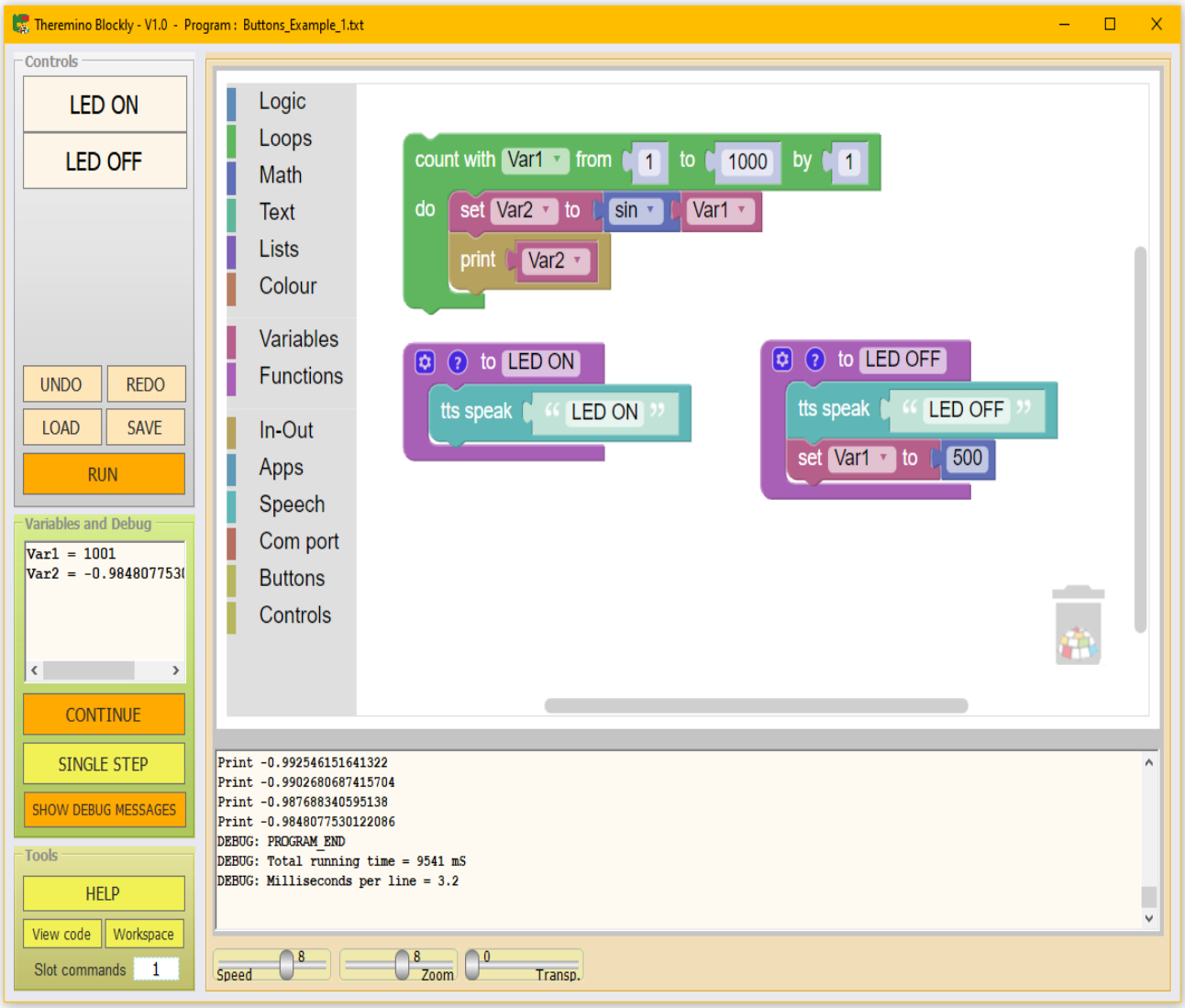
Sistema theremino



Theremino Blockly V1.0

Indice delle pagine

Premesse

Con **Blockly** si impara a programmare senza bisogno di studiare la sintassi dei comandi.

Collegare i blocchi è facile e intuitivo ***anche per i più piccoli*** e il programma viene creato automaticamente.

Theremino_Blockly è il fratello minore di ***Theremino_Automation***, è più semplice ma può comunque accedere a tutte le funzioni principali del nostro sistema, dai semplici **input-output** alla **sintesi** e al **riconoscimento vocale,** al controllo dei **robot collaborativi**, alla misura delle **radiazioni** e del **radon**, al controllo delle **aritmie** cardiache, alle **analisi chimiche**, alle misure con **oscilloscopi** e **analizzatori di spettro**, ecc...

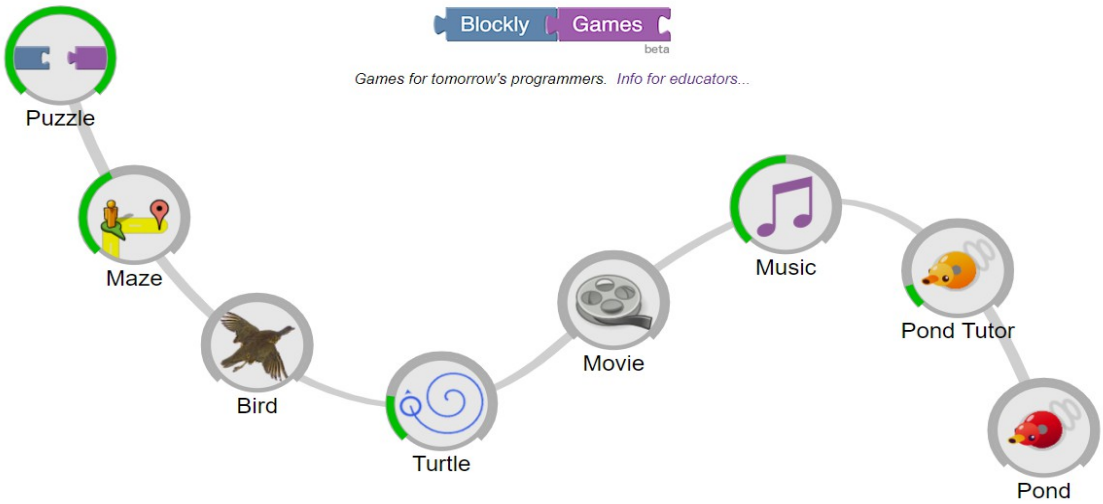
In ***questa pagina*** trovate un indice delle (quasi duecento) applicazioni del nostro sistema, tutte **gratuite** e **Open-Source**.

Primi passi con Blockly

Blockly è stato creato da **Google** e sono disponibili unità didattiche, esempi e piccoli giochi che facilitano l'apprendimento dei suoi fondamenti.

Connettere i blocchi per principianti
https://www.ivana.it/jm/software-on-line/358-blockly

Giochi
https://blockly.games
https://www.brainpop.com/games/blocklymaze
https://blockly.games/music



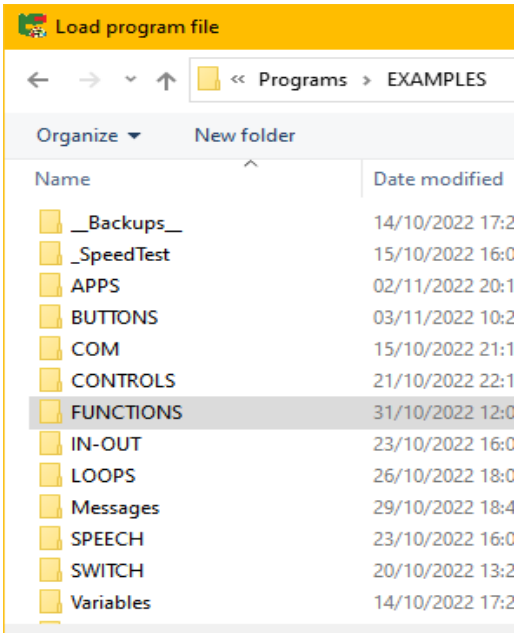
Caricare ed eseguire i programmi

Il pulsante **LOAD** apre una finestra con programmi di esempio da scegliere.

Dopo aver caricato un programma lo si esegue con il pulsante **RUN**, oppure avviando una funzione con i pulsanti che si trovano nella zona superiore.

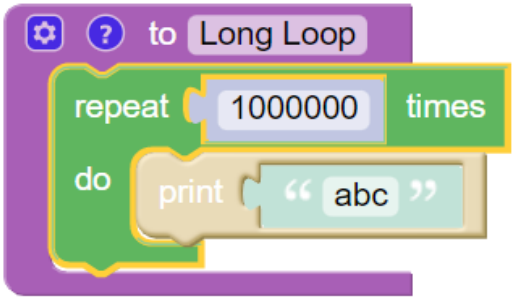
Per fermare il programma si preme il tasto **RUN** (che è diventato **STOP**).

Lo **STOP** avviene in modo automatico anche quando si fa click sulla zona dei blocchi per modificarli.



Semplici programmi per iniziare

Consigliamo di iniziare con i semplici esempi che si trovano nella cartella **Programs**. Eseguite i programmi per vedere cosa fanno. Provate a spostare i blocchi con il mouse, ad aggiungerne altri prendendoli dal menu verticale di sinistra e infine a ripristinare il programma originale con **UNDO** e il **cestino**.



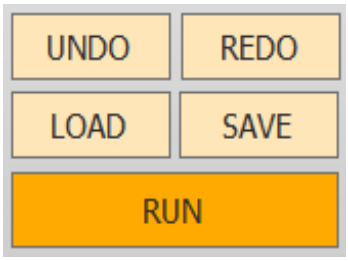
Tutti i file di esempi sono doppi, ad esempio **NomeFile_1** e **NomeFile_2,** così potete fare modifiche e prove, con meno rischi di perdere gli esempi originali.

Ricordatevi comunque di salvare il programma con un nuovo nome prima di iniziare a modificarlo. Comunque se necessario potrete ripristinare tutti gli esempi originali dai file che si scaricano dalla ***pagina di Blockly***.

Modificare e salvare i programmi

Ogni modifica che si fa al programma viene automaticamente scritta nel file, per cui non c'è bisogno di ricordarsi di salvare il lavoro, prima di chiudere la applicazione **Blockly**, o di caricare un altro programma.

Quindi per evitare di modificare gli esempi, **prima di iniziare a fare modifiche** è bene salvare il programma con un nuovo nome, utilizzando il tasto **SAVE**.



- Premendo **SAVE** con il **pulsante sinistro** del mouse, si salva il programma con un nuovo nome.
- Premendo **LOAD** con il **pulsante sinistro** del mouse, si apre una finestra con molti programmi di esempio da scegliere.

- Premendo **SAVE** con il **pulsante destro** del mouse, si salva velocemente il programma.
- Premendo **LOAD** con il **pulsante destro** del mouse, si salva il programma e poi lo si ricarica immediatamente.
- Il pulsante **UNDO** serve per tornare indietro, quando si sono fatte modifiche al programma e si vuole eliminarle.
- Il pulsante **REDO** ricostruisce le modifiche eliminate con **UNDO**.

Al posto di **UNDO** e **REDO** si possono anche usare le combinazioni di tasti **CTRL-Z** e **CTRL-Y** (si preme **CTRL** e tenendolo premuto si preme **Z** o **Y**).

- - - - -

Quando si riaprirà il programma sarà esattamente come lo si era lasciato.
Se si chiude il programma in RUN allora **ripartirà con RUN attivo**.

E se il programma è composto da sole funzioni allora
la prima funzione viene eseguita quando si avvia il programma.

L'area dei blocchi

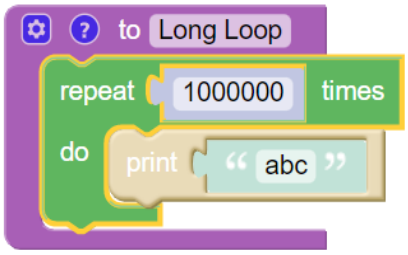
Il funzionamento di **Blockly** è intuitivo il modo migliore di capirlo è provare.

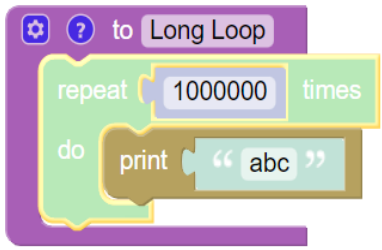
Caricate e provate gli esempi che sono nella cartella **Programs**

Consultate anche la documentazione, gli esempi e i giochi creati da **Google**, che abbiamo indicato all'inizio di questo documento.

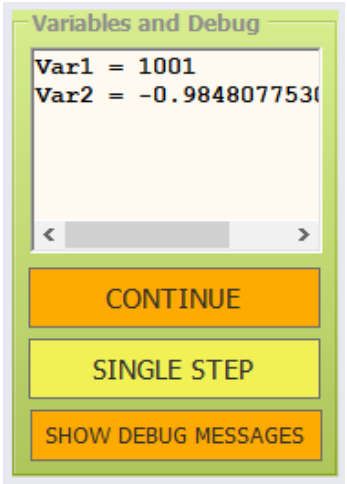
Esecuzione del programma

Durante l'esecuzione i blocchi si schiariscono, come si vede nel blocco verde di queste due immagini. Per vederli meglio abbassate la velocità a 4 o anche meno.

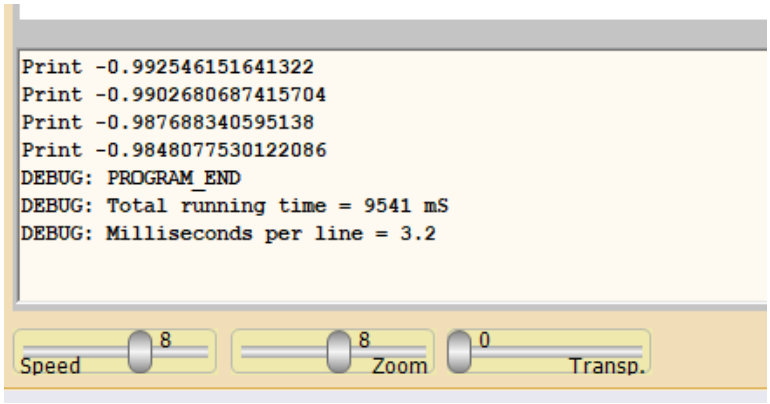




Durante le prove è utile tenere aperto il pannello **Variables and Debug** e abilitare il pulsante **Show debug messages**.

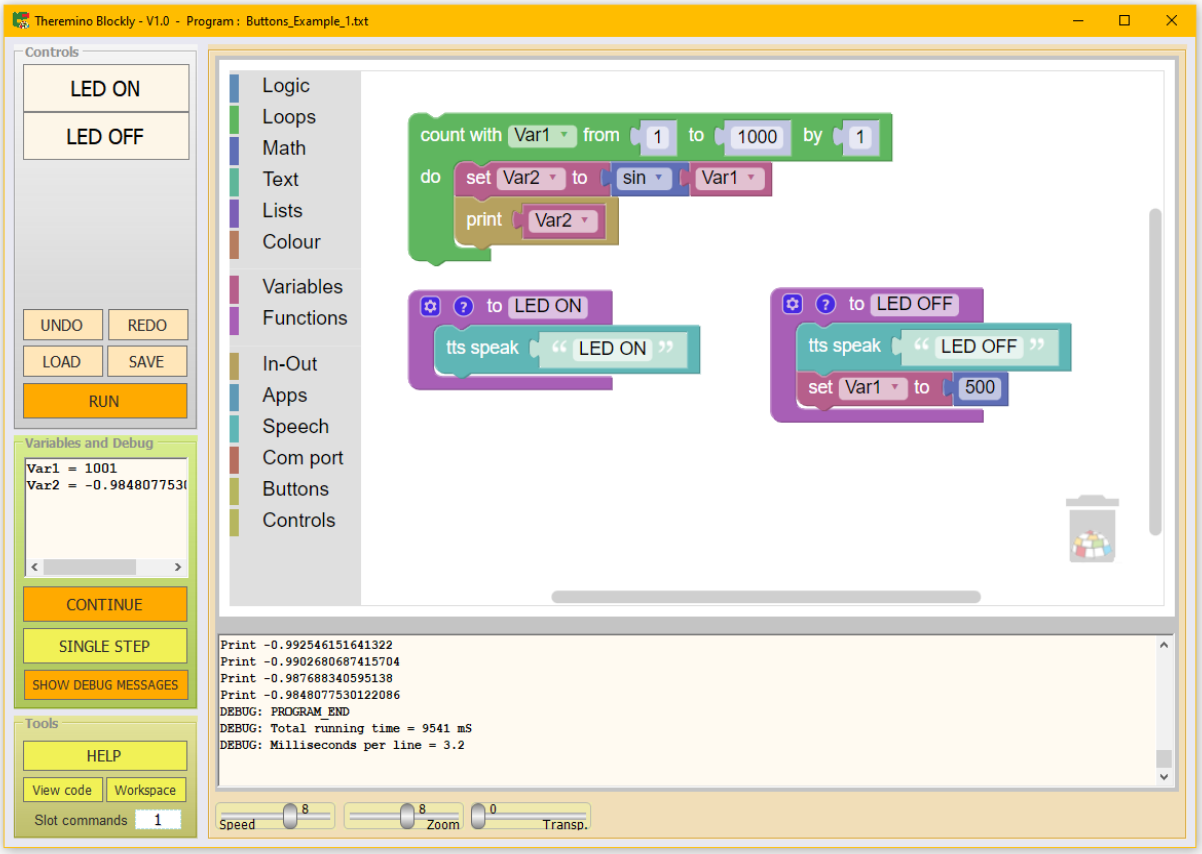


In questo modo nell'area di testo inferiore, oltre al testo inviato con **PRINT**, si vedranno anche le informazioni di servizio precedute dal prefisso **DEBUG**.



I particolari del funzionamento di questi pulsanti sono spiegati nella pagina sul funzionamento del ***pannello "Variables and Debug"***.

Struttura dei programmi



I programmi sono composti da funzioni (in colore viola in questa immagine) e ogni funzione fa apparire un pulsante per eseguirla.

Alcune sezioni possono anche stare al di fuori da ogni funzione, come ad esempio la parte racchiusa in colore verde in questa immagine.

Le parti di programma esterne alle funzioni vengono eseguite (dall'alto in basso) quando si preme RUN e anche quando si avviano le funzioni.

Per evitare che le funzioni eseguano anche le zone di programma esterne, si consiglia di racchiudere queste zone dentro a una funzione. Poi si chiameranno queste funzioni da programma, oppure con i pulsanti.

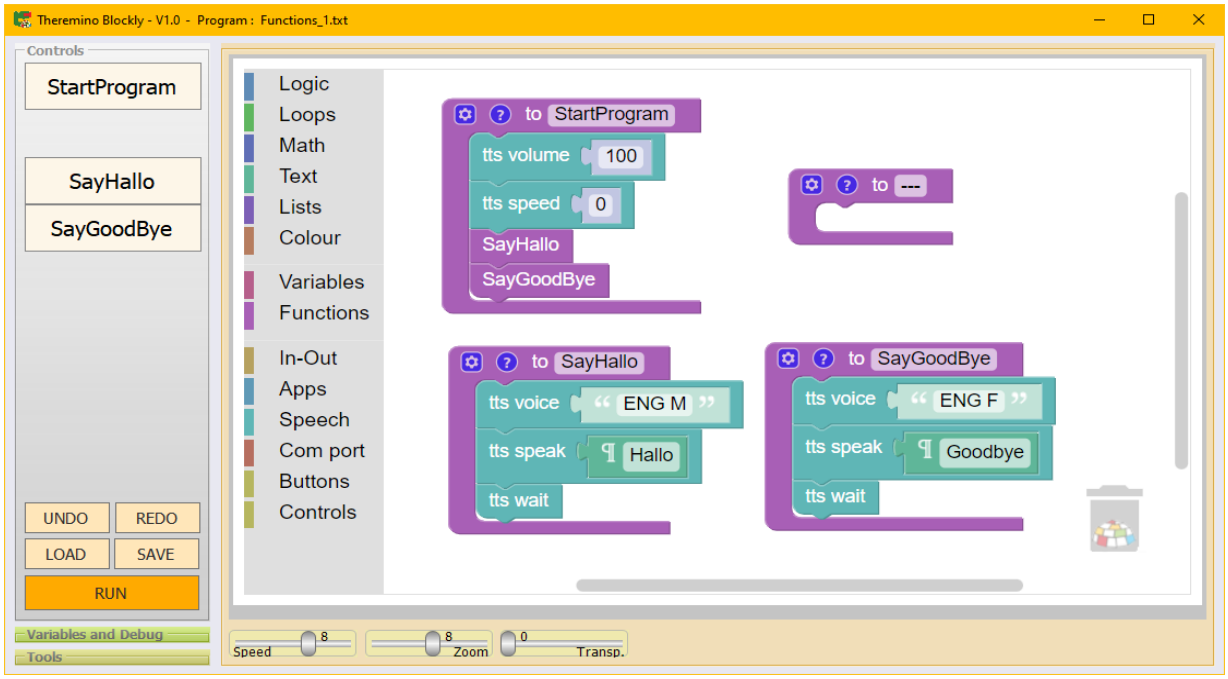
Se non si vuole che le funzioni creino un pulsante basta aggiungere un carattere non alfabetico all'inizio del loro nome. Per vedere un esempio aprite il programma: **EXAMPLES \ SPEECH \ TTS_Example_1.txt**

Se il programma è composto da sole funzioni allora **la prima funzione viene eseguita quando si avvia il programma con RUN.**

I pulsanti delle funzioni

Nel pannello Controls appaiono i pulsanti delle funzioni.

In questa immagine i pulsanti sono **StartProgram**, **SayHello** e **SayGoodbye**.



Utilizzando il menu **Functions** che si trova a metà della **barra dei Tools** di **Blockly** si possono creare le funzioni e dargli un nome significativo.

Quando si creano i blocchi funzione, i pulsanti per eseguirli appaiono automaticamente nel pannello Controls**.**

Quando si modificano i nomi dei blocchi funzione, i pulsanti di controllo vengono automaticamente aggiornati.

Per migliorare la visibilità del testo nei pulsanti si può spezzare il testo su due righe. Si aggiungono **due spazi consecutivi** in un punto opportuno del nome della funzione e il testo si spezza in quel punto.

Tutti gli altri spazi dovranno essere singoli, altrimenti il testo diventerebbe di tre righe e alcune parole sparirebbero in basso.

Se non si vuole che una funzione crei un pulsante si aggiungono caratteri non alfabetici (ad esempio **__** oppure **---**) all'inizio del suo nome.

Se il programma è composto da sole funzioni allora **la prima funzione viene eseguita anche quando si avvia il programma con RUN.**

Pulsanti e funzioni

L'ordine dei pulsanti segue l'ordine dei blocchi (dall'alto in basso e da sinistra a destra) per cui se si vogliono scambiare di posto i pulsanti basta spostare i blocchi.

Si possono anche lasciare posti vuoti tra i pulsanti per raggruppare i pulsanti che eseguono funzioni simili. Per lasciare posti vuoti si aggiungono caratteri non alfabetici all'inizio del nome della funzione, ad esempio **__** o **---** , e poi si spostano i blocchi in modo da portare i posti vuoti nelle posizioni giuste.

Numerazione dei pulsanti delle funzioni

Alcuni blocchi (**disable button, enable button, button colour, button slot** e **button text**) oltre a individuare i pulsanti in base al loro identificatore (il nome della funzione di **Blockly**), possono identificarli anche con un numero.

I pulsanti sono numerati dall'alto in basso. La prima colonna di pulsanti va da 1 a 16. Se ci sono più di sedici pulsanti la seconda colonna va da 16 a 32, la terza da 33 a 48 e così via fino a 128.

Se si cambia l'ordine dei blocchi, e quindi dei pulsanti, si devono controllare e eventualmente modificare, anche le istruzioni che individuano i pulsanti in base al loro numero. Per questo motivo è preferibile specificare i pulsanti con il testo e non con un numero.

L'unico caso in cui potrebbe essere meglio utilizzare un numero è quando si vuole fare la stessa operazione su molti pulsanti, ad esempio utilizzare un ciclo che incrementa una variabile numerica per disabilitarli o colorarli.

Esecuzione delle funzioni

Premendo il pulsante relativo a una funzione ogni altra operazione viene interrotta e la funzione viene eseguita fino alla fine. Quando la funzione finisce la esecuzione riprende da dove era stata interrotta.

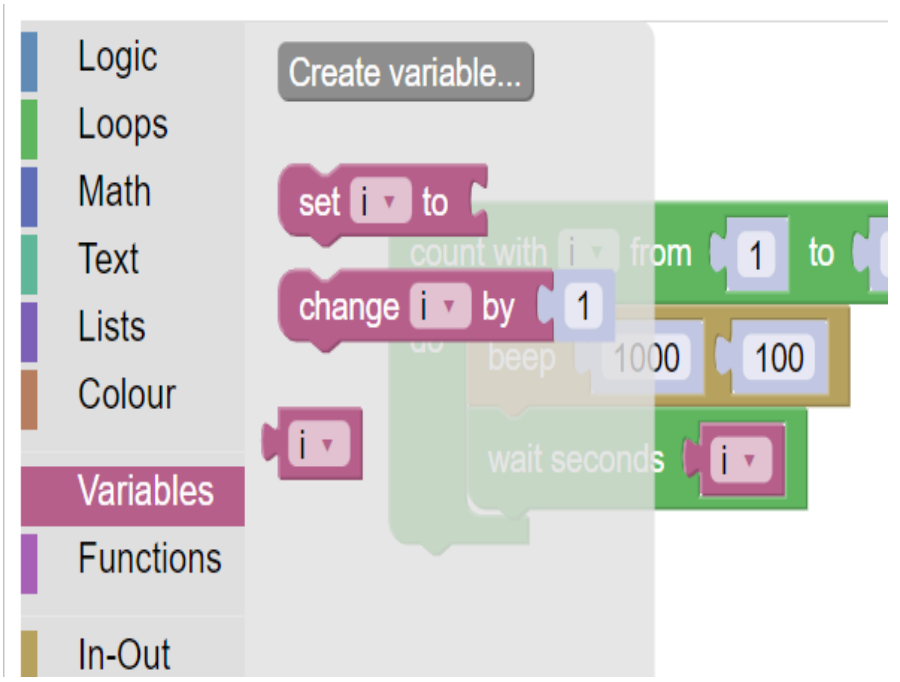
Le funzioni possono essere eseguite anche dal programma.

Le funzioni che si eseguono da programma non dovrebbero creare un pulsante, quindi si aggiungono i caratteri (**__** o **---**) all'inizio del suo nome.

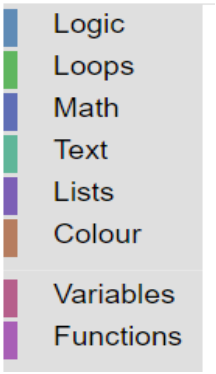
Se il programma è composto da sole funzioni allora **la prima funzione viene eseguita anche quando si avvia il programma con RUN.**

Menu dei blocchi

Facendo click sul menu di sinistra si può scegliere tra oltre cento blocchi diversi, divisi per argomenti e colori.

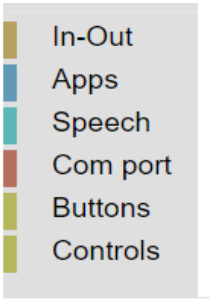


Le prime voci da **Logic** a **Functions** contengono i blocchi fondamentali del linguaggio.



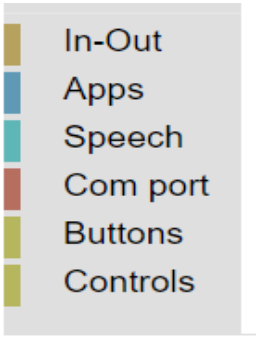
Le istruzioni di questi blocchi sono simili alle istruzioni che si utilizzano nei linguaggi di programmazione più famosi, come ad esempio: VisualBasic, CSharp, Java, C++, Pascal, Python, ecc...

Le sei voci in basso da **In-Out** a **Controls** contengono blocchi specifici, per comunicare con le applicazioni del sistema theremino e quindi anche con l'esterno del PC.



Le spiegazioni dettagliate su queste istruzioni sono nelle prossime pagine.

Menu per le comunicazioni con l'esterno

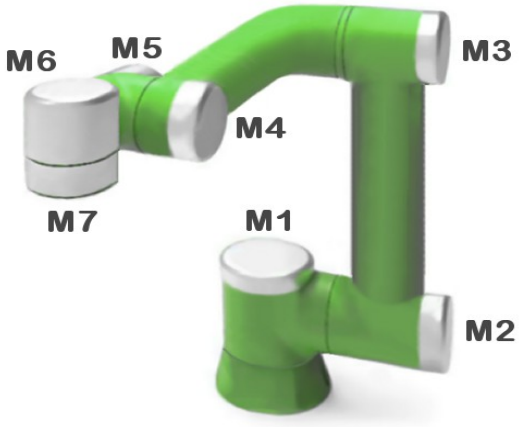


Le ultime sei voci del menu dei blocchi contengono istruzioni specifiche, per comunicare con le applicazioni del sistema theremino e con l'esterno del PC.

Comunicare in tempo reale è importante, tutti gli esseri viventi lo fanno, ma in genere i computer attuali, pur essendo velocissimi, trascurano questi aspetti.

Le applicazioni del nostro sistema lavorano in locale su un PC che utilizziamo come ***PLC***.

Nel nostro sistema il PC stesso controlla direttamente i sensori e i motori e si elimina gran parte dell'hardware di controllo che altrimenti sarebbe costoso e complesso.



Ecco alcuni esempi di quello che si può fare con il nostro software gratuito e Open Source e con poche decine di dollari di motori.

Video1.mp4

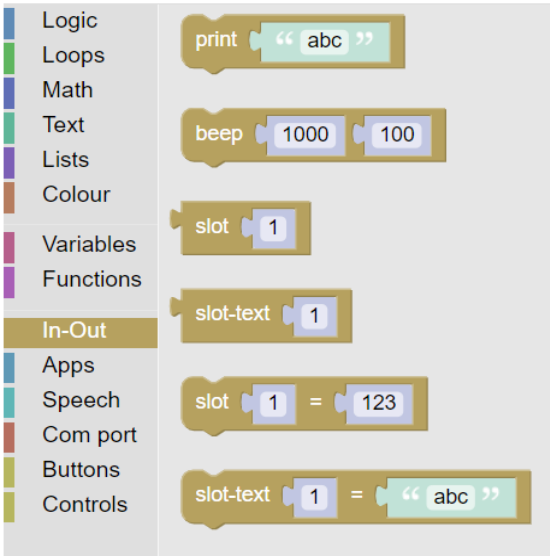
Video2.mp4

Video3.mp4

Video4.mp4

Le pagine web non potrebbero reagire con tempi di risposta adeguati alla sensoristica e alla robotica (dai centesimi ai millesimi di secondo) quindi tutte le applicazioni del nostro sistema lavorano in locale su un PC.

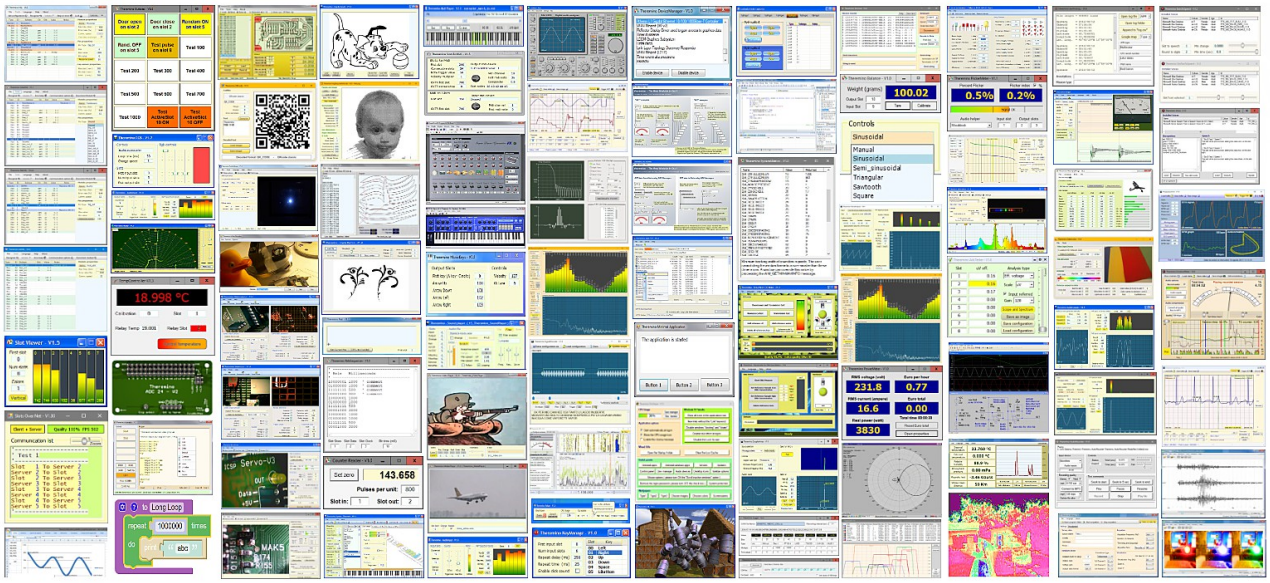
Menu In-Out



I blocchi di questa sezione comunicano con le applicazioni del nostro sistema, ma anche con i sensori, gli attuatori e con gli umani che stanno all'esterno del PC.

Comunicando attraverso gli ***Slot numerici*** e gli ***Slot di testo***, che sono una invenzione esclusiva del nostro sistema, si possono costruire applicazioni complesse anche senza essere esperti in programmazione e in elettronica.

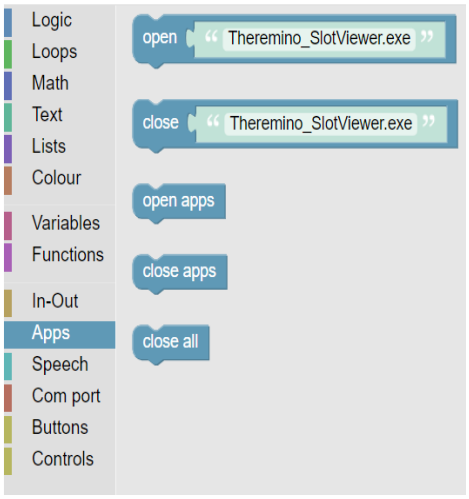
Attraverso gli ***Slot numerici*** e gli ***SlotText*** si comunica con le numerose applicazioni del nostro sistema scritte appositamente per facilitare le operazioni di ***Input-Output***, questa immagine mostra le più importanti.



Ognuna di queste applicazioni si occupa di un compito specifico. Non sarebbe possibile fare tutte queste operazioni in Blockly e nemmeno con i linguaggi più evoluti, perché la complessità diventerebbe ingestibile. Ma suddividendo i compiti su più applicazioni, si possono costruire sistemi molto complessi, ma nello stesso tempo robusti e affidabili.

Tutto questo **software**, completamente **gratuito** e **OpenSource**,
è frutto di oltre dieci anni di continuo lavoro del ***nostro team***.
Per scaricarle e per i particolari di funzionamento aprite ***questa pagina***.

Menu Apps - Aprire files



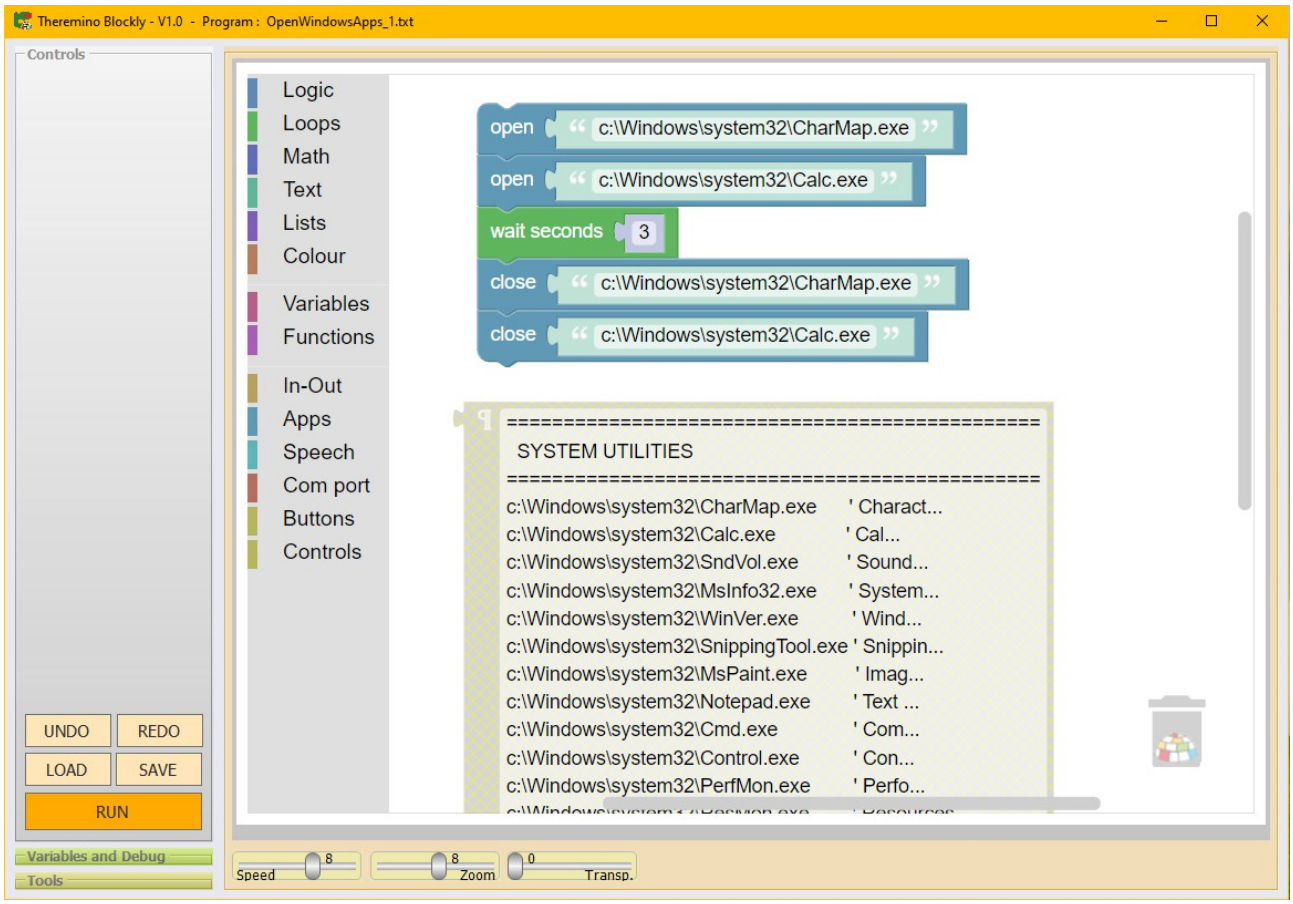
Questi esempi aprono alcune applicazioni del sistema Windows.

```
Open "C:\Windows\Notepad.exe"

Open "C:\Windows\system32\Calc.exe"

Open "C:\Windows\system32\mspaint.exe"
```

L'immagine seguente, presa dall'esempio "OpenWindowsApps", mostra come appaiono i blocchi "Load".

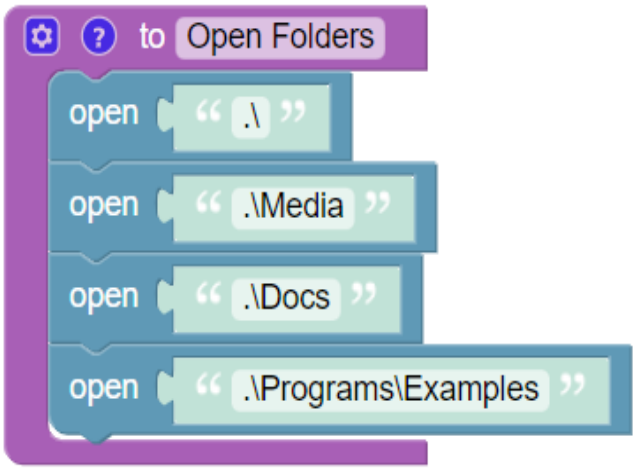


- - - - -

Sperimentate con gli esempi che si trovano in:
Programs \ EXAMPLES \ APPS

Menu Apps - Aprire cartelle

Questi esempi aprono alcune cartelle della applicazione **Theremino_Blockly**. Notare che il punto più una barra rovesciata indica la cartella principale della applicazione **Blockly**, cioè la cartella del file **Theremino_Blockly.exe**




```
Open ".\"
Open ".\Files"
Open ".\Media"
```

Per indicare cartelle e sottocartelle della applicazione **Blockly** il punto e la barra si possono omettere, per cui i tre esempi precedenti possono diventare semplicemente:

```
Open ""    Open "Files"    Open "Media"
```

Per indicare la cartella superiore si utilizza il doppio punto. Ad esempio per aprire la cartella che contiene la cartella del file **Theremino_Blockly.exe** si può scrivere:

```
Open ".."
```

I prossimi esempi aprono il disco C e alcune cartelle del sistema Windows.

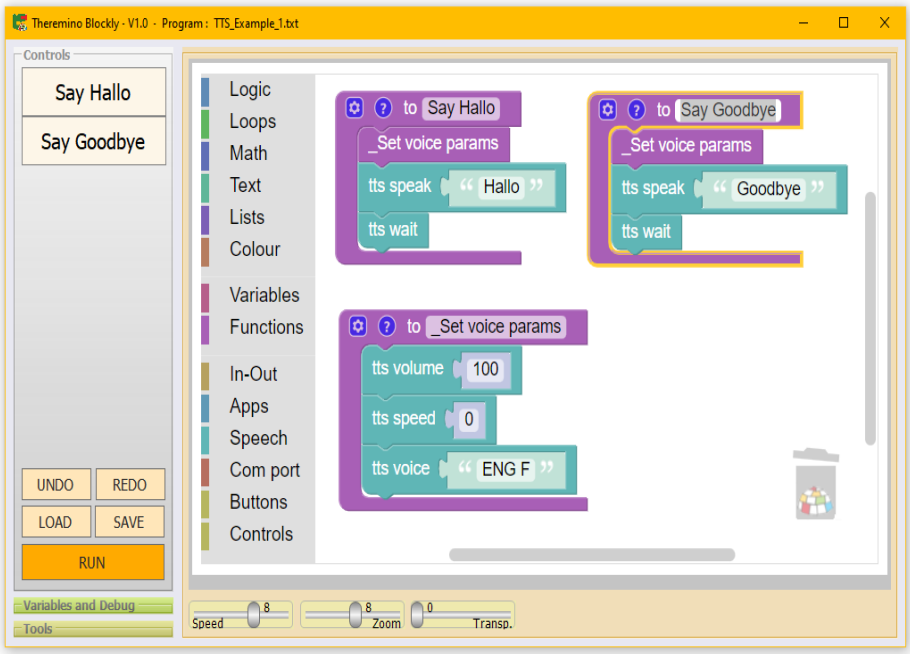
```
Open "C:\"
Open "C:\windows"
Open "C:\windows\system32"
```

- - - - -

Sperimentate con gli esempi che si trovano in:
Programs \ EXAMPLES \ APPS

Menu Speech - Sintesi vocale

Gli esempi della cartella **Programs \ EXAMPLES \ SPEECH** mostrano tutti i comandi che si utilizzano per la sintesi vocale **TTS** (**Text To Speech**).



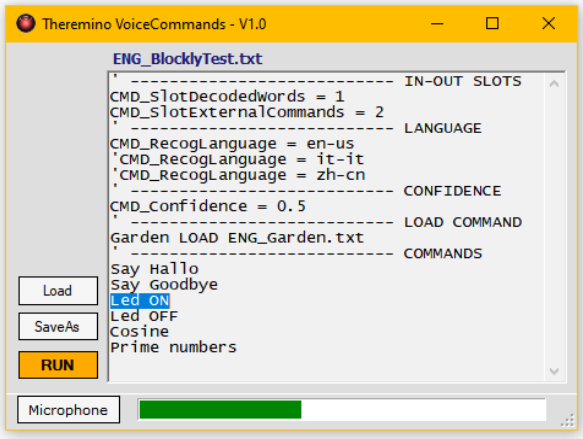
Le lingue disponibili dipendono da Windows e le si impostano con il suo pannello di impostazione delle lingue.

Comandi vocali

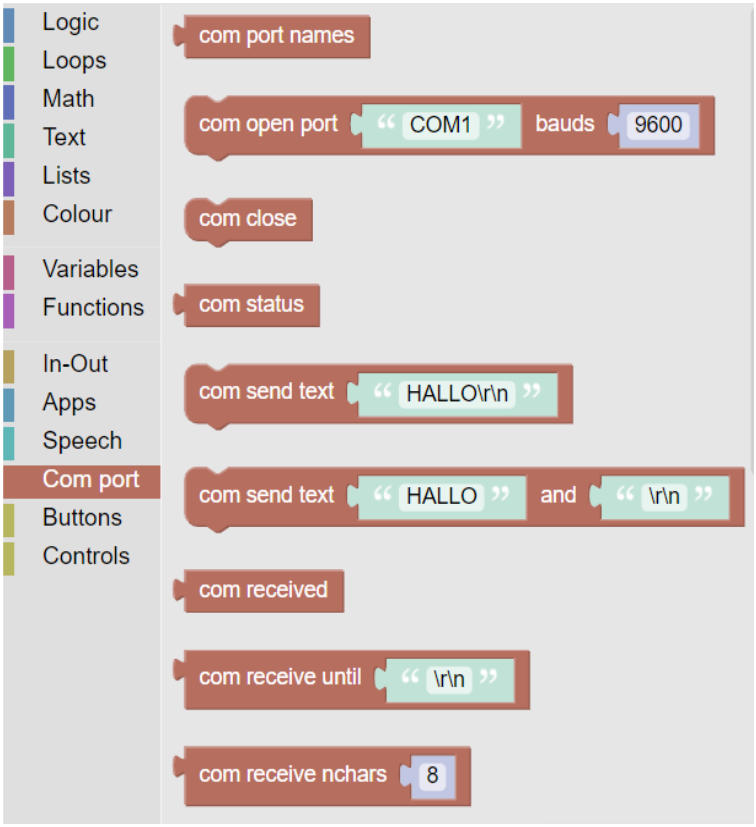
Per i comandi vocali si utilizza la applicazione **VoiceCommands** che si scarica da ***questa pagina***.

I nomi dei ***comandi esterni*** da eseguire si ricevono nello **SlotText** dei comandi di **Blockly**, quindi si deve impostare lo stesso **SlotText** sia in **Blockly** che in **VoiceCommands**.

A sua volta **Blockly** può inviare alla app. **VoiceCommands** i seguenti comandi: **Run**, **Stop** e **Load NomeFile**, e anche in questo caso gli **SlotText** di trasmissione e ricezione devono corrispondere.



Menu Com port - Porte seriali



Per mezzo della porta seriale si possono controllare molti sensori, ad esempio bilance. o sistemi GPS, o attuatori, come ad esempio i motori intelligenti.

In alcuni casi comunicare via seriale può essere molto complesso, quindi prima di scrivere programmi lunghi e difficili, controllate in ***questa pagina***.

Nel nostro sistema esistono già molte comode applicazioni per tutti i casi più comuni.

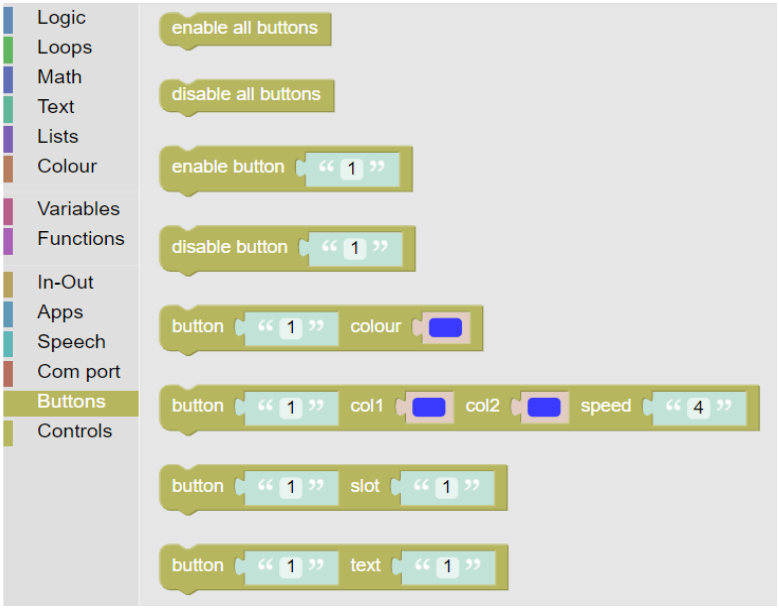
Quando si invia del testo per andare a capo su una nuova linea si utilizzano i caratteri speciali **CR** (ritorno carrello) e **NL** (nuova linea). Ma dato che potrebbero essere parte di parole come **CR**omo o o**NL**us, li si sostituiscono con le sequenze **\r** e **\n**.

In genere nel sistema Windows si utilizzano ambedue e quindi si scrive **\r\n** ma alcuni sistemi o sensori potrebbero utilizzare il solo **CR**, o solo **NL**.

Sperimentate con gli esempi che si trovano nella cartella:
Programs \ EXAMPLES \ COM

Per le prove di comunicazione potrebbe essere utile
la applicazione ***Theremino_Terminal***.

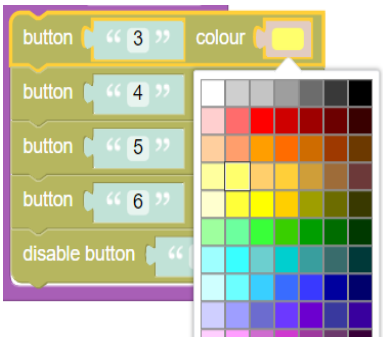
Menu Buttons - Pulsanti di controllo



Questi blocchi agiscono sui pulsanti di controllo per:

- Abilitarli
- Disabilitarli
- Colorarli
- Farli lampeggiare
- Associarli a uno Slot

Colorare i pulsanti può rendere più facile e intuitivo l'uso del programma. Si possono anche colorare diversamente i pulsanti in esecuzione o farli lampeggiare.



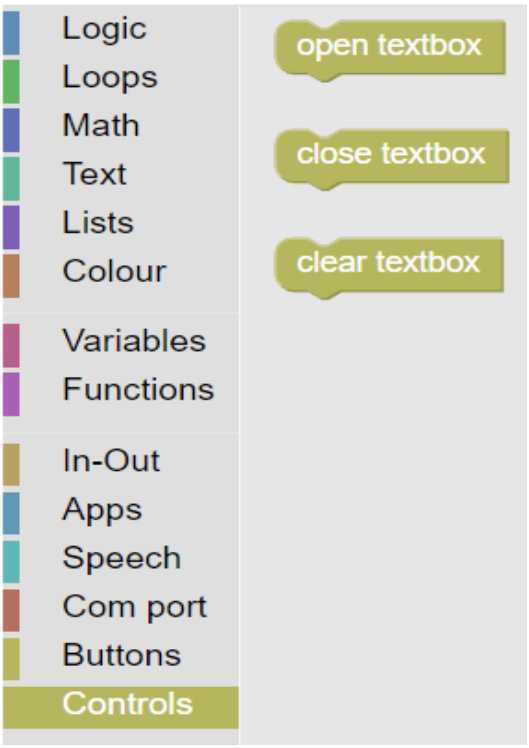
Disabilitare i pulsanti in esecuzione può essere utile per evitare che vengano premuti due volte per sbaglio.

Associare i pulsanti agli Slot apre grandi possibilità. Si possono eseguire funzioni in base a condizioni esterne, ad esempio quando un sensore di distanza individua la presenza di un umano nella zona di lavoro. Oppure quando una temperatura supera un certo livello.

La funzione associata al pulsante, e quindi allo Slot, viene eseguita quando il valore dello Slot supera il valore 500 (metà del campo dei valori che nel nostro sistema i valori vanno da 0 a 1000) e non viene più eseguita fino a che il valore dello Slot torna sotto al 500 e lo supera nuovamente.

Sperimentate con gli esempi che si trovano nella cartella:
Programs \ EXAMPLES \ BUTTONS

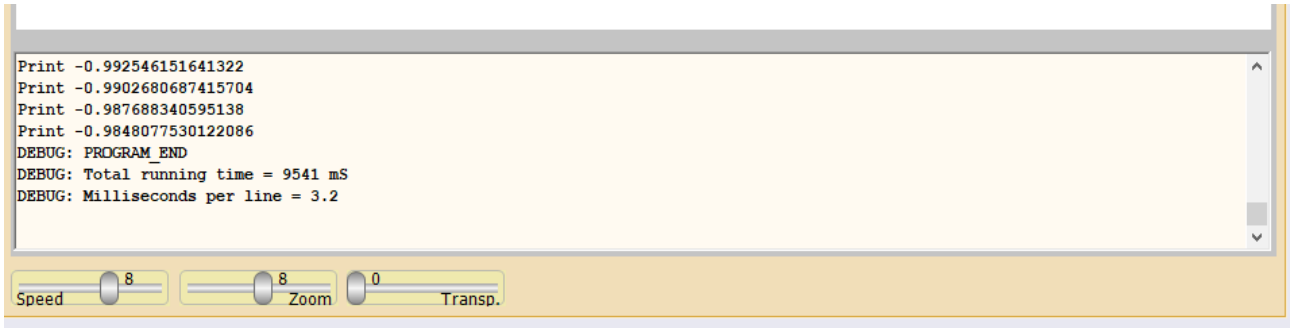
Menu Controls - Controlli di visualizzazione



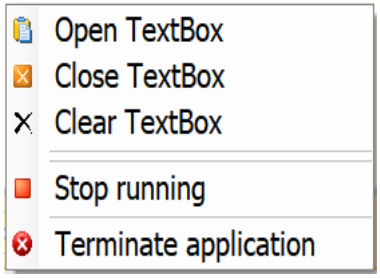
Probabilmente in futuro aggiungeremo altri controlli in questa sezione ma attualmente gli unici controlli sono una grande casella di testo orizzontale.

Con questi comandi il programma stesso può aprire la casella di testo (**open textbox**), chiuderla (**close textbox**) e eliminare tutto il testo che contiene (**clear textbox**).

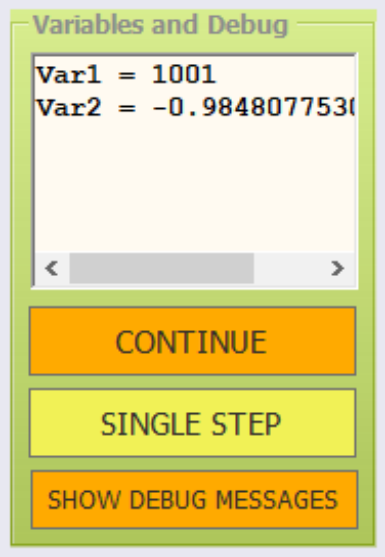
Si può agire sulla casella di testo anche manualmente, premendo il pulsante sinistro del mouse sulla sua riga grigia superiore e trascinandola in alto o in basso.



Oppure si può fare click con il tasto destro del mouse sulla casella di testo e usare le voci del menu.



Il pannello "Variables and Debug"



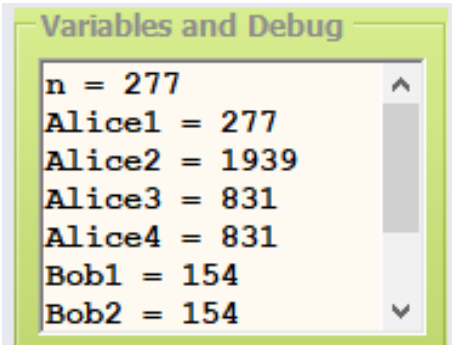
I comandi di questo pannello facilitano il controllo e la messa a punto del software.

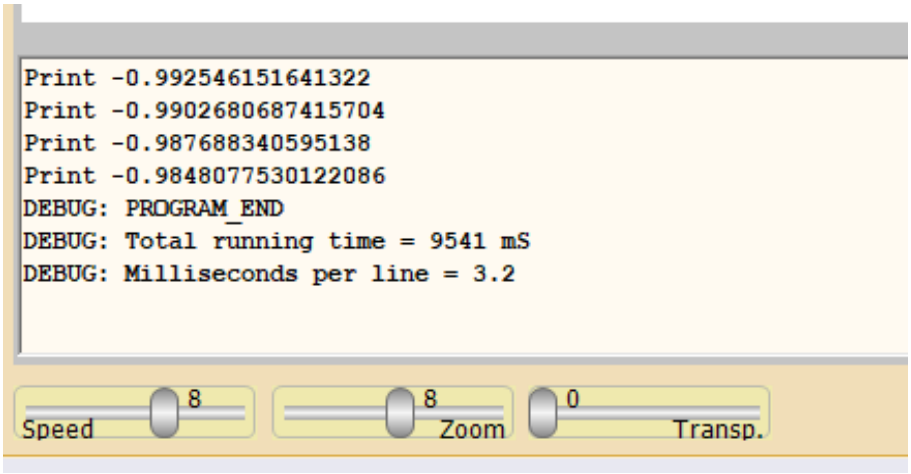
- **CONTINUE** Quando il programma è in pausa questo pulsante lo fa ripartire.
- **SINGLE STEP** Quando il programma è in pausa si può eseguirlo una riga per volta.
- **SHOW DEBUG** Se abilitato (arancione) mostra le informazioni di servizio nell'area di testo inferiore.

Volendo si può anche usare un ***comando esterno*** per mettere in pausa la applicazione e poi eseguirla passo passo.

Verificare il funzionamento dei programmi

Se il pulsante **Show debug messages** è abilitato (arancione) e il pannello **Variables and Debug** è visibile, allora durante l'esecuzione del programma i valori delle variabili e tutti i comandi di **debug** vengono continuamente aggiornati.

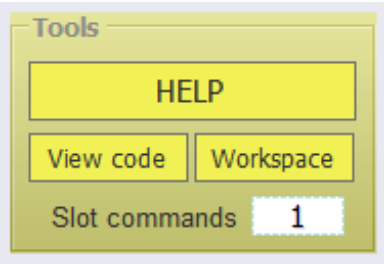




E' quindi possibile ridurre la velocità, abbassando il cursore **Speed**, o eseguire il programma una riga per volta con **Single step**, per esplorare i particolari del funzionamento, migliorare il programma ed eliminare gli errori.

Il pannello "Tools"

Questo pannello contiene comandi di servizio, utili per l'apprendimento e per inviare comandi da applicazioni esterne.



- **HELP** apre la cartella che contiene la documentazione in varie lingue.
- **View-code** visualizza il codice del programma nel formato Java Script.
- **Workspace** visualizza i blocchi del programma nel formato XML, che è il formato scelto da Google per visualizzare **Blockly** nelle pagine WEB.
- **Slot commands** serve per impostare lo Slot di testo da utilizzare per inviare comandi alla applicazione **Blockly** da altre applicazioni del nostro sistema. Per disabilitare i comandi dall'esterno si imposta a -1. Altrimenti lo si imposta con un numero da 1 a 999.

Esecuzione dei comandi esterni

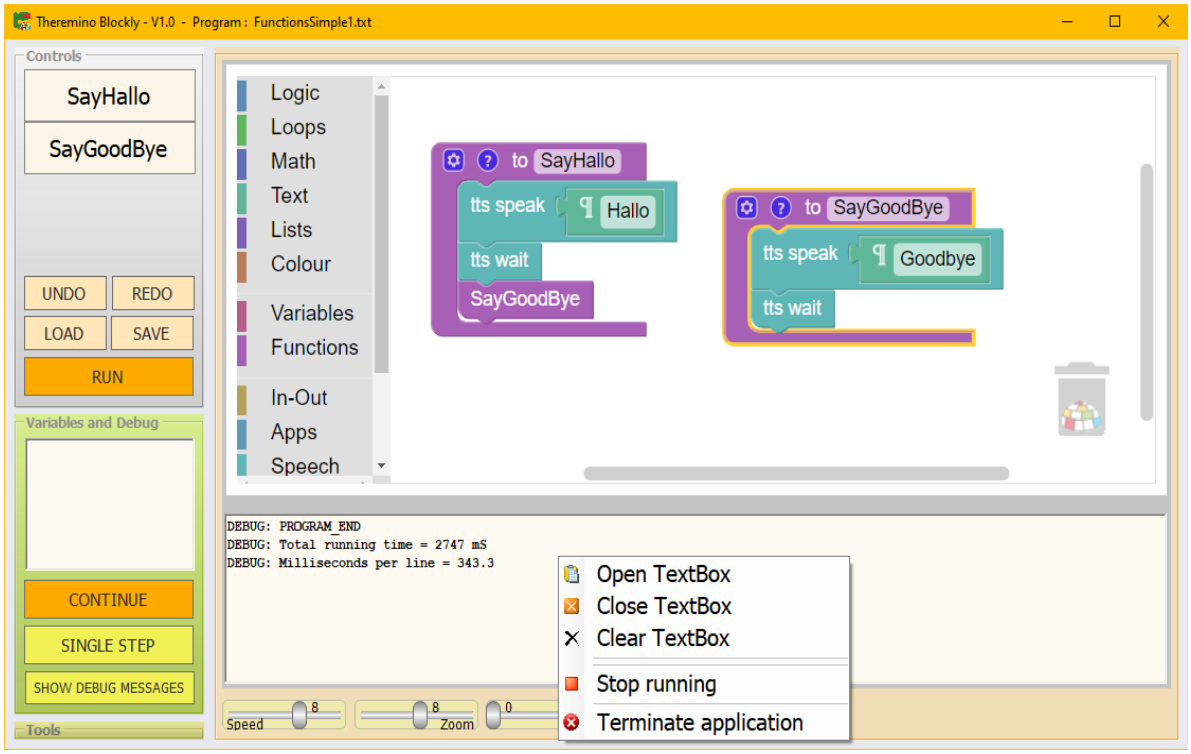
Lo Slot di testo che da usare per i comandi esterni si imposta come spiegato qui sopra.

I comandi in arrivo da altre applicazioni possono essere **RUN**, **STOP**, **PAUSE**, **STEP**, **CONTINUE** e anche tutti i **nomi delle funzioni di Blockly**.

I comandi non fanno distinzione tra caratteri maiuscoli e minuscoli.

Il menu della applicazione

Facendo click con il tasto destro del Mouse (o toccando lo schermo tattile senza staccare il dito per due secondi), si apre il menu che si vede qui sotto.



- **Open TextBox** apre l'area orizzontale inferiore, dove si scrive con le istruzioni PRINT e dove si inviano le informazioni di servizio (DEBUG) se il pulsante "Show debug messages" è abilitato (di colore arancio).

- **Close TextBox** chiude l'area orizzontale inferiore.

- **Clear TextBox** elimina tutto il testo dall'area orizzontale inferiore.

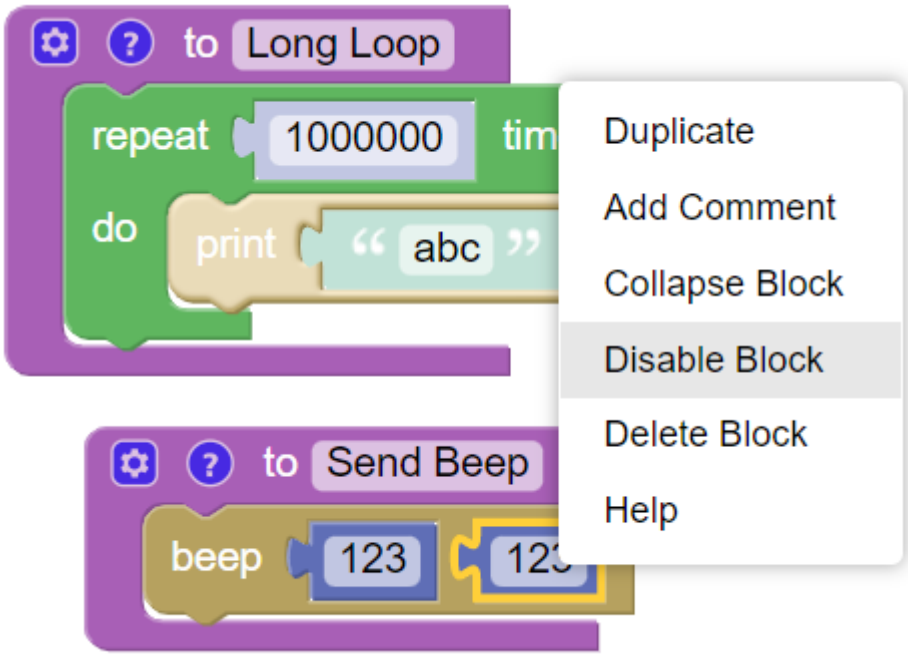
- **Stop running** e **Terminate application** fermano l'esecuzione del programma e chiudono la applicazione **Blockly**.

Per modificare l'altezza dell'area di testo orizzontale
si utilizza il pulsante sinistro del mouse e si trascina in alto e in basso
la riga grigia orizzontale che divide l'area di testo dalla zona dei blocchi.

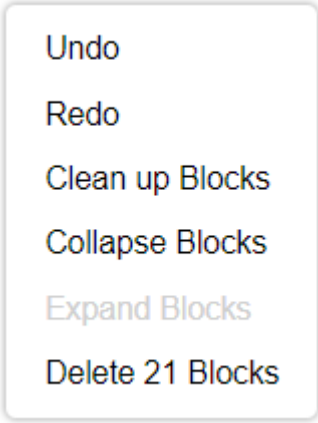
Esistono anche tre blocchi, nel menu di Blockly **Controls**,
per aprire l'area di testo, chiuderla e cancellare il testo che contiene.

I menu della zona dei blocchi

Facendo click con il tasto destro del mouse su uno dei blocchi, appare un menu che permette di duplicare i blocchi, ridurli, disabilitarli, abilitarli, eliminarli e anche aprire pagine web di informazioni sul loro funzionamento.



Facendo invece click nell'area bianca tra i blocchi appare un altro menu che agisce su tutti i blocchi, per riordinarli, ridurre le loro dimensioni al minimo, espanderli e eliminarli.



Il cestino

Per eliminare i blocchi li si trascina con il mouse nella zona verticale sinistra, o si usano le voci dei menu, oppure li si trascina sopra al cestino.

In tutti i casi si potranno recuperare i blocchi eliminati facendo click sul cestino e spostandoli nell'area di lavoro.

In alternativa si potrebbero recuperare i blocchi utilizzando il pulsante **UNDO** spiegato nelle prime pagine di questo documento.

I controlli della barra inferiore

Il cursore **SPEED** regola la velocità di esecuzione.

Le velocità vanno da ”1” (circa tre righe al secondo), a ”8” (cinquecento righe al secondo), e fino a ”12” (tremila righe al secondo su computer abbastanza veloci).

Mentre si scrive il programma è bene utilizzare una velocità media. Di solito la velocità ”4” (20 righe al secondo), che è abbastanza lenta da poter seguire visivamente l'esecuzione del programma.

Poi, durante l'esecuzione dei programmi, si consiglia di utilizzare la velocità "8", che impegna poco la CPU del computer, e le velocità superiori solo nel caso si debbano eseguire programmi particolarmente complessi e che devono controllare hardware in tempo reale, ad esempio per i robot collaborativi.

Il cursore **ZOOM** stabilisce la dimensione del testo, sia nella finestra del programma, sia nelle caselle di servizio.

Questo cursore regola la trasparenza della finestra principale e permette di vedere anche sotto di essa.

Eseguire più istanze di Blockly

Se provate a lanciare una seconda istanza di **Blockly** dallo stesso file, questa non si aprirà. Si tratta di un comportamento voluto per impedire di aprirne due per sbaglio e quindi evitare che le opzioni delle due istanze si mischino. Se necessario si può forzare l'apertura di una seconda istanza tenendo premuto **SHIFT** durante l'avvio.

Se si vogliono due o più copie di **Blockly** che funzionino contemporaneamente, ciascuna con opzioni indipendenti, basta copiare il file **Theremino_Blockly.exe** con un diverso nome.

Se si preferisce si possono tenere i file **Theremino_Blockly.exe** in cartelle separate o anche tutti nella stessa cartella.

Limiti di Theremino Blockly

In **Blockly** le istruzioni sono limitate all'essenziale, quindi per compiti complessi si deve passare ad ambienti di programmazione più potenti, ad esempio **Theremino_Automation** e, in alcuni casi particolarmente difficili, anche ai linguaggi **CSharp** e **VbNet** (con **Visual Studio**).

Come linea guida si possono contare i blocchi, se diventano più di cento o duecento è bene passare a **Theremino Automation** e poi, oltre le mille o duemila righe è meglio passare a **Visual Studio**.

Velocità di esecuzione

Nei linguaggi “interpretati”, a differenza di quelli “compilati”, le istruzioni vengono lette, carattere per carattere, e interpretate durante l'esecuzione stessa. L'esecuzione di un linguaggio interpretato è quindi più lenta e inoltre la parte **Blockly** è stata scritta da Google in **JavaScript** che è più lento di **DotNet**.

Quindi **Theremino-Blockly** è circa dieci volte più lento di **Theremino-Automation**, che a sua volta è dieci volte più lento di **Visual Studio** (**VbNet**, **Csharp**, **C++**).

La velocità è comunque sovrabbondante per i compiti a cui sono destinati questi linguaggi. Infatti abbiamo dovuto aggiungere la possibilità di rallentarli, anche di migliaia di volte, per facilitare la comprensione di quello che accade.

Documentazione tecnica

La applicazione **Theremino_Blockly** è scritta in **DotNet** ma utilizza il codice di **Blockly** e del **JS-Interpreter** che sono scritti in **Java-Script.**

La parte dei blocchi viene eseguita in un controllo **WebView2** supportato da **Microsoft Edge.**

Tutti i file necessari sono già presenti sia in **Windows 10** che in **Windows 11**, senza installazione e senza Internet.

- - - - -

Eventuali sviluppatori che volessero espandere **Theremino_Blockly** con nuovi blocchi dovranno aggiungere sezioni nei seguenti file:

blockyToolbox.xml

- Questo file contiene la parte di definizioni dei blocchi per visualizzarli nella pagina HTML

theremino_custom_blocks.js

- La prima metà di questo file contiene le funzioni di esecuzione dei blocchi che generano chiamate alla applicazione DotNet.
- La seconda metà del file contiene le definizioni delle caratteristiche dei blocchi.

Alcuni blocchi speciali potrebbero aver bisogno di funzioni aggiuntive o di modifiche anche nel file **theremino.js** e in alcuni casi anche nel **Module_Blockly.vb**

Per approfondire questi argomenti consultate la documentazione che abbiamo preparato nel file **References.rtf** che si trova nella cartella **Docs** che contiene questo stesso file di **HELP**.

- - - - -

Chi volesse aggiornare i file di **Blockly** dovrà sostituire **blockly_compressed.js**, **blocks_compressed.js** e **javascript_compressed.js** con le ultime versioni che si scaricano da qui: ***https://github.com/google/blockly/releases***

Dopo aver sostituito questi file nella cartella **Blockly \ Files \ JS** ci sono buone probabilità che la nostra applicazione funzioni male o parzialmente. In questo caso un programmatore esperto dovrà utilizzare Visual Studio Community 2022, risolvere i problemi e ricompilare la applicazione.