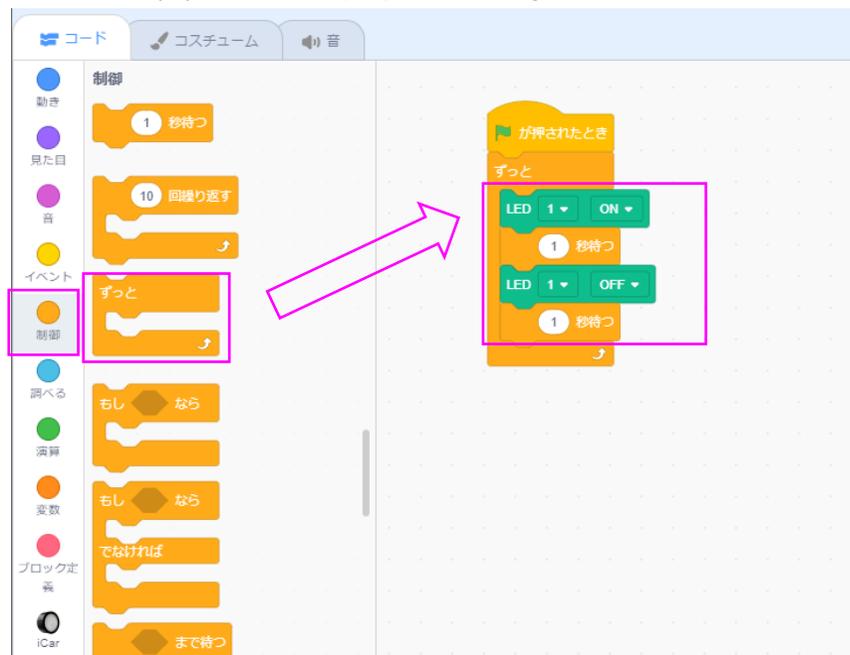
(1)ブロックパレットの「制御」から、

 をドラッグして、LED1 OFFの後に接続します。



(2)ブロックパレットの「制御」から、

 をドラッグして、LED1 ON～1秒待つまでを囲むように配置します。



プログラムが完成したら、 ボタンを押します。LED1が1秒点灯、1秒消灯を繰り返します。停止するには、 ボタンの横の  ボタンを押すと、プログラムが停止します。

●MEMO ずっと繰り返すプログラムでは、ずっとブロックの中に0.1秒待つなどの待ちを入れると動作が安定します。

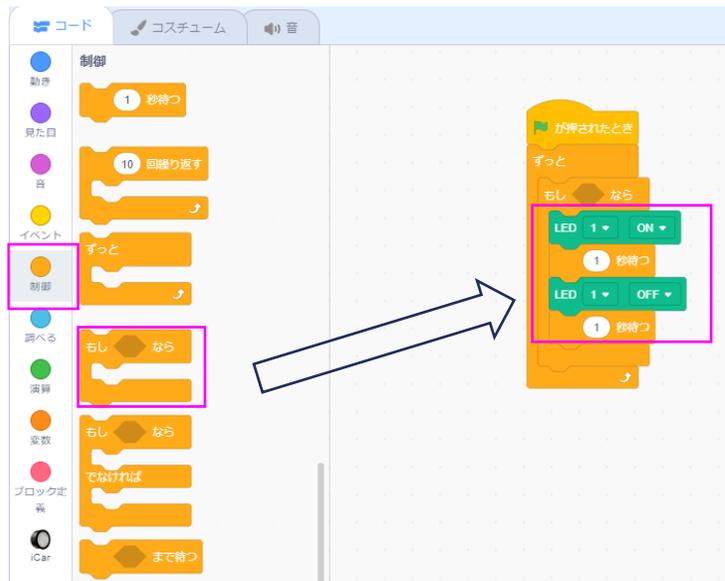
プログラムを変更しよう その2(もし~なら(分岐)を使う)(1/2)

前のページで作成したLEDがずっと点滅するプログラムを、分岐処理を使って、ボタン1を押している時だけ、LEDが点滅するプログラムに変更してみましょう。

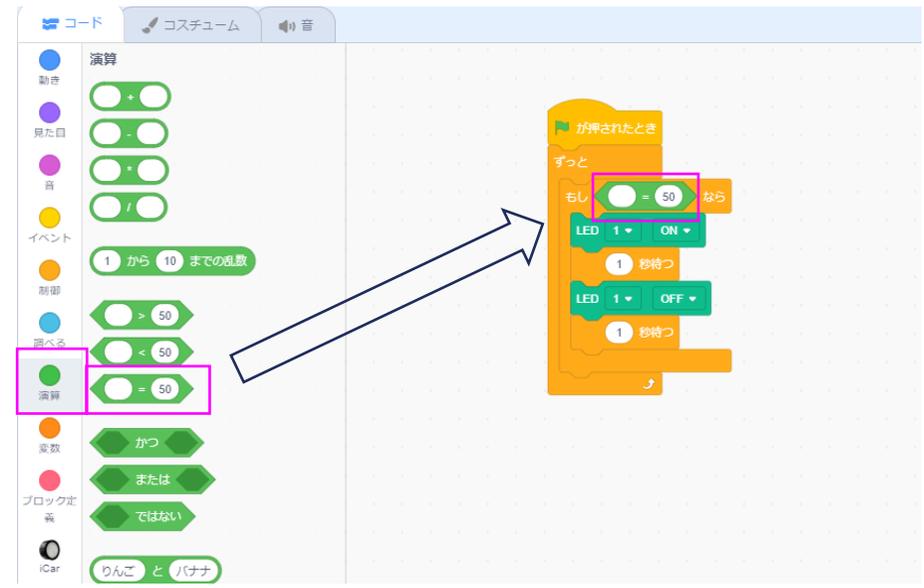
※このプログラムは、「3_ボタンを押している間LED点滅」というファイル名でサンプルプログラムに含まれています。

(1)ブロックパレットの「制御」から、

もし~なら をドラッグして、「LED1 ON」と「1秒待つ」の間に配置します

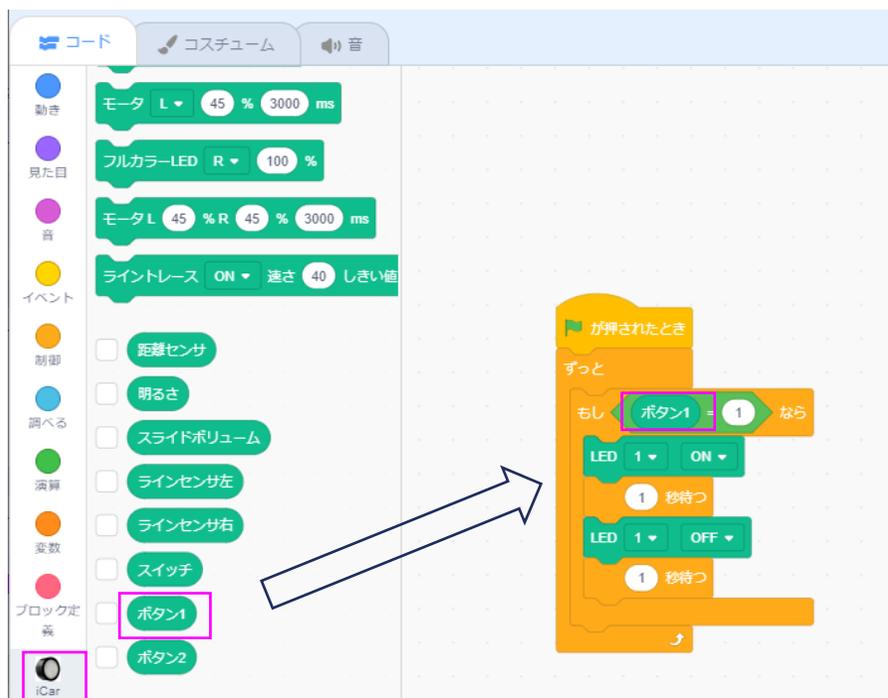


(2)ブロックパレットの「演算」から、 をドラッグして、もし~なら の中に入れます

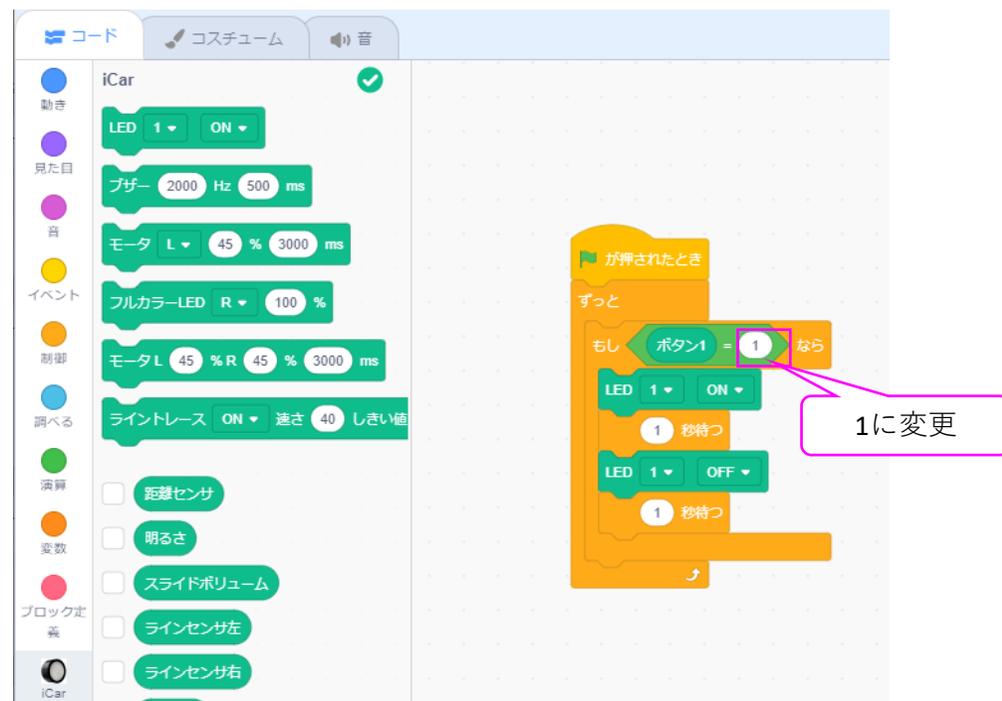


プログラムを変更しよう その2(もし~なら(分岐)を使う)(2/2)

(3)ブロックパレットの「iCar」から、**ボタン1** をドラッグして、**ボタン1 = 50** の中に入れます



(4) **ボタン1 = 50** の部分の数字(50)の部分をクリックして1に変更します (1は英数入力で入力してください)

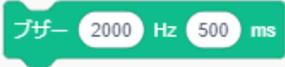


プログラムが完成したら、**旗** ボタンを押します。そして、ボタン1(BTN_1)を押すと、LED1が1秒点灯、1秒消灯を繰り返し、離すと点滅が止まります。

● ボタンを押すと、プログラムが停止します。

iCar ブロックの機能 (1/3)

このページでは、ScratchでのiCarブロックの機能について説明します。プログラム作成の参考としてください。

ブロック	機能	引数
	LEDを点灯・消灯させます	1: 操作するLED番号 2: ON, OFF (ON=点灯, OFF=消灯)
	ブザーを指定時間鳴らします	1: 周波数Hz (8~16000 ※0または範囲外は停止) 2: 鳴らす時間 ミリ秒 ((0=ずっと)~16383ミリ秒) ※1ms(ミリ秒)=0.001秒 ※10ミリ秒以下は切り捨てされます
	モータL,Rを指定出力・時間で動かします	1: 動かすモータ L, R (L=左, R=右) 2: 速度(%) (-100=後退~0=停止~100=前進) ※範囲外指定でブレーキをかける 3: 動かす時間 ミリ秒 ((0=ずっと)~16383ミリ秒) ※10ミリ秒以下は切り捨てされます
	フルカラーLEDの明るさを設定します	1: 明るさを変えるLED (R=赤, G=緑, B=青) 2: 明るさ(0(消灯)~100(%))
	モータL,Rを同時に動かします	1: モータLの速度(%) (-100~100), ※範囲外はブレーキ 2: モータRの速度(%) (-100~100), ※範囲外はブレーキ 3: 動かす時間 ミリ秒 ((0=ずっと)~16383ミリ秒) ※10ミリ秒以下は切り捨て
	LCDを更新します	なし
	LCDをクリアします ※LCDに変更内容を反映するには、 LCD更新ブロックを呼び出しする必要があります	1: クリアする行, (全部, 1行目, 2行目)
	LCDに文字を設定します	1: 書き込む行 2: 書き込む文字 (※指定可能文字: アルファベット, 数字)

iCar ブロックの機能 (2/3)

このページでは、ScratchでのiCarブロックの機能について説明します。プログラム作成の参考としてください。

ブロック	機能	引数
	<p>ライントレース (白い地面にひかれた黒い線に沿って走行) (白検知でモータ回転、黒検知でモータ停止)</p> <p>※ライントレースON中は、モータブロックは動作しません。 モータブロックを使う場合はライントレースOFFとしてください</p>	<p>1: ライントレースON, OFF 2: 速度 (0~100) 3: しきい値 (0~4095)</p> <p>速度・・・ラインセンサが、白を検知しているときに設定するモータ速度</p> <p>しきい値・・・ラインセンサで地面の白、黒を判定するための値(しきい値以下で白と判定)</p>
	<p>センサ値をLCDに表示します</p> <p>下記のようにセンサ値が表示されます</p> <div style="border: 1px solid black; padding: 5px; display: inline-block;"> <p>L (1) (2) R (3) V (4) B (5) SW (6)</p> </div> <p>(1) ラインセンサL (2)距離センサ (3) ラインセンサR (4)スライドボリューム (5) 明るさセンサ (6)スイッチ・ボタンの状態を3桁表示 (スイッチ, ボタン1, ボタン2の順)</p>	<p>1: センサモニタ ON/OFF</p>

iCar ブロックの機能 (3/3)

このページでは、ScratchでのiCarブロックの機能について説明します。プログラム作成の参考としてください。

ブロック	機能	取得できる値の意味
距離センサ	前方の物との距離を取得する	前方障害物との距離 [mm] (6027・・・未検出)
明るさ	周囲の明るさを取得する	周囲の明るさ (0=明るい～4095=暗い)
スライドボリューム	スライドボリュームの位置を取得	スライドボリュームの位置 (0=後方～100=前方)
ラインセンサL	ラインセンサLの明るさを取得	ラインセンサLの直下の明るさ (0=明るい～4095=暗い)
ラインセンサR	ラインセンサRの明るさを取得	ラインセンサRの直下の明るさ (0=明るい～4095=暗い)
スイッチ	スイッチの状態を取得	スイッチの位置 (0=後退方向, 1=中点, 2=前進方向)
ボタン1	ボタン1の状態を取得	1=押されている, 0=押されていない
ボタン2	ボタン2の状態を取得	1=押されている, 0=押されていない