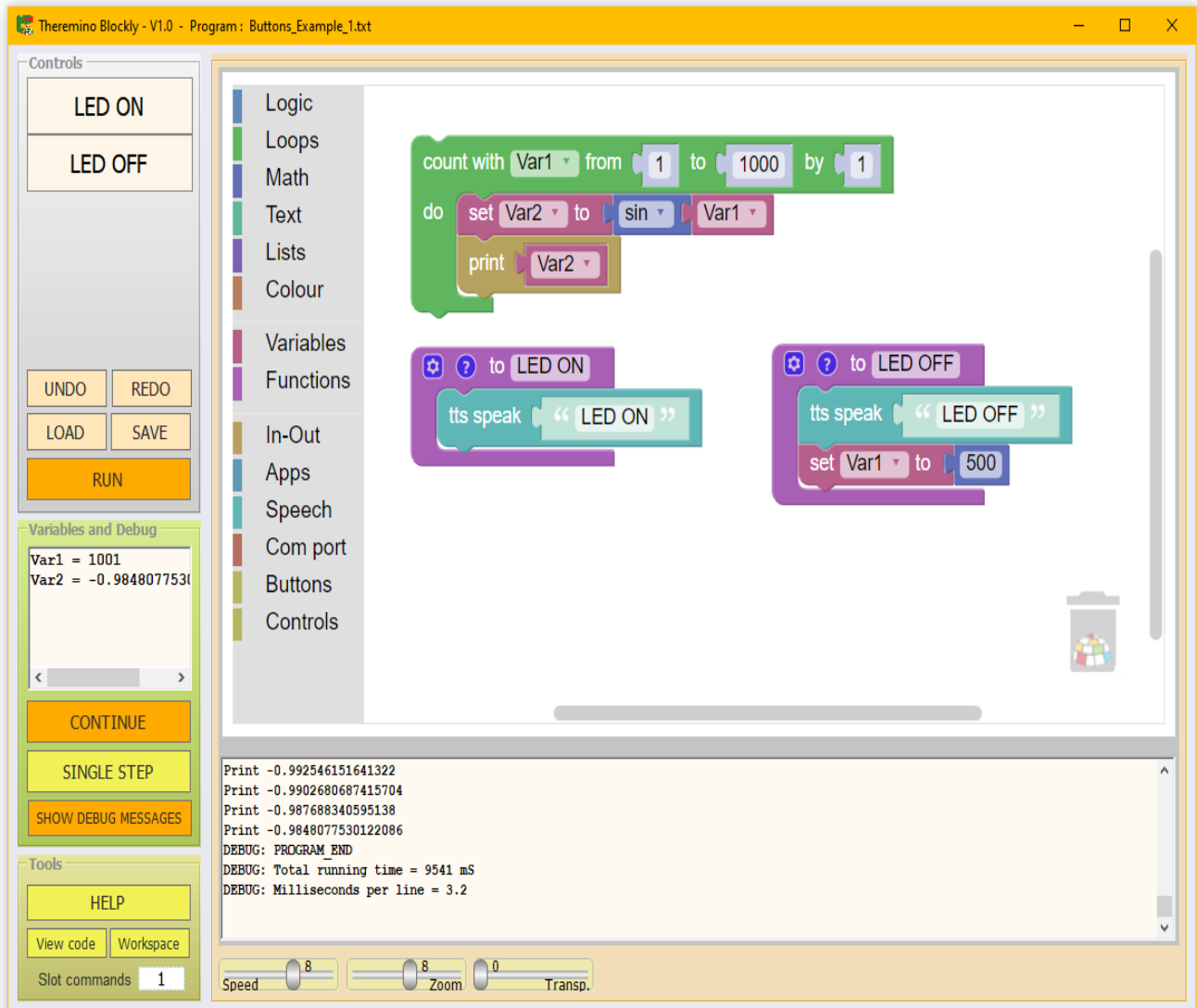


# システムテレミノ



# テレミノ ブロックリー V1.0

## ページ インデックス

敷地内.....	3
<b>Blockly</b> の最初のステップ.....	3
プログラムのロードと実行.....	4
始めるための簡単なプログラム.....	4
プログラムの編集と保存.....	5
ブロックエリア.....	6
プログラムの実行.....	6
プログラム構造.....	7
機能ボタン.....	8
ボタンと機能.....	9
ブロックメニュー.....	10
外部通信用メニュー.....	11
アプリ メニュー - ファイルを開く.....	13
アプリ メニュー - フォルダを開く.....	14
音声メニュー - 音声合成.....	15
音声コマンド.....	15
<b>Com</b> ポート メニュー - シリアル ポート.....	16
メニューボタン - コントロールボタン.....	17
コントロール メニュー - ビュー コントロール.....	18
「変数とデバッグ」パネル.....	19
プログラムの機能を確認する.....	19
「ツール」パネル.....	20
外部コマンドの実行.....	20
アプリケーションメニュー.....	21
ブロックエリアのメニュー.....	22
ゴミ.....	22
下部のバー コントロール.....	23
<b>Blockly</b> の複数のインスタンスを実行する.....	24
<b>Theremino Blockly</b> の制限.....	24
実行速度.....	24
技術文書.....	25

## 敷地内

と**ブロックリー**コマンド構文を勉強しなくてもプログラミングを学ぶことができます。

ブロックの接続は簡単で直感的**小さなお子様にも**そして、プログラムは自動的に作成されます。

**Theremino\_Blockly** の弟です **Theremino オートメーション**、より単純ですが、それでも可能ですシステムのすべての主要機能にアクセスする、簡単に来て**入出力で合成**そしてアル認識 **ボーカル**、の制御に**ロボット 共同作業**、程度まで**放射線**彼は・・・で生まれた**ラドン**、の制御へ**不整脈心臓**、で分析する **化学薬品**、で測定する**オシロスコープとアナライザー**から**スペクトラム**など...

の**このページ**私たちのシステムの (ほぼ 200 の) アプリケーションのインデックスを見つけてください。**自由とオープンソース**.

## Blockly の最初のステップ

**ブロックリー**によって作成されました**グーグル**また、基礎を学ぶのに役立つ教育ユニット、例、および小さなゲームが用意されています。

初心者向けコネクティングブロック

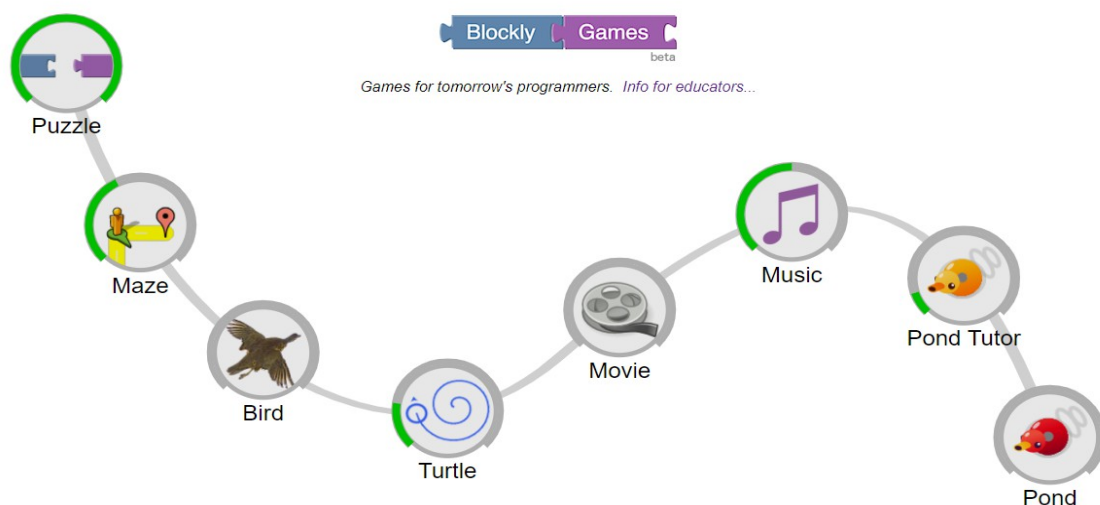
<https://www.ivana.it/jm/software-on-line/358-blockly>

ゲーム

<https://blockly.games>

<https://www.brainpop.com/games/blocklymaze>

<https://blockly.games/music>



# プログラムのロードと実行

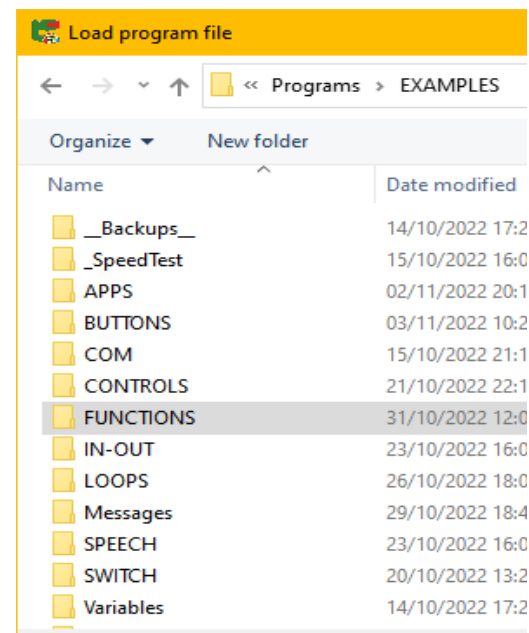


ボタン**ロード**サンプル プログラムを選択できるウィンドウが開きます。

プログラムを読み込んだ後、ボタンで実行します**走る**、または上部エリアにあるボタンで機能を開始します。

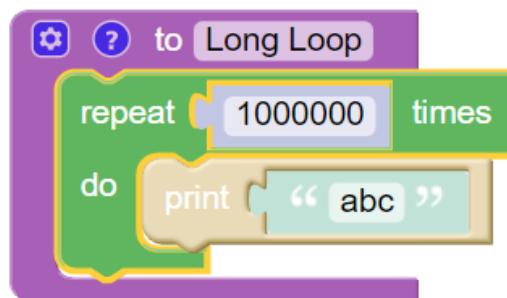
プログラムを停止するには、キーを押します**走る** ( となりました**止まる** )。

の**止まる**それが起こる自動的にブロック領域をクリックして編集するとき。



## 始めるための簡単なプログラム

フォルダにある簡単な例から始めることをお勧めします **プログラム**. プログラムを実行して、その動作を確認します。ブロックをマウスで動かし、左側の垂直メニューからブロックを追加して、最後に元のプログラムを復元してみてください **元に戻す**そしてその**パスケット**.



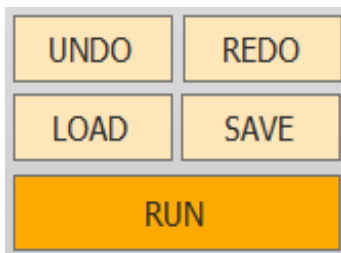
サンプル ファイルはすべて double です。たとえば、**ファイル名\_1**と**ファイル名\_2**、そのため、元の例を失うリスクが少なくなり、変更やテストを行うことができます。

ただし、覚えておいてください**編集を開始する前に、プログラムを新しい名前**で保存してください。ただし、必要に応じて、元の例をすべて復元できます。ダウンロード元のファイル [ブロックリーページ](#).

# プログラムの編集と保存

プログラムに加えたすべての変更は自動的にファイルに書き込まれるため、アプリケーションを閉じる前に作業を保存することを忘れないでください。**ブロックリー**、または別のプログラムをロードします。

したがって、例の変更を避けるために、**変更を開始する前にキーを使用して、プログラムを新しい名前で保存することをお勧めします保存**。



◆ 押す **保存**とともに**左のボタン**マウスを離すと、プログラムは新しい名前で保存されます。

◆ 押す **ロード**とともに**左のボタン**マウスをクリックすると、ウィンドウが開き、選択できる多くのサンプル プ

ログラムが表示されます。

◆ 押す **保存**とともに**右ボタン**マウスの、プログラムはすぐに保存されます。

◆ 押す **ロード**とともに**右ボタン**マウスをクリックしてプログラムを保存し、すぐに再ロードします。

◆ ボタン **元に戻す**プログラムに変更が加えられ、それらを削除したい場合に戻るために使用されます。

◆ ボタン**やり直し**破棄された変更を再構築します**元に戻す**。

それ以外の**元に戻す**と**やり直し**キーの組み合わせも使用できます**CTRL-Z**と**CTRL-Y** (プレス**CTRL**押し続ける**Z**また**よ**)。

-----

プログラムを再度開くと、終了したときとまったく同じになります。

**RUNでプログラムを閉じた場合それでRUNがアクティブな状態で再起動します。**

プログラムが関数だけで構成されている場合、**最初の機能プログラムを起動すると実行されます。**

# ブロックエリア



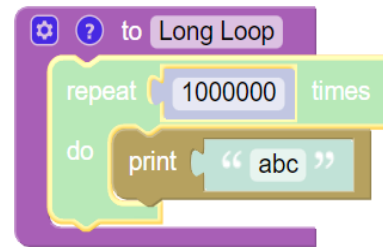
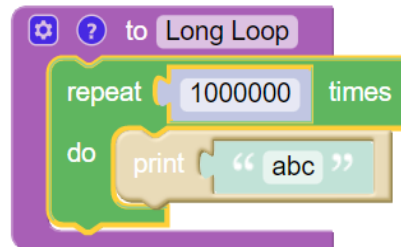
の操作**ブロックリー**それを理解するための最良の方法は、試してみることです。

フォルダにある例を読み込んで試してください**プログラム**

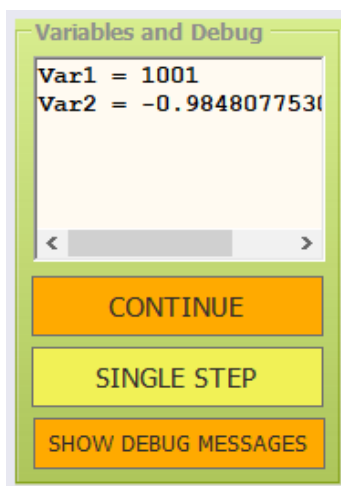
によって作成されたドキュメント、例、およびゲームも参照してください。**グーグル**、このドキュメントの冒頭で示しました。

# プログラムの実行

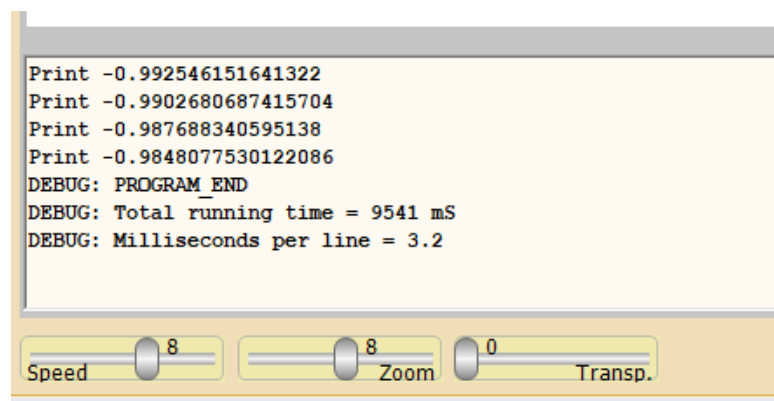
これら 2 つの画像の緑色のブロックに見られるように、実行中はブロックが明るくなります。それらをよく見るには、速度を 4 またはさらに遅くします。



リハーサル中はパネルを開いたままにしておくと便利です **変数とデバッグ** ボタンを有効にします **デバッグ メッセージを表示する**。

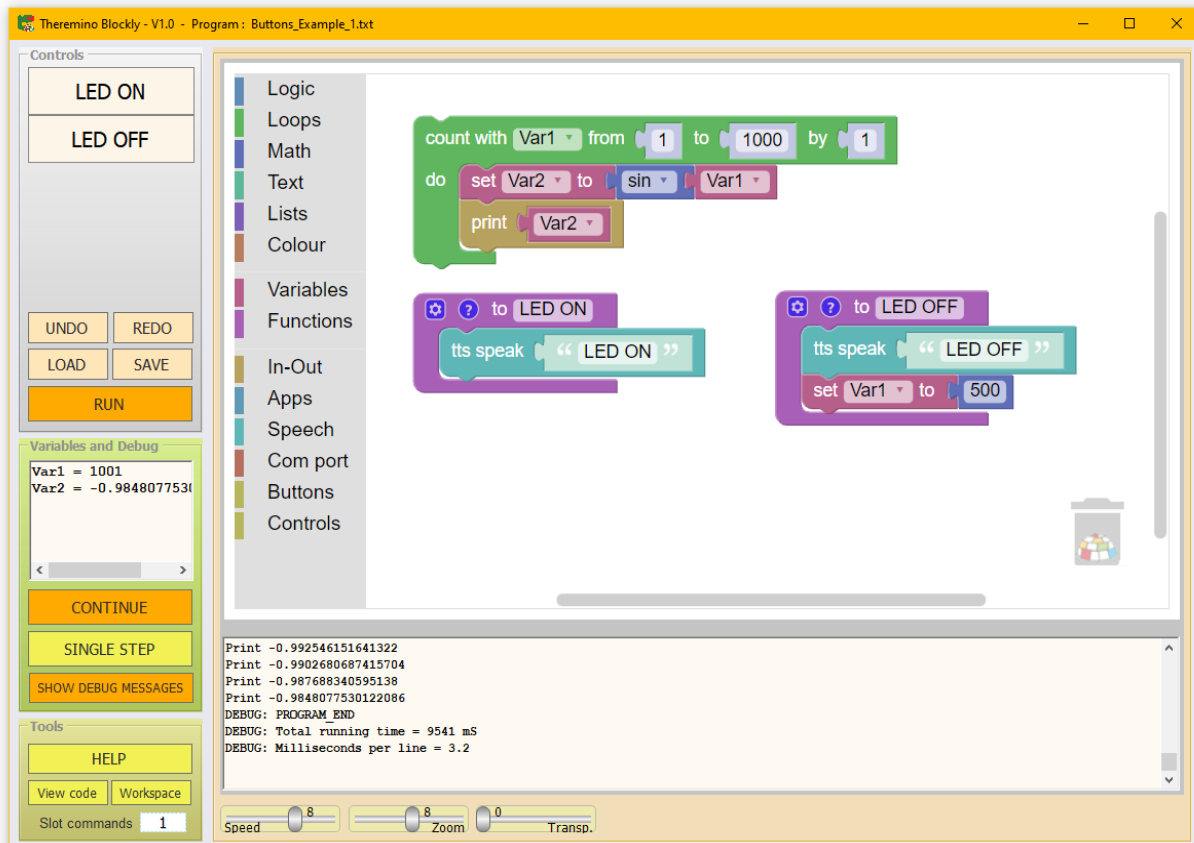


で送信されたテキストに加えて、下部のテキスト領域でこのように **印刷する**、接頭辞が付いたサービス情報も表示されます **デバッグ**。



これらのボタンの操作の詳細は、操作ページで説明されています。 [「変数とデバッグ」パネル](#)。

# プログラム構造



プログラムは関数 (この画像では紫色) と各機能はそれを実行するためのボタンを表示します。

一部のセクションは、この画像の緑色で囲まれた部分など、関数の外側にある場合もあります。

RUN が押されたとき、および関数が開始されたときに、関数外のプログラム部分が (上から下へ) 実行されます。

関数が外部プログラムゾーンも実行しないようにするには、これらのゾーンを関数内に囲むことをお勧めします。次に、これらの関数をプログラムから、またはボタンで呼び出します。

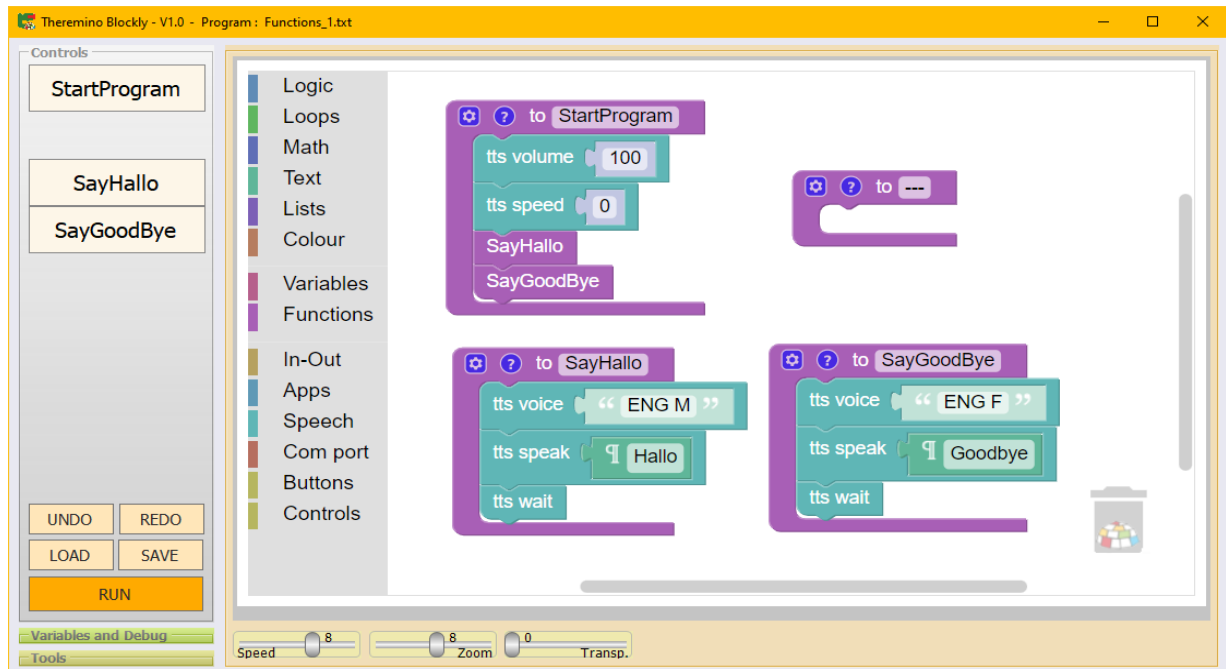
関数でボタンを作成したくない場合は、名前の先頭にアルファベット以外の文字を追加してください。例を表示するには、プログラムを開きます。 **EXAMPLES\SPEECH\TTS\_Example\_1.txt**

プログラムが関数だけで構成されている場合、**最初の機能 RUN でプログラムを起動すると実行される。**

# 機能ボタン

コントロールパネルに機能ボタンが表示されます。

この写真のボタンは**スタートプログラム**、**こんにちはと言う**と**さよならを言う**。



メニューの使用**機能**の真ん中にある**ツールバー** から**ブロックリー** 関数を作成して意味のある名前を付けることができます。

ファンクション ブロックを作成すると、それらを実行するためのボタンがコントロールパネルに自動的に表示されます。

機能ブロック名を変更すると、コントロール ボタンが自動的に更新されます。

ボタン内のテキストを見やすくするために、テキストを 2 行に分割できます。彼らは合計します**連続する 2 つのスペース**関数名の適切な位置で、テキストがその位置で改行されます。

他のすべてのスペースは 1 つにする必要があります。そうしないと、テキストが 3 行になり、一部の単語が下部に表示されなくなります。

関数でボタンを作成したくない場合は、追加します  
アルファベット以外の文字 (例: **\_\_**また**---**) をその名前の先頭に付けます。

プログラムが関数だけで構成されている場合、**最初の機能**  
**RUNでプログラムを起動したときにも実行されます。**



# ボタンと機能

ボタンの順序はブロックの順序 (上から下、左から右) に従うため、ボタンを入れ替える場合はブロックを移動するだけです。

同様の機能を実行するボタンをグループ化するために、ボタン間に空白スペースを残すこともできます。空白のままにするには、関数名の先頭にアルファベット以外の文字を追加します。たとえば、`__`また`---`、次にブロックを移動して、空の場所を正しい位置に移動します。

## 機能ボタンの番号付け

一部のブロック (**無効化ボタン**、**有効化ボタン**、**ボタンの色**、**ボタンスロットとボタンのテキスト**) だけでなく、識別子 (の関数名**ブロックリー**)、番号でそれらを識別することもできます。

ボタンには上から順に番号が付けられています。ボタンの最初の列は 1 から 16 までです。ボタンが 16 個を超える場合、2 番目の列は 16 から 32 まで、3 番目の列は 33 から 48 まで、というように 128 まで続きます。

ブロックの順序、つまりボタンの順序が変更された場合は、番号に従ってボタンを識別する命令も確認し、必要に応じて変更する必要があります。このため、数字ではなくテキストでボタンを指定することをお勧めします。

数値を使用した方が良いのは、多くのボタンで同じことを行いたい場合のみです。たとえば、数値変数をインクリメントしてボタンを無効にしたり色を付けたりするループを使用する場合などです。

## 関数の実行

機能に関連するボタンを押すと、他のすべての操作が中断され、その機能が最後まで実行されます。関数が終了すると、中断したところから実行が再開されます。これらの機能は、プログラムによって実行することもできます。

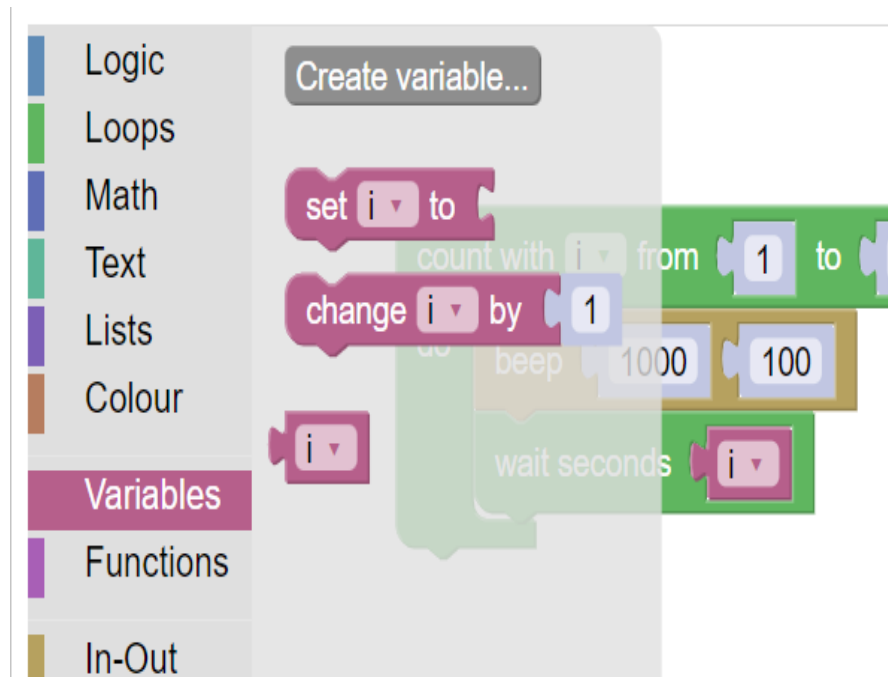
プログラムで実行する関数はすべきではありませんボタンを作成し、文字を追加します (`__`また`---`) をその名前の先頭に付けます。

プログラムが関数だけで構成されている場合、**最初の機能**  
**RUNでプログラムを起動したときにも実行されます。**

# ブロックメニュー



左側のメニューをクリックすると、トピックや色ごとに分けられた 100 を超えるさまざまなブロックから選択できます。



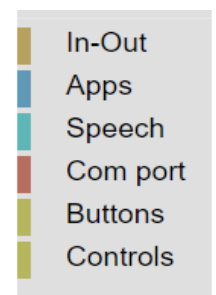
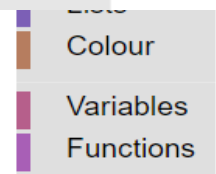
からの  
最初の  
エント

り論理に機能言語の構成要素が含まれています。

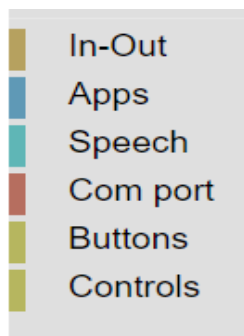
これらのブロックの命令は、VisualBasic、CSharp、Java、C++、Pascal、Python などの最も有名なプログラミング言語で使われる命令に似ています。

からの以下の6つのエントリインアウトにコントロール含む特定のブロック、Theremino システムのアプリケーションと通信するため、PC の外部とも通信します。

これらの手順の詳細な説明は、次のページにあります。



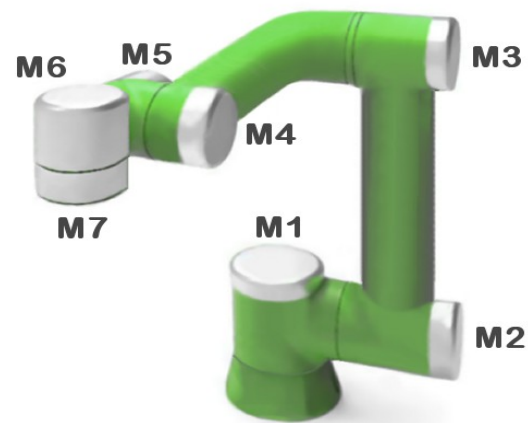
## 外部通信用メニュー



ブロックメニューの最後の6項目には、特定の指示、Theremino システムのアプリケーションと PC の外部と通信します。

リアルタイムで通信することは重要であり、すべての生物がそれを行っていますが、一般に、今日のコンピュータは非常に高速であるにもかかわらず、これらの側面を無視しています。

私たちのシステムのアプリケーションは、私たちが使用する PC 上でローカルに動作します。[シーケンサ](#). 私たちのシステムでは、PC 自体がセンサーとモーターを直接制御します。それ以外の場合は高価で複雑な制御ハードウェアの多くが排除されます。



以下は、当社の無料およびオープンソースソフトウェアと数十ドル相当のエンジンで実現できることの例です。

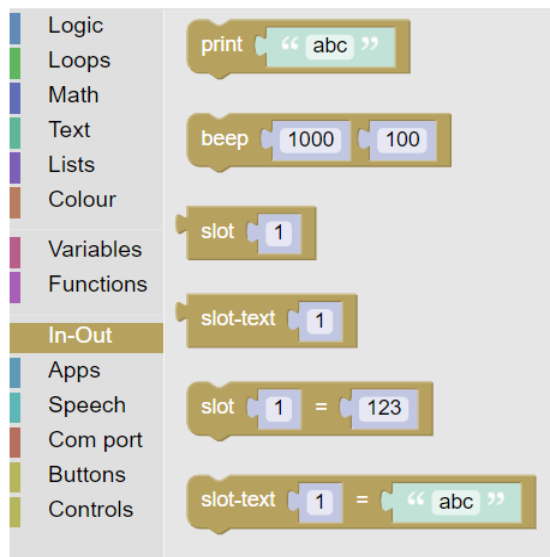
[Video1.mp4](#)  
[Video2.mp4](#)  
[Video3.mp4](#)  
[Video4.mp4](#)



Web ページは、センサーとロボット工学の適切な応答時間 (100 分の 1 から 1000 分の 1 秒) で反応できませんでした。

したがって、システムのすべてのアプリケーションは PC 上でローカルに動作します。

# インアウトメニュー



このセクションのブロックは、システムのアプリケーションと通信するだけでなく、PC 外のセンサー、アクチュエーター、人間とも通信します。

を通じて通信する [ナンバーズロット](#) 彼ら [テキストスロット](#)、彼らは私たちのシステムの独占的な発明アプリケーションを構築できますプログラミングやエレクトロニクスの専門家でなくても複雑です。

を通して [ナンバーズロット](#) 彼ら [スロットテキスト](#) あなたはと通信します多数のアプリケーションの操作を容易にするために特別に作成された私たちのシステムの [入出力](#)、この写真は最も重要なものを示しています。

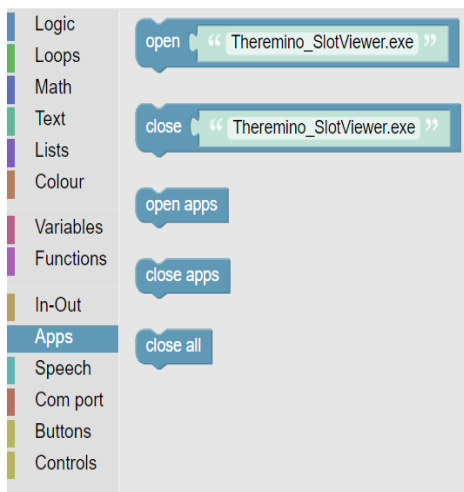


これらのアプリケーションはそれぞれ、特定のタスクを処理します。これらすべての操作を Blockly で実行することは不可能であり、最も高度な言語を使用しても、複雑さが管理できなくなるためです。しかし、複数のアプリケーションにタスクを分割することで、堅牢で信頼性の高い非常に複雑なシステムを同時に構築することができます。

このすべてソフトウェア、完全に自由とオープンソース、  
は、10 年以上にわたって継続的に取り組んできた [私たちのチーム](#)。  
それらをダウンロードして操作の詳細を開くには [このページ](#)。



# アプリ メニュー - ファイルを開く



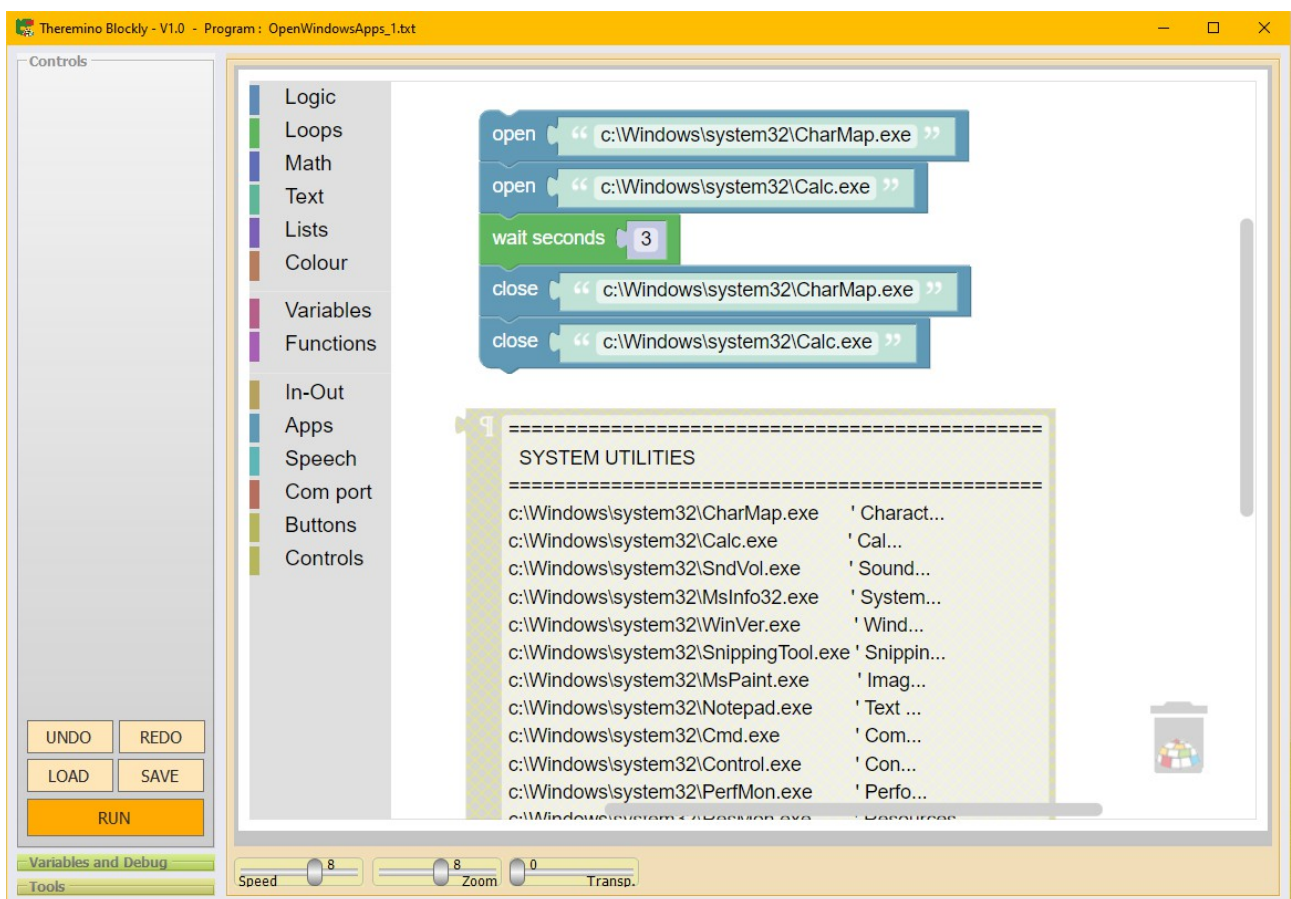
これらの例では、Windows システムのいくつかのアプリケーションを開きます。

「C:\windows\Notepad.exe」を開く

「C:\windows\system32\Calc.exe」を開く

「C:\windows\system32\mspaint.exe」を開く

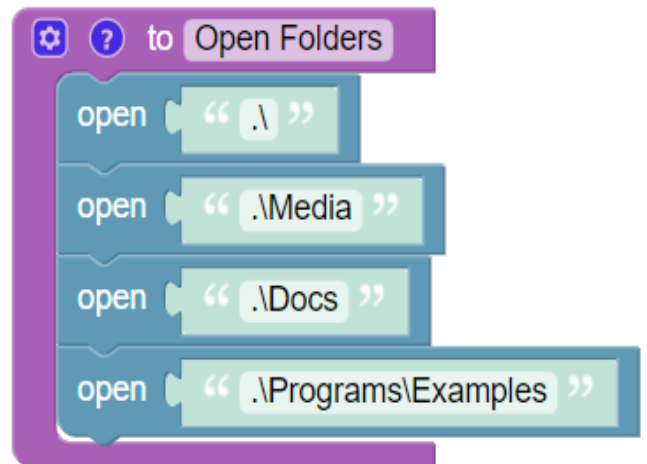
「OpenWindowsApps」の例から取った次の画像は、「Load」ブロックがどのように見えるかを示しています。



以下にある例を試してください。 **プログラム\EXAMPLES\APPS**

# アプリ メニュー - フォルダを開く

これらの例では、アプリケーションのいくつかのフォルダを開きます **Theremino\_Blockly**. ドットとバックスラッシュは、アプリケーションのルートフォルダを示すことに注意してください。 **ブロックリー**、つまりファイルのフォルダ **Theremino\_Blockly.exe**



開ける ".\"

「.\ファイル」を開く

「.\Media」を開く

アプリケーションのフォルダとサブフォルダを示すには **ブロックリー** ドットとスラッシュは省略できるため、前の 3 つの例は単純に次のようになります。

開ける "" 「ファイル」を開く 「平均」を開く

二重ポイントは、親フォルダを示すために使用されます。たとえば、ファイルフォルダを含むフォルダを開くには **Theremino\_Blockly.exe** あなたは書ける：

開ける ".."

次の例では、C ドライブと Windows システムのいくつかのフォルダを開きます。

「C:\」を開く

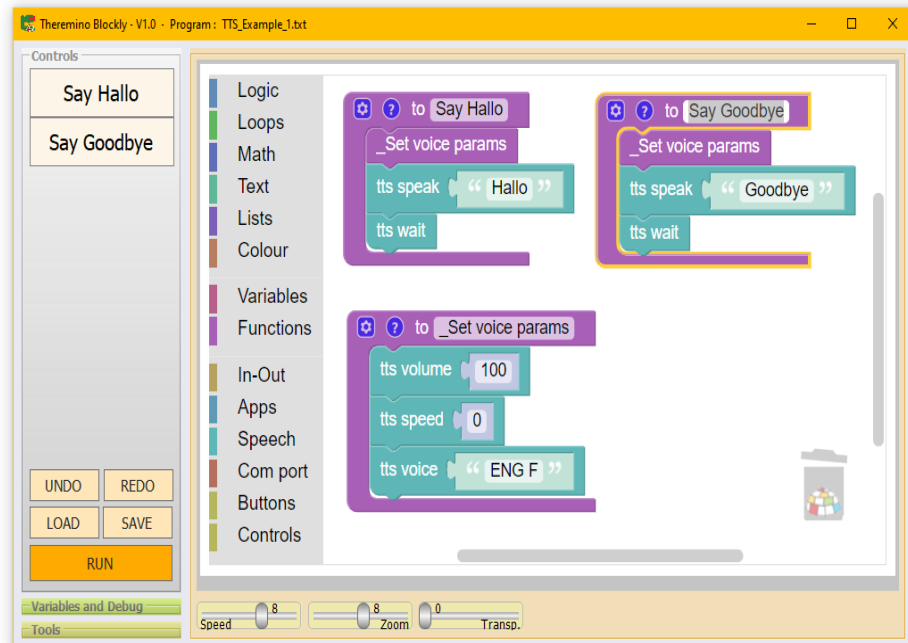
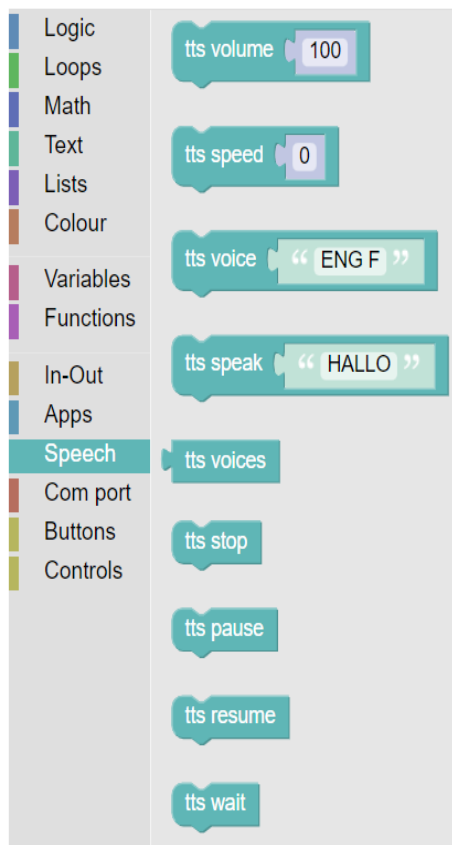
「C:\windows」を開く

「C:\windows\system32」を開く

以下にある例を試してください。 **プログラム\EXAMPLES\APPS**

# 音声メニュー - 音声合成

フォルダーの例 **プログラム\EXAMPLES\SPEECH** 音声合成に使用するすべてのコマンドを表示します **TTS 拡張子(テキスト読み上げ)**。



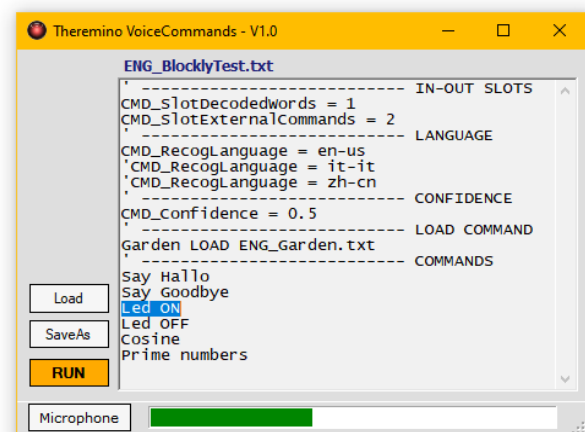
利用可能な言語は Windows によって異なり、言語設定パネルで設定します。

## 音声コマンド

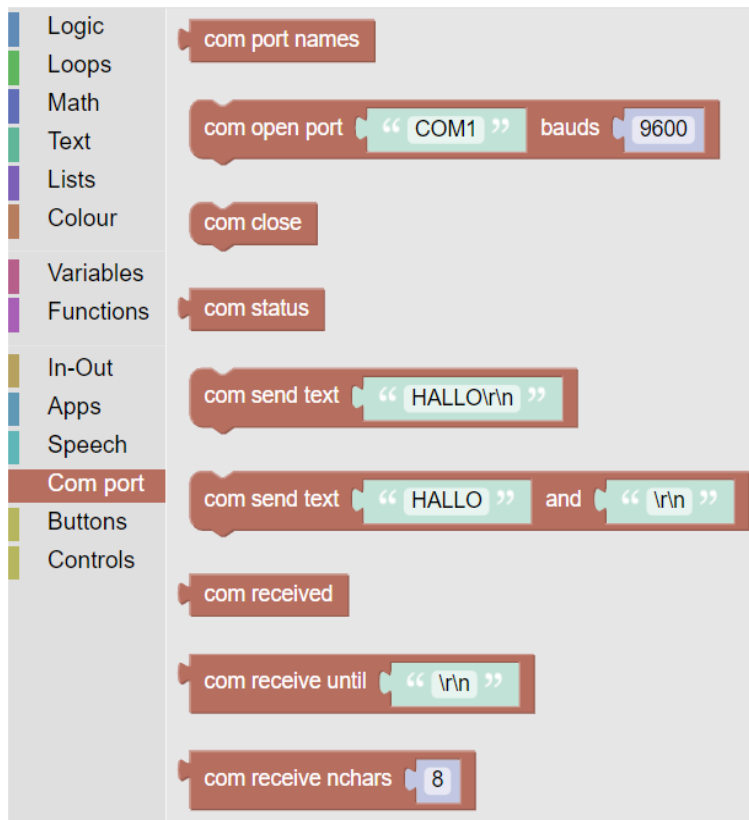
音声コマンドの場合は、アプリケーションを使用します **音声コマンド** からダウンロードします [このページ](#)。

の名前 **外部コマンド** 実行されることは受け取られます **スロットテキスト** コマンドの **ブロックリー**、同じように設定する必要があります **スロットテキスト** 両方で **ブロックリー** その中 **音声コマンド**。

順番に **ブロックリー** アプリに送信できます。 **音声コマンド** 次のコマンド: **走る**、**止まる** と **ファイル名をロード**、またこの場合、**スロットテキスト** 送信と受信が一致している必要があります。



# Com ポート メニュー - シリアル ポート



シリアルポートを使用して、スケールなどの多くのセンサーを制御できます。またはGPSシステム、またはスマートモーターなどのアクチュエーター。

場合によっては、シリアル経由の通信が非常に複雑になることがあるため、長くて難しいプログラムを作成する前に、チェックインしてください。[このページ](#)。

私たちのシステムには、最も一般的なすべてのケースに対して、すでに多くの便利なアプリケーション

ンがあります。

テキストを送信して改行するときに特殊文字が使用される **CR** ( カートリターン ) へ **NL** ( 改行 )。しかし、それらは次のような単語の一部になる可能性があるため、**CR** ホモ・オー **NL** 私たち、それらをシーケンスに置き換えます **\r** と **\n**。

通常、Windows システムでは、両方が使用されてから書き込まれます。**\r\n** ただし、一部のシステムまたはセンサーはソ口を使用する場合があります **CR**、またはのみ **NL**。

次のフォルダにある例を試してください。

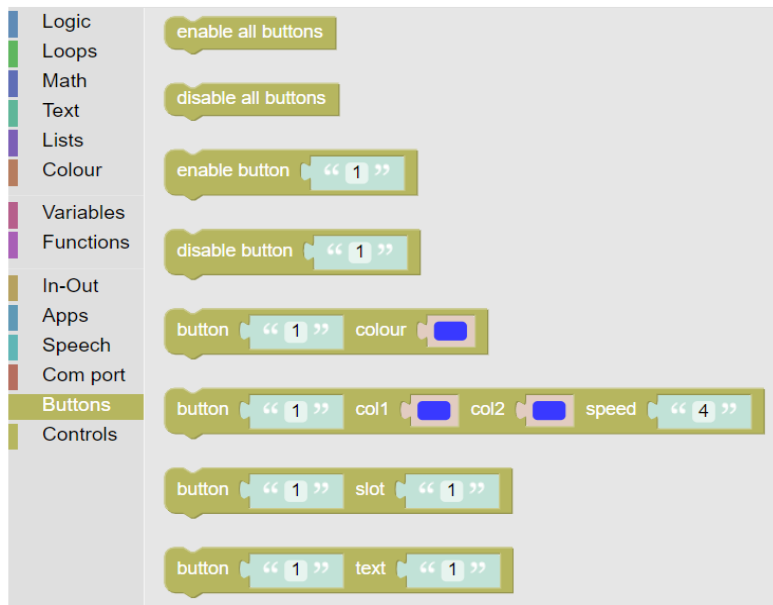
**プログラム\EXAMPLES\COM**

通信テストに役立つかも

アプリケーション [Theremino\\_Terminal](#)。



# メニューボタン - コントロールボタン

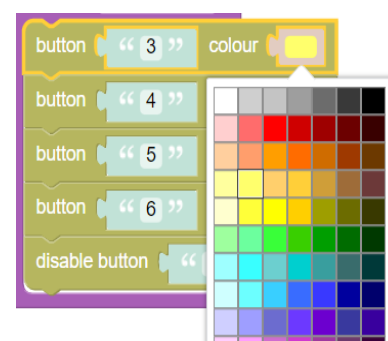


これらのブロックは、コントロール ボタンに作用して次のことを行います。

- それらを有効にする
- それらを無効にする
- それらに色を付ける
- それらを点滅させる
- それらをスロットに関連付けます



ボタンに色を付けると、プログラムをより簡単かつ直感的に使用できるようになります。ボタンの色を変えることもできます実行するか、点滅させます。



実行中のボタンを無効にすると、誤って 2 回押されるのを防ぐのに役立ちます。

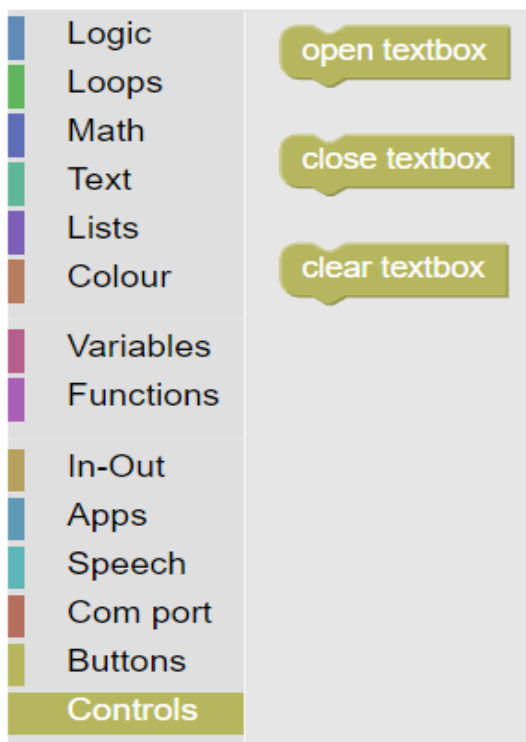
ボタンをスロットにバインドすると、大きな可能性が開かれます。外部条件に基づいて機能を実行できます。たとえば、距離センサーが作業領域に人の存在を検出した場合などです。または、温度が一定のレベルを超えたとき。

ボタンに関連付けられた機能、したがってスロットに関連付けられた機能は、スロットの値が値 500 (当社のシステムでは値の範囲が 0 から 1000 までの値の範囲の半分) を超えたときに実行されます。Slot の値が 500 未満に戻り、再びそれを超えるまで実行されなくなりました。

次のフォルダにある例を試してください。

**プログラム\EXAMPLES\BUTTONS**

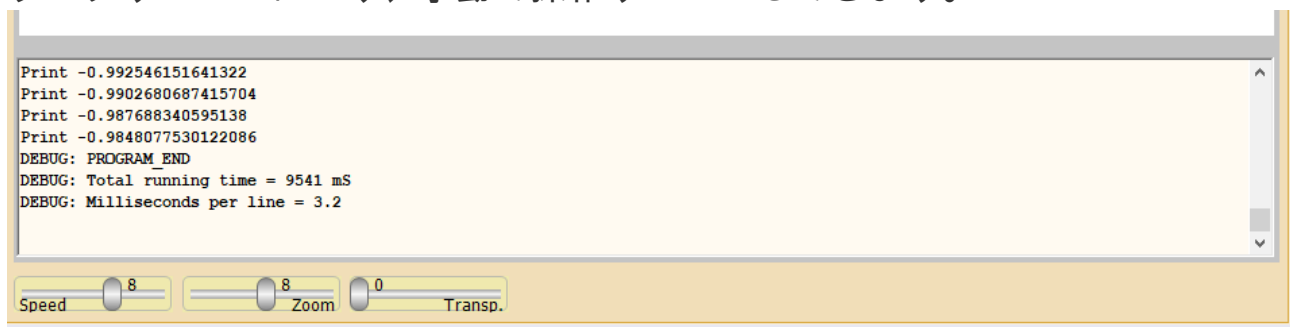
## コントロール メニュー - ビュー コントロール



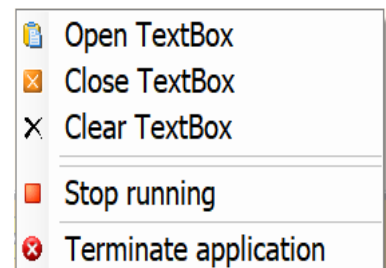
将来的には、このセクションにさらにコントロールを追加する予定ですが、現在のところ、コントロールは大きな水平テキストボックスのみです。

これらのコマンドを使用すると、プログラム自体テキストボックスを開くことができます ( **オープンテキストボックス** )、閉じます ( **近いテキストボックス** ) を含むすべてのテキストを削除し、 ( **クリア テキストボックス** ) 。

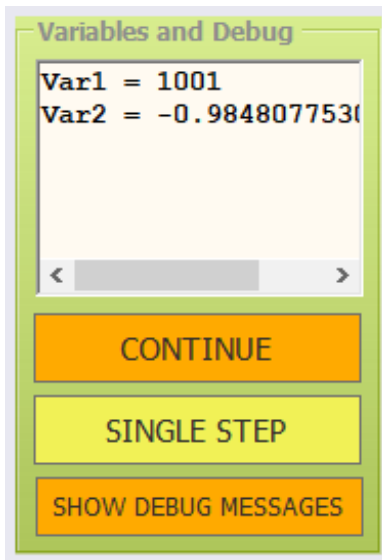
テキスト ボックスは、上部の灰色の線でマウスの左ボタンを押して上下にドラッグすることにより、手動で操作することもできます。



または、テキスト ボックスを右クリックしてメニュー項目を使用することもできます。



# 「変数とデバッグ」パネル



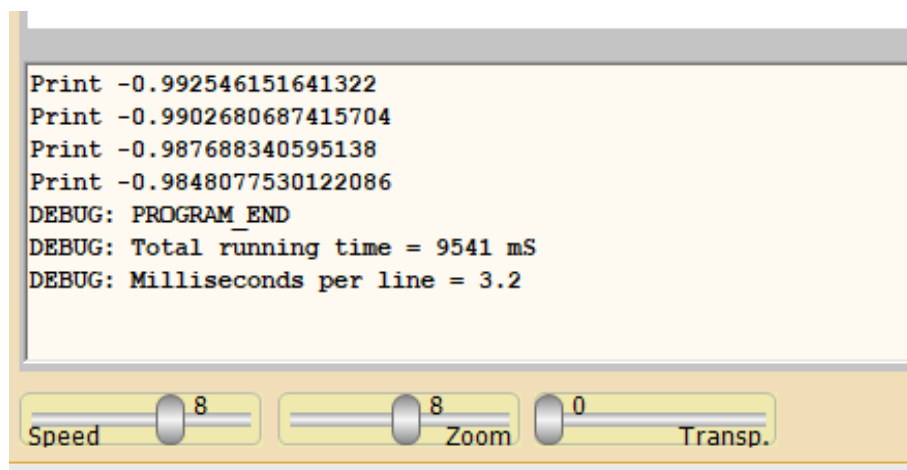
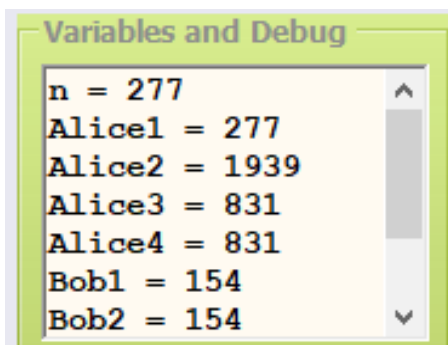
このパネル上のコントロールにより、ソフトウェアの確認とセッアップが容易になります。

- **継続する** プログラムが一時停止している場合、このボタンで再開します。
- **単一段階** プログラムを一時停止すると、一度に 1 行ずつ実行できます。
- **デバッグを表示** 有効 (オレンジ色) の場合、下部のテキスト領域にサービス情報が表示されます。

また、[外部コマンド](#)アプリケーションを一時停止してから段階的にそれを行います。

## プログラムの機能を確認する

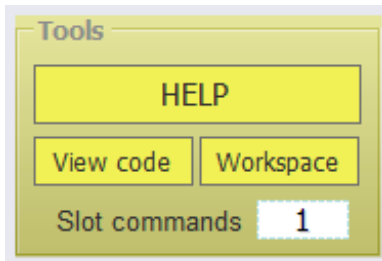
ボタンの場合**デバッグ メッセージを表示する**有効 (オレンジ) で、パネル**変数とデバッグ**表示されている場合、プログラムの実行中に変数の値とすべてのコマンド**デバッグ**彼らが来る継続的に更新されます。



したがって、スライダーを下げることで速度を下げることができます。**スピード**、または一度に 1 行ずつプログラムを実行します**単一段階**、操作の詳細を調査し、プログラムを改善し、エラーを排除します。

# 「ツール」 パネル

このパネルには、学習や外部アプリケーションからのコマンドの送信に役立つサービス コマンドが含まれています。



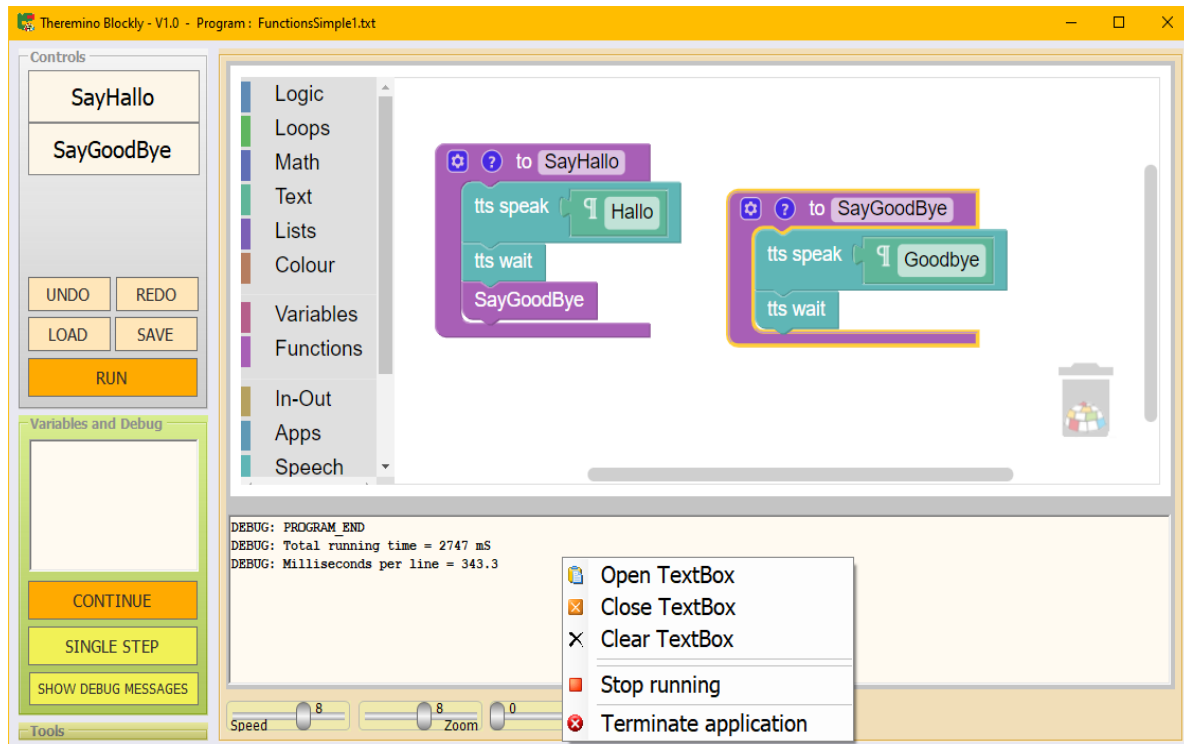
- ◆ **ヘルプ**さまざまな言語のドキュメントを含むフォルダーを開きます。
- ◆ **ビューコード**プログラム コードを Java Script 形式で表示します。
- ◆ **ワークスペース** プログラム ブロックを XML 形式で表示します。これは、Google が表示するために選択した形式です。**ブロックリー**ウェブページで。
- ◆ **スロット コマンド**アプリケーションにコマンドを送信するために使用されるテキストスロットを設定するために使用されます**ブロックリー**私たちのシステムの他のアプリケーションから。外部コマンドを無効にするには、-1 に設定します。それ以外の場合は、1 から 999 までの数値で設定されます。

## 外部コマンドの実行

外部コマンドに使用するテキスト スロットは、上記のように設定されます。他のアプリケーションからの着信コマンドは、**走る**、**止まる**、**一時停止**、**ステップ**、**継続する**そしてまたすべて**関数名ブロックリー**。コマンドは大文字と小文字を区別しません。

# アプリケーションメニュー

マウスの右ボタンをクリックすると (またはタッチ スクリーンに 2 秒間指を離さずに触れると)、次のメニューが表示されます。



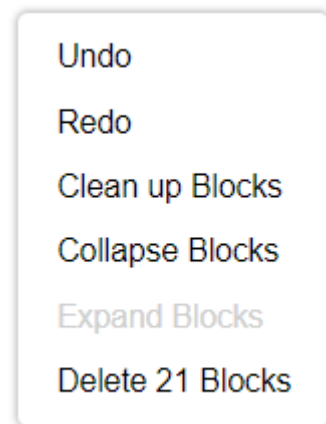
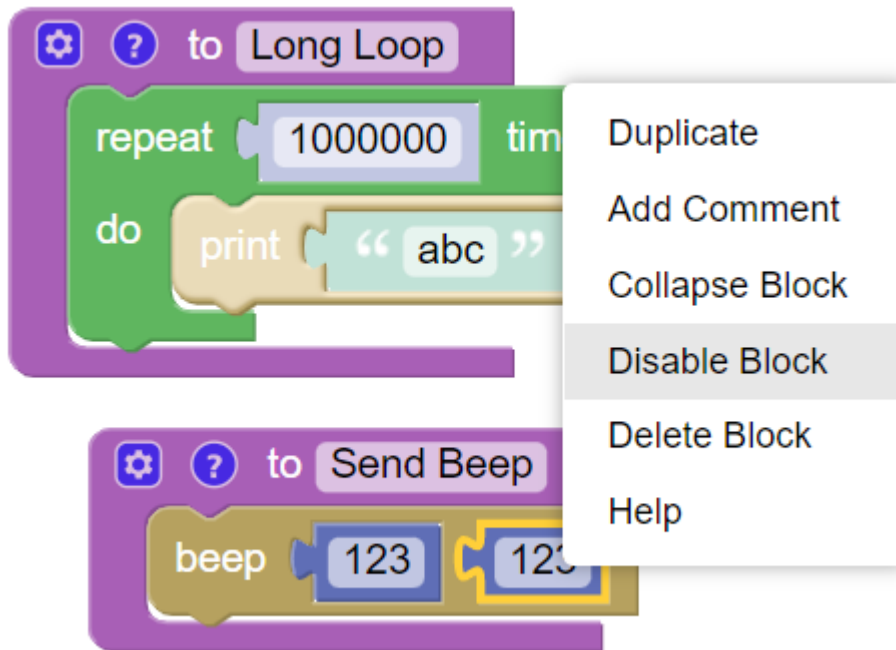
- ◆ **開いた テキストボックス** 下部の水平領域が開きます。ここには、PRINT 命令を記述し、[デバッグ メッセージを表示] ボタンが有効になっている場合 (オレンジ色) にサービス情報 (DEBUG) を送信します。
- ◆ **近い テキストボックス** 下の水平領域を閉じます。
- ◆ **クリア テキストボックス** 下の水平領域からすべてのテキストを削除します。
- ◆ **止まる ランニング**と**終了 応用**プログラムの実行を停止し、アプリケーションを閉じます**ブロックリー**。

横書きテキスト領域の高さを変更するには  
マウスの左ボタンを使用して上下にドラッグします  
テキスト領域とブロック領域を分ける灰色の水平線。

Blockly メニューには 3 つのブロックもあります**コントロール**、  
テキスト領域を開き、閉じて、そこに含まれるテキストを削除します。

## ブロックエリアのメニュー

ブロックの 1 つをマウスの右ボタンでクリックすると、ブロックの複製、縮小、無効化、有効化、削除、機能に関する情報を含む Web ページを開くことができるメニューが表示されます。



代わりにブロック間の白い領域をクリックすると、すべてのブロックに作用する別のメニューが表示され、ブロックを再配置し、サイズを最小に縮小し、展開して削除します。

## ゴミ

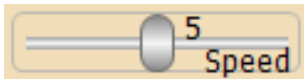
ブロックを削除するには、左側の垂直領域でマウスをドラッグするか、メニュー項目を使用するか、ゴミ箱にドラッグします。

いずれの場合も、バスケットをクリックして作業領域に移動することで、削除されたブロックを復元できます。

または、ボタンを使用してブロックを回復することもできます **元に戻す** このドキュメントの最初のページで説明されています。



## 下部のバー コントロール

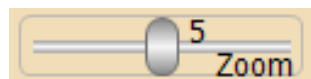


カーソル**速度**実行速度を調整します。

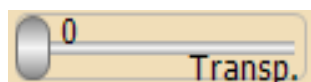
速度範囲「1」(毎秒約 3 行)、「8」(毎秒 500 行)、最大で「12」(かなり高速なコンピュータで毎秒 3,000 行)。

プログラムを書いている間は、中程度の速度を使用するのがよいでしょう。通常速度「4」(毎秒 20 行)、プログラムの実行を視覚的に追跡できるほど遅いです。

次に、プログラムの実行中は、コンピュータの CPU をほとんど使用しない速度「8」を使用することをお勧めします。特に複雑なプログラムを実行する必要があり、ハードウェアを実際に制御する必要がある場合にのみ、より高い速度を使用することをお勧めしますたとえば協働ロボットの場



カーソル**ズーム**プログラム ウィンドウとサービス ボックスの両方で、テキストのサイズを設定します。



このスライダーは、メイン ウィンドウの透明度を調整し、その下も見えるようにします。

# Blockly の複数のインスタンスを実行する

の 2 番目のインスタンスを起動しようとする、**Blockly** 同じファイルから、それは開きません。この動作は、誤って 2 つを開くことを防ぎ、2 つのインスタンスのオプションが混在するのを防ぐことを目的としています。必要に応じて、長押しして 2 つ目のインスタンスを強制的に開くことができます**シフト** 起動中。

2 枚以上ご希望の場合**Blockly** 同時に働き、それぞれに独立したオプションがあり、ファイルをコピーするだけです **Theremino\_Blockly.exe** 別の名前で。

必要に応じてファイルを保持できます **Theremino\_Blockly.exe** 別々のフォルダーに、またはすべて同じフォルダーに。

## Theremino Blockly の制限

の**Blockly** 命令は本質的なものに限定されているため、複雑なタスクの場合は、より強力なプログラミング環境に切り替える必要があります。たとえば、**Theremino\_オートメーション**そして、いくつかの特に困難なケースでは、言語にも **C シャープ**と **VbNet** (と**ビジュアルスタジオ**)。

目安として、ブロックを数えることができます。ブロックが 100 または 200 以上になったら、次のように切り替えるとよいでしょう。**Theremino オートメーション**そして、1,000 行または 2,000 行を超えると、**ビジュアルスタジオ**。

## 実行速度

「解釈された」言語では、「コンパイルされた」言語とは異なり、命令は文字ごとに読み取られ、実行中に解釈されます。したがって、インタープリター言語の実行は遅く、さらにパーティーです。**Blockly**で Google によって書かれました **JavaScript** よりも遅い**ドットネット**。

したがって **Theremino Blockly** よりも約 10 倍遅い **Theremino オートメーション**、それ自体はより 10 倍遅い**ビジュアルスタジオ**(**VbNet**、**C シャープ**、**C++**)。

ただし、これらの言語が意図するタスクの速度は非常に豊富です。実際、何が起きているのかを理解しやすくするために、速度を何千倍も遅くする可能性を追加する必要があります。



# 技術文書

アプリケーション **Theremino\_Blockly** で書かれています **ドットネット** しかし、それはのコードを使用します **ブロックリー** 彼は・・・で生まれた **JS インタープリター** に書かれている **JavaScript**.

ブロック部分はコントロールで実行されます **WebView2** による支援 **マイクロソフトエッジ**.

必要なファイルはすべてすでに存在しています 両方で **ウィンドウズ10** その中 **Windows11**、インストールなし、インターネットなし.

-----

拡張したいすべての開発者 **Theremino\_Blockly** 新しいブロックを使用すると、次のファイルにセクションを追加する必要があります。

## **blockyToolbox.xml**

- このファイルには、ブロックを HTML ページに表示するためのブロックの定義部分が含まれています。

## **theremino\_custom\_blocks.js**

- このファイルの前半には、DotNet アプリケーションへの呼び出しを生成するブロック実行関数が含まれています。
- ファイルの後半には、ブロック特性の定義が含まれています。

一部の特別なブロックでは、追加の機能やファイルの変更も必要になる場合があります

## **theremino.js** 場合によっては、**Module\_Blockly.vb**

これらのトピックの詳細については、ファイルに用意されているドキュメントを参照してください。 **参照** **.rtf** フォルダにあります **ドキュメント** この同じファイルを含む **ヘルプ**.

-----

アップグレードしたい人 **ブロックリー** 交換する必要があります

**blockly\_compressed.js**、**blocks\_compressed.js**と**javascript\_compressed.js**ここからダウンロードできる最新バージョン:<https://github.com/google/blockly/releases>

フォルダ内のこれらのファイルを置き換えた後 **BlocklyFiles\JS** アプリケーションがうまく機能していないか、部分的に機能している可能性があります。この場合、専門のプログラマーが Visual Studio Community 2022 を使用し、問題を修正してアプリケーションを再コンパイルする必要があります。