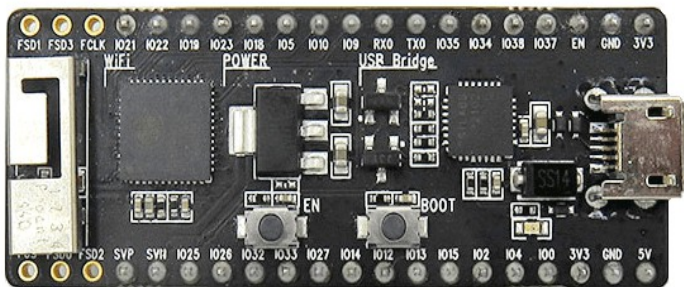


theremino
•the•real•modular•in-out•

Sistema theremino

Theremino IoT HAL V2.x Istruzioni

Principio di funzionamento

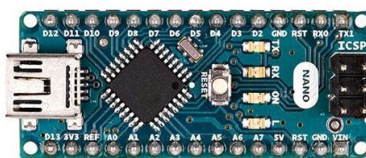


Gli **IoTModule (ESP32)** si alimentano a 5 volt e comunicano via WiFi con un Access Point, oppure direttamente con i PC dotati di un modulo WiFi che supporti la funzione HotSpot mobile. Ad esempio un Notebook o un Tablet recente e con Windows 10.

Utilizzando un Access Point che fa da ponte, qualunque dispositivo con Windows può collegarsi agli IoTModule.

Si possono quindi leggere i dati sul PC e anche trasferirli da uno IoTModule a un altro. Si potrebbe ad esempio collegare un potenziometro a uno IoTModule e con questo ruotare un servo motore che si trova su un altro.

Si possono anche trasferire i dati, da e verso i moduli Master, i NetModule e gli Arduino, in qualunque numero e combinazione.



Con la applicazione [IoT HAL](#) si configurano i Pin di ingresso uscita, per leggere sensori, muovere motori ecc..

A seconda dei modelli, si possono collegare fino a 26 o 28 sensori e/o attuatori.

Si utilizzano le oltre cento app. del sistema Theremino che coprono quasi tutti i campi, dagli esperimenti scientifici alla musica, alla radioattività, alla didattica... vedere [questa pagina](#).

Tutto funziona subito, senza scrivere una sola riga di firmware o di software, e senza installazioni.

Theremino IoT HAL - V0.5

File Strumenti Lingua Aiuto Informazioni Calibrate

Riconosci Valida Errori Opzioni di comunicazione Disconnetti modulo

Type	ID	Subtype	Slot	Value	Notes
Module	1	Test_ADC			Version 60-1
Pin	36	Adc_16	0	0.0	
Pin	39	Adc_16	1	0.0	
Pin	25	Unused			
Pin	26	Unused			
Pin	32	Adc_16	4	195.4	
Pin	33	Adc_16	5	61.6	
Pin	27	Unused			
Pin	14	Unused			
Pin	12	Unused			
Pin	13	Unused			
Pin	15	Unused			
Pin	2	Unused			
Pin	4	Unused			
Pin	0	Unused			
Pin	37	Adc_16	14	0.0	
Pin	38	Adc_16	15	0.0	
Pin	34	Adc_16	16	0.0	
Pin	35	Adc_16	17	0.0	
Pin	1	Unused			
Pin	3	Unused			
Pin	9	Unused			
Pin	10	Unused			
Pin	5	Dig_in	22	0.0	
Pin	18	Dig_in	23	0.0	
Pin	23	Dig_in	24	0.0	
Pin	19	Dig_in	25	0.0	
Pin	22	Dig_in	26	0.0	
Pin	21	Dig_in	27	0.0	

Proprietà del Modulo

Nome FillCAP

Ripetizione (fps) 311

Tasso errore 0.00

Velocità 12

Primo utilizzo dei moduli

La prima volta si deve preparare l'IDE di Arduino, con questa procedura

1) Installare l'IDE di Arduino e avviarlo.

2) Seguire queste istruzioni:

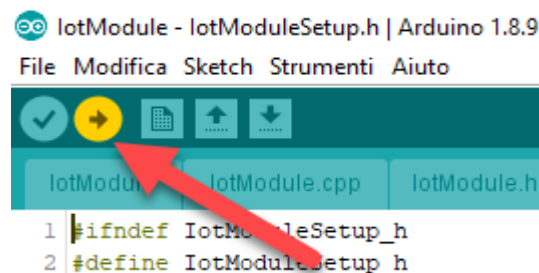
- Aprire "File" / "Impostazioni"
 - Nelle riga "URL aggiuntive..." copiare la riga seguente:
`https://dl.espressif.com/dl/package_esp32_index.json`
 - Andare in "Strumenti" / "Scheda: xxx" / "Gestore schede..."
 - Scrivere "ESP32" nella barra superiore, selezionarlo e installarlo.
 - Selezionare "Strumenti" / "Scheda: xxx" / "ESP32 Arduino" / "ESP32 Pico Kit"
 - Selezionare "Strumenti" / "Partition scheme" / "Minimal SPIFFS ..."
- (istruzioni tratte da: [Installing the ESP32 board in Arduino IDE](#))

Per programmare lo **lotModule**, seguire questa procedura

- ◆ Avviare l'IDE di Arduino.
- ◆ Selezionare "File" / "Apri" e caricare il file "lotModule.ino"
- ◆ Selezionare il file "lotModuleSetup.h" (nella barra superiore)
- ◆ Impostare il nome della propria rete WiFi e la password.
- ◆ Modificare la riga `IOTMODULE_PINOUT_TYPE` a seconda del tipo di modulo usato (impostare 0/1/2 per WROOM / PICO-KIT / TTGO-T7)
- ◆ Le altre opzioni del "Setup" sono spiegate in [questa pagina](#).
- ◆ Controllare la lista di porte in "Strumenti" / "Porta", poi collegare il modulo, individuare la nuova porta COM, che si è aggiunta alla lista, e sceglierla.

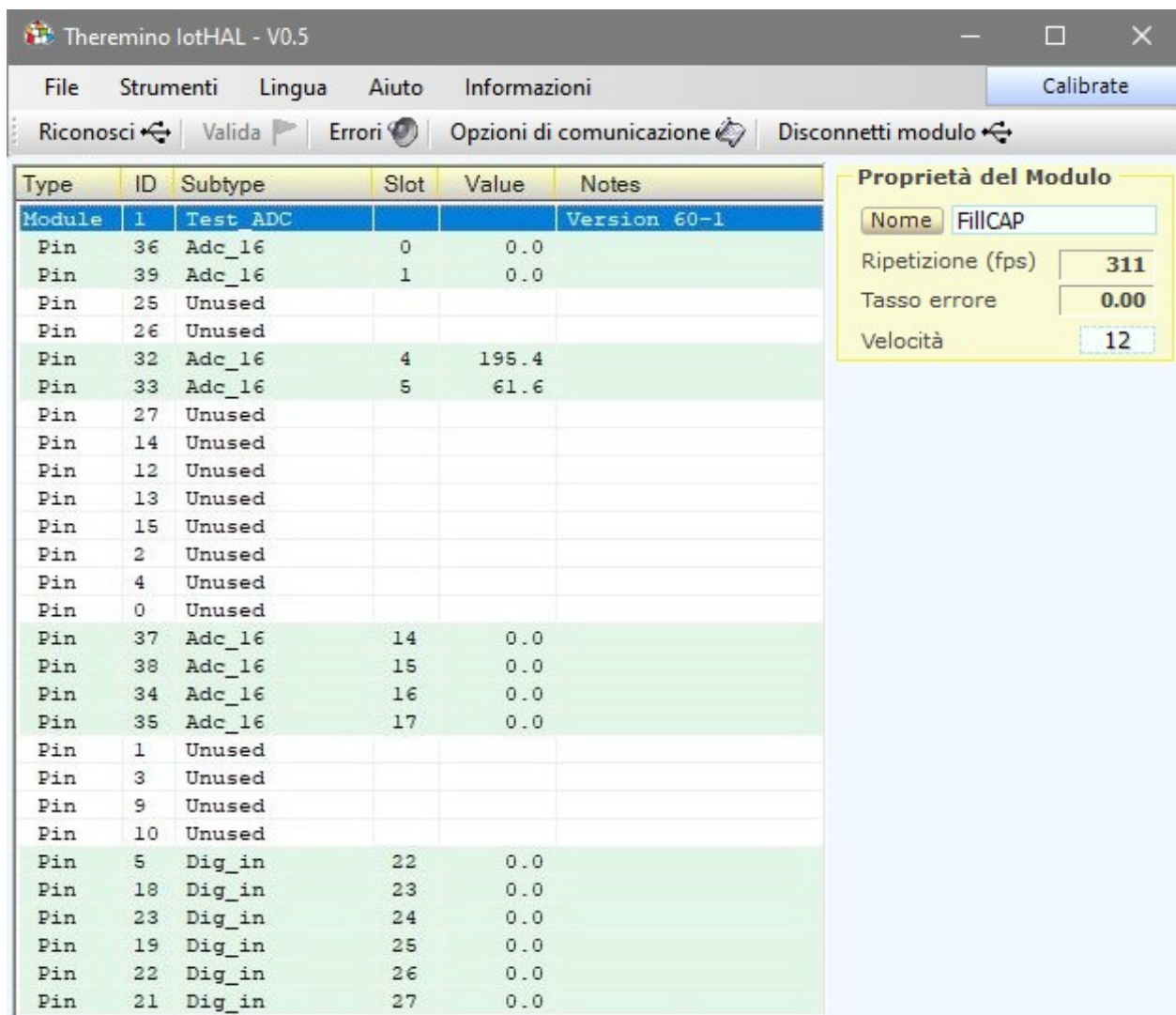
Se la porta COM non appare [leggere qui](#).

- ◆ Programmare lo **lotModule** e attendere qualche secondo.
- ◆ Lanciare la applicazione "Theremino_lotHAL.exe" e controllare che lo **lotModule** appaia nella sua lista, come spiegato nella prossima pagina.



Collegarsi allo IotModule con lo IotHAL

Quando il modulo è alimentato e collegato alla rete, aprendo la applicazione IotHAL, si dovrebbe vederlo apparire con la sua lista di Pin, come nella immagine seguente.



The screenshot shows the 'Theremino IotHAL - V0.5' application window. It has a menu bar with 'File', 'Strumenti', 'Lingua', 'Aiuto', and 'Informazioni'. Below the menu is a toolbar with buttons: 'Riconosci', 'Valida', 'Errori', 'Opzioni di comunicazione', and 'Disconnetti modulo'. The main area contains a table with columns: Type, ID, Subtype, Slot, Value, and Notes. The table lists various pins, some with ADC or digital input types and their current values. On the right, a 'Proprietà del Modulo' sidebar shows fields for 'Nome' (FillCAP), 'Ripetizione (fps)' (311), 'Tasso errore' (0.00), and 'Velocità' (12).

Type	ID	Subtype	Slot	Value	Notes
Module	1	Test_ADC			Version 60-1
Pin	36	Adc_16	0	0.0	
Pin	39	Adc_16	1	0.0	
Pin	25	Unused			
Pin	26	Unused			
Pin	32	Adc_16	4	195.4	
Pin	33	Adc_16	5	61.6	
Pin	27	Unused			
Pin	14	Unused			
Pin	12	Unused			
Pin	13	Unused			
Pin	15	Unused			
Pin	2	Unused			
Pin	4	Unused			
Pin	0	Unused			
Pin	37	Adc_16	14	0.0	
Pin	38	Adc_16	15	0.0	
Pin	34	Adc_16	16	0.0	
Pin	35	Adc_16	17	0.0	
Pin	1	Unused			
Pin	3	Unused			
Pin	9	Unused			
Pin	10	Unused			
Pin	5	Dig_in	22	0.0	
Pin	18	Dig_in	23	0.0	
Pin	23	Dig_in	24	0.0	
Pin	19	Dig_in	25	0.0	
Pin	22	Dig_in	26	0.0	
Pin	21	Dig_in	27	0.0	

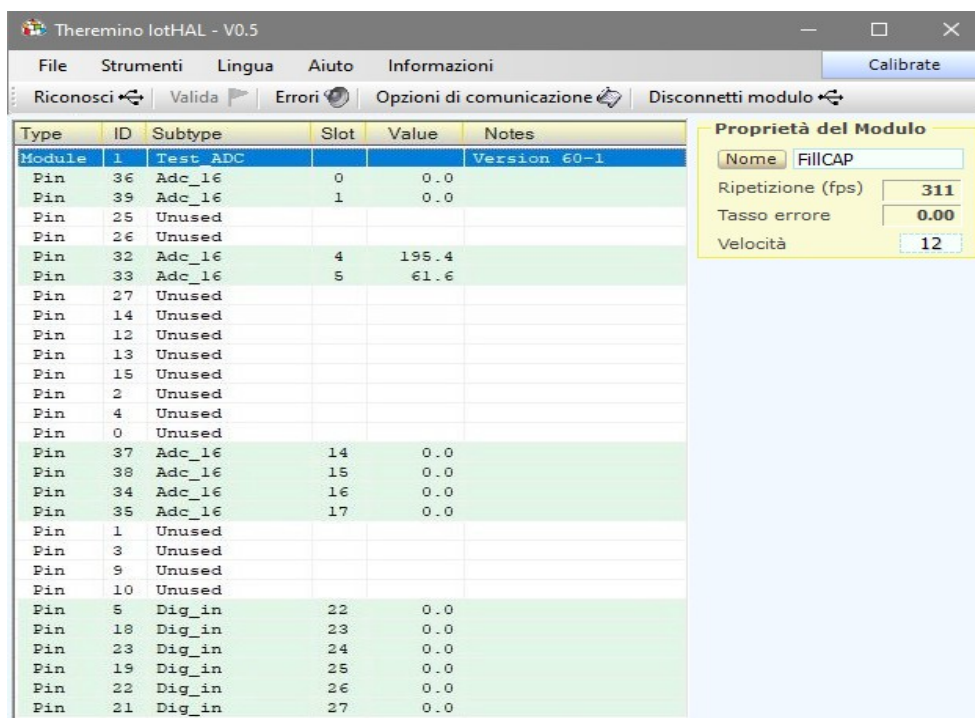
La prima riga è relativa al modulo, con il suo nome e la versione. Il primo numero "60" è la versione del firmware, e il secondo "1" indica il tipo di modulo (WROOM o PICO ecc..)

Le righe seguenti sono relative ai singoli Pin, con il nome del Pin, il tipo di InOut, lo Slot a cui è associato e il valore attuale.

Selezionando la prima riga si possono cambiare le proprietà del modulo e selezionando le righe seguenti si possono modificare le proprietà dei singoli Pin.

Se il modulo non dovesse apparire (lista completamente bianca), riprovare con il pulsante "Riconosci". Provare anche a ripetere le procedure della pagina precedente, controllando nel file "IotModuleSetup.h", che il nome della rete WiFi, la Password e le altre proprietà siano giuste. E eventualmente consultare anche i consigli nelle ultime pagine di questo documento.

Theremino lotHAL



Type	ID	Subtype	Slot	Value	Notes
Module	1	Test_ADC			Version 60-1
Pin	36	Adc_16	0	0.0	
Pin	39	Adc_16	1	0.0	
Pin	25	Unused			
Pin	26	Unused			
Pin	32	Adc_16	4	195.4	
Pin	33	Adc_16	5	61.6	
Pin	27	Unused			
Pin	14	Unused			
Pin	12	Unused			
Pin	13	Unused			
Pin	15	Unused			
Pin	2	Unused			
Pin	4	Unused			
Pin	0	Unused			
Pin	37	Adc_16	14	0.0	
Pin	38	Adc_16	15	0.0	
Pin	34	Adc_16	16	0.0	
Pin	35	Adc_16	17	0.0	
Pin	1	Unused			
Pin	3	Unused			
Pin	9	Unused			
Pin	10	Unused			
Pin	5	Dig_in	22	0.0	
Pin	18	Dig_in	23	0.0	
Pin	23	Dig_in	24	0.0	
Pin	19	Dig_in	25	0.0	
Pin	22	Dig_in	26	0.0	
Pin	21	Dig_in	27	0.0	

Proprietà del Modulo
Nome: FillCAP
Ripetizione (fps): 311
Tasso errore: 0.00
Velocità: 12

Theremino lotHAL collegato a uno lotModule via WiFi

Il Theremino lotHAL (Hardware Abstraction Layer), che si scarica da [questa pagina](#), appare con una interfaccia abbastanza semplice, ma svolge operazioni complesse, con algoritmi ottimizzati.

lotHAL è il cuore della comunicazione con l'hardware, sa comunicare con molti lotModule contemporaneamente, conosce il protocollo USB e conosce tutti i più comuni tipi di Input-Output.

Senza lotHAL comunicare con l'hardware sarebbe difficile (come con Arduino), richiederebbe molto tempo e lavoro (come con Arduino) e infine, per ogni tipo di InOut, ad esempio per muovere un motore o anche solo accendere un LED, si dovrebbe scrivere del firmware apposito (come con Arduino).

Nel sistema Theremino esistono anche altri tre HAL, il primo si chiama semplicemente HAL e comunica via USB con i moduli Master, il secondo si chiama NetHAL e comunica via WiFi, rete e Internet con i moduli NetModule, il terzo si chiama ArduHAL e comunica con i moduli Arduino. A volte utilizzeremo il nome generico HAL, per indicare tutte e quattro le applicazioni.

Se si usano i moduli hardware allora l'HAL è indispensabile e deve rimanere acceso, si può minimizzarlo, ma deve restare in funzione.

Se non si usa hardware allora l'HAL non è necessario, le applicazioni del sistema possono comunicare tra di loro, attraverso gli Slot, anche senza HAL.

Quando si aggiungono o tolgono moduli, alcune righe rosse avvertono che la configurazione è cambiata. Con il pulsante "Valida" si sceglie di perdere la vecchia configurazione e adeguarsi all'hardware attuale.

Collegare più moduli

La applicazione Theremino loTHAL può comunicare con un qualunque numero di moduli.

Tutti i Pin vengono letti e scritti tutti contemporaneamente e alla massima velocità consentita da ogni modulo. In altre parole, un modulo lento non provoca un rallentamento della comunicazione con gli altri.

Esempio di loTHAL collegato a due lotModule via WiFi

In questa immagine si vedono due moduli con i nomi "Test_ADC" e "Test_Pwm_8ch".

Si può anche notare che le due versioni del firmware sono diverse, non sempre questo è possibile ma in questo caso erano compatibili.

Se si collegano altri moduli, questi vengono riconosciuti e listati di seguito. Se il numero di linee supera l'altezza della finestra appare una barra di scorrimento verticale che permette di scorrerli.

I moduli vengono riconosciuti all'avvio della applicazione loTHAL, oppure premendo "Recognize", se la applicazione è già avviata.

I nomi vengono memorizzati nei moduli stessi. Per cui anche se si cambiano reti, Access Point e indirizzi IP, i moduli vengono riconosciuti per nome, e ad ognuno viene assegnata la sua configurazione.

Nella lista i moduli vengono presentati in ordine alfabetico, per cui la lista apparirà sempre con lo stesso ordine, anche se si scambiano gli indirizzi IP.

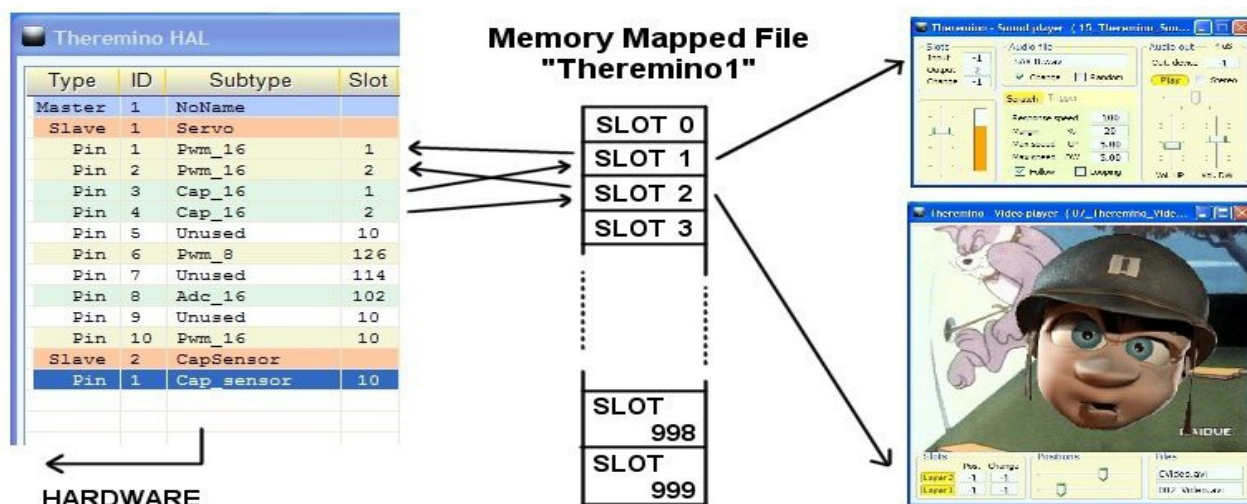
The screenshot shows the 'Theremino loTHAL - V0.23' application window. The interface includes a menu bar (File, Tools, Language, Help, About), a toolbar with buttons for 'Recognize', 'Validate', 'Error beep', 'Communication options', and 'Disconnect module', and a 'Calibrate' button. The main area displays a table of modules and their pins.

Type	ID	Subtype	Slot	Value	Notes
Module	1	Test_ADC			Version 107-0
Pin	36	Dig_in	1	0.0	
Pin	39	Adc_16	2	0.0	
Pin	34	Dig_in	3	0.0	
Pin	35	Dig_in	4	0.0	
Pin	32	Adc_16	5	0.0	
Pin	33	Adc_16	6	0.0	
Pin	25	Dig_in	7	0.0	
Pin	26	Dig_in	8	0.0	
Pin	27	Dig_in	9	0.0	
Pin	14	Dig_in	10	0.0	
Pin	12	Dig_in	11	0.0	
Pin	13	Dig_in	12	0.0	
Pin	15	Dig_in	13	0.0	
Pin	2	Dig_in	14	0.0	
Pin	0	Adc_16	15	0.0	
Pin	4	Adc_16	16	0.0	
Pin	16	Adc_16	17	0.0	
Pin	17	Adc_16	18	0.0	
Pin	5	Unused			
Pin	18	Unused			
Pin	19	Dig_in	21	0.0	
Pin	21	Dig_in	22	0.0	
Pin	3	Dig_in	23	0.0	
Pin	1	Dig_in	24	0.0	
Pin	22	Dig_in	25	0.0	
Pin	23	Dig_in	26	0.0	
Module	2	Test_Pwm_8ch			Version 109-0
Pin	36	Dig_in	50	0.0	
Pin	39	Dig_in	51	0.0	
Pin	34	Pwm	52	0.0	
Pin	35	Pwm	53	0.0	
Pin	32	Pwm	54	0.0	
Pin	33	Pwm	55	0.0	
Pin	25	Pwm	56	0.0	
Pin	26	Pwm	57	0.0	
Pin	27	Dig_in_pu	58	0.0	
Pin	14	Pwm	59	0.0	
Pin	12	Pwm	60	0.0	
Pin	13	Pwm	61	0.0	
Pin	15	Pwm	62	0.0	
Pin	2	Pwm	63	0.0	
Pin	0	Dig_in	64	0.0	
Pin	4	Dig_in	65	0.0	
Pin	16	Dig_in	66	0.0	
Pin	17	Dig_in	67	0.0	
Pin	5	Dig_out	68	0.0	
Pin	18	Dig_out	69	0.0	
Pin	19	Dig_out	70	0.0	
Pin	21	Dig_out	71	0.0	
Pin	3	Dig_out	72	0.0	
Pin	1	Dig_out	73	0.0	
Pin	22	Dig_out	74	0.0	
Pin	23	Dig_out	75	0.0	

On the right side, the 'Module properties' panel is visible for the selected 'Test_ADC' module. It shows fields for 'Name' (Test_ADC), 'Rep freq. (fps)' (202), 'Error rate (%)' (0.00), and 'Comm. speed' (9). A 'Calibrate' button is also present at the top right of the window.

Gli "Slot"

Gli "Slot" del sistema Theremino sono identificati con un numero da 0 a 999 e fanno tutti parte del MemoryMappedFile con nome "Theremino1". Ogni Slot contiene un numero "Float" che può essere letto o scritto da qualunque componente del sistema Theremino.



In questa immagine soltanto l'HAL scrive negli Slot ma in realtà tutti i componenti del sistema possono sia leggere che scrivere in uno qualunque degli Slot, anche se già usato da altri.

Nella scelta degli Slot da usare si deve fare attenzione a due cose:

- Controllare di non usare lo stesso Slot per sbaglio per due funzioni diverse.
- Evitare di scrivere in due sullo stesso Slot.

I Pin di ingresso, i quali scrivono negli Slot sono evidenziati in verde chiaro. Se due o più Pin di ingresso hanno lo stesso Slot, allora la applicazione HAL avverte con righe rosse e con la dicitura **SLOT CONFLICT**.

Molte applicazioni e molti Pin possono leggere lo stesso Slot ma si deve evitare di configurare più di un Pin in scrittura sul medesimo Slot, facendolo non si rompe nulla ma si ottengono risultati indefiniti.

Se si inviano più flussi di dati verso lo stesso Slot allora i dati si mischiano e vince l'ultimo a scrivere, se si vogliono unire i dati in modo ordinato sono necessarie delle regole.

Type	ID	Subtype	Slot	Value	Notes
Master	1	TestSlotCo...			
Slave	1	MasterPins			Firmware V5.0
Pin	1	Adc_16	1	105.3	
Pin	2	Adc_16	2	99.5	
Pin	3	Dig_in	4	0.0	SLOT CONFLICT
Pin	4	Dig_in	4	0.0	SLOT CONFLICT
Pin	5	Dig_in	5	0.0	
Pin	6	Dig_in	6	0.0	
Pin	7	Dig_in_pu	7	1000.0	
Pin	8	Unused			

Per stabilire regole matematiche e logiche tra gli Slot, ed anche per scrivere algoritmi di comportamento complessi, si usano Theremino_Automation o Theremino_Script, oppure qualunque linguaggio di programmazione come C++, CSharp, VbNet, VB6 o Excel, ma è anche possibile usare linguaggi visuali come MaxMSP, Processing, PureData, LabView e EyesWeb. Per MaxMSP sono pronti i Plugin e gli esempi qui: www.theremino.com/downloads/foundations

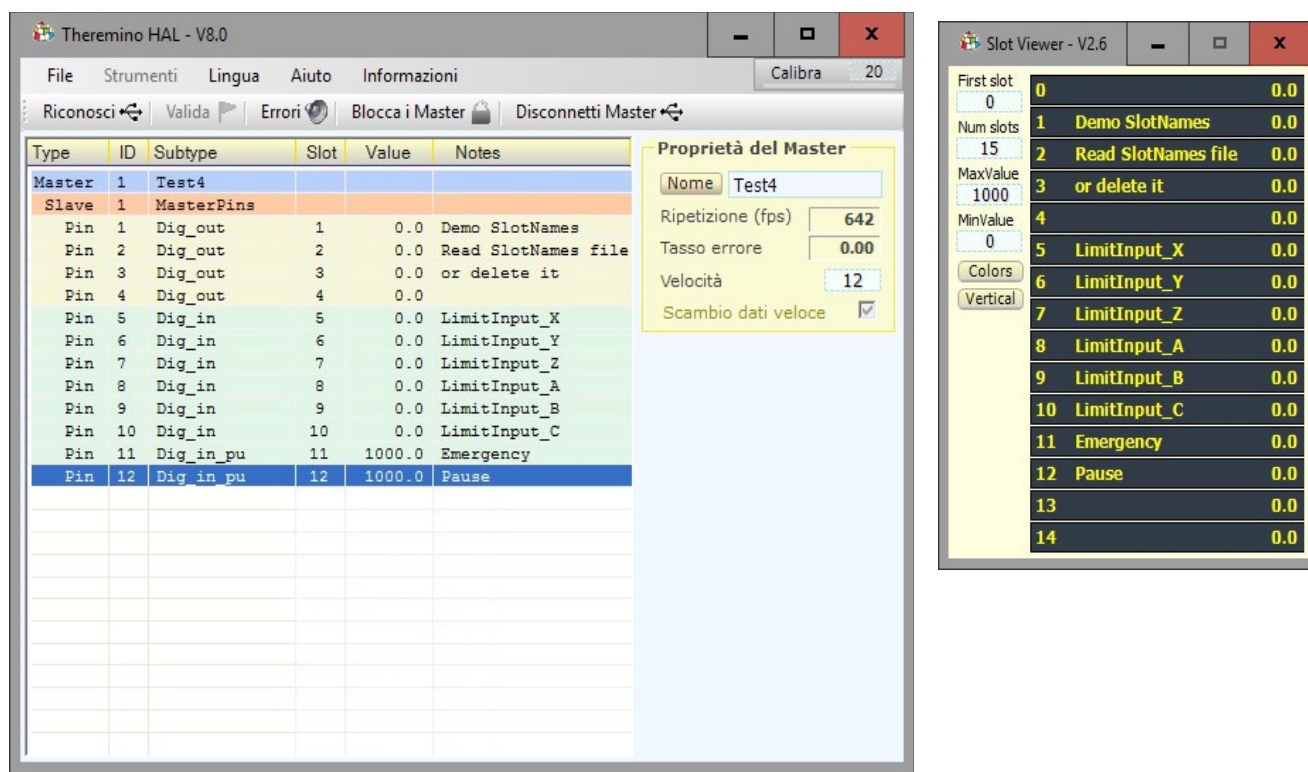
Altre informazioni sulle comunicazioni in queste pagine:

www.theremino.com/technical/communications

www.theremino.com/technical/pin-types

I nomi degli Slot

Tutte le applicazioni HAL e anche lo SlotViewer possono visualizzare i nomi degli Slot (oppure annotazioni o commenti).



Importante notare che i nomi non sono legati ai Pin fisici, ma agli Slot.

I nomi si scrivono in un file, che deve chiamarsi "SlotNames.txt" e che deve stare nella stessa cartella di "Theremino_lotHAL.exe" e di "Theremino_SlotViewer.exe". Se il file "SlotNames.txt" non esiste il campo dei commenti rimarrà vuoto.

Per modificare i nomi degli Slot, si apre il menu "File", si sceglie "Modifica il file SlotNames.txt", e lo si modifica con l'editor predefinito (normalmente NotePad o WordPad). Infine si salva il file e viene ricaricato automaticamente.

Le regole di scrittura sono semplici e sono mostrate nel file di esempio, che si trova nelle ultime versioni di HAL e SlotViewer.

Ogni linea del file inizia con il numero dello Slot, seguito da uno spazio e dal testo da visualizzare. La linea può anche continuare con una parte di commento, che non verrà visualizzata, preceduta da un apice singolo.

Se si vuole usare lo stesso file di commenti, sia per il lotHAL che per lo SlotViewer, si devono tenere i file "SlotNames.txt", "SlotViewer.exe" e "lotHAL.exe", tutti nella stessa cartella.

Gli Slot per i comandi di testo

Dagli inizi del 2024 le applicazioni del sistema Theremino e le applicazioni create dagli utenti possono inviare comandi **in forma di testo** e ricevere risposte dall'HAL, utilizzando due **TextSlot** per comunicare.

Nelle prossime pagine vedremo che esiste anche uno **Slot** per inviare comandi numerici, ma con i comandi testuali si possono fare molte più cose, praticamente tutto quello che si potrebbe fare con il mouse e la tastiera sulla interfaccia dell'HAL.

Quando utilizzare gli Slot numerici al posto dei TextSlot

Esiste un solo caso in cui è preferibile usare lo Slot dei comandi numerici.

Quando si vuole conoscere il numero di Master (ovvero moduli hardware) che sono stati riconosciuti e sono attivi, con gli Slot numerici basta leggere il valore dello Slot e questo avviene in pochi microsecondi, mentre se lo facessimo con i TextSlot sarebbe necessario inviare un comando per chiedere il valore e poi attendere la risposta. E questo richiederebbe decine di millisecondi.

Per questo motivo e anche per facilitare chi li sta già usando nelle proprie applicazioni, abbiamo mantenuto attivi anche i comandi degli Slot numerici.

Numeri da utilizzare per i TextSlot

I **TextSlot** hanno numeri da 0 a 999 come gli Slot numerici ma sono due cose diverse. Anche se si usa lo stesso numero **TextSlot** e **Slot numerici** non interferiscono tra loro.

Se non si impostano diversamente vengono utilizzati il **TextSlot zero** per i comandi e il **TextSlot uno** per le risposte.

Utilizzare altri Slot al posto dei TextSlot zero e uno

Solitamente i TextSlot dei comandi sono zero e uno, ma potrebbe accadere di voler utilizzare più applicazioni indipendenti sullo stesso PC. In questi casi ogni applicazione risiederebbe in cartelle separate insieme al suo HAL e accede ai suoi moduli Master utilizzando il comando "Blocca master". In questi casi si può assegnare ad ogni HAL uno Slot dei comandi diverso. Per i comandi **si può utilizzare qualunque TextSlot (da 0 a 999) facendo però attenzione a non usarlo per altro.**

Per assegnare i numeri agli Slot dei comandi, si modificano manualmente le ultime righe del file "Theremino_HAL_INI.txt". Quindi per utilizzare, ad esempio, i TextSlot 300 e 301, si scriverebbe: **CommandTextSlot= 300 e ResponseTextSlot= 301.**

Attenzione a non eliminare i segni "=". In caso di errori vengono utilizzati gli slot zero e uno e l'HAL riscrive la linea corretta nel file INI.

Come inviare i comandi

Dopo aver inviato un comando, prima di inviarne un altro è bene attendere almeno 50 o 100 ms per dare tempo alle applicazioni di eseguire le scritture e letture anche nei casi peggiori.

Per minimizzare questa attesa si controlla il testo nel CommandTextSlot e si prosegue appena viene eliminato dall'HAL.

Vedere l'esempio di Automation: **Demo Programs\SlotText Commands\Commands_to_HAL.txt**

Comandi con i TextSlot

Nelle prossime pagine elencheremo tutti i comandi utilizzabili ma **solo la applicazione HAL li riconosce tutti.**

ArduHAL, IotHAL e NetHAL utilizzano solo una parte dei comandi e molti dei comandi sono validi solo quando sono selezionati i tipi di Pin che li utilizzano.

I comandi che finiscono per ON/OFF sono abilitazioni. Qualunque valore che non sia uno zero oppure la parola OFF vale come ON. Normalmente si utilizzano 1/0 oppure ON/OFF.

Sia i comandi che i loro parametri aggiuntivi possono essere scritti indifferentemente con caratteri maiuscoli o minuscoli in qualunque combinazione.

I parametri dei comandi, come ad esempio i nomi dei Master, i tipi di Pin, i tipi di Filtri e i tipi di canali dell'Adc24, vengono riconosciuti solo in base ai loro caratteri alfanumerici. Per i parametri gli spazi e gli altri segni non contano. Quindi ad esempio "Dig_in_pu" potrebbe essere scritto come "diginpu" o anche "DIG - IN _ _ PU" e verrebbe comunque riconosciuto e considerato valido.

Comandi della barra superiore

- **Recognize** Ha lo stesso effetto del pulsante Recognize.
- **Validate** Ha lo stesso effetto del pulsante Validate.
- **ErrorBeep ON/OFF** Abilita o disabilita il pulsante Error Beep.
- **LockMasters ON/OFF** Abilita o disabilita il pulsante Lock Masters.
- **Disconnect** Disconnette il Master (modulo) selezionato.
- **NoEdits ON/OFF** Abilita o disabilita il pulsante NoEdits.
- **Calibrate** Abilita o disabilita il pulsante Calibrate.
- **CalibrationValue N (1..100)** Valore numerico per la casella "calibrate".

Comandi dei Master (o moduli)

- **SelectMaster N (nota1)** Seleziona il Master (se ce ne sono più di uno).
- **LoadConfig XXX** Carica la configurazione XXX per il Master selezionato.
- **Speed N (1..12)** Velocità di comunicazione per il Master selezionato.
- **FastDataExchange ON/OFF** Abilita o disabilita il FastExchange (solo per i moduli Slave)

(Nota1)

Il comando SelectMaster viene riconosciuto anche se si scrive SelectModule che potrebbe essere più appropriato per i moduli Arduino e ESP32.

Comandi con i TextSlot (parte 2)

Selezionare i PIN

Per selezionare il Pin si utilizza una stringa di testo anche se in molti casi questa stringa conterrà semplicemente dei numeri da 1 a 12.

Però quando si utilizzano i moduli Arduino e Esp32 i Pin non sono numerati in sequenza, ad esempio 22, 35, 28, 60 ecc.. e in alcuni casi sono alfanumerici, ad esempio A0, A1, A3, D0, D1, D3 ecc..

Quindi in tutti i casi invece di un numero si indica il nome del Pin come stringa di testo.

Comando per selezionare i PIN

 **SelectPin XXX** Si utilizza una stringa di testo per il nome del Pin

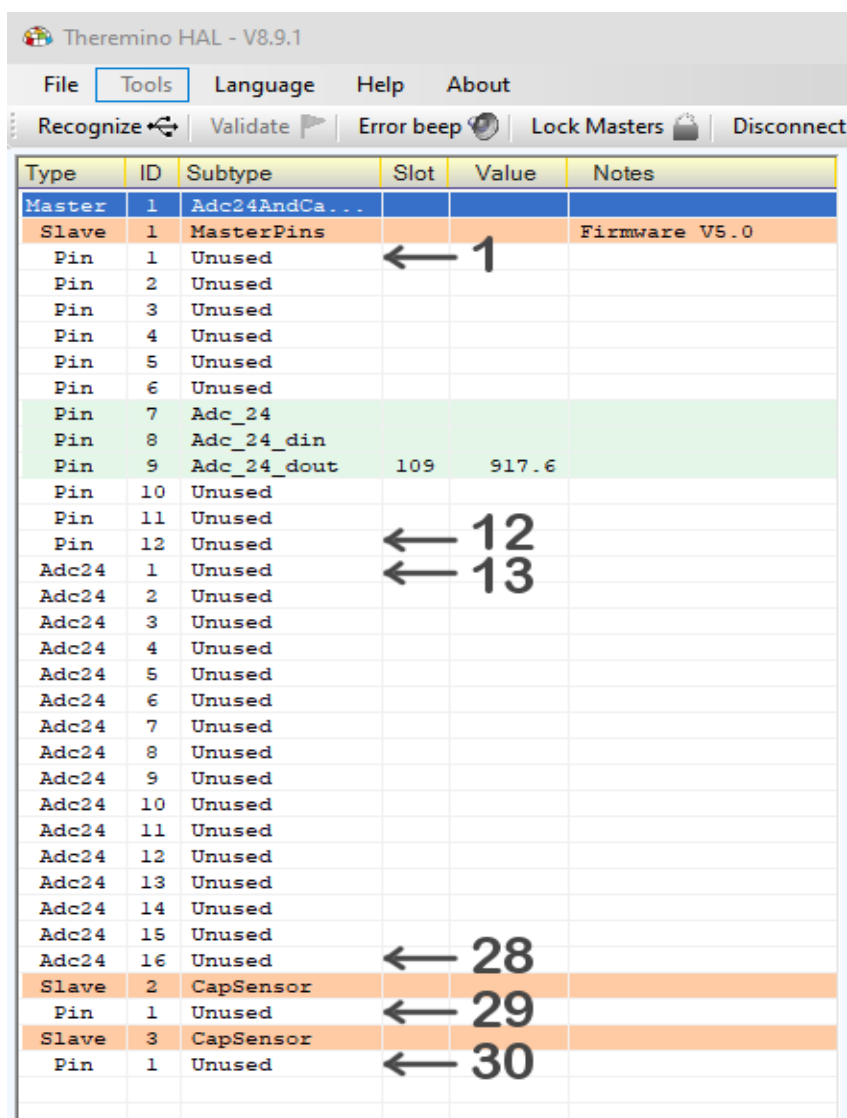
Numerazione per i Pin ADC24 e CAP-SENSOR

Quando si utilizzano i Master con la applicazione Theremino_HAL e si collega un Adc24 e/o uno o più CapSensors, la numerazione dei Pin è diversa da quello che si potrebbe immaginare.

I Pin del Master vanno da 1 a 12.

Poi si prosegue la numerazione indicando gli Adc24 con 13, 14 e seguenti.
(in questo caso vanno da 13 a 28)

E anche i CapSensor proseguono con la numerazione.
(in questo caso si indicano con 29 e 30 ma senza Adc24 sarebbero 13 e 14)



Type	ID	Subtype	Slot	Value	Notes
Master	1	Adc24AndCa...			
Slave	1	MasterPins			Firmware V5.0
Pin	1	Unused	← 1		
Pin	2	Unused			
Pin	3	Unused			
Pin	4	Unused			
Pin	5	Unused			
Pin	6	Unused			
Pin	7	Adc_24			
Pin	8	Adc_24_din			
Pin	9	Adc_24_dout	109	917.6	
Pin	10	Unused			
Pin	11	Unused			
Pin	12	Unused	← 12		
Adc24	1	Unused	← 13		
Adc24	2	Unused			
Adc24	3	Unused			
Adc24	4	Unused			
Adc24	5	Unused			
Adc24	6	Unused			
Adc24	7	Unused			
Adc24	8	Unused			
Adc24	9	Unused			
Adc24	10	Unused			
Adc24	11	Unused			
Adc24	12	Unused			
Adc24	13	Unused			
Adc24	14	Unused			
Adc24	15	Unused			
Adc24	16	Unused	← 28		
Slave	2	CapSensor			
Pin	1	Unused	← 29		
Slave	3	CapSensor			
Pin	1	Unused	← 30		

Comandi con i TextSlot (parte 3)

Proprietà dei Pin (pannello Pin properties)

- **SetPinType XXX** Imposta il tipo per il Pin attualmente selezionato.
- **SetSlot N (0..999)** Imposta lo Slot per il Pin selezionato.
- **SetMaxValue N** Imposta MaxValue per il Pin selezionato.
- **SetMinValue N** Imposta MinValue per il Pin selezionato.
- **SetPinType XXX** Imposta il tipo per il Pin selezionato.
- **SetResponseSpeedButton ON/OFF** Abilita o disabilita il pulsante ResponseSpeed.
- **SetResponseSpeed N (1..100)** Imposta il valore di ResponseSpeed.

Proprietà per i Pin PWM e SERVO (pannelli PWM properties e SERVO properties)

- **MaxTime N** Imposta Max time per il Pin selezionato.
- **MinTime N** Imposta Min time per il Pin selezionato.
- **LogarithmicResponse ON/OFF** Abilita o disabilita il pulsante Logarithmic response.

Proprietà per i Pin di tipo STEPPER (pannello Stepper properties)

- **MaxSpeed N** Imposta Max sp. (mm/min) per il Pin selezionato.
- **MaxAcc N** Imposta Max acc. (mm/sec/s) per il Pin selezionato.
- **StepsPerMm N** Imposta Min time per il Pin selezionato.
- **LinkedtoPrevious ON/OFF** Abilita o disabilita il pulsante Linked to previous.

Proprietà per i Pin di tipo PWM FAST (pannello Pwm_Fast properties)

- **PwmFastFrequency N** Imposta la frequenza del PWM.
- **PwmFastDutyCycle N (0..1000)** Imposta il Duty Cycle.
- **FrequencyFromSlot ON/OFF** Abilita o disabilita il controllo di frequenza dallo Slot.
- **DutyCycleFromSlot ON/OFF** Abilita o disabilita il controllo del duty cycle dallo Slot.

Proprietà per i Pin di tipo CAP (pannello Touch properties)

- **MinVariation N (-1000..+500)** Imposta la variazione minima.
- **ProportionalArea N (-500..+1000)** Imposta l'area proporzionale.

Proprietà per i Pin di tipo COUNTER e PERIOD (pannello Freq properties)

- **ConvertToFrequency ON/OFF** Abilita o disabilita la conversione in frequenza.
- **MaxFrequency N** Imposta la frequenza massima.
- **MinFrequency N** Imposta la frequenza minima.

Comandi con i TextSlot (parte 4)

Proprietà dei Pin di tipo CapSensor e Usound (pannelli CapSensor e Usound properties)

- **MaxDist N** (50..9999) Imposta la distanza massima in mm.
- **MinDist N** (0..5000) Imposta la distanza minima in mm.
- **Area N** (0..9999) Imposta l'area in mmq (solo per i CapSensor).
- **RemoveErrors ON/OFF** Abilita o disabilita il controllo degli errori per il tipo Usound

Proprietà dei Pin di tipo Adc24 (pannello Adc24 properties)

- **NumberOfPins N** (0..16) Imposta il numero di PIN (solo per i Master, non ESP32).
- **SamplesSec N** (10..19200) Imposta la frequenza di campionamento.
- **Filter XXX** Imposta il tipo di filtro (MaxSpeed, Fast, Medium ecc..)

Proprietà dei Pin di tipo Adc24_ch e Adc24_cn_b (pannello Adc24_channel properties)

- **Type XXX** Imposta il tipo (Differential, Pseudo Diff, Single ended).
- **Gain N** Imposta il guadagno (0, 2, 4, 8, 16, 32, 64, 128)
- **Biased ON/OFF** Abilita o disabilita la polarizzazione a 1.65 volt.

RICHIESTE DEI VALORI

- **GetMasterName (nota 1)** Richiesta del nome del Master (modulo) selezionato.
- **GetFPS** Richiesta della frequenza attuale di comunicazione.
- **GetPinType** Richiesta del tipo per il Pin selezionato.

RISPOSTE

- **OK** Il comando è stato eseguito con successo.
- **Testo dell'errore** Comprende anche il comando che ha generato l'errore.
- **Parametro richiesto** Il valore che era stato richiesto con uno dei comandi **GET**.

(Nota 1)

Il comando GetMasterName viene riconosciuto anche se si scrive GetModuleName che potrebbe essere più appropriato per i moduli Arduino e ESP32.

Comandi attraverso gli Slot numerici

Le applicazioni del sistema Theremino, o altre applicazioni create dagli utenti, possono inviare comandi e ricevere dati dall'HAL, utilizzando uno Slot numerico per comunicare.

Per esempio una applicazione potrebbe modificare i parametri di tutti i Pin, riscrivendo il file delle configurazioni e poi inviando il comando "Riconosci". Oppure una applicazione potrebbe verificare quanti Master sono effettivamente collegati, inviando il comando Riconosci, e poi leggendo il loro numero sullo slot dei comandi. Oppure una applicazione musicale potrebbe calibrare i CapSensor o i pulsanti capacitivi, inviando il comando "Calibra" (attualmente non implementati negli ESP32).

Utilizzare altri Slot al posto dello Slot zero

Solitamente lo Slot dei comandi è lo zero, ma potrebbe accadere di voler utilizzare più applicazioni indipendenti sullo stesso PC. In questi casi ogni applicazione risiederebbe in cartelle separate insieme al suo lotHAL e accedrebbe ai suoi moduli. In questi casi si può assegnare ad ogni lotHAL uno Slot dei comandi diverso. Per i comandi si può utilizzare qualunque Slot (da 0 a 999) facendo però attenzione a non assegnarlo a nessun Pin.

Per assegnare un numero diverso da zero allo Slot dei comandi, si modifica manualmente l'ultima riga del file "Theremino_lotHAL_INI.txt". Quindi per utilizzare, ad esempio, lo slot 300, si scriverebbe: `CommandSlot= 300`. Attenzione a non eliminare il segno "=". Se si sbaglia qualcosa il lotHAL utilizza lo Slot zero e riscrive la linea corretta nel file INI.

Come inviare i comandi

Attualmente sono definiti due comandi:

- ◆ Riconosci Si invia "NAN_Recognize", oppure il numero "1"
- ◆ Calibra (*) Si invia "NAN_Calibrate", oppure il numero "2"

Le applicazioni che non sono in grado di inviare i numeri speciali NAN (Not A Number), possono utilizzare i numeri "1" e "2" al posto dei valori "NAN_Recognize" e "NAN_Calibrate".

Per sicurezza, i comandi "1" e "2" devono essere preceduti da una sequenza. La sequenza prevede due numeri (333 e 666) che corrispondono in realtà ai numeri in virgola mobile, con sette cifre di precisione, 333.0000 e 666.0000. Quindi è praticamente impossibile che un ADC o altri dispositivi possano inviare questa sequenza per errore.

Messaggi di risposta

Le risposte, e i messaggi di errore, vengono comunicati con numeri nello slot dei comandi.

- ◆ -1 Il comando "Riconosci" è ancora in esecuzione.
- ◆ 0 Non sono stati trovati dei lotModule, la lista dei moduli è vuota.
- ◆ da 1 in su Il numero di lotModule che sono stati riconosciuti.
- ◆ NAN_MasterError Uno degli lotModule ha smesso di comunicare.

Lo Slot dei comandi - Esempi

Per inviare il comando “Riconosci”, si scrive:

```
----- VbNet
Slots.WriteSlot (0, NAN_Recognize)

----- CSharp
Slots.WriteSlot(0, NAN_Recognize);

----- Theremino Script
WriteSlot (0, NAN_Recognize)
```

Come spiegato nella pagina precedente, alcune applicazioni (ad esempio Theremino Automation), potrebbero non essere in grado di utilizzare i numeri speciali NAN. Se non si utilizzano i NAN gli esempi precedenti diventano:

```
----- VbNet
Slots.WriteSlot (0, 333)
System.Threading.Thread.Sleep(50)
Slots.WriteSlot (0, 666)
System.Threading.Thread.Sleep(50)
Slots.WriteSlot (0, 1)

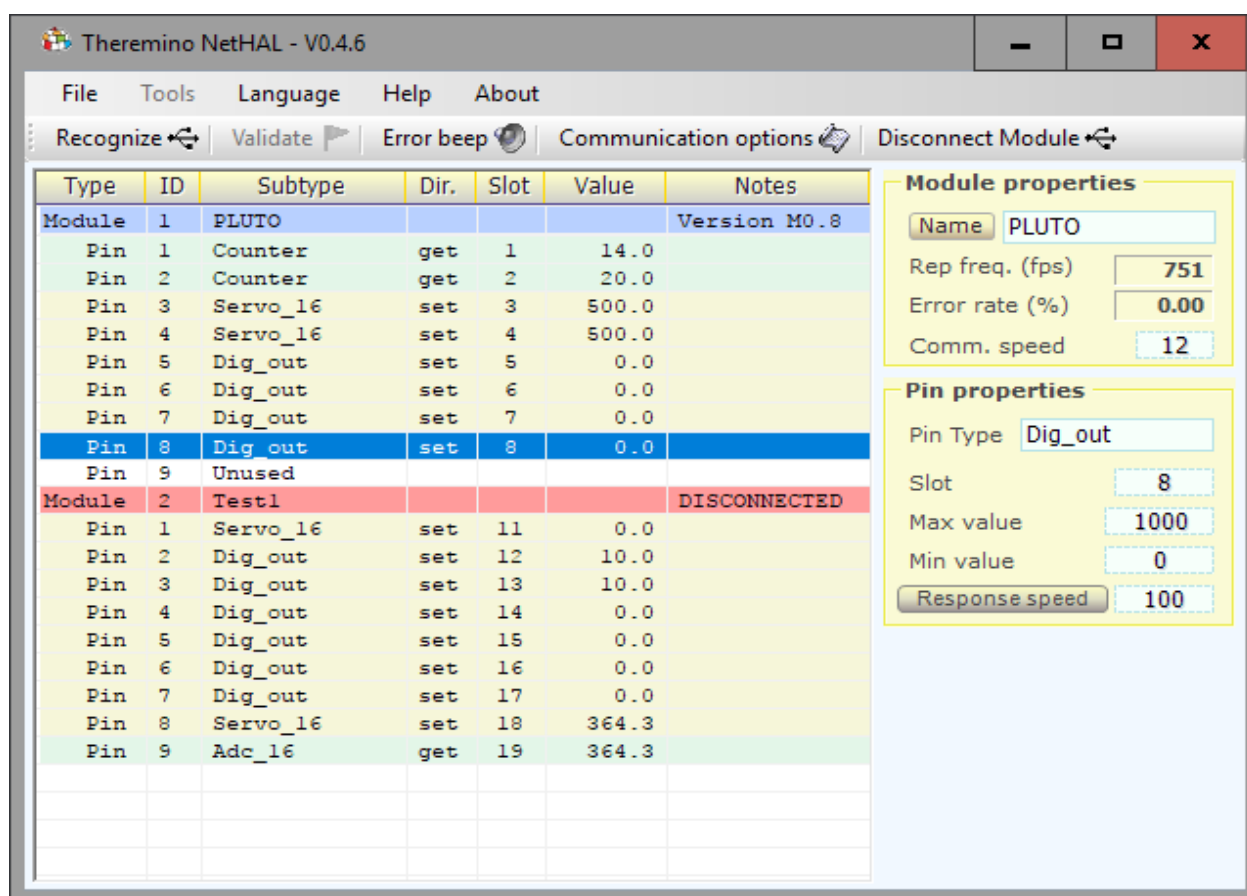
----- CSharp
Slots.WriteSlot(0, 333);
System.Threading.Thread.Sleep(50);
Slots.WriteSlot(0, 666);
System.Threading.Thread.Sleep(50);
Slots.WriteSlot(0, 1);

----- Theremino Automation
Slot 0 = 333
Wait Seconds 0.05
Slot 0 = 666
Wait Seconds 0.05
Slot 0 = 1

----- Theremino Script
WriteSlot (0, 333)
Threading.Thread.Sleep(50)
WriteSlot (0, 666)
Threading.Thread.Sleep(50)
WriteSlot (0, 1)
```

Le pause di 50 millisecondi servono per dare tempo all'HAL di leggere lo Slot.

I colori della applicazioni HAL



Lo schema di colori aiuta a riconoscere i componenti e la loro configurazione

Il primo modulo (con nome PLUTO) fornisce:

Un "Pin" configurato come "Counter" (il colore verde chiaro indica un Input)

Un "Pin" configurato come "Servo_16" (il colore giallo chiaro indica un Output)

Sei "Pin" configurati come "Dig_out" (di cui il Pin 8 è **Selezionato**)

Un "Pin" configurato come "Unused" (il colore di sfondo bianco indica "non usato")

Il secondo modulo (con nome Test1) fornisce:

Un "Pin" configurato come "Servo_16" (il colore giallo chiaro indica un Output)

Sei "Pin" configurati come "Dig_out" (il colore giallo chiaro indica un Output)

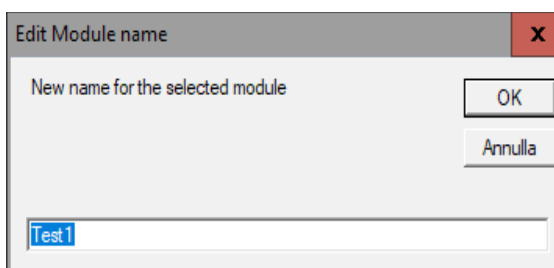
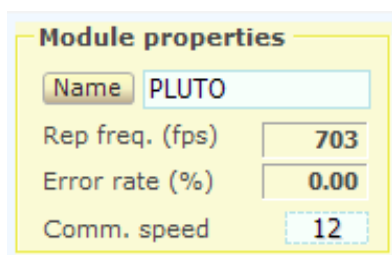
Un "Pin" configurato come "Servo_16" (il colore giallo chiaro indica un Output)

Un "Pin" configurato come "Adc_16" (il colore verde chiaro indica che è un Input)

La riga "Pin 8 - DigOut" di colore blu è la riga selezionata e le sue proprietà vengono mostrate sulla destra

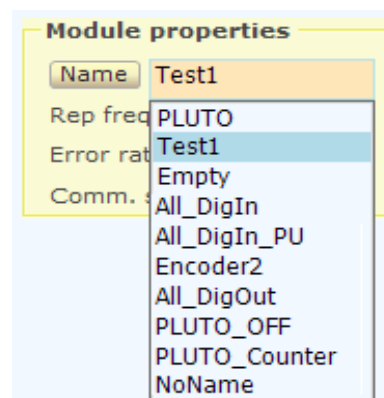
La riga "Module 2 - Test1" ha lo sfondo rosso perchè il modulo "Test1" è disconnesso (spento)

Le proprietà del modulo - Il nome



Il nome del modulo selezionato può essere modificato in due modi:

- ◆ Premendo il bottone "Name" e poi modificando il suo nome.
- ◆ Cliccando sulla casella del nome e scegliendo una diversa configurazione dal menù a cascata.



Il nome del modulo viene scritto nel modulo hardware e serve per riconoscerlo quando lo si ricollega.

Un nuovo modulo appena collegato viene chiamato "NoName". E' bene prenderle l'abitudine di dargli subito un nome diverso, per distinguerlo da tutti gli altri.

Nei nomi dei moduli il "case" delle lettere (maiuscole o minuscole) non conta.

Se nel database dei moduli ci sono due moduli con lo stesso nome allora viene usata la configurazione del primo per ambedue. E' quindi importante dare nomi diversi ad ogni modulo (a meno che si vogliano avere moduli di ricambio con lo stesso nome di quello principale)

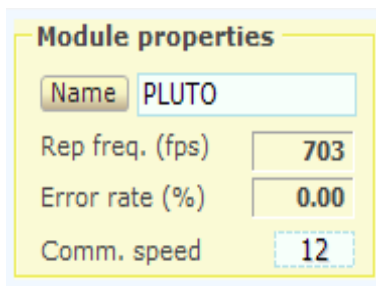
I moduli vengono sempre listati in ordine alfabetico, per cui se cambiano gli IP l'ordine dei moduli non cambia.

Il programma HAL riesce quasi sempre a usare la giusta configurazione quando si scollegano, sostituiscono e ripristinano i componenti, ma se si cambia nome ai moduli usando un diverso computer o con un'altra applicazione HAL (situata una cartella separata - quindi con parametri separati) o in altri casi difficili e complessi, allora l'allineamento tra configurazione e hardware si perde.

Se si perde l'allineamento si dovrebbe ripristinare la configurazione manualmente, un Pin alla volta, ma gli esperti possono editare il file di configurazione e eventualmente copiare questo file interamente, o solo parte delle configurazioni, da una applicazione HAL a un'altra, su un altro computer o in un'altra cartella.

Quando la configurazione non è valida modificare il nome dei moduli non modifica il file di configurazione ma solo il nome scritto nell'hardware è quindi possibile modificare i nomi dei moduli fino a farli coincidere a quelli giusti nella configurazione.

Le proprietà del modulo - Comunicazione



Module properties	
Name	PLUTO
Rep freq. (fps)	703
Error rate (%)	0.00
Comm. speed	12

- Numero di comunicazioni al secondo
- Percentuale di errori di trasmissione (normalmente zero)
- Regolazione della velocità di comunicazione

Il numero di comunicazioni al secondo "fps" normalmente è superiore a 500 e spesso anche più di 800, aumentando il numero di Pin utilizzati questa frequenza potrebbe diminuire leggermente.

Per molte applicazioni una frequenza di 100 fps è più che sufficiente, per alcune applicazioni particolari è bene mantenere fps più alto possibile, almeno 400 o 500.

La percentuale di errori normalmente è zero. Si dovrebbero avere errori solo in caso di disturbi causati da un'altra rete sullo stesso canale o se la distanza dall'access point supera i 20..50 metri.

Se il segnale è molto debole o disturbato la percentuale di errori aumenta e si potrebbe perdere il collegamento. **Ma appena il segnale migliora il collegamento viene ripristinato automaticamente.**

Suggerimenti per aumentare la frequenza "fps"

- Avvicinarsi all'Access Point
- Modificare il canale WiFi nell'Access Point in modo da evitare i disturbi dei vicini di casa.
- Utilizzare la applicazione [ThereminoWiFi](#) per vedere quali canali sono meno usati.
- Utilizzare un Access Point migliore.
- Aumentare la "Comm speed".
- Diminuire il numero di byte usati configurando come "Unused" tutti i Pin che non servono.

Regolare la frequenza "fps"

Con il valore "Comm speed" è possibile regolare la frequenza di aggiornamento "fps".

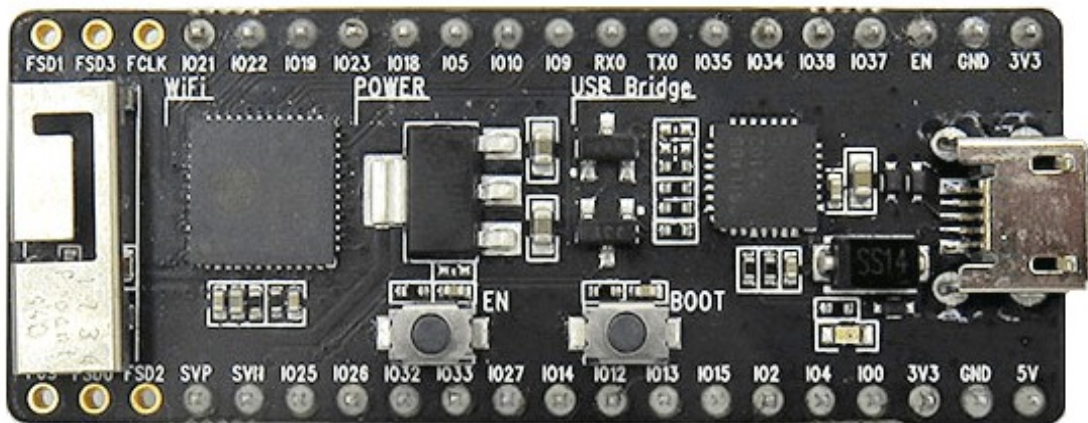
Per aumentare la velocità di risposta è bene aumentare al massimo la frequenza di scambio, quindi impostare "Comm speed" a "12". Ma per molte applicazioni cento scambi al secondo sono più che sufficienti, per cui normalmente si può regolare "Comm speed" da 8 a 10 e caricare meno la CPU.

Numero di moduli collegabili

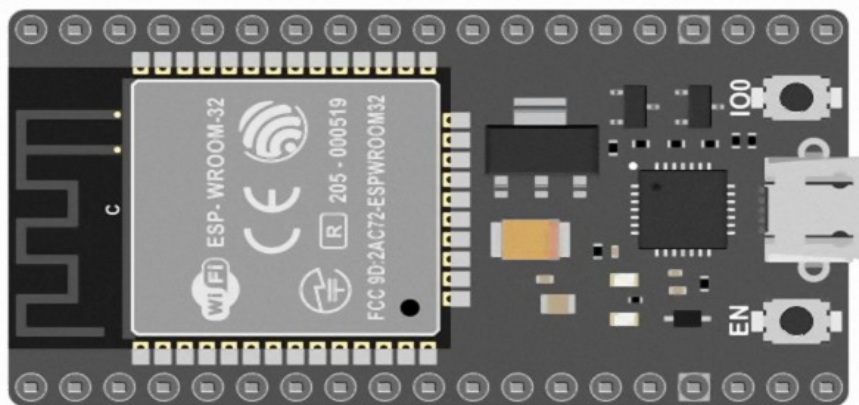
I moduli comunicano tutti sullo stesso canale WiFi e condividono necessariamente la banda. Per cui la velocità di comunicazione con l'Access Point diminuisce aumentando il numero di moduli collegati. A seconda del modello di Access Point si possono avere comportamenti differenti.

In un esperimento (con un LinkSys e firmware DD-WRT) la velocità era 600 Fps con un solo modulo, e scendeva a 320 Fps con sei moduli collegati. Aumentando i moduli fino a 8 e oltre, abbiamo sperimentato ulteriori rallentamenti e anche notevoli ritardi nella comunicazione.

I diversi tipi di moduli ESP32



Questo è il modello di ESP32 che preferiamo, comunica più velocemente degli altri, ha un numero maggiore di Pin e la disposizione dei Pin è migliore.

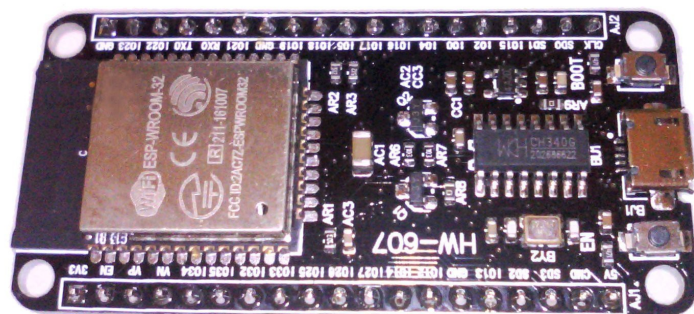


Per utilizzare questi moduli si deve abilitare la riga `"#define IOTMODULE_PINOUT_TYPE 1"` nel file `"IotModuleSetup.h"` Questo modello di ESP32 è da utilizzare come seconda scelta.

Per utilizzare questi moduli si deve abilitare la riga `"#define IOTMODULE_PINOUT_TYPE 0"` nel file `"IotModuleSetup.h"`

Gran parte degli ESP32 sono simili ai due precedenti. Ma ne esistono anche di diversi (che non abbiamo provato), e anche di apparentemente simili, ma non funzionanti o che richiedono particolari accorgimenti.

Ad esempio il modulo di questa immagine si programma senza errori, ma poi non si collega all'Access Point. Inoltre i suoi led non si accendono mai, sia programmandolo che premendo il pulsante di Reset.



Abbiamo scoperto che far funzionare questo modello di moduli bisogna collegare il Pin "0" al 3.3 volt attraverso un resistore da 1k (se lo si collega direttamente non si riesce più a programmarlo). Inoltre questo modello non ha nemmeno un LED e ha il WiFi che funziona male, appena ci si allontana di pochi metri dall'Access Point, comincia a fare errori e si disconnette.

Consumi e tensioni dello ESP32

Corrente di alimentazione	Condizioni di misura
10 mA	Non connesso
200 mA	In alcuni istanti durante la connessione
140 mA	WiFi attivo
150 mA	HAL collegato
160 mA	Tutti DigIn
160 mA	Tutti DigIn pu
160 mA	Tutti DigOut
280 mA	Alcuni Pwm a 5 Mhz
320 mA	In qualche caso, con Pwm e fili lunghi

Quando il consumo supera i 200 mA possono verificarsi errori di comunicazione e in alcuni casi anche la perdita della comunicazione. Accade più facilmente in particolari condizioni, con fili lunghi e con Pwm impostati a frequenze da 1 MHz in su.

*Usando alte frequenze di PWM e bene tenere i fili cortissimi o aggiungere un resistore in serie al segnale, posizionandolo vicino al modulo ESP32. **Vedere la prossima pagina.***

Impostare i Pin 1 e 3 come DigIn, DigOut o Pwm, interrompe la comunicazione seriale (il debug con la seriale via USB) ma non fa aumentare il consumo.

Portare a metà tensione gli ingressi DigIn o disturbare gli ingressi DigIn aperti, con segnali ad alta frequenza, non aumenta il consumo in modo misurabile.

Gli ingressi non tollerano tensioni più alte di 3.5 volt circa. Se li si collega al 5 volt assorbono molta corrente e potrebbero rompersi. I segnali da 5 volt possono essere collegati agli ingressi attraverso un resistore da almeno 1 k ohm.

Un condensatore elettrolitico da 1000 uF o anche da 2200 uF (meglio se low-esr) tra 5V e GND, migliora di molto le forme d'onda, quando le si guardano all'oscilloscopio. Aumenta anche la stabilità e si possono collegare servo mediamente potenti al 5 volt, senza causare buchi di alimentazione. Un condensatore elettrolitico tra 3.3V e GND, non ha effetti visibili sulle forme d'onda, ma potrebbe aumentare la stabilità in alcuni casi.

ATTENZIONE: I grossi elettrolitici possono essere pericolosi per i componenti del modulo, scaricarli prima di collegarli e collegarli da spento. Espressif dice che se si aggiungono condensatori di grande capacità si potrebbero avere problemi di Reset parziali, in caso di brevi mancanze di tensione.

Se si utilizzano motori o altri dispositivi che assorbono molta corrente, il connettore USB andrebbe usato solo per la programmazione e non per alimentare anche i dispositivi collegati. Altrimenti la comunicazione WiFi potrebbe interrompersi non appena si muovono i Servo o altri utilizzatori che consumano molta corrente.

Connettere i segnali di uscita

Fare attenzione ai casi seguenti:

- ◆ Quando con le uscite dello Esp32 si pilotano carichi capacitivi superiori a poche decine di pF, ad esempio un cavo schermato di un metro che fa già dai 50 ai 300 pF, o anche solo fili di collegamento lunghi qualche decina di centimetri.
- ◆ E peggio ancora, quando su queste uscite si inviano segnali che cambiano con molta frequenza, come i PWM o STEPPER.

In questi casi possono accadere difetti di ogni genere, errori sui segnali e anche perdita della comunicazione.

Questi difetti accadono perché nelle commutazioni lo ESP32 non limita la corrente di uscita a valori ragionevoli (ad esempio 10 mA) ma fa tutto quel che può per caricare la capacità con la corrente più alta che riesce a dare. Dalle simulazioni si direbbe che arriva a 3 ampere, sicuramente non sarà vero ma comunque “spara” fuori una corrente esagerata creando due tipi di problemi:

- (1) Le armoniche delle commutazioni interferiscono con il segnale WiFi.
- (2) La alimentazione cede durante le commutazioni e si verificano errori, o anche continui reset del modulo e disconnessioni della applicazione HAL.

Fortunatamente esiste una semplice soluzione: aggiungere un resistore in serie ad ogni uscita dell'Esp32 che deve pilotare carichi capacitivi superiori a poche decine di pF.

Un resistore da 330 ohm può andare bene nella maggioranza dei casi. Con questo valore la corrente viene limitata a circa 10 mA e si riesce lo stesso a pilotare carichi da molte centinaia di pF (ad esempio un cavo schermato da alcuni metri), con segnali ad onda quadra fino a 300 KHz, e anche un po' oltre.

Volendo pilotare capacità più alte, o aumentare la frequenza trasmissibile, si potrebbe diminuire il resistore fino a 100 ohm. Meglio non scendere oltre, in modo da evitare i problemi detti all'inizio.

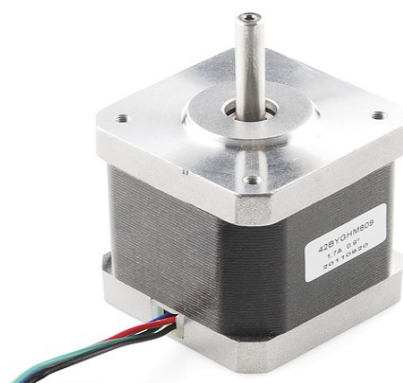
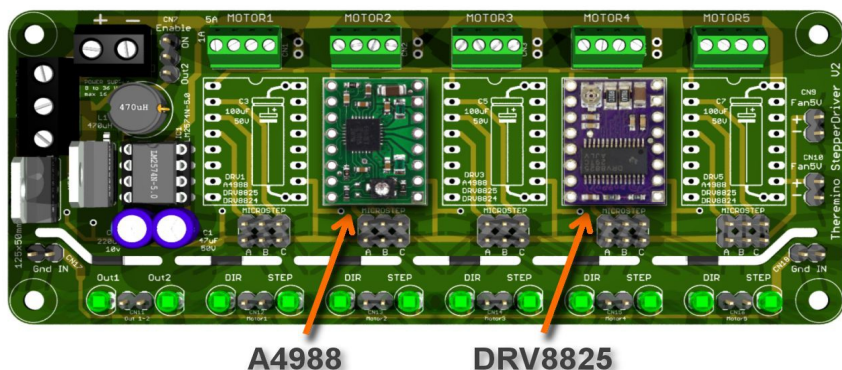
Quindi con 100 ohm e capacità da 300 pF si potrebbe arrivare fino a 1 o 2 MHz. Mentre con 100 ohm e capacità di 30 pF, si potrebbe arrivare fino a 10 o 20 MHz.

Questi calcoli sono per garantire una onda quadra con fronti abbastanza ripidi, cioè tempi di commutazione inferiori a un decimo del tempo totale del periodo.

Connettere i segnali di uscita per Motori Stepper

I motori stepper si collegano attraverso appositi moduli driver.

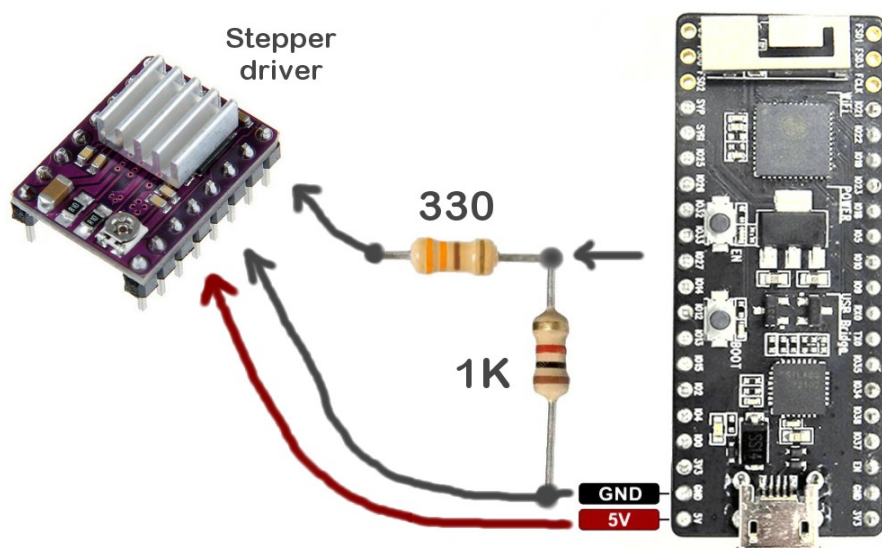
Per informazioni sui driver vedere [questa pagina](#).



Per assicurare che i motori Stepper stiano perfettamente fermi durante l'accensione o il riavvio del modulo, si dovrebbe aggiungere un resistore di pull-down ai segnali di uscita che vanno ai driver dei motori.

Il valore del resistore di pull-down deve essere compreso tra 1k e 4.7k e lo si deve collegare tra il Pin di uscita e la massa (GND).

L'eventuale resistore da 330 ohm (vedere la pagina precedente) deve essere connesso dopo, lungo il filo di segnale che va verso il driver del motore, ma non troppo lontano dal modulo (pochi centimetri al massimo).



I due resistori di questa immagine devono essere ripetuti per ogni filo STEP e DIR che va dal modulo IOT verso i driver dei motori Stepper.

I tipi di Pin

I Pin sono configurabili come:

- ◆ Non usato
- ◆ Uscita digitale
- ◆ Uscita PWM (da 0.02 Hz a 40 MHz)
- ◆ Uscita DAC (tensione di uscita regolabile)
- ◆ Uscita per i Servo-comandi
- ◆ Uscita per motori Stepper
- ◆ Ingresso digitale
- ◆ Ingresso di conteggio, frequenza e periodo
- ◆ Ingresso per Encoders a due/quattro fasi
- ◆ Ingresso ADC per potenziometri e trasduttori
- ◆ Ingresso CAP per pulsanti capacitivi
- ◆ Adc24

Pin properties	
Pin Type	Unused
	Unused
	Dig_out
	Pwm
	Servo
	Dac_8
	Dig_in
	Dig_in_pu
	Cap_16
	Counter
	Counter_pu
	Period
	Period_pu
	Encoder_a
	Encoder_a_pu

I **tipi con pullup**, il cui nome termina per "_pu", permettono di collegare facilmente interruttori, pulsanti e dispositivi open-collector, senza dover aggiungere resistori esterni (resistore di PullUp tipico = 45k, che produce una corrente di 70 uA quando l'ingresso è basso).

Tutti i Pin in uscita possono fornire correnti di circa +/-20 mA. Se si utilizzano contemporaneamente molti Pin in uscita la corrente disponibile per ognuno scende progressivamente fino a +/-10 mA.

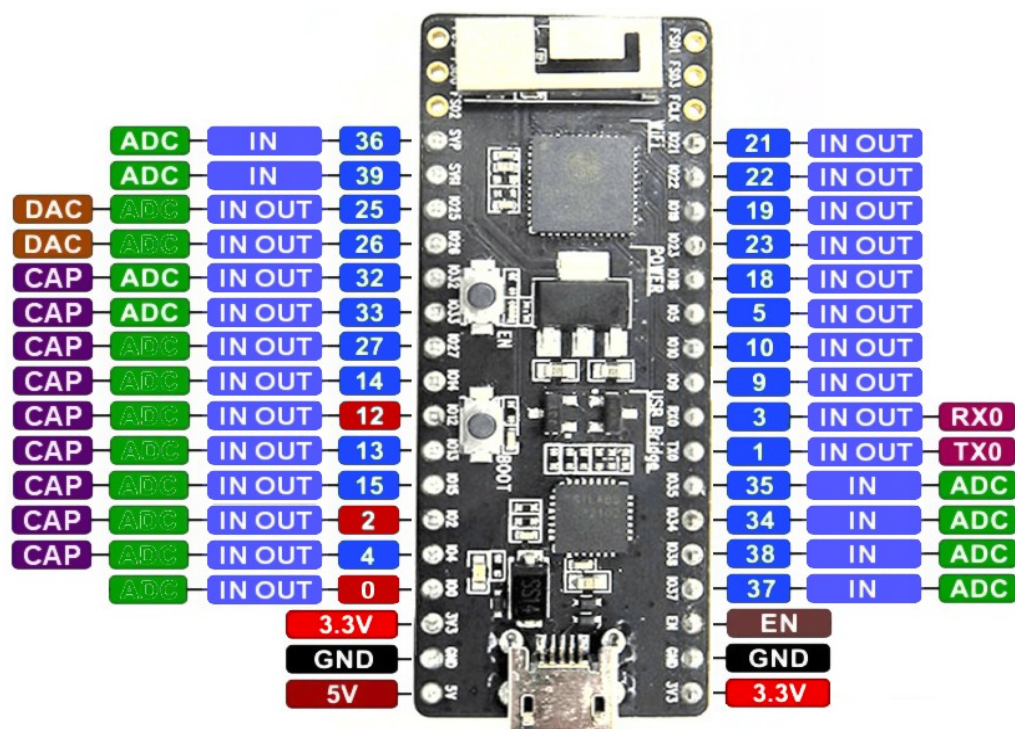
Tutti i Pin sono configurabili come "Unused", questo permette di diminuire il numero di byte che transitano e massimizzare il numero di scambi per secondo.

Non tutti i Pin sono configurabili in tutti i modi possibili. Le immagini delle prossime pagine mostrano le funzioni disponibili per ogni Pin, per i modelli di lotModule più comuni.

I tipi di Pin per le schede ESP32 - Pico V4

Questo è il modello di ESP32 che preferiamo, comunica più velocemente degli altri, ha un numero maggiore di Pin e la disposizione dei Pin è migliore.

Per utilizzare questi moduli si deve abilitare la riga
"#define IOTMODULE_PINOUT_TYPE 1" nel file "IotModuleSetup.h"



IN = Inputs only, and no pullups

IN OUT = DigIn / Counter / Period / Encoder
DigOut / Pwm / Servo

0 = Pin "0" must be high at startup and low while programming

2 = Pin "2" must be low while programming

12 = Pin "12" must be low at startup and while programming

ADC = Adc when WiFi is not active

RX0 TX0 = Used by serial debugger

Leave Pins 0, 2 and 12 open, or use them as output only, to control high impedance devices
Pins 2 and 12 could be input-pull-up, connected to GND with a button or an open collector

I Pin con l'etichetta **IN** possono essere programmati come : DigIn, Counter, Period e Encoder.

I Pin con l'etichetta **IN OUT** sono programmabili anche come : DigOut, Pwm e Servo.

I Pin con l'etichetta **DAC** danno in uscita una tensione regolabile da 0 a 3.3 volt.

I Pin con la parola **ADC** cancellata potrebbero essere utilizzati come ADC, ma solo comunicando via USB e disattivando la comunicazione WiFi (nelle versioni attuali non si può fare).

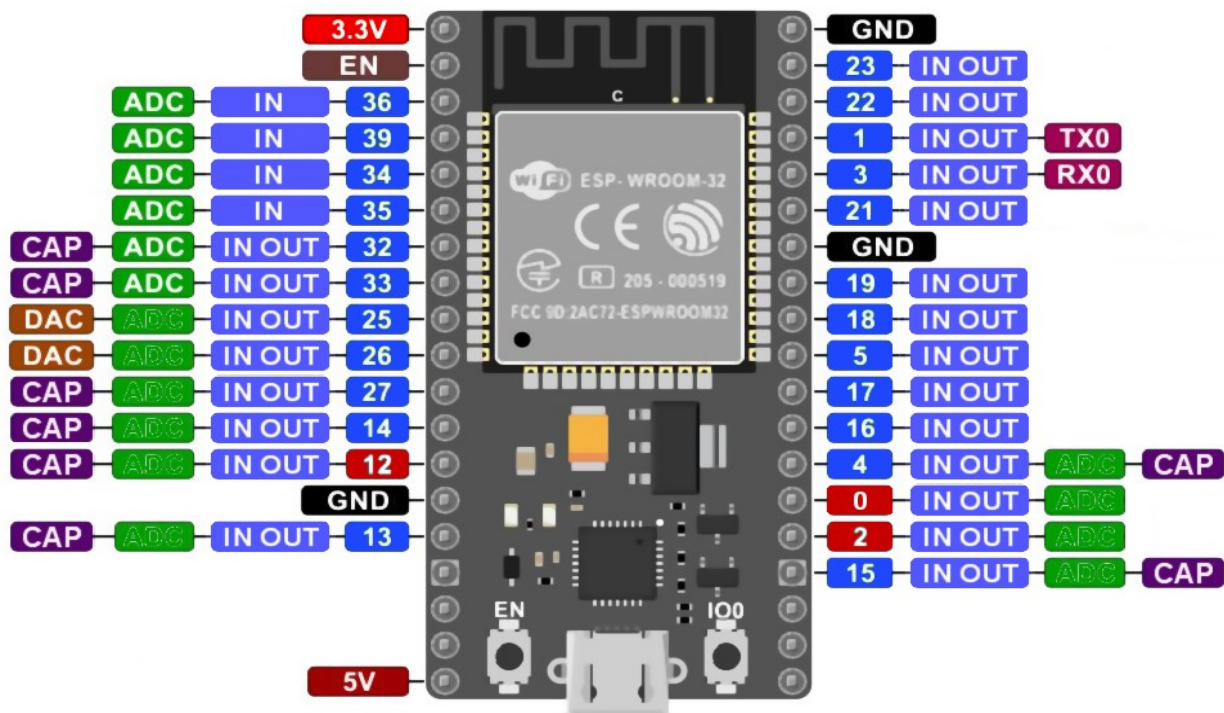
I Pin "0", "1", "2", "3" e "12" sono utilizzabili solo con particolari attenzioni, altrimenti il modulo non funzionerà più bene. Vedere la prossima pagina: [Collegare i Pin speciali](#).

I tipi di Pin per le schede ESP32 – WROOM

Questo modello di ESP32 è da utilizzare come seconda scelta.

Per utilizzare questi moduli si deve abilitare la riga

"#define IOTMODULE_PINOUT_TYPE 0" nel file lotModuleSetup.h



IN = Inputs only, and no pullups

ADC = Adc when WiFi is not active

IN OUT = DigIn / Counter / Period / Encoder
DigOut / Pwm / Servo

RX0 TX0 = Used by serial debugger

0 = Pin "0" must be high at startup and low while programming

2 = Pin "2" must be low while programming

12 = Pin "12" must be low at startup and while programming

Leave Pins 0, 2 and 12 open, or use them as output only, to control high impedance devices

Pins 2 and 12 could be input-pull-up, connected to GND with a button or an open collector

.....

I Pin con l'etichetta **IN** possono essere programmati come : DigIn, Counter, Period e Encoder.

I Pin con l'etichetta **IN OUT** sono programmabili anche come : DigOut, Pwm e Servo.

I Pin con l'etichetta **DAC** danno in uscita una tensione regolabile da 0 a 3.3 volt.

I Pin con la parola **ADC** cancellata potrebbero essere utilizzati come ADC, ma solo comunicando via USB e disattivando la comunicazione WiFi (nelle versioni attuali non si può fare).

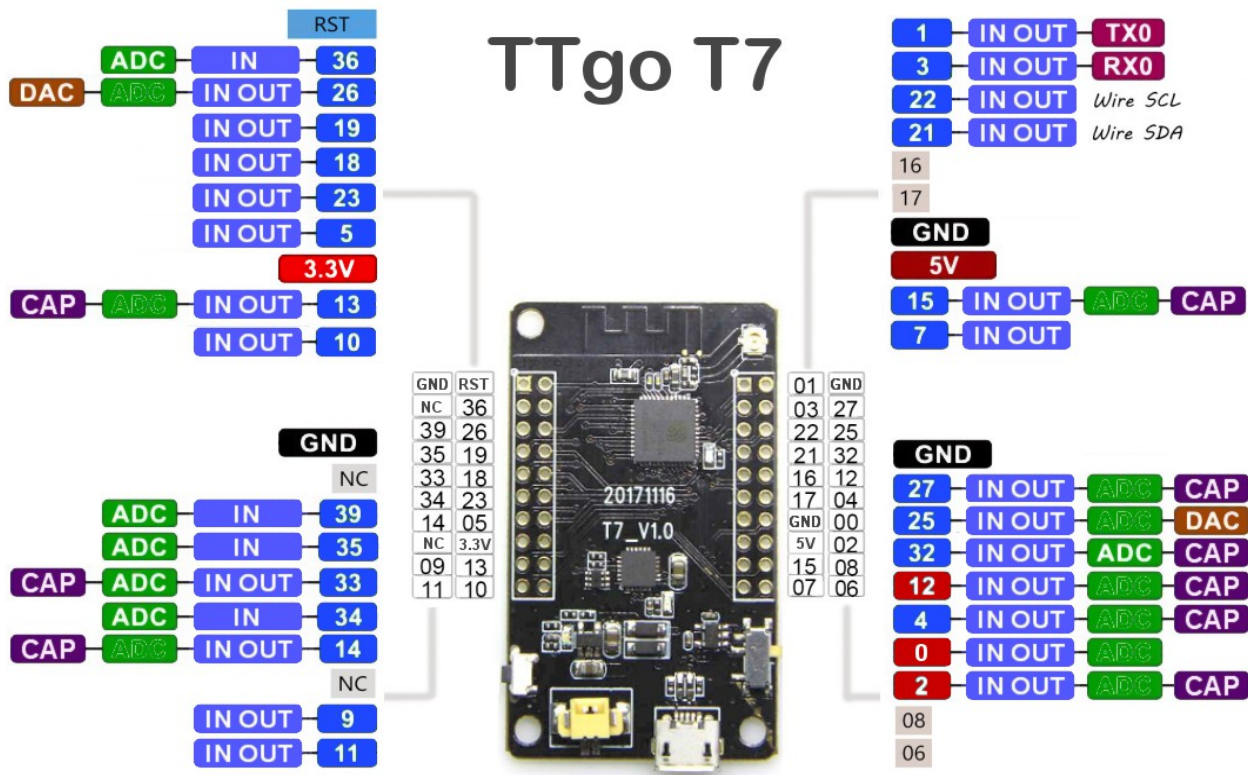
I Pin "0", "1", "2", "3" e "12" sono utilizzabili solo con particolari attenzioni, altrimenti il modulo non funzionerà più bene. Vedere la prossima pagina: [Collegare i Pin speciali](#).

I tipi di Pin per le schede - TTGO T7

Questo modello ha anche il connettore per una antenna esterna e una funzione UPS (con batteria al Litio da 3.7 volt) che viene ricaricata dal 5 volt proveniente dal connettore USB.

Per utilizzare questi moduli si deve abilitare la riga

"#define IOTMODULE_PINOUT_TYPE 2" nel file lotModuleSetup.h



IN = Inputs only, and no pullups

IN OUT = DigIn / Counter / Period / Encoder
DigOut / Pwm / Servo

0 = Pin 0 must be high at startup and low while programming

2 = Pin 2 must be low while programming

12 = Pin 12 must be low at startup and while programming

ADC = Adc when WiFi is not active

RX0 TX0 = Used by serial debugger

RST = Reset when high

NC NC = Not internally connected

06 08 16 17 = Do not connect !

Leave Pins 0, 2 and 12 open, or use them as output only, to control high impedance devices

Pins 2 and 12 could be input-pull-up, connected to GND with a button or an open collector

.....

I Pin con l'etichetta **IN** possono essere programmati come : DigIn, Counter, Period e Encoder.

I Pin con l'etichetta **IN OUT** sono programmabili anche come : DigOut, Pwm e Servo.

I Pin con l'etichetta DAC danno in uscita una tensione regolabile da 0 a 3.3 volt.

I Pin con la parola **ADC** cancellata potrebbero essere utilizzati come ADC, ma solo comunicando via USB e disattivando la comunicazione WiFi (nelle versioni attuali non si può fare).

I Pin "0", "1", "2", "3" e "12" sono utilizzabili solo con particolari attenzioni, altrimenti il modulo non funzionerà più bene. Vedere la prossima pagina: [Collegare i Pin speciali](#).

Collegare i Pin speciali

Pin 1 e 3

Questi due Pin sono utilizzati per il Debug del firmware, per mezzo di una porta seriale collegata attraverso il cavo USB.

Impostandoli come DigIn, DigOut o Pwm, o collegandoli a dispositivi a bassa impedenza, la comunicazione seriale smette di funzionare.

Se si rinuncia al collegamento seriale si possono usare questi due Pin sia in ingresso che in uscita, configurandoli come: DigIn, Counter, Period, Encoder, DigOut, Pwm e Servo.

Li si possono anche usare come Generic IN/OUT, 8/16/24 e Float. Utilizzandoli come Generic il collegamento seriale continua a funzionare.

Pin 0

Questo Pin deve essere mantenuto **alto** durante la **accensione**

Questo Pin deve essere mantenuto **basso** durante la **programmazione**

Tenerlo scollegato oppure usarlo in uscita per pilotare dispositivi ad alta impedenza.

Pin 2

Questo Pin deve essere mantenuto **basso** durante la **programmazione**

Tenerlo scollegato oppure usarlo in uscita per pilotare dispositivi ad alta impedenza.

Dato che forzarlo basso non crea problemi si potrebbe anche usarlo per un pulsante normalmente aperto collegato a massa, oppure collegarlo al collettore di un transistor open-collector.

Pin 12

Questo Pin deve essere mantenuto **basso** durante la **accensione**

Questo Pin deve essere mantenuto **basso** durante la **programmazione**

Tenerlo scollegato oppure usarlo in uscita per pilotare dispositivi ad alta impedenza.

Dato che forzarlo basso non crea problemi si potrebbe anche usarlo per un pulsante normalmente aperto collegato a massa, oppure collegarlo al collettore di un transistor open-collector.

Collegamenti dell'Adc24

Per collegare l'Adc24 abbiamo utilizzato i Pin 0, 2, 4, che sono speciali e difficili da usare. In questo modo non occupiamo nessuno dei Pin che si usano normalmente.

Utilizzare i Pin speciali come "Generic"

Tutti i Pin speciali possono essere utilizzati anche come "Generic", per comunicare dati con l'HAL, e in questo caso non danno nessun effetto collaterale.

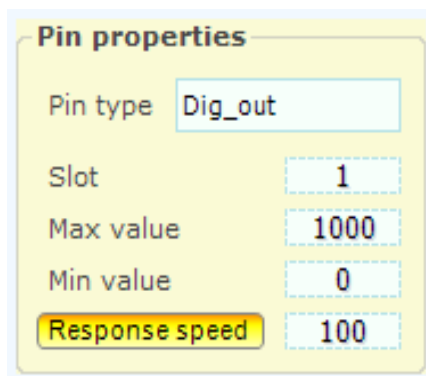
Stato dei Pin dopo il reset del modulo

Quando il modulo viene riavviato, o per mancanza di alimentazione, o perché qualcuno ha premuto il pulsantino di reset, i Pin si trovano in uno stato iniziale che non è ancora sotto il controllo della applicazione IoT HAL. Questa tabella indica lo stato iniziale di ogni Pin.

GPIO	Input mode after reset
0	InPullUp
1	InPullUp
2	InFloat
3	InPullUp
4	InPullDown
5	InPullUp
6	InPullUp
7	InPullUp
8	InPullUp
9	InPullUp
10	InPullUp
11	InPullUp
12	InPullDown
13	InFloat
14	InFloat
15	InPullUp
16	InFloat
17	InFloat
18	InFloat
19	InFloat
21	InFloat
22	InFloat
23	InFloat
25	Float
26	Float
27	InFloat
32	Float
33	Float
34	Float
35	Float
36	Float
37	Float
38	Float
39	Float

Per evitare di accendere dispositivi pericolosi o fastidiosi, si consiglia di aggiungere un resistore tra il loro Pin e GND. Un resistore da circa 10k ohm è adatto per tutti i Pin "float", mentre per i Pin con "pullup", o che sono collegati a resistori sul circuito stampato, si dovrà usare un valore più basso. Resistori da 1k dovrebbero andare bene, ma è meglio controllare caso per caso.

Il parametri comuni di tutti i Pin



Pin properties	
Pin type	Dig_out
Slot	1
Max value	1000
Min value	0
Response speed	100

"Slot" indica dove scrivere o leggere i dati. Gli Slot sono mille, numerati da 0 a 999, e possono essere letti o scritti da tutti i Pin e da tutte le applicazioni del sistema Theremino.

Attenzione: molte applicazioni e molti Pin possono leggere dallo stesso Slot, ma si deve evitare di configurare più di un Pin in scrittura sul medesimo Slot, facendolo non si rompe nulla ma si ottengono risultati indefiniti.

"Max value" normalmente si tiene a mille e indica il valore che il Pin deve avere quando è al massimo.

"Min value" normalmente si tiene a zero e indica il valore che il Pin deve avere quando è al minimo.

Regolando Max value e Min value con valori diversi da 0 e 1000 si può ottenere qualunque rapporto di scala e taratura. Se si scambiano i due valori (valore min maggiore di max) allora la scala si ribalta, questo è utile per invertire il movimento dei servocomandi o ribaltare la lettura di sensori che agiscono al contrario.

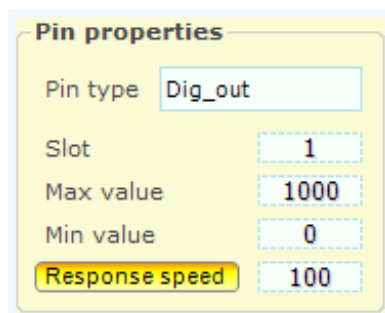
"Response speed" regola il filtro IIR (Infinite Impulse Response) per il migliore compromesso tra rumore e velocità di risposta. Con il valore 100 il filtro è disabilitato e si ha la massima velocità di risposta, con il valore 1 si ha il massimo filtraggio (eliminazione di ogni tremolio) ma una risposta molto lenta (circa un secondo). Normalmente si usa il valore 30 che fornisce un buon filtraggio e risponde abbastanza velocemente.

Se il pulsante "Response speed" è premuto, il filtro IIR si adatta alle variazioni in modo da ottenere una maggiore reattività quando vi sono ampie variazioni e un maggiore smorzamento, quando le variazioni sono minori. Come risultato si ottiene una buona stabilità delle cifre, senza penalizzare troppo il tempo di assestamento.

I segnali di alcuni sensori potrebbero funzionare male con "velocità di risposta" premuto. Questo è particolarmente vero per i sensori che producono un segnale con piccole variazioni intorno ad un alto valore di base. In questo caso il segnale non arriva al valore finale, o è molto lento ad arrivare. Se si sperimenta questo, disabilitare "Response speed".

I tipi di Pin in "Output" --> Dig_out / Dac_8

◆ Dig_out



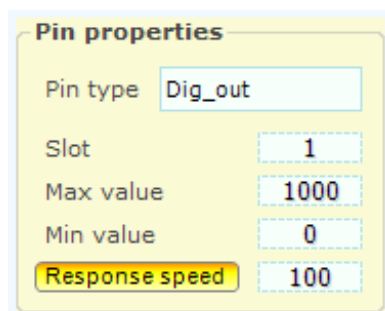
Pin properties	
Pin type	Dig_out
Slot	1
Max value	1000
Min value	0
Response speed	100

Questo tipo di Pin fornisce una uscita digitale.

Il valore in arrivo da uno Slot, limitato tra "Min value" e "Max value" e filtrato da "Response speed", viene confrontato con il valore intermedio tra "Min value" e "Max value", se lo supera il Pin si accende altrimenti si spegne.

Il Pin può assumere solo le tensioni 0 Volt (spento) e 3.3 Volt (acceso) e la corrente di uscita è limitata a circa +/- 20 mA

◆ Dac_8



Pin properties	
Pin type	Dig_out
Slot	1
Max value	1000
Min value	0
Response speed	100

Questo tipo di Pin fornisce in uscita una tensione analogica regolabile da 0 a 3.3 volt.

Il valore in arrivo da uno Slot, viene scalato con i valori "Min value" e "Max value" e trasformato in una tensione di uscita variabile.

La tensione di uscita sul Pin è regolabile da 0 Volt e 3.3 Volt e la corrente di uscita è limitata a circa +/- 20 mA

La risoluzione dei DAC è solo 8 bit, quindi si possono ottenere solo 256 livelli di tensione (con gradini di circa 13 mV tra un livello e il successivo).

Per ottenere una risoluzione maggiore (fino a 20 bit), si possono usare le uscite PWM e un semplice adattatore composto da un resistore e un condensatore, come spiegato nelle prossime pagine.

I tipi di Pin in "Output" --> Servo / Pwm

◆ Servo

Pin properties	
Pin type	Servo_16
Slot	1
Max value	1000
Min value	0
Response speed	100
Servo properties	
Max time (uS)	2500
Min time (uS)	500

Questo tipo di Pin pilota direttamente i servo comandi.

Il valore in arrivo da uno Slot, limitato tra "Min value" e "Max value" e filtrato da "Response speed" viene trasformato in impulsi di larghezza tra "Min time (uS)" e "Max time (uS)"

Il tempo di ripetizione è adeguato ai normali servocomandi per modellistica che tra min e max ruotano di circa 180 gradi.

Il Pin fornisce tensioni di 0 e 3.3 Volt, adeguate a tutti i normali servo alimentati da 3 a 6 Volt e una corrente sufficiente a pilotare decine di servo in parallelo.

◆ Pwm

Pin properties	
Pin type	Pwm_16
Slot	1
Max value	1000
Min value	0
Response speed	100
PWM properties	
Max time (uS)	4000
Min time (uS)	0
Logarithmic response	<input type="checkbox"/>

Questo tipo di Pin fornisce una uscita PWM (Modulazione della larghezza degli impulsi)

Il valore in arrivo da uno Slot, limitato tra "Min value" e "Max value" e filtrato da "Response speed", viene trasformato in impulsi di larghezza tra "Min time ()" e "Max time (uS)"

Il Pin fornisce impulsi tra le tensioni 0 Volt (spento) e 3.3 Volt (acceso) e la corrente di uscita è limitata a circa +/- 20 mA

Le frequenze impostabili vanno da 0.02 Hz fino a 40 Mhz ma la precisione scende man mano che si sale di frequenza, come spiegato nelle prossime pagine.

I Pin di tipo Servo e Pwm hanno delle limitazioni, leggere le due pagine seguenti.

I tipi di Pin in "Output" --> Servo e Pwm

Il processore ESP32 usato negli IoTModule ha risorse limitate, per cui si possono programmare i Pin di tipo Stepper, Servo e Pwm con le seguenti regole:

- ◆ Ci sono sedici timers in tutto e questi timer vengono usati per gli Stepper, i Servo e i Pwm.
- ◆ Ogni Stepper utilizza due timer, quindi gli Stepper disponibili sono al massimo otto.
- ◆ Gli Stepper vengono allocati per primi, per cui ad ogni Stepper impostato si perdono due Servo o Pwm.
- ◆ I canali rimanenti possono essere impostati tutti come Servo, oppure tutti come Pwm, o in qualunque combinazione dei due tipi. Si potrebbero ad esempio impostare fino a otto Servo e otto Pwm, oppure dodici Servo e quattro Pwm, e così via.
- ◆ Se i Servo non sono in numero pari si perde un canale. Quindi se si impostano sette Servo, poi non si potranno impostare nove Pwm, ma solo otto. E se si impostano quindici Servo il sedicesimo non lo si potrà impostare come Pwm.
- ◆ Quando i timer non bastano più per mantenere le frequenze dei Pwm separate, gli ultimi Pwm vengono collegati a coppie e cambiando la frequenza di uno si cambia la frequenza di ambedue i Pwm della coppia.

Questa immagine dà una idea di come i canali PWM si accoppiano e condividono la stessa frequenza (in rosso).

Nel caso si utilizzassero anche gli Stepper, ogni Stepper occuperebbe due righe. Per cui il numero di Pwm (e di Servo) disponibili diminuirebbe di due e i Pwm si accoppierebbero due righe prima.

		ledc channel															
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
numero pwm	0																
	1	Y															
	2	Y		Y													
	3	Y		Y		Y											
	4	Y		Y		Y		Y									
	5	Y		Y		Y		Y		Y							
	6	Y		Y		Y		Y		Y		Y					
	7	Y		Y		Y		Y		Y		Y		Y			
	8	Y		Y		Y		Y		Y		Y		Y		Y	
	9	Y		Y		Y		Y		Y		Y		Y		Y	
	10	Y		Y		Y		Y		Y		Y		Y		Y	
	11	Y		Y		Y		Y		Y		Y		Y		Y	
	12	Y		Y		Y		Y		Y		Y		Y		Y	
	13	Y		Y		Y		Y		Y		Y		Y		Y	
	14	Y		Y		Y		Y		Y		Y		Y		Y	
	15	Y		Y		Y		Y		Y		Y		Y		Y	
	16	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Alcuni Pin sono utilizzabili solo in ingresso e non possono essere impostati come Servo o Pwm. Vedere le mappe dei Pin in [questa pagina](#) e nella [pagina seguente](#).

I tipi di Pin in "Output" --> Pwm

The image shows two configuration windows. The top window, titled "Pin properties", has fields for "Pin Type" (set to "Pwm_fast"), "Slot" (3), "Max value" (1000), "Min value" (0), and "Response speed" (100). The bottom window, titled "Pwm_Fast properties", has fields for "Frequency (Hz)" (1000), "Duty cycle 0-1000" (0), "Frequency from slot" (unchecked), and "Duty cycle from slot" (checked).

La frequenza minima generabile è 0.02 Hz e la massima è 40 MHz circa. Il Duty Cycle va da zero (segnale di uscita sempre basso) fino al 100% (segnale di uscita sempre alto).

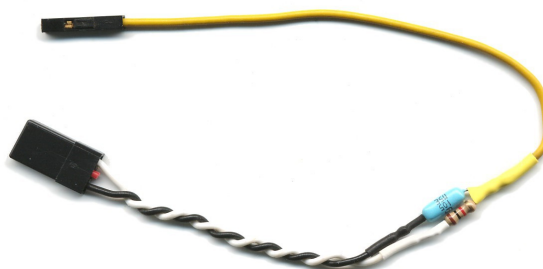
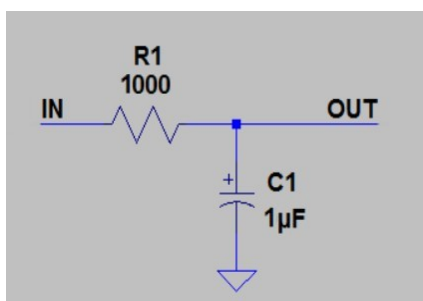
Il valore in arrivo dallo Slot imposta il rapporto di tempo, tra segnale basso e alto.

Il valore Slot, solitamente tra 0 e 1000, viene prima filtrato e poi scalato con "Min value" e "Max value". Normalmente si impostano Min=0 / Max=1000 e si regola il PWM, fornendo valori da 0 a 1000.

La granularità delle regolazioni dipende dalla frequenza impostata.

- ◆ A 1000 Hz la precisione del Duty Cycle è 16 bits (errori: 0.0015%) e quella della frequenza è di 14 bit (errori: 0.006%)
- ◆ A 16 KHz la precisione del Duty Cycle è 12 bits (errori: 0.024%) e quella della frequenza è di 10 bit (errori: 0.1%)
- ◆ A 1 MHz la precisione del Duty Cycle scende a soli 6 bits (errori: 1.5%) e quella della frequenza a soli 4 bit (errori: 6%)

Con i Pin di tipo Pwm e un semplice adattatore si ottiene una tensione di uscita analogica, regolabile con precisione da 0 a 3.3 Volt. Impostare la frequenza a 15 KHz.



Per ulteriori informazioni sugli adattatori, leggere questa pagina:

<https://www.theremino.com/hardware/adapters#pwm>

Frequenza e risoluzione dei segnali Pwm

Frequenza	Risoluz.	Canali High-Speed	Canali Low-Speed	Min. step
40 Mhz	1bit	Fisso 50%	Fisso 50%	
20 Mhz	2bit	0% 50% 100%	50% 100%	
10 Mhz	3bit	0% 25% 50% 75%100%	25% 50% 75% 100%	
5 Mhz	4bit			
2500 kHz	5bit			
1250 kHz	6bit	64 valori	64 valori	51 mV
625 kHz	7bit			
312 kHz	8bit	256 valori	256 valori	13 mV
156 kHz	9bit			
78 kHz	10bit	1024 valori	1024 valori	3.2 mV
39 kHz	11bit			
19 kHz	12bit	4096 valori	4096 valori	800 uV
9 kHz	13bit			
4 kHz	14bit	16384 valori	16384 valori	200 uV
2 kHz	15bit			
1 kHz	16bit	65536 valori	65536 valori	50 uV
610 Hz	17bit			
305 Hz	18bit			
152 Hz	19bit			
76 Hz	20bit	oltre un milione di valori	oltre un milione di valori	3 uV
38 Hz	20bit			
19 Hz	20bit			
9 Hz	20bit			
4 Hz	20bit			
2 Hz	20bit			
1 Hz	20bit			

Quando si utilizzano frequenze basse, i segnali PWM hanno una risoluzione (Nota 1) molto alta, addirittura dai 16 ai 20 bit quando si scende sotto al KHz.

Salendo di frequenza la risoluzione (Nota 1) decresce progressivamente, fino ad arrivare a zero per frequenze oltre i 20 MHz

(Nota 1) La risoluzione rappresenta la precisione con cui è possibile regolare, sia i tempi che la frequenza del segnale PWM.

Particolarità dei segnali PWM

Problemi con le alte frequenze

Regolando i canali di uscita PWM con frequenze alte (dai 100 KHz in su), possono avvenire errori di comunicazione. Questi errori sono dovuti, sia alle cadute di tensione provocate dal forte consumo dei segnali PWM, sia alle armoniche emesse dai fili collegati alle uscite PWM. Le armoniche del segnale disturbano i sensibili circuiti di ingresso collegati alla antenna.

Aumentando la frequenza dei PWM e collegando fili lunghi alle uscite PWM, il problema si aggrava. E utilizzando particolari frequenze di PWM, il modulo potrebbe anche disconnettersi.

Nel caso di fili lunghi non è nemmeno possibile utilizzare cavo schermato a causa della sua alta capacità. In alcuni casi, dovendo necessariamente utilizzare frequenze di PWM alte, si potrebbe pensare di aggiungere un resistore in serie al segnale PWM (vicino al modulo ESP32) e poi utilizzare cavo schermato. Il resistore limiterà la corrente di uscita, e in collaborazione con la capacità del cavo, formerà un passa basso che eliminerà i disturbi a radio frequenza. Il resistore va scelto di valore più alto possibile, compatibilmente con le esigenze del circuito da pilotare e con la banda passante necessaria.

Questi problemi si eliminano completamente utilizzando frequenze basse per i segnali PWM, e mantenendo i fili corti e lontani dalla antenna.

Vedere la pagina: [Connettere i segnali di uscita](#)

Con alte frequenze di PWM, i LED non si accendono più.

Quando si alza di molto la frequenza dei segnali PWM basta poca capacità (e i led ne hanno molta) per sovraccaricare i circuiti di uscita. Superata la corrente massima le uscite PWM producono una tensione continua che è la media del segnale PWM. In queste condizioni i LED cominciano ad accendersi solo quando si superano i 2.5 o 3 volt, cioè quando il segnale PWM è verso il massimo.

Invece, con frequenze basse, i LED si accendono con lampi di luce di durata pari alla parte alta del segnale PWM. In queste condizioni la luce media emessa dai LED è proporzionale ai tempi del segnale PWM e il LED si accendono anche con il PWM al 20% o meno.

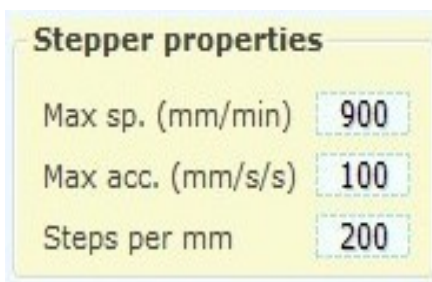
I tipi di Pin in "Output" --> Stepper

Ogni motore stepper ha bisogno di due Pin, uno per gli STEP e uno per la DIREZIONE. Lo IotModule permette di controllare fino a 8 motori configurando una coppia adiacente di pin per ogni motore.

I Pin di tipo Stepper leggono dallo Slot un valore, che è semplicemente la destinazione in millimetri. Le applicazioni più semplici, possono specificare una destinazione lontana, e lasciar fare tutto al firmware. Applicazioni più esigenti, possono calcolare loro stesse il percorso e inviare frequenti destinazioni intermedie. Con questa tecnica, una applicazione può controllare la velocità di lavoro (feed), e stabilire con precisione il percorso, anche in più dimensioni. Per ottenere un movimento fluido bastano 20 destinazioni al secondo (fino a 50 per le applicazioni più esigenti).

Per invertire la direzione di movimento di un asse basta scambiare i valori "1000" e "0", delle caselle "1000 means mm" and "0 means mm".

Parametri specifici dei Pin di tipo Stepper



The image shows a software window titled "Stepper properties". It contains three input fields with numerical values: "Max sp. (mm/min)" set to 900, "Max acc. (mm/s/s)" set to 100, and "Steps per mm" set to 200. Each field has a small up/down arrow icon to its right.

Stepper properties	
Max sp. (mm/min)	900
Max acc. (mm/s/s)	100
Steps per mm	200

Max Speed - Massima velocità in millimetri al minuto. Il firmware controlla continuamente le destinazioni inviate dal software. Se il software sta chiedendo troppo al motore, il firmware limita la velocità, per evitare che perda passi. Alzare questo valore fino a vedere che il motore perde passi (fa un rumore acuto e si ferma) e poi diminuirlo di un 20..50%, per tornare in una zona sicura. Ripetere le prove anche sotto carico, o frenando il motore manualmente, in modo da assicurarsi di avere un certo margine.

Max Acc - Questa è la massima accelerazione (e decelerazione), in millimetri al minuto. Il firmware controlla continuamente le destinazioni inviate dal software. Se il software sta chiedendo troppo al motore, il firmware limita la sua accelerazione, per evitare che perda passi. Alzare questo valore fino a vedere che il motore perde passi durante i cambi di direzione (fa un rumore acuto e si ferma) e poi diminuirlo di un 20..50%, per tornare in una zona sicura. Ripetere le prove anche sotto carico, o frenando il motore manualmente, in modo da assicurarsi di avere un certo margine.

Steps per mm - Qui si devono impostare gli step, che il motore fa in una rotazione, moltiplicati per i microstep, impostati nel controller, e divisi per i millimetri, prodotti da una rotazione del motore. Se ogni rotazione, produce un millimetro di spostamento, e il motore è da duecento passi per giro, e non si usano i microstep, allora si imposta il valore: $200 \text{ (passi per giro)} \times 1 \text{ (microstep)} / 1 \text{ (mm per giro)} = 200$. Se si usano sedici microstep allora si imposta il valore: $200 \text{ (passi per giro)} \times 16 \text{ (microstep)} / 1 \text{ (mm per giro)} = 3200$

I Pin di tipo Stepper_Dir non hanno parametri da regolare. Sono solo un segnaposto per il Pin fisico di uscita, che stabilisce la direzione del motore. Non è necessario usare il valore, che questi Pin scrivono nello Slot, ma alcune applicazioni potrebbero trovarlo utile. Il valore che viene scritto nello Slot, è la distanza dalla destinazione, in millimetri (e fino alle frazioni di millesimo di millimetro). Questa informazione può essere usata a scopo diagnostico, o per algoritmi che devono rispettare una tolleranza specificata. Disponendo di questa informazione il software può lavorare ad anello chiuso e sempre alla massima velocità. Controllando continuamente la distanza di ogni motore dalla destinazione, il software può rallentare esattamente quando serve, senza fare complessi calcoli di velocità, traiettorie e accelerazioni.

I tipi di Pin in "Output" --> Stepper e Stepper_Dir

I valori dei Pin di tipo Stepper

Il valore letto dallo Slot, viene rapportato (con "1000 means mm" e "0 means mm") e trasformato in un valore tra zero e uno.

Se si impostano "1000 means mm" = 1000 e "0 means mm" = 0, allora non vengono eseguite conversioni di scala e il valore proveniente dallo Slot viene considerato "millimetri".

Da qui in poi il valore è sempre in millimetri. "Zero" indica zero millimetri e "uno" indica 1000 millimetri. Questo valore non è limitato tra zero e uno, ma tra due miliardi di step positivi, e due miliardi di step negativi. Se si usa "Steps per mm = 200" i limiti sono: +10 Km e -10 Km.

Il valore viene quindi filtrato con un filtro IIR (lineare o esponenziale), regolabile con "Response Speed". Il valore di uscita del filtro viene chiamato "Filtrato"

Il valore finale che viene inviato all'hardware è un numero di STEP (pre-moltiplicato per il valore "Steps per mm") e rappresenta la "destinazione".

Il valore speciale NAN_Reset, ha il significato speciale di azzeramento dell'asse. Quando si scrive un Reset, nello Slot di un Pin di tipo Stepper, il motore si ferma immediatamente. Successivamente, il primo valore che verrà scritto nello Slot, rappresenterà il valore "zero di riferimento". Il NAN_Reset è disponibile in Theremino Automation come "Reset", oppure nella nuova classe "ThereminoSlots", scaricabile con i sorgenti di Theremino Automation.

I valori dei Pin di tipo Stepper_Dir

Ogni Pin Stepper_Dir è sempre associato a un Pin Stepper.

Si tratta di un Pin particolare, sia di uscita che di ingresso. Fornisce al motore il segnale elettrico di direzione (quindi una uscita) ma nello stesso tempo si comporta come ingresso e fornisce informazioni al software.

Il valore grezzo che viene letto dall'hardware, è il numero di STEP (positivi o negativi), che mancano per arrivare alla "destinazione" specificata. La applicazione HAL calcola i millimetri (e frazioni), dividendo il valore grezzo, per il valore "Steps per mm". Infine questo valore in millimetri, viene scritto nello Slot, e può essere letto dalle altre applicazioni, solitamente da una applicazione CNC.

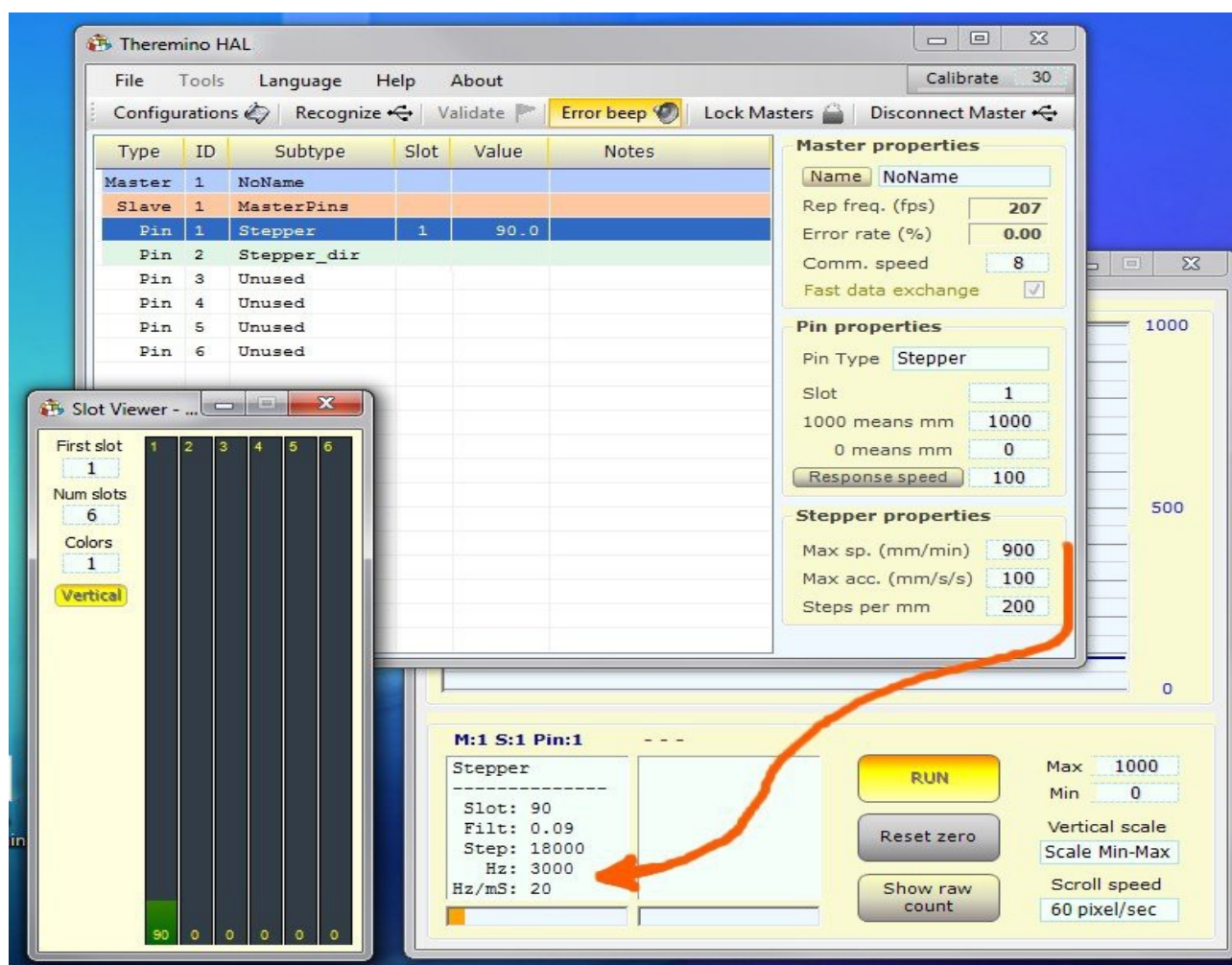
La applicazione CNC, conoscendo la distanza rimanente e la destinazione (da lei stessa specificata), può calcolare, con una semplice sottrazione, la posizione effettiva del motore. Conoscendo la posizione di ogni motore in ogni istante, gli algoritmi di controllo sono semplificati e il loro funzionamento è più preciso.

Informazioni più dettagliate sui Pin di tipo Stepper e sui driver per i motori stepper, in queste pagine:

<http://www.theremino.com/hardware/outputs/motors>

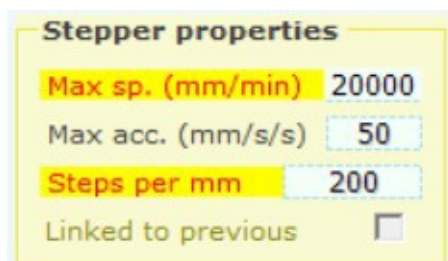
<http://www.theremino.com/technical/pin-types>

I tipi di Pin in "Output" --> Dettagli degli Stepper



Per aprire la finestra dei dettagli fare doppio click sulla linea del Pin (di tipo Stepper). Nella seconda finestra, in basso, si leggono i dettagli del Pin selezionato.

I dettagli del Pin mostrano il prodotto della velocità massima e degli step per millimetro in Hz (step per secondo). Questi valori sono utili durante le prove per stabilire quanti micro-step usare. In alcuni casi, può essere utile conoscere la destinazione grezza (in steps), al posto della destinazione in millimetri.



La massima frequenza degli step è definita nel file "IotModuleSetup.h". A seconda dell'intervallo scelto la frequenza massima ottenibile va da 1200 Hz a 307200 Hz. Se con i parametri impostati si supera la frequenza massima, si viene avvertiti dalle caselle "MaxSpeed" e "Steps per mm", che si colorano di giallo e arancione.

Se le caselle si colorano, allora si deve ridurre "MaxSpeed", oppure ridurre "Steps per mm", diminuendo anche la regolazione di microstep sul driver.

I tipi di Pin in "Input" <-- Dig / Adc / Cap

◆ Dig_in e Dig_in_pu

Pin properties	
Pin type	Dig_in
Slot	1
Max value	1000
Min value	0
Response speed	100

Questo tipo di Pin fornisce un ingresso digitale.

Il valore di tensione viene letto con un ingresso digitale (basso se minore di 0.8 volt e alto se maggiore di 2.5 volt) e trasformato in una informazione Acceso-Spento che infine diventano "Max value" e "Min value". Il valore viene infine filtrato con "Response speed" e poi scritto nello Slot. Il filtraggio produce valori intermedi e approssimativamente proporzionali al rapporto di tempo tra Acceso e Spento

◆ Adc_16

Pin properties	
Pin type	Adc_16
Slot	1
Max value	1000
Min value	0
Response speed	30

Questo tipo di Pin fornisce un ingresso analogico.

Il valore di tensione da 0 Volt a 3.3 Volt viene trasformato in un numero tra "Min value" e "Max value". Il valore viene infine filtrato con "Response speed" e poi scritto nello Slot. Il filtraggio riduce il rumore presente nel segnale di ingresso, ma rallenta la risposta. Il valore 30 rappresenta un buon compromesso tra velocità e rumore.

◆ Cap_16

Pin properties	
Pin type	Cap_16
Slot	1
Max value	1000
Min value	0
Response speed	30

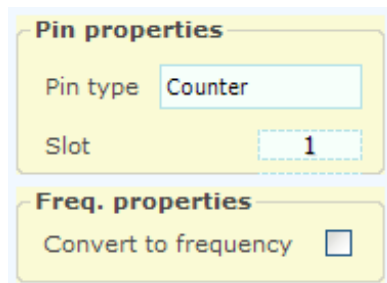
Touch properties	
Min variation	10
Proportional area	0

Questo tipo di Pin permette di leggere semplici tasti come con un Makey Makey (<http://vimeo.com/60307041#>) ma con prestazioni superiori. (i tasti non sono resistivi ma capacitivi e quindi possono essere regolati per funzionare anche solo sfiorandoli, senza contatto, attraverso un isolante e senza filo aggiuntivo di collegamento della terra)

Oltre al classico funzionamento tipo MakeyMakey è anche possibile ottenere un controllo graduale, come con i cursori "slider", il controllo di "espressione" determinato dalla la velocità di pressione dei tasti o la lettura di valori capacitivi grezzi, come i sensori di umidità. Ulteriori informazioni sui tasti capacitivi in [queste pagine](#).

I tipi di Pin in "Input" <-- Counter

◆ Counter e Counter_pu

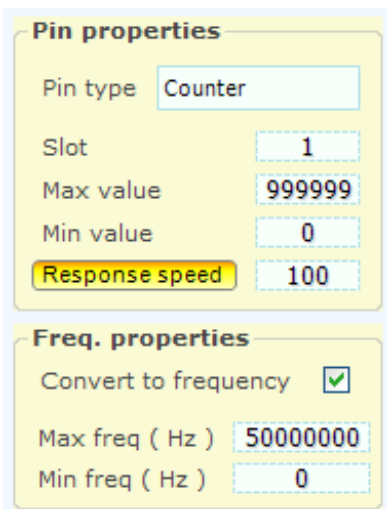


The screenshot shows the 'Pin properties' dialog box. Under the 'Pin properties' section, 'Pin type' is set to 'Counter' and 'Slot' is set to '1'. Under the 'Freq. properties' section, the 'Convert to frequency' checkbox is unchecked.

Tutti i Pin possono essere programmati come Counter o Counter_pu.

Il conteggio viene fatto con un contatore hardware che funziona fino a frequenze di molte decine di MHz.

◆ Counter e Counter_pu con l'opzione "Freq"



The screenshot shows the 'Pin properties' dialog box with additional frequency-related settings. Under 'Pin properties', 'Pin type' is 'Counter', 'Slot' is '1', 'Max value' is '999999', 'Min value' is '0', and 'Response speed' is '100'. Under 'Freq. properties', the 'Convert to frequency' checkbox is checked, 'Max freq (Hz)' is '50000000', and 'Min freq (Hz)' is '0'.

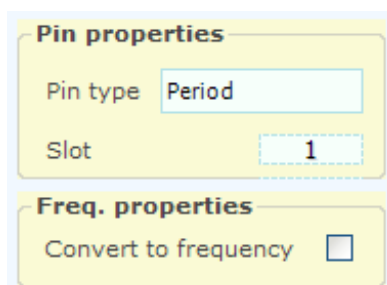
I Pin programmati come Counter o Counter_pu possono essere trasformati da contatori a frequenzimetri.

Il valore di frequenza rapportato da "Min freq" e "Max freq", viene poi rapportato tra "Min vale" e "Max value", filtrato con "Response speed" e infine inviato allo Slot.

I counter con l'opzione frequenza sono adatti alla misura di frequenze abbastanza alte (dai 10 KHz in su). Per le frequenze basse (da qualche KHz fino a 0.02 Hz) è preferibile utilizzare i "Period" spiegati nella prossima pagina, che forniscono una misura più stabile e meno rumorosa.

I tipi di Pin in "Input" <-- Period

◆ Period e Period_pu



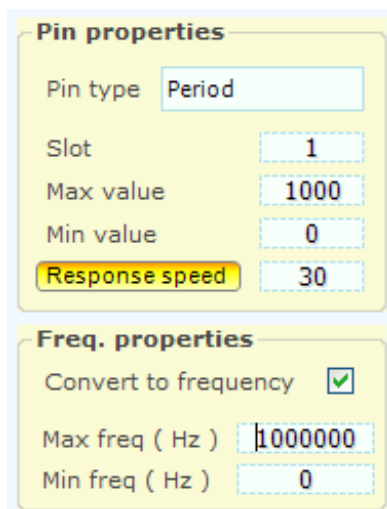
The screenshot shows the 'Pin properties' dialog box. Under 'Pin type', 'Period' is selected. Under 'Slot', the value '1' is entered. Under 'Freq. properties', the 'Convert to frequency' checkbox is unchecked.

Questo tipo di Pin misura il periodo di una forma d'onda ripetitiva, da salita a salita, da un periodo minimo di 50 uS, fino ad un massimo di circa 260 secondi.

La risoluzione è di tre microsecondi e la precisione è del +/- 1% in un range di temperatura ambiente da 0C a 50C

Il valore che si legge è in microsecondi e frazioni di microsecondo. Se il valore è inferiore a 50 uS viene scartato. In questo modo si eliminano eventuali errori dovuti a disturbi elettrici durante i fronti del segnale.

◆ Period e Period_pu con l'opzione "Freq"



The screenshot shows the 'Pin properties' dialog box. Under 'Pin type', 'Period' is selected. Under 'Slot', the value '1' is entered. Under 'Max value', the value '1000' is entered. Under 'Min value', the value '0' is entered. Under 'Response speed', the value '30' is entered. Under 'Freq. properties', the 'Convert to frequency' checkbox is checked. Under 'Max freq (Hz)', the value '1000000' is entered. Under 'Min freq (Hz)', the value '0' is entered.

I Pin programmati come Period o Period_pu possono essere trasformati da contatori a frequenzimetri.

Questa tecnica permette di misurare frequenze molto basse (fino a meno di un decimo di Hertz) con altissima risoluzione. La frequenza massima misurabile è 20 KHz.

Il valore di frequenza limitato tra "Min freq" e "Max freq", viene poi rapportato tra "Min value" e "Max value", filtrato con "Response speed" e infine inviato allo Slot.

Se il valore di frequenza supera i 20 KHz viene scartato. In questo modo si eliminano eventuali errori dovuti a disturbi elettrici durante i fronti del segnale.

Le misure che si effettuano con i Pin di tipo Period sono affette da un rumore di alcuni microsecondi. Per cui alla frequenza di 1000 Hz si ottengono circa dieci bit di rapporto segnale rumore. Queste caratteristiche sono leggermente inferiori a quelle ottenibili con i moduli Master, perché i Master sono basati su un Micro-Controllore mentre lo ESP32 contiene un sistema operativo (Free-RTOS).

I tipi di Pin in "Input" <-- Encoder

Pin properties
Pin Type
Slot
Max value
Min value

Pin properties
Pin Type

Tutti i Pin possono essere programmati come Encoder.

Per ogni Encoder sono necessari due Pin di ingresso: Encoder_A e Encoder_B, oppure Encoder_A_Pu, e Encoder_B_Pu. Questa coppia di ingressi legge le due fasi degli Encoder "in quadratura".

Il conteggio dell'encoder viene scritto nello Slot associato al Pin "Encoder_A". Ogni Pin di tipo "Encoder_A" usa 16 bit per la trasmissione dei dati, mentre gli "Encoder_B" sono solo un segnaposto e non inviano dati.

Gli Encoder leggono la posizione angolare di un perno, come i potenziometri, ma il numero di giri è illimitato. Il conteggio generato va da 0 a 65535 (16 bit). Quando il conteggio supera 65535 il numero riparte da zero. Questo sistema permette a molte applicazioni, asincrone tra loro, di leggere il numero progressivo senza perdere conteggi.

Per ulteriori informazioni sugli Encoder, leggere questa pagina:

<http://www.theremino.com/hardware/inputs/sensors#encoders>



Esistono Encoder simili a piccoli potenziometri (i più noti sono i KY-040 di questa immagine). Questi modelli sono meccanici e forniscono 18, 20 o 24 impulsi per giro a seconda del costruttore. Il firmware ricava da questi impulsi 72, 80 o 96 posizioni angolari per ogni giro, se regolato nel modo software (x2), oppure 36, 40 o 48, se regolato nel modo hardware (x4).

Gli Encoder professionali (ottici o magnetici) funzionano senza problemi, ma quelli con contatti meccanici, come quello di questa immagine, probabilmente funzioneranno male, perché gli ingressi dello IotModule non sono di tipo Schmitt Trigger. Per farli funzionare senza errori, si dovrebbe aggiungere il modulo Schmitt Trigger che si trova sul sito Theremino.

Impostare gli encoder del modulo come "Software" o "Hardware"

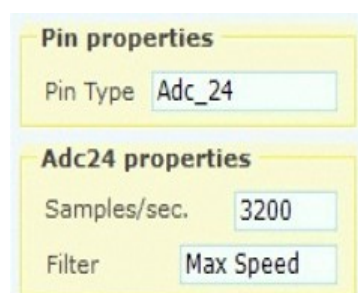
Modificando "IOTMODULE_ENCODERS_TYPE" nel file "IotModuleSetup.h" (e riprogrammando il modulo), si può modificare il funzionamento degli encoder. Con gli encoder hardware non ci sono limiti di frequenza. Invece con gli encoder software la massima velocità di conteggio è limitata a qualche decina di KHz, ed è dipendente dal carico sul micro-controllore. Se necessario si può limitare la frequenza usando Encoders con meno passi per giro.

In compenso gli encoders software hanno una maggiore immunità ai disturbi e danno un numero doppio di conteggi (x4), rispetto a quelli hardware (x2), quindi si ottiene una maggiore risoluzione.

I tipi di Pin in "Input" <-- Adc24

Per attivare l'Adc24 il firmware deve essere programmato con IOTMODULE_ADC24_PINS da 1 a 16, poi si deve **selezionare la linea del Pin 60 e impostare il suo Pin Type come "Adc24"**.

Quando l'Adc24 funziona, il valore del Pin 60 (Adc_24) cresce proporzionalmente alla velocità di campionamento (con 100 sps e filtro MaxSpeed, il valore cresce di 100 valori al secondo).



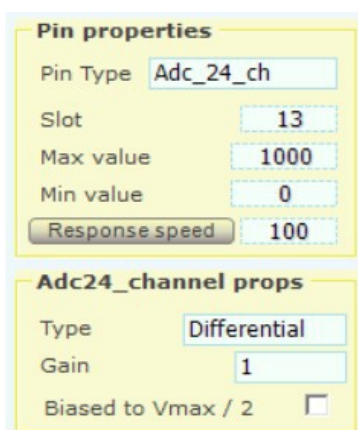
Pin properties
Pin Type: Adc_24

Adc24 properties
Samples/sec.: 3200
Filter: Max Speed

Proprietà del Pin 60 configurato come Adc24

Samples per sec: Velocità di campionamento che si suddivide tra tutti gli ingressi attivi (per esempio con 600 sps e tre ingressi, ogni ingresso viene campionato 200 volte al secondo).

Filter: Sono disponibili otto filtri da scegliere per il migliore compromesso tra rumore e velocità di risposta.



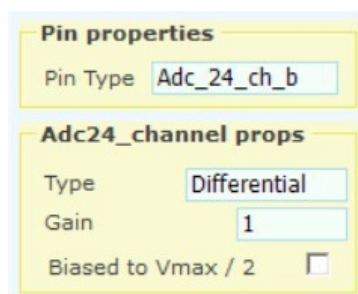
Pin properties
Pin Type: Adc_24_ch
Slot: 13
Max value: 1000
Min value: 0
Response speed: 100

Adc24_channel props
Type: Differential
Gain: 1
Biased to Vmax / 2: ☐

Proprietà dei Pin "Adc_24_ch"

Importante notare che i 16 Pin di ingresso sono a coppie (1-2, 3-4 ... 15-16) e quindi il loro tipo (Differential, Pseudo e Single) e il guadagno (da 1 a 128), valgono per ambedue gli ingressi della coppia.

Il "Biased to Vmax/2" può invece essere attivato separatamente, su ognuno dei sedici ingressi.



Pin properties
Pin Type: Adc_24_ch_b

Adc24_channel props
Type: Differential
Gain: 1
Biased to Vmax / 2: ☐

Proprietà dei Pin "Adc_24_ch_b"

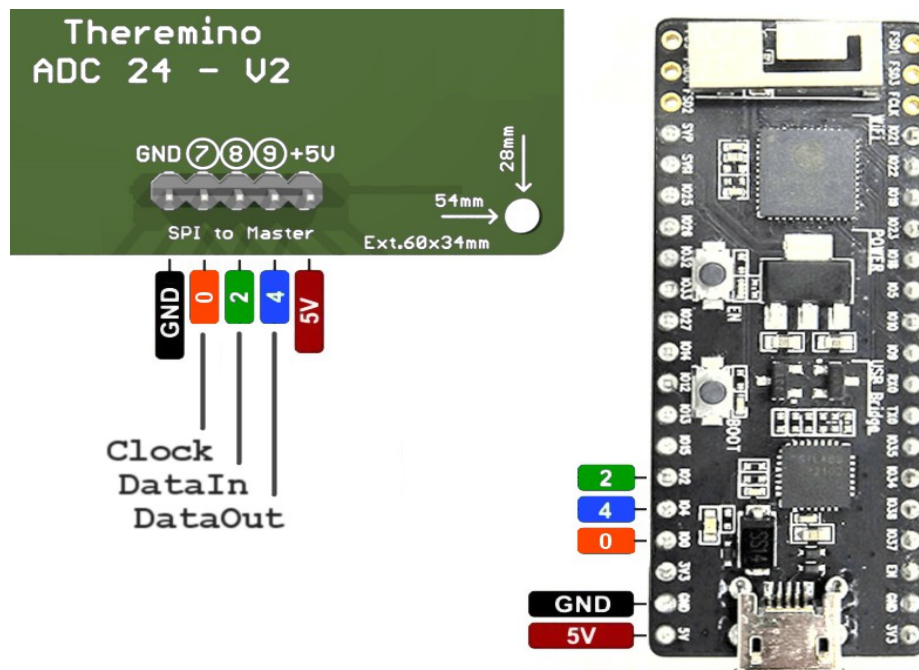
Se la coppia di ingressi è impostata come Differential, il secondo ingresso della coppia diventa "Adc_24_ch_b".

Questo è un tipo speciale che non invia dati agli Slot, ed è solo un segnapposto, per la connessione di riferimento del sensore.

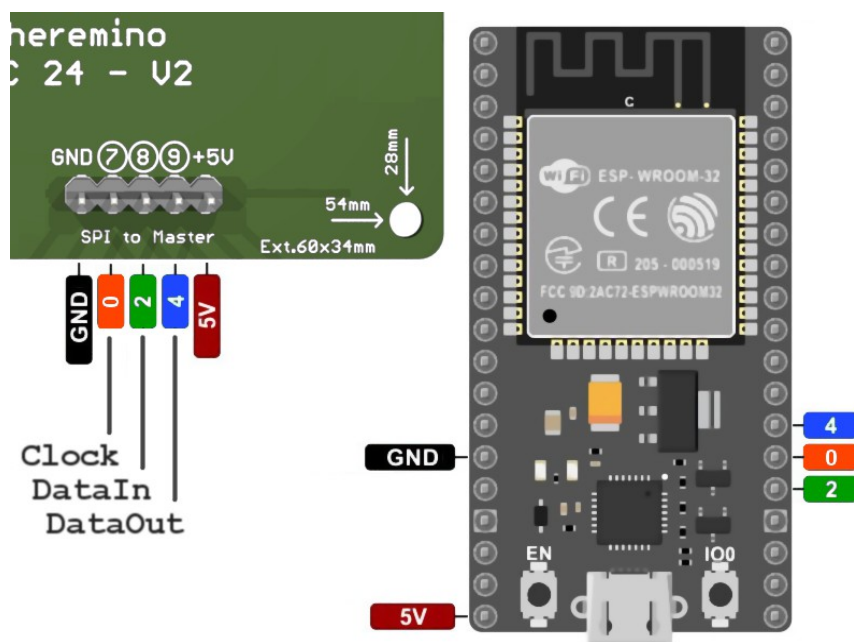
Informazioni sul modulo Adc24 e manuale di utilizzo con spiegazioni dettagliate ed esempi:
<http://www.theremino.com/hardware/adapters#adc24>

Il modulo Adc24 è collegato a cinque Pin dello IotModule come illustrato nella prossima pagina. Fili adiacenti percorsi da forti segnali potrebbero disturbare questo collegamento. Se i fili sono lunghi è bene utilizzare cavo schermato e diminuire la frequenza di comunicazione come spiegato in [questa pagina](#).

Connessioni del modulo Adc24



Connessioni tra l'Adc24 e il PICO KIT V4



Connessioni tra l'Adc24 e lo ESP WROOM 32

I tipi di Pin “Generici”

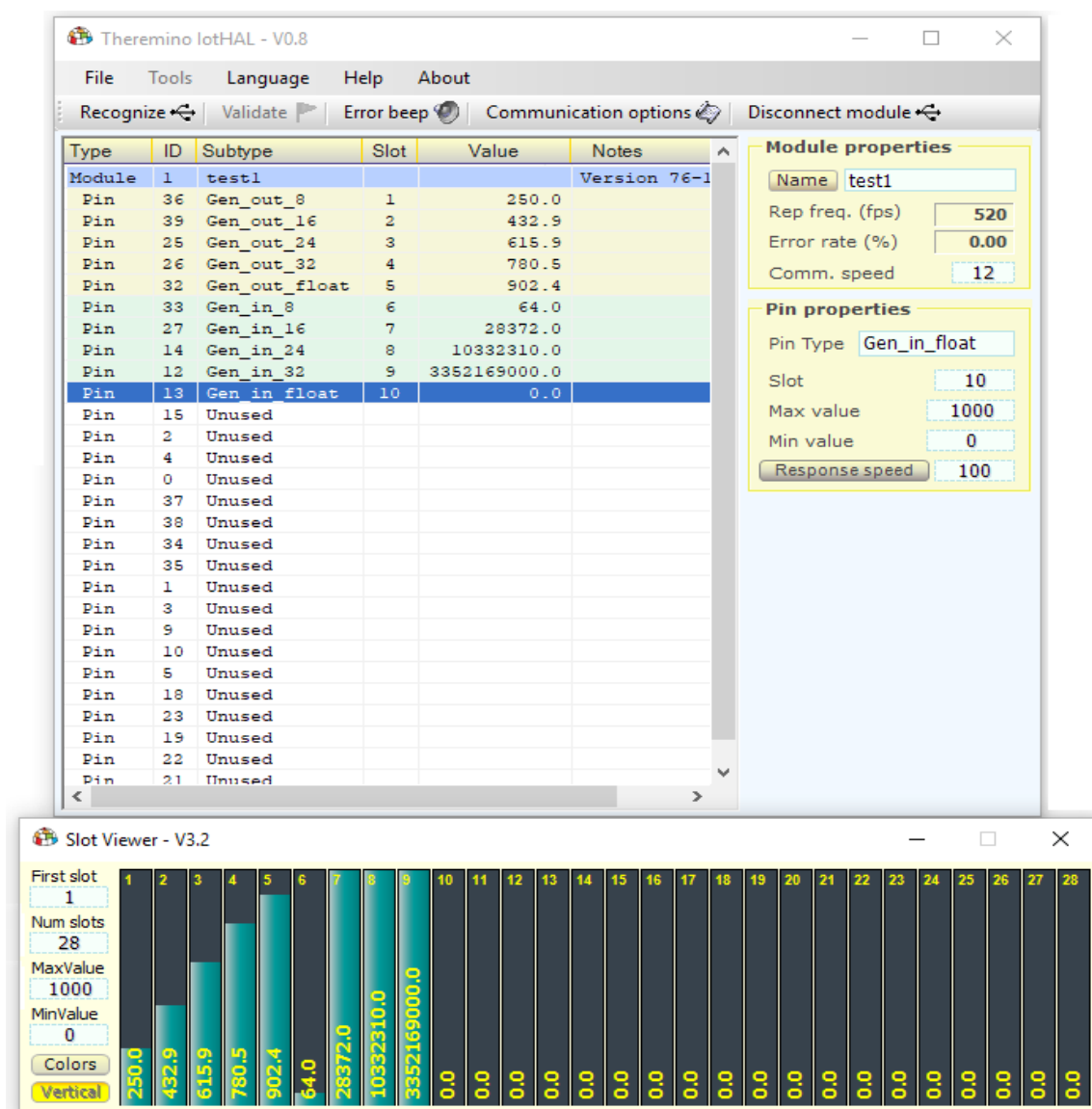
Per leggere e scrivere i normali Pin di Ingresso-Uscita dello **IoTModule** non è necessario utilizzare le funzioni spiegate in questa pagina, basta configurare i Pin con lo **IoTModule** e tutto funziona immediatamente.

Invece per i Pin generici si deve anche scrivere il firmware. Quindi si deve saper pianificare e organizzare senza errori, e ci vuole una buona esperienza nella programmazione.

Quello che spiegheremo in questa pagina vale solo per leggere o scrivere sensori o attuatori speciali, ad esempio sensori o motori che comunicano in I2C. Oppure per pre-condizionare i dati con un proprio firmware prima di inviarli al PC, o dal PC verso gli attuatori.

I Pin generici dello **IoTModule** (o Pin Virtuali), sono stati introdotti per permettere lo scambio dati bidirezionale di variabili numeriche tra i dati interni al firmware dello **IoTModule** e gli slot gestiti dallo **IoTModule**. Sono variabili aggiuntive, slegate dalle funzioni dei Pin fisici e dal loro valore.

Nelle funzioni **GenericWrite** e **GenericRead** si indica un Pin e quello stesso Pin deve essere configurato nel **IoTModule** con il tipo **Gen_in** o **Gen_out** corrispondente. Se si fanno errori si avranno risultati imprevedibili.



Programmare i Pin “Generici”

Il numero di Pin generici necessari dipende dalle caratteristiche della periferica hardware, e dal numero dei valori che si vogliono trasferire. Qualsiasi Pin fisico del modulo può essere usato come Pin generico. Se tutti i Pin fisici sono usati oppure si deve usare un numero elevato di Pin generici se ne possono aggiungere un certo numero “virtuali” ricompilando il firmware. Questo numero è impostato nella costante `IOTMODULE_GENERIC_PINS` definita nel file `lotModuleSetup.h` e può variare da 0 fino ad un massimo di 20, indipendentemente dal tipo di Pin generico utilizzato (float, 24, 16 o 8 bit). I nomi dei Pin generici “virtuali” partono dal Pin 80 e terminano al Pin 99.

Qualsiasi modifica alla costante `IOTMODULE_GENERIC_PINS` deve essere seguita dalla ricompilazione del firmware nell'IDE Arduino, seguita dalla programmazione dello `lotModule` e dalla validazione della nuova configurazione con il pulsante "Valida" sulla applicazione `lotHAL`.

Inviare dati dai sensori hardware verso le applicazioni sul PC

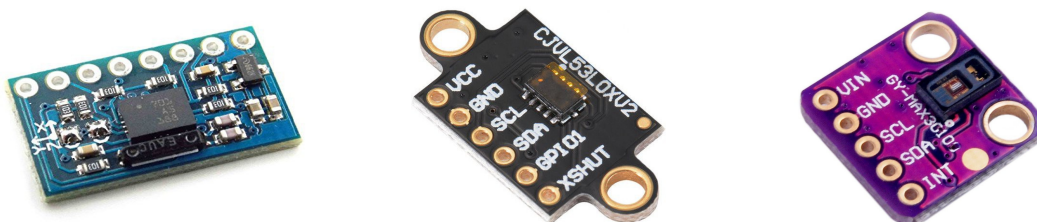
- Si sceglie un Pin da utilizzare
- Nel Loop che si trova nel file ".ino", si legge il sensore e si interpretano i suoi dati
- Nel Loop che si trova nel file ".ino", si invia il dato con “GenericWrite” (8, 16, 24 o Float)
- Sullo `lotHal` si legge il dato configurando il Pin come `Gen_in` (8, 16, 24 o Float)
- Sullo `lotHal` si imposta il Pin scelto con uno Slot attraverso il quale il dato del sensore andrà alle applicazioni che lo utilizzano.

Inviare dati dalle applicazioni sul PC agli attuatori hardware

- Sullo `lotHAL` si sceglie un Pin da utilizzare
- Sullo `lotHal` si imposta il Pin scelto con uno Slot attraverso il quale inviare i dati numerici
- Sullo `lotHal` si inviano i dati al modulo configurando il Pin come `Gen_out` (8, 16, 24 o Float)
- Nel Loop che si trova nel file ".ino", si legge il dato con `GenericRead` (8, 16, 24 o Float)
- Nel Loop che si trova nel file ".ino", si interpreta il dato numerico e lo si invia all'hardware.

Leggere sensori speciali con i Pin generici

Il documento “Leggere un sensore I2C”, che si scarica da [questa pagina](#), è un buon esempio di utilizzo dei "Pin Generici".



Esempi di utilizzo dei Pin “Generici”

Nelle prossime righe si può vedere un esempio di codice da inserire nel Loop Arduino per il trasferimento dei quattro tipi fondamentali di dato dei Pin generici.

Nelle prime due righe il valore a 8 bit dello slot associato al Pin Generico 32 (slot e Pin di input nel IotHAL, funzione GenericRead nel IotModule) viene trasferito indietro sullo slot associato al Pin generico 33 (funzione GenericWrite nel IotModule, Pin e slot di output nel IotHAL).

Le altre due coppie fanno lo stesso manipolando valori a 16 e a 24 bit e utilizzando i Pin 27, 14, 80 e 81. L'ultima coppia fa lo stesso con numeri “Float” e utilizzando i Pin 82 e 83.

```
uint32_t n;  
float f;  
  
n = IotModule.genericRead8(32); // IotHAL Pin 32 configured as gen_out_8  
IotModule.genericWrite8(33, n); // IotHAL Pin 33 configured as gen_in_8  
  
n = IotModule.genericRead16(27); // IotHAL Pin 27 configured as gen_out_16  
IotModule.genericWrite16(14, n); // IotHAL Pin 14 configured as gen_in_16  
  
n = IotModule.genericRead24(80); // IotHAL Pin 80 configured as gen_out_24  
IotModule.genericWrite24(81, n); // IotHAL Pin 81 configured as gen_out_24  
  
f = IotModule.genericReadFloat(82); // IotHAL Pin 82 configured as gen_out_float  
IotModule.genericWriteFloat(83, f); // IotHAL Pin 83 configured as gen_in_float
```

Una volta scritte queste righe nella funzione Loop() di Arduino, le si possono provare con lo IotHal, modificando il valore della colonna Value con il mouse, o aiutandosi con lo SlotViewer.

Trasferire numeri interi dallo IotModule agli Slot sul PC e viceversa

Spesso si usano interi che stanno in 16 bit (da 0 a 65535). Ma in alcuni casi potrebbero essere valori piccoli e quindi stare in 8 bit (da 0 a 255), oppure valori molto grandi e quindi aver bisogno di 24 bit (da 0 a 16777215).

Lo standard del sistema Theremino è riportare tutti i valori ad una escursione “normalizzata” da 0 a 1000. Ma in alcuni casi, ad esempio un sensore che fornisce un valore intero, può essere utile trasferire il valore prodotto senza modificarlo. Per ottenere questo comportamento si modifica la casella “Max value” con un valore adeguato al numero di bit impostati (255 per gli 8 bit / 65535 per i 16 bit / 16777215 per i 24 bit).

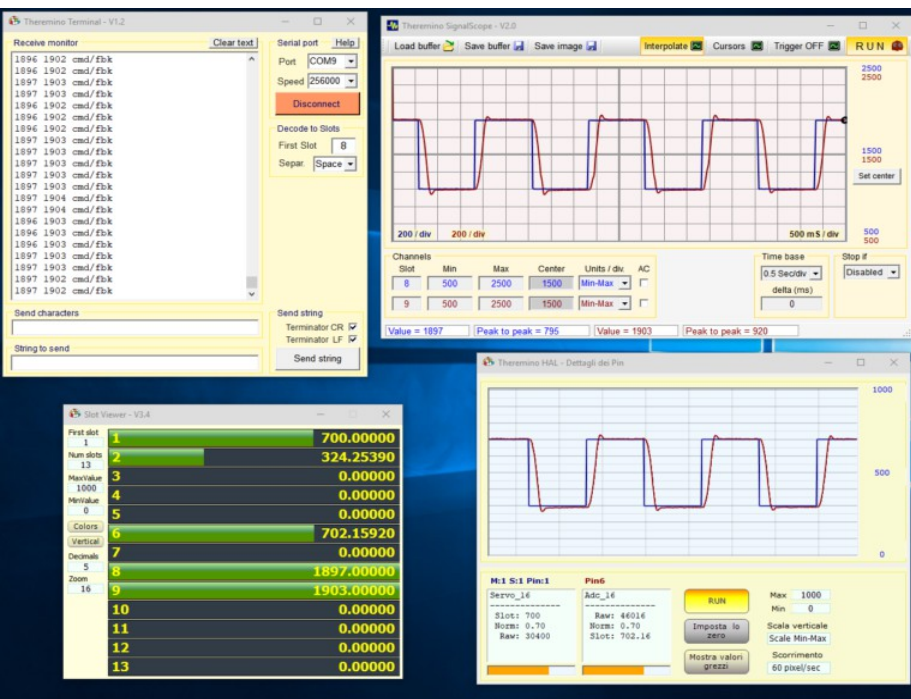
Trasferire numeri Float dallo IotModule agli Slot sul PC e viceversa

I numeri Float, chiamati "Float" o "Double" nell'IDE di Arduino, oppure "Single" nelle applicazioni sul PC, possono essere enormi o piccolissimi e hanno sempre la stessa precisione. Questi numeri vengono trasferiti così come sono, se si impostano “Min value = 0” e “Max value = 1000”. Altrimenti si possono utilizzare “Min value” e “Max value” per scalarli a piacere.

Fare il debug con i Pin "Generici"

Quando si scrive il firmware del modulo (sia quello che gli utenti possono scrivere nel file xxxx.ino, sia quello scritto da noi nella cartella "IotModule"), una comune esigenza è di conoscere i valori delle variabili interne.

Storicamente per questo compito si utilizzavano i debugger hardware, che erano costosi e anche difficili da collegare. Un secondo metodo (che usa anche Arduino) è quello di inviare i valori delle variabili attraverso il collegamento seriale.



Nel firmware si inviano alla seriale i valori delle variabili che si vogliono seguire e la [applicazione Terminal](#) interpreta i valori e li invia agli Slot. In pratica si riescono a vedere le variazioni delle variabili interne del firmware, come se si collegasse ad esse un oscilloscopio.

Si può quindi utilizzare il [Signal Scope](#) (o altre applicazioni del sistema theremino), per vedere la variazione nel tempo dei loro valori.

Sembra di collegare un oscilloscopio dentro al firmware.

Per chi utilizza i micro, e soprattutto per chi scrive firmware di controllo dei servomeccanismi, questa è una possibilità eccezionalmente utile!

Per mezzo della applicazione IotHAL si possono utilizzare i Pin Generic e ottenere le stesse prestazioni anche via WiFi (senza collegare il cavo USB per la seriale).

I valori delle variabili interne al firmware vengono trasferiti direttamente negli slot desiderati utilizzando una delle istruzioni Generic, ad esempio la seguente:

```
IotModule.genericWriteFloat(Numero dello slot, Valore da scrivere);
```

Tasti resistivi o capacitivi

Per vedere cosa è possibile fare con semplici tasti guardate questo bellissimo video del Makey Makey: <http://vimeo.com/60307041#>

Però i tasti del Makey Makey non sono capacitivi, ma resistivi, funzionano solo se la resistenza è inferiore a circa 4 Mega Ohm, hanno bisogno di un filo aggiuntivo come riferimento di terra e non funzionano attraverso materiali isolanti come la plastica. Inoltre i tasti del Makey Makey sono solo sei (non espandibili), ogni Makey Makey potrebbe fornire anche 20 tasti e si potrebbero collegare anche più Makey Makey, ma alla fine i tasti vanno sulla tastiera che ne sopporta al massimo sei: www.makeymakey.com/faq Infine i Makey Makey hanno un funzionamento solo Acceso/Spento, senza regolazioni intermedie e non sentono la velocità di pressione dei tasti (Velocity).



I tasti capacitivi del sistema Theremino possono fare molto di più, sono espandibili a piacere aggiungendo moduli Master (6 tasti per modulo) e lotModule (9 tasti per modulo) in numero illimitato, come visibile qui: www.youtube.com/watch?v=NbC5klRS_6s e anche qui: www.youtube.com/watch?v=2RzwUfXhFZY

Inoltre i tasti del sistema Theremino possono anche fornire un controllo graduale, come con un cursore di tipo slider, e anche il controllo di "espressione", determinato dalla velocità di pressione dei tasti.

I tre tipi di tasti capacitivi

Touch properties	
Min variation	20
Proportional area	0

Touch properties	
Min variation	20
Proportional area	150

Touch properties	
Min variation	40
Proportional area	-30

- **Tasti on/off**
"Min variation" da 10 a 50
"Proportional area" deve essere a ZERO
- **Tasti proporzionali**
"Min variation" da 10 a 100
"Proportional area" da 100 a 200 (per massimo circa a 1000)
- **Tasti con velocity**
"Min variation" da 25 a 50 (regolare per massima uscita)
"Proportional area" -30 (regolare per massimo circa a 1000)

Misura generica della capacità

- **Sensori capacitivi (sensori di umidità, cond. variabili, ecc.)**
"Min variation" da -1 a -1000 (taratura del valore minimo)
"Proportional area" da 1 a 1000 (taratura del valore massimo)

Attenzione: con questo tipo di Pin non si ottiene una misura della capacità elettrica, ma solo un valore proporzionale a un sensore o a una posizione. Molti fattori contribuiscono a rendere non lineare la misura, primo tra tutti la capacità del cavo di collegamento. Il cavo deve essere molto corto e dopo aver effettuato la taratura non si deve più spostarlo. In tutti i casi si dovranno fare le regolazioni di scala e le linearizzazioni adeguate nel software.

I parametri Min Variation e Proportional Area

Min variation

Elimina le variazioni piccole e evita che i disturbi elettrici possano far scattare i tasti anche senza toccarli. Alzando questo parametro, i tasti diventano meno sensibili per cui è bene tenerlo più basso possibile, compatibilmente con l'esigenza di eliminare tutti i disturbi.

Per i tasti con velocity la migliore regolazione di questo parametro si ottiene premendo velocemente e ripetutamente il tasto e regolando "Min variation" con la rotella del mouse in modo da ottenere il massimo segnale in uscita. Per facilitare questa regolazione impostare temporaneamente "Proportional area" con un numero negativo abbastanza grande, ad esempio -50.

Proportional Area

Si regola per avere circa mille quando il dito è nella posizione di massimo sullo slider oppure quando si premono i tasti alla massima velocità possibile.

Normalmente questo valore deve essere più alto per i Pin 1 e 2 (meno sensibili) e in caso di fili lunghi e oggetti grandi.

Taratura dello zero dei tasti capacitivi

Se si cambia la disposizione meccanica dei tasti, o la loro posizione, o si spostano i fili che li collegano o si avvicinano oggetti metallici con il programma HAL in funzione, può accadere di perdere la taratura dello zero dei tasti. Se lo zero non è ben tarato i tasti capacitivi possono diventare meno sensibili o anche non funzionare del tutto.

Se si toglie capacità dai tasti (si accorciano i fili o li si allontana da oggetti metallici) la taratura viene rifatta immediatamente in modo automatico ma non è possibile distinguere un aumento di capacità dovuto a un dito o a uno spostamento dei fili.

Abbiamo provato molti metodi di riconoscimento automatico, metodi con lenta deriva e metodi con taratura temporizzata, ma nessuno ha funzionato bene e tutti peggiorano la precisione del normale funzionamento dei tasti.

Per cui si consiglia di non spostare i fili dei tasti, i tasti stessi e gli oggetti conduttivi, nel raggio di una decina di centimetri, durante il funzionamento.

Per verificare se un tasto è tarato si allontanano le mani dal tasto e si verifica nei dettagli del suo Pin che i valori "Smoot" e "Mean" siano uguali tra loro o molto vicini (non più di un punto di differenza)

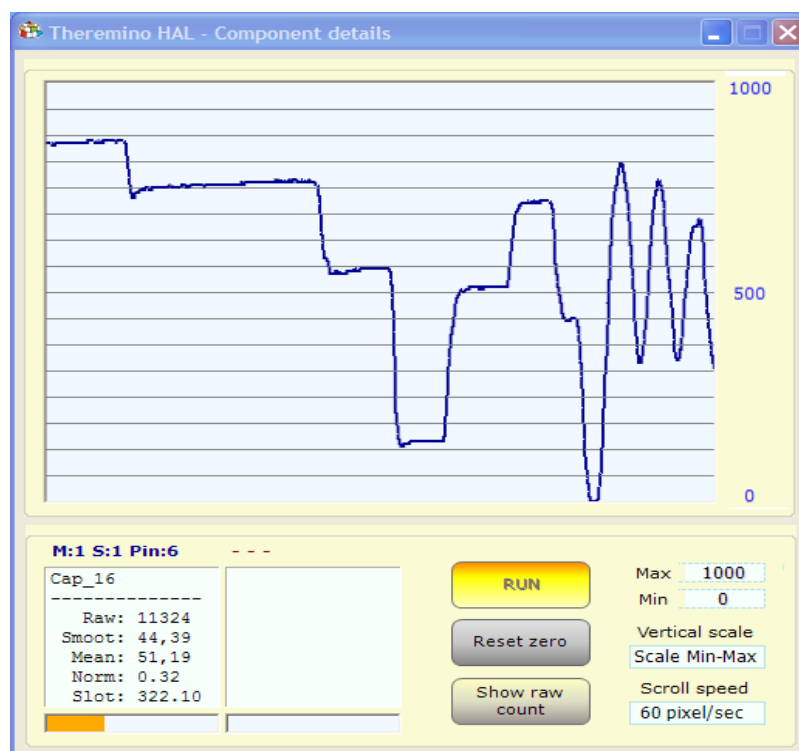
Nel dubbio premere Calibrate (tenere le mani lontane dai tasti mentre si esegue la taratura dello zero)

Lettura dei sensori capacitivi

Impostando Min Variation con un valore negativo, si cambia completamente il modo di funzionamento e si possono collegare sensori di umidità (del tipo a variazione di capacità e senza circuito di controllo). Si potrebbero anche improvvisare sensori per leggere la rotazione di un perno o uno spostamento lineare. Sensori di questo tipo potrebbero essere semplici ma anche molto affidabili.

Impostando Min Variation con un valore negativo, il significato di Min Variation e Proportional Area cambia:

- ◆ Min Variation imposta il minimo e Proportional Area il massimo, del campo di capacità misurabili.
- ◆ Il tasto Calibration viene disabilitato. La calibrazione è fissa ed è il valore stesso di Min Variation.
- ◆ I valori di capacità usabili vanno da qualche picoFarad a qualche nanoFarad.



I Tasti capacitivi di tipo "Slider"



"Proportional area" deve essere un numero positivo, questo determina il funzionamento di tipo "Proporzionale"

Con un tasto capacitivo di questa forma si ottiene una regolazione continua simile a un cursore "slider"

Il controllo si effettua con un dito, tutto in alto = 1000, tutto in basso = 0

Questi tasti sono adatti per il controllo di volume e agiscono anche da "Panic button" (quando si allontana il dito dal tasto il volume si azzerava)

Queste sono le regolazioni normali per i tasti di tipo "Slider" (nota 1)

Pin properties	
Pin type	Cap_16
Slot	2
Max value	1000
Min value	0
Response speed	30

Touch properties	
Min variation	20
Proportional area	150

"Max value" si tiene normalmente a 1000 (Nota 2)

"Min value" si tiene normalmente a zero (Nota 2)

"Response speed" si imposta normalmente a 30 (leggero filtraggio)

"Min variation" si imposta normalmente da 10 a 100 (meglio alzarlo poco per avere la massima sensibilità nella parte bassa)

"Proportional area" si imposta normalmente a 200 (circa 100 per tasti meno sensibili o con fili lunghi)

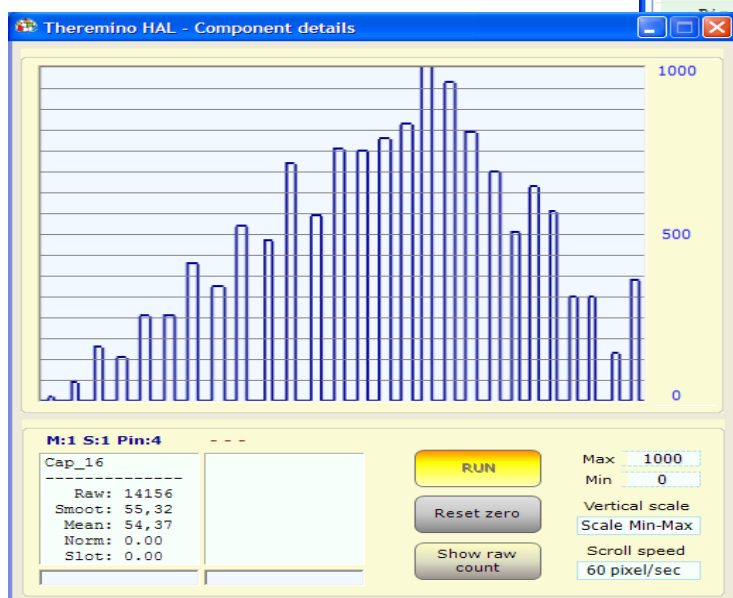
(Nota 1) Per i tasti di tipo "Slider" è sempre meglio usare "Cap_16"

(Nota 2) Per invertire il segnale di uscita si possono scambiare Max con Min (Max = 0 e Min = 1000)

I Tasti capacitivi con "Velocity"

"Proportional area" deve essere un numero negativo.

Questo determina il funzionamento di tipo "Velocity"



Theremino HAL - V3.0						
File Tools Help About						
Edit configurations Recognize Validate Beep on errors Calibrate 20						
Type	ID	Subtype	Dir.	Slot	Value	Notes
Master	1	Theremin				
Slave	1	MasterPins				
Pin	1	Unused				
Pin	2	Cap_16	get	2	0.0	
Pin	3	Cap_16	get	3	0.0	
Pin	4	Cap_16	get	4	0.0	
Pin	5	Cap_16	get	5	0.0	
Pin	6	Cap_16	get	6	0.0	
	2	CapSensor				
	1	Cap_sensor	get	0	12.8	
	2	Theremin2				
	1	MasterPins				
	1	Dig_out	set	1	Sleep	
	2	Unused				
	3	Cap_16	get	89	0.0	
	4	Cap_16	get	57	0.0	
	5	Cap_16	get	55	0.0	
	6	Adc_8	get	66	1000.0	

Le tastiere che permettono di suonare piano o forte, a seconda di come si premono i tasti, sono molto apprezzate per le applicazioni musicali.

I tasti capacitivi possono essere regolati per misurare la velocità di pressione del tasto e trasformarla in un valore da 0 a 1000 (circa). Per un buon funzionamento della "Velocity" la velocità di comunicazione deve essere alta (da 200 a 500 fps) e si devono regolare i tasti, uno per uno, in modo da ottenere un valore massimo di poco superiore a 1000

Pin properties	
Pin type	Cap_16
Slot	2
Max value	1000
Min value	0
Response speed	30
Touch properties	
Min variation	40
Proportional area	-30

Queste sono le regolazioni per i tasti con "Velocity"

"Max value" si tiene normalmente a 1000 (Nota 1)

"Min value" si tiene normalmente a "0" (Nota 1)

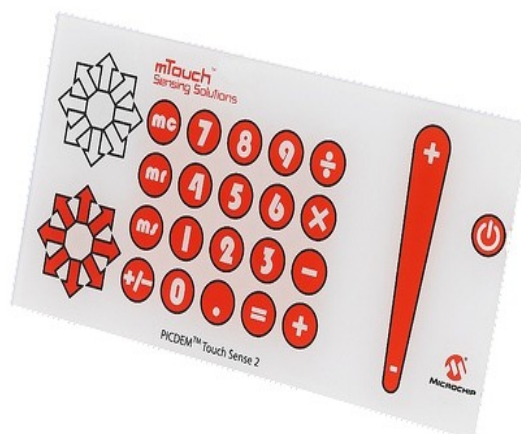
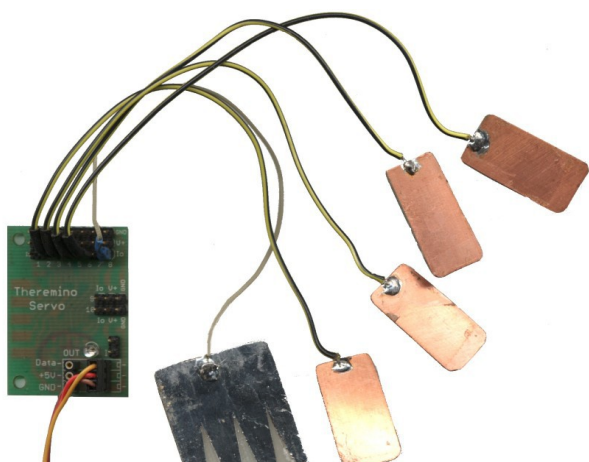
"Response speed" si tiene normalmente a 30 (meglio non modificarlo)

"Min variation" si imposta normalmente a 50 (e circa a 25 per i tasti 1 e 2 che sono meno sensibili o per tasti con fili lunghi)

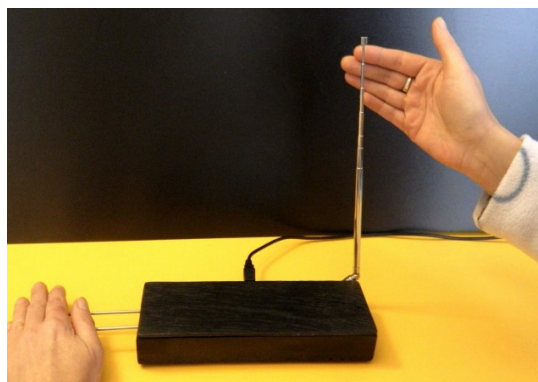
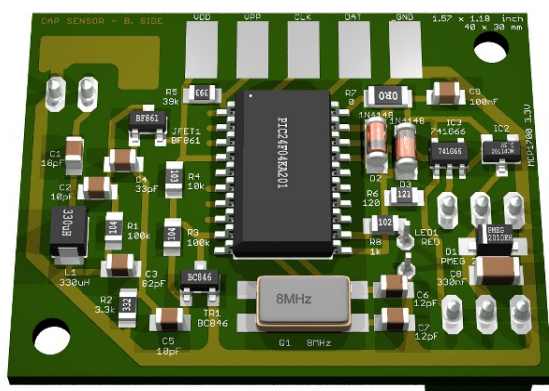
"Proportional area" si imposta normalmente a -40 (o circa -20 per i Pin 1 e 2 che sono meno sensibili, o per tasti con fili lunghi)

(Nota 1) Per invertire il segnale di uscita si scambiano Max con Min (Max = 0 e Min = 1000)

Differenze tra i tasti capacitivi e i moduli CapSensor



I tasti capacitivi non sostituiscono i moduli CapSensor, i primi funzionano solo a brevi distanze (da pochi millimetri fino a pochi centimetri) mentre i CapSensor funzionano fino a distanze di molti metri e possono essere regolati per una risposta quasi perfettamente lineare. I tasti capacitivi in compenso costano molto meno e sono più adatti per comporre tastiere con molti tasti.



Costruzione meccanica dei tasti capacitivi

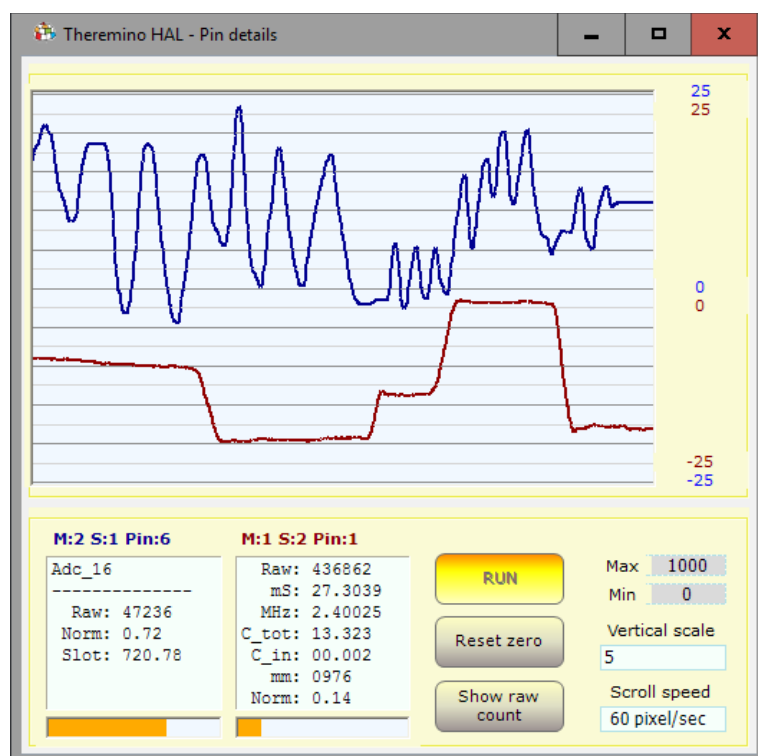


Fare attenzione che i tasti capacitivi devono essere isolati, altrimenti può bastare una piccola scintilla di elettricità statica per produrre errori di comunicazione, non si rompe nulla ma la comunicazione si blocca e si deve premere il pulsante "Recognize"

Quindi normalmente li si gira con il rame sotto e la vetronite (sottile) sopra o, meglio ancora, si tiene il rame sopra e si aggiunge un sottile foglio di plastica isolante stampato a colori con la forma dei tasti, come nella immagine all'inizio di questa pagina.

I fili che vanno dai tasti ai Pin devono essere più corti possibile e stare almeno a 5 o 10 millimetri tra di loro. Il funzionamento dei tasti e l'isolamento dai disturbi, migliorano diminuendo la capacità elettrica. Sono state fatte prove anche in situazioni "impossibili", con fili lunghissimi e ogni genere di tasto, come piante di fiori in vaso e frutti vari e una regolazione individuale ha sempre fatto funzionare tutti i tasti.

Il visualizzatore dei dettagli dei Pin



Con doppio click sulla linea di un Pin attivo, si apre questo strumento. Per due segnali, cliccare prima su un Pin e poi sul secondo, con un click singolo.

La scala verticale può essere impostata a "Scale Min-Max", che corrisponde al valore delle caselle Min e Max.

Oppure si imposta in 24 livelli da 0.01 a 50000 punti per divisione verticale (dieci linee scure). Se si utilizzano queste impostazioni, per centrare le tracce si preme "Reset zero".

In alcuni casi può essere utile visualizzare i valori grezzi ("Raw" in inglese) Per visualizzare i valori "Raw" si abilita il tasto "Show raw count".

Il controllo "Scorrimento" regola la velocità di scorrimento del grafico da 0.1 Pixel al secondo fino a 60 Pixel al secondo.

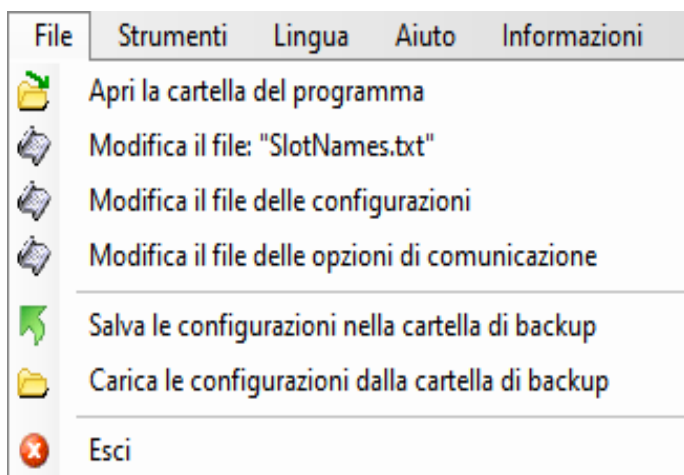
I due riquadri di testo mostrano i dettagli interni dei Pin, il titolo indica di quale Pin si tratta, in questa immagine il testo "M:1 Pin:2" significa "Modulo 1, Pin 2" ("S" non è usato nello IotHAL)

I dettagli dei Pin possono aiutare nel controllo e la regolazione dei dispositivi di Ingresso Uscita (Sensori e Attuatori).

Alcuni tipi di Pin sono più complessi e presentano più valori intermedi. In genere esiste un valore "Raw" con valori molto variabili a seconda del tipo di Pin, un valore "Normalizzato" che va sempre da 0 a 1 e un valore "Slot" che normalmente va da 0 a 1000 e che è il valore "semplificato" disponibile sugli Slot e usabile facilmente da tutti i software ad alto livello.

- ◆ **Raw** Valore "grezzo" che può essere un conteggio, un tempo, una tensione o altro.
- ◆ **mS** Tempo in millisecondi
- ◆ **uSec** Tempo in microsecondi
- ◆ **Smoot** Valore passato attraverso un filtro FIR di smoothing (usato solo nei Cap8 e Cap16)
- ◆ **Mean** Valore medio (usato nei tipo Cap8 e Cap16 come calibrazione dello zero)
- ◆ **Norm** Valore normalizzato tra zero e uno
- ◆ **Slot** Valore scritto o letto dallo Slot associato al Pin (normalmente da 1 a 1000)
- ◆ **Out** Valore digitalizzato che può valere solo "0" o "1" (usato solo da DigOut)

I comandi dei menu



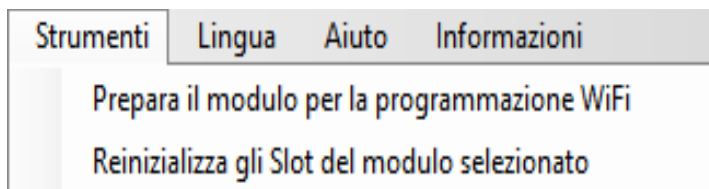
Aprire la cartella di lavoro può essere utile per modificare i file di documentazione e delle lingue.

Modificare il file: “SlotNames” i commenti (o nomi degli slot) sono spiegati a pagina 8.

Modificare il file delle configurazioni può essere comodo in certi casi. Per maggiori informazioni leggere anche "Domande e risposte" nelle ultime pagine di questo documento.

Modificare le opzioni di comunicazione Viene aperto il file delle opzioni di comunicazione per modificare gli indirizzi IP validi, la porta di comunicazione e i nomi degli IoTModule validi. Le istruzioni e gli esempi di opzioni, sono nello stesso file delle opzioni: “CommOptions.txt”.

Salvare le configurazioni come backup permette di fare delle copie di sicurezza delle configurazioni. Se i file di configurazione vengono modificati per errore sarà poi possibile ricaricarli da una versione precedente. Le versioni precedenti mostrano la data e l'ora in cui sono state salvate.



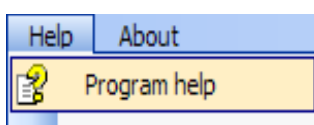
Prepara per la programmazione WiFi vedere i particolari nelle ultime pagine di questo documento.

Re-inizializza gli Slot rinumera gli Slot del modulo selezionato.



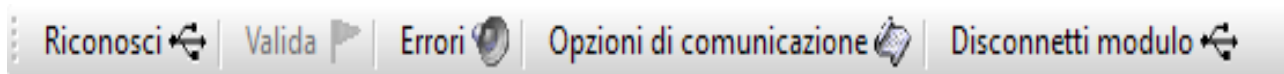
I file delle lingue si trovano nella cartella “Docs” vicino alla applicazione ThereminoHAL.exe.

Per fare nuovi file delle lingue basta copiare il file Language_ENG.txt, cambiare “ENG” con “FRA”, “ESP”, “DEU” o “JPN” e modificare il testo con BloccoNote.



Questo comando apre la documentazione (questo stesso file).

I comandi della Toolbar



Riconosci

Serve per riconoscere i Moduli attivi.

Valida

Quando si modificano i moduli, si viene avvertiti che la configurazione è cambiata con righe rosse nella lista. Se si sceglie di perdere la vecchia configurazione e adeguarsi all'hardware attuale, con questo pulsante si rende valida la nuova configurazione.

Errori

Se premuto gli errori di comunicazione vengono evidenziati con un suono.

Opzioni di comunicazione

Viene aperto il file delle opzioni di comunicazione per modificare gli indirizzi IP validi, la porta di comunicazione e i nomi degli IotModule validi. Le istruzioni e gli esempi di uso, sono nello stesso file delle opzioni di comunicazione, che si chiama "CommOptions.txt".

Disconnetti Modulo

Elimina il Modulo selezionato dalla lista. In questo modo si possono eliminare i Moduli indesiderati senza doverli spegnere fisicamente.

Applicazioni isolate

Alcune applicazioni del sistema Theremino lanciano automaticamente un proprio HAL. Questo accade se esiste un file Net_HAL.exe, nella cartella lotHAL, situata accanto al file EXE della applicazione. Si potrebbe anche collocare il solo file Net_HAL.exe accanto al file exe della applicazione, ma è meglio che l'HAL abbia una sua cartella, con la sotto-cartella Docs contenente i file di documentazione e delle lingue.

Questi HAL usano una propria configurazione privata e si connettono solo ai Moduli i cui nomi sono nel file CommOptions. Una applicazione composta in questo modo, continuerà a funzionare anche se copiata su un diverso computer e anche se altre applicazioni del sistema Theremino stanno comunicando con i propri Moduli, su altre porte USB.

Le applicazioni che più si avvantaggiano da queste possibilità, sono le applicazioni con un compito preciso, come ad esempio: Theremino Geiger, Theremino OilMeter, Theremino Meteo, Theremino Theremin, Theremino Arm, Theremino Geo e Theremino EmotionMeter.

Questo non vuol dire che le applicazioni isolate non possano comunicare con le altre. La comunicazione modulare è sempre possibile e avviene attraverso gli Slot, che sono in comune per tutte le applicazioni.

Per evitare di usare gli stessi Slot per compiti diversi abbiamo definito uno schema di massima.

Experimental 100 slots	000 - 099
- - -	
Theremino_Theremin	100 - 199
Theremino_SlotsToMidi	200 - 299
Theremino_MusicKeys	300 - 329
- - -	
469 free slots	330 - 799
- - -	
Theremino_OilMeter	800 - 809
Theremino_EEG	810 - 819
Theremino_Meteo	820 - 839
Theremino_Arm	840 - 849
10 free slots	850 - 859
10 free slots	860 - 869
10 free slots	870 - 879
Theremino_EmotionMeter	880 - 889
Theremino_Geiger	900 - 909
Theremino_Bridge	900 - 909
Theremino_GEO	910 - 919
Theremino_GeoPreamplifier	920 - 929
Theremino_Radar	930 - 939
10 free slots	940 - 949
10 free slots	950 - 959
10 free slots	960 - 969
10 free slots	970 - 979
10 free slots	980 - 989
10 free slots	990 - 999

Questo schema è solo indicativo. Si possono usare gli Slot a piacere, basta che nello stesso PC non si usi contemporaneamente lo stesso Slot, per due compiti diversi. Se si sbaglia non si rompe nulla, ma i dati si sovrappongono con risultati indefiniti.

Regolazione delle caselle numeriche

Draw speed (fps) 5

Le caselle numeriche dell'HAL (e di tutte le altre applicazioni del sistema Theremino) sono state sviluppate da noi (Nota1) per essere più comode e flessibili delle TextBox originali di Microsoft.

I valori numerici sono regolabili in molti modi

- Cliccando, e tenendo premuto, il bottone sinistro del mouse e muovendo il mouse su e giù
- Con la rotella del mouse
- Con i tasti freccia-su e freccia-giù della tastiera
- Con i normali metodi che si usano per scrivere numeri con la tastiera
- Con i normali metodi di selezione e di copia-incolla
- Premendo SHIFT la velocità di variazione viene moltiplicata per cento
- Premendo CTRL la velocità di variazione viene moltiplicata per dieci
- Premendo ALT la velocità di variazione viene divisa per dieci

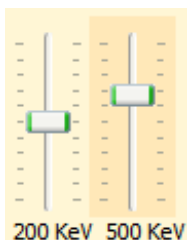
Muovere il mouse su e giù permette ampie e veloci regolazioni

La rotella del mouse permette una regolazione comoda e immediata

I tasti freccia permettono regolazioni fini senza dover distogliere lo sguardo da ciò che si sta regolando

(Nota1) Come tutto il nostro software i loro file sorgente sono disponibili (Freeware e OpenSource) e sono scaricabili da qui: www.theremino.com/downloads/uncategorized (sezione "Custom controls") Questi controlli possono essere usati a piacere in ogni progetto anche senza nominarne la fonte. I sorgenti "Open" servono anche come garanzia che non vi abbiamo incluso malware.

Regolazione dei cursori



Questi sono i cursori originali di Microsoft, sono abbastanza comodi per cui abbiamo solo aggiunto il colore arancio e la possibilità di azzerarli.

<<< I cursori non a zero sono evidenziati con un colore arancio, per azzerarli basta fare click con il bottone destro del mouse (non tutti i cursori hanno uno zero e in tal caso non si colorano e non sono azzerabili con il mouse)

I cursori sono regolabili nei modi seguenti

- Cliccando sul cursore con il bottone destro del mouse per "azzerarli"
- Cliccando sul cursore con il bottone sinistro del mouse e muovendo il mouse su e giù
- Con la rotella del mouse
- Con i tasti freccia-sinistra e freccia-destra della tastiera
- Con i tasti freccia-su e freccia-giù della tastiera

Il metodo di muovere il mouse su e giù permette ampie e veloci regolazioni.

La rotella del mouse permette una regolazione comoda e immediata.

I tasti freccia permettono regolazioni fini senza distogliere lo sguardo da ciò che si sta regolando.

I tasti freccia sinistra/destra o su/giù hanno lo stesso effetto, ma può essere più intuitivo usare i primi per i cursori orizzontali e i secondi per i cursori verticali.

Domande e Risposte

Modificare il testo dei pannelli del programma nelle varie lingue

Modificare i file: "..\Docs\Language_Eng.txt" e "..\Docs\Language_Ita.txt"

Per le lingue Tedesco, Francese e Spagnolo basta copiare il file inglese tre volte con i nomi seguenti: "..\Docs\Language_Deu.txt", "..\Docs\Language_Fra.txt", "..\Docs\Language_Esp.txt"

Editare il file di configurazione

Normalmente la associazione tra configurazioni e moduli viene mantenuta allineata dal ThereminoHAL, che usa i nomi dei moduli per stabilire le giuste configurazioni da adottare.

Normalmente l'HAL riesce a usare la giusta configurazione anche se si scollegano e sostituiscono moduli. Ma in alcuni casi, se si cambia nome ai moduli con un HAL che si trova su un diverso computer, o in una cartella diversa, allora l'allineamento tra configurazione e hardware si perde. In questi casi si può cliccare sulla tendina a discesa del nome del modulo e ripristinare l'allineamento scegliendo la giusta configurazione per ogni Modulo.

Per fare modifiche più complesse, si può aprire il file "Theremino_HAL_ConfigDatabase.txt" con un editor di testo come il "Notepad" e editare manualmente le configurazioni che sono abbastanza semplici.

Come ridurre il lavoro della CPU?

- Chiudere o minimizzare la finestra "Component details" e minimizzare la finestra principale.
- Ridurre la "Comm speed", come spiegato nelle prime pagine di questo documento.

Cosa fare se il modulo non funziona

Il modulo si collega alla rete ma non appare nella lista dello IoT HAL

A volte la rete potrebbe trovarsi in uno stato anomalo, alcuni dicono che sia dovuto al Proxy, altri che sia dovuto ad errori di configurazione. Consigliamo di eseguire la sequenza seguente:

- ◆ Fare click sulla casella di ricerca (in basso a sinistra)
- ◆ Scrivere CMD e attendere che appaia "Prompt dei comandi"
- ◆ Cliccare con il tasto destro del mouse e scegliere "Esegui come amministratore"
- ◆ Copiare i seguenti due comandi, uno per uno, e dare ENTER ad ognuno.
`netsh int ip reset`
`netsh winsock reset`
- ◆ Riprovare se i moduli appaiono sul IoT HAL (o avviandolo o premendo "Riconosci")
- ◆ Se non appaiono ancora provare a fare un riavvio.
- ◆ Se non appaiono ancora provare anche il comando seguente (che però resetta completamente il firewall e quindi potrebbe bloccare alcuni programmi).
`netsh advfirewall reset`

Controllare la procedura di avviamento dello IoT Module

- ◆ Collegare lo IoT Module a una porta USB
- ◆ Riprogrammare il modulo con "ESP_LOG_VERBOSE" nel file "IoTModuleSetup.h"
- ◆ Avviare la app. [*Theremino Terminal*](#) (oppure il "Monitor seriale" dell'IDE di Arduino)
- ◆ Connettere la porta COM cui è collegato lo IoT Module con Speed = 115200
- ◆ Premere il pulsantino di reset e leggere le righe in arrivo nel terminal (o monitor).
- ◆ Verificare se si collega alla rete (in caso contrario verificare il nome della rete o la password)
- ◆ Eventualmente copiare tutto quello che arriva e farlo analizzare da un esperto.

Controllare il tipo di modulo

Seguire i consigli di [*questa pagina*](#) sui modelli di moduli. In alcuni casi si deve collegare il Pin 0 con il 3.3 volt attraverso un resistore da 10k.

Il collegamento USB non funziona e non appare una porta COM

In alcuni computer potrebbe mancare il driver per il chip CH340 o per il chip CP2102. Leggere le istruzioni per installarli nella prossima pagina.

Se non si riesce a comunicare con il modulo

Su alcuni computer potrebbe accadere di non riuscire a comunicare con lo lotModule attraverso la porta USB e la porta seriale (COM).

Sintomi

1. Il sistema operativo segnala un errore quando si collega il cavo USB.
2. Collegando il cavo USB non appare una nuova porta COM.

La comunicazione con lo lotModule passa attraverso un chip CH340 o uno CP2102. Questi chip convertono i dati da USB a seriale e hanno bisogno di un driver apposito, che in alcuni computer potrebbe mancare. In questi casi si risolve il problema installando manualmente il driver.

Scaricare e installare il driver per il chip CH340

Questa è la pagina ufficiale del produttore "Quin Heng Technology" per il download del driver, per tutti i tipi di CH340. Il driver è valido per i sistemi operativi Windows 10/8.1/8/7/VISTA/XP e Server 2016..2000/ME/98, sia a 32 che a 64 bit, e certificato da Microsoft Digital Signature: http://www.wch.cn/download/CH341SER_EXE.html

- ◆ *Aprire la pagina con un click sul link qui sopra*
- ◆ *Premere il tastone azzurro "DOWNLOAD"*
- ◆ *Attendere il completamento del download*
- ◆ *Fare click sul file CH341SER.EXE*
- ◆ *Acconsentire alle modifiche*
- ◆ *Premere "Install"*

Scaricare e installare il driver per il chip CP2102

Il driver per questo chip è sempre preinstallato su Windows. Comunque nei rari casi che la porta COM non apparisse, basta andare su questa pagina e installarlo:

<https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers>

- ◆ *Aprire la pagina con un click sul link qui sopra*
- ◆ *Scaricare "Download Windows 10 Universal" (premendo "Download VCP")*
- ◆ *Scompattare lo zip in una cartella a piacere*
- ◆ *Lanciare "CP010xVCPiInstaller_x86.exe" (oppure "_x64.exe" su sistemi a 32bit)*

Il file "IotModuleSetup.h" (opzioni principali)

IOTMODULE_LOG_LEVEL ESP_LOG_NONE

Scrivendo VERBOSE al posto di NONE il modulo invia alla seriale molte informazioni utili. In alcuni casi possono aiutare a capire perché il modulo non si collega alla rete WiFi. Per vedere i messaggi si può utilizzare una applicazione di tipo "Terminale seriale".

IOTMODULE_PINOUT_TYPE 1

Impostare questa opzione con "0" se si utilizza un modulo di tipo ESP32-WROOM-32.

Impostare questa opzione con "1" se si utilizza un modulo di tipo ESP32-PICO-KIT V4

Impostare questa opzione con "2" se si utilizza un modulo di tipo ESP32-TTGO-T7 v1.0

Impostare questa opzione con "3" se si utilizza un modulo di tipo ESP32-TTGO-T7 v1.3.

Le immagini dei moduli sono nelle prime pagine di questo documento.

IOTMODULE_INOUT_PINS 99

Con questa opzione si regola il numero di Pin che verranno visualizzati nella applicazione IoT HAL. Per vedere tutti i Pin si imposta 28 per i moduli PICO e TTGO, oppure 26 per il modulo WROOM. Impostando un numero maggiore di 28 (ad esempio 99) si è sicuri di abilitare tutti i Pin, indipendentemente dal modulo usato.

IOTMODULE_ADC24_PINS 0

Con questa opzione (da 0 a 16), si regola il numero di Pin del modulo Adc24 che verranno visualizzati nella applicazione IoT HAL. Con "0" il modulo Adc24 viene completamente disabilitato.

IOTMODULE_GENERIC_PINS 0

Con questa opzione (da 0 a 20), si regola il numero di Pin "generici" che verranno visualizzati nella applicazione IoT HAL. Normalmente si tiene questa opzione a "0", perché di solito il numero di Pin fisici inutilizzati è sufficiente per utilizzarli come segnaposto per i Pin "generici".

Il file "IotModuleSetup.h" (Configurazione di rete)

La principale modifica introdotta nella versione v.191 è la configurazione multi SSID in IotModule.h.

Questa funzionalità è molto utile se si conosce in anticipo la credenziale di accesso (SSID, password) di alcuni Access Point che si intende collegare, insieme al tipo di connessione IP (DHCP, IP statico); è possibile preparare la configurazione in anticipo, programmando il firmware una sola volta.

Ad ogni accensione il modulo cercherà una corrispondenza tra gli SSID riconosciuti e configurati e aggiornerà continuamente l'elenco degli SSID riconosciuti alla ricerca di una corrispondenza. Se viene trovata una corrispondenza e la connessione va a buon fine, l'SSID abbinato verrà quindi utilizzato come riferimento: se il modulo perde temporaneamente la connettività wifi proverà a riconnettersi allo stesso SSID.

Affinché il modulo possa connettersi a un'altra rete, è necessario spegnere e riaccendere l'alimentazione in ingresso oppure premere il pulsante di ripristino.

Questa nuova versione introduce alcuni nuovi parametri e alcune incompatibilità con i parametri utilizzati in precedenza.

I parametri della stringa di più parole consentono di specificare più elementi su una riga; ogni elemento è separato dall'altro da uno o più spazi.

Come vedrai negli esempi, i parametri multiword formano una sorta di configurazione organizzata in colonne, dove ogni colonna identifica una possibile configurazione.

Vedere l'appendice **Configurazione di rete** per una descrizione completa dei parametri

Il file "IotModuleSetup.h" (opzioni secondarie)

Configurazioni speciali

IOTMODULE_ENCODERS_TYPE 1

Selezionando "1" si utilizza un metodo "software" per leggere gli Encoder. Con gli encoder software il numero di passi dell'encoder viene moltiplicato per quattro, invece con gli encoder hardware i passi vengono moltiplicati per due. Quindi gli encoder software sono più precisi e sono anche più resistenti ai disturbi elettrici. In compenso quelli hardware sopportano frequenze maggiori.

IOTMODULE_FUOTA_ENABLE 1

Selezionando "1" si abilita la possibilità di aggiornare il firmware del modulo via WiFi. Di solito lo si lascia abilitato, ma chi ha paura che gli extraterrestri gli riprogrammino il modulo per i loro scopi malefici, può impostare questo parametro a zero. In questo modo nemmeno San Gennaro potrà riprogrammare il modulo e si dovrà andare ogni volta sul posto a collegare il modulo via USB.

IOTMODULE_ADC24_COMMUNICATION_SPEED 16

Con questa opzione si può impostare la velocità di collegamento con il modulo Adc24 da 1 MHz a 16 Mhz. Di solito si usa la massima velocità (16) e questo garantisce le massime prestazioni per tutti i Pin usati (si minimizza il rumore sui Pin di tipo "Period" e si migliora leggermente qualche altra funzione). Abbassare la velocità di comunicazione potrebbe essere necessario nel caso che i fili di collegamento tra lo ESP32 e il modulo Adc24 fossero lunghi. Con 1 MHz (e cavi schermati) si potrebbe arrivare anche fino a dieci metri e oltre.

IOTMODULE_OUTPUT_WATCHDOG_ENABLE = 1

Questa riga imposta un dispositivo di sicurezza che, in caso di perdita della comunicazione, porta a zero tutti i Pin configurati come uscita (DigOut, DAC, PWM, SERVO). Con ENABLE = 1 l'azzeramento avviene dopo un secondo, con ENABLE = 0 l'azzeramento non avviene mai.

Il file "IotModuleSetup.h" (opzione WiFi Mode Pin)

IOTMODULE_WIFI_MODE_PIN = n

Questa caratteristica può risultare utile in fase di test in quanto permette di connettere lo IotModule ad un access point, oppure ad un PC senza dover riprogrammare il firmware.

- ◆ Se WIFI_MODE_PIN vale -1 o non è definito, allora la modalità WiFi di funzionamento del IotModule, Station o SoftAP, dipende esclusivamente dal valore della variabile IOTMODULE_WIFI_MODE descritta in precedenza
- ◆ Se alla variabile IOTMODULE_WIFI_MODE_PIN si assegna un **valore valido** che identifica un Pin fisico dello IotModule (36 per esempio), allora la modalità di funzionamento è decisa esclusivamente dallo stato del relativo Pin fisico, nel momento in cui il IotModule è avviato o resettato.

I Pin validi sono tutti i Pin fisici (non Generic o Adc24 da 60 in su). Inoltre per essere valido il Pin deve essere abilitato, cioè visibile nella lista dello IotHAL, ma non è necessario che sia configurato.

Se il livello del Pin è basso il WiFi viene inizializzato in modalità Station, se il livello del Pin è alto il WiFi viene inizializzato in modalità SoftAP.

Il tutto funziona facilmente e automaticamente nel contesto DHCP, ma anche con configurazione IP statica se la configurazione di rete dell'access point risulta compatibile.

Avvertenza : nel caso che il IotModule sia avviato in modalità SoftAP e nei pressi vi sia anche l'access point, il PC visualizzerà due SSID identici e difficilmente distinguibili : uno è quello dell'AP e l'altro dello IotModule.

Il file "IotModuleSetup.h" (opzioni per gli stepper)

IOTMODULE_STEPPER_TIMING 0

Questa impostazione permette di scegliere l'intervallo di frequenza degli step per i motori Stepper

IOTMODULE_STEPPER_TIMING = 8 :	320...307200 Hz -	92160 mm/Sec
IOTMODULE_STEPPER_TIMING = 7 :	160...153600 Hz -	46080 mm/Sec
IOTMODULE_STEPPER_TIMING = 6 :	80...76800 Hz -	23040 mm/Sec
IOTMODULE_STEPPER_TIMING = 5 :	40...38400 Hz -	11520 mm/Sec
IOTMODULE_STEPPER_TIMING = 4 :	20...19200 Hz -	5760 mm/Sec
IOTMODULE_STEPPER_TIMING = 3 :	10...9600 Hz -	2880 mm/Sec
IOTMODULE_STEPPER_TIMING = 2 :	5...4800 Hz -	1440 mm/Sec
IOTMODULE_STEPPER_TIMING = 1 :	3...2400 Hz -	720 mm/Sec
IOTMODULE_STEPPER_TIMING = 0 :	2...1200 Hz -	360 mm/Sec

FUOTA - Firmware Update On The Air

Utilizzando un modulo collegabile senza fili (a parte l'alimentazione che può anche essere generata localmente vicino al modulo, per esempio, tramite un pannello fotovoltaico), è possibile che la collocazione finale sia fisicamente molto scomoda da raggiungere.

Se questo fatto può essere da una lato un vantaggio, dall'altro si può rivelare un grosso problema, quando si dovesse aggiornare il firmware.

La modalità FUOTA viene incontro a questa problematica, ed è disponibile sull'Arduino IDE, come alternativa alla programmazione USB.

Per poter usufruire di questa modalità, lo `IoTModule` deve essere già stato programmato una volta (via USB), con il firmware `IoTModule`, e deve già essere in grado di connettersi correttamente alla rete WiFi. Ovvero l'Applicazione `IoT HAL` deve poterlo visualizzare tra i moduli connessi.

Per quanto sia possibile il riconoscimento contemporaneo di più moduli, la applicazione Arduino IDE permette la programmazione di un solo modulo per volta.

Attenzione

Il modulo accetta l'aggiornamento via Wifi solo se la costante `"IOTMODULE_FUOTA_ENABLE"` è posta a 1 nel file `"IoTModuleSetup.h"`

Se per sicurezza si vuole impedire l'aggiornamento WiFi, porre la costante a 0

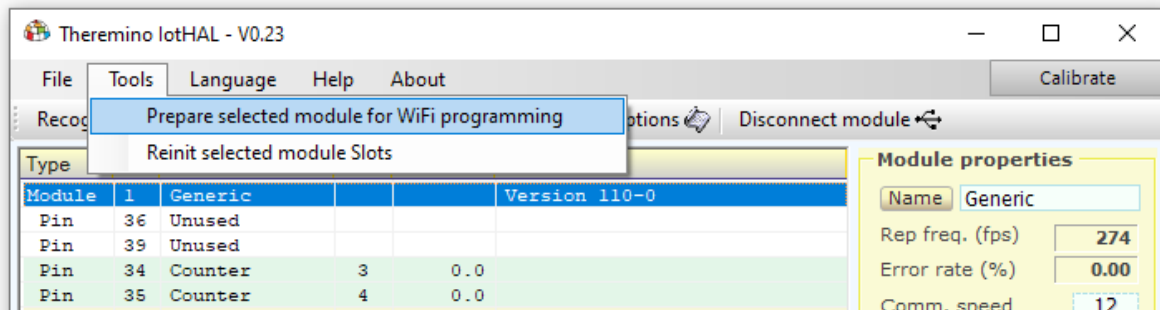
Prima di lanciare la programmazione FUOTA si deve essere sicuri che la nuova versione di firmware funzionerà correttamente.

Se ci sono errori che compromettono la connessione alla rete WiFi (per esempio il nome della rete o la password sbagliati), non sarà più possibile comunicare con lo `IoTModule` via WiFi.

Se si sbaglia qualcosa si dovrà per forza andare dove si trova il modulo e collegarlo a una porta USB.

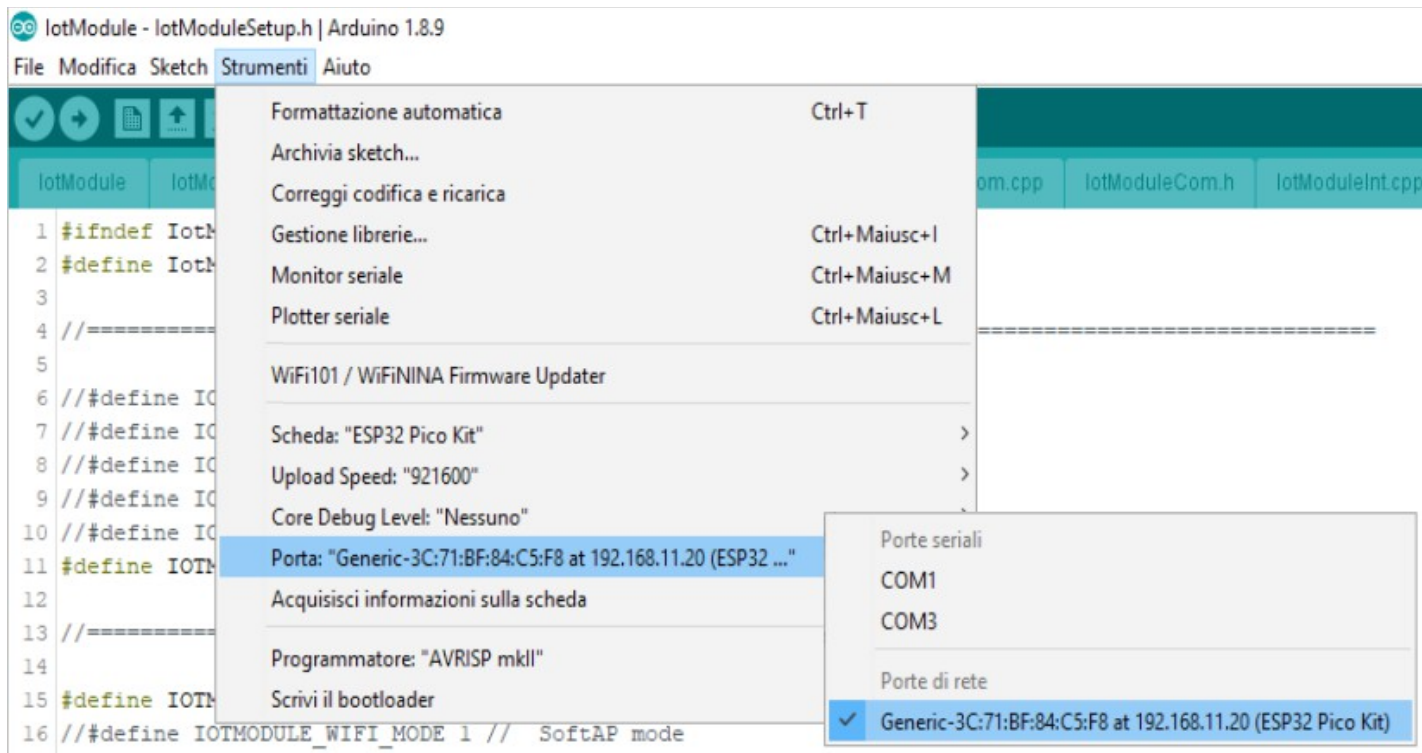
FUOTA - Procedura per l'aggiornamento

- ◆ Selezionare, sull'applicazione IoT HAL, lo IoTModule da aggiornare.
- ◆ Selezionare il menù "Tools"/"Prepare selected module for WiFi programming"



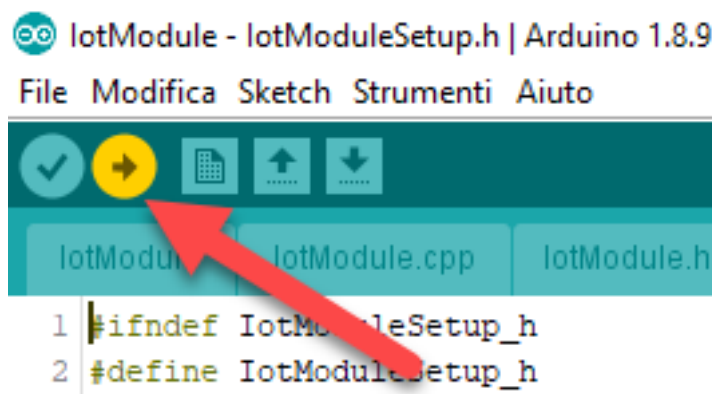
- ◆ Il modulo a questo punto si sconnetterà dallo IoT HAL e rimarrà per 120 secondi visibile dall'IDE Arduino tra le **porte di rete**, in attesa della programmazione.
- ◆ Dopo 120 secondi il modulo si riconnette automaticamente con lo IoT HAL, in modo da poterlo recuperare senza andare sul posto a resettarlo, nel caso che la richiesta di FUOTA sia avvenuta per errore.
- ◆ Selezionare il modulo, identificabile dalla porta di rete :

<nome> - <MAC> at <IP> (<tipo scheda>)

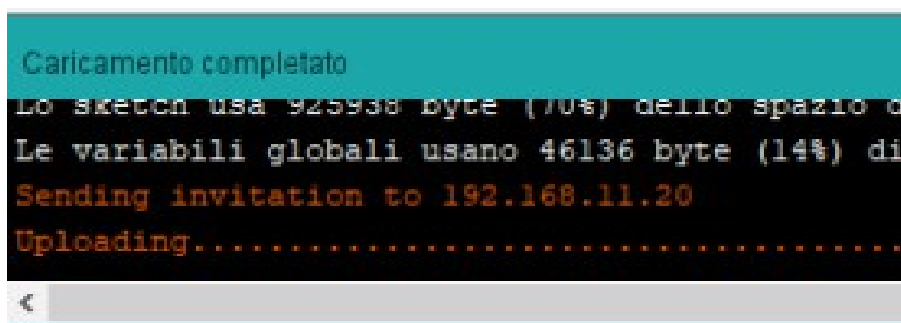


FUOTA - Inviare l'aggiornamento via WiFi

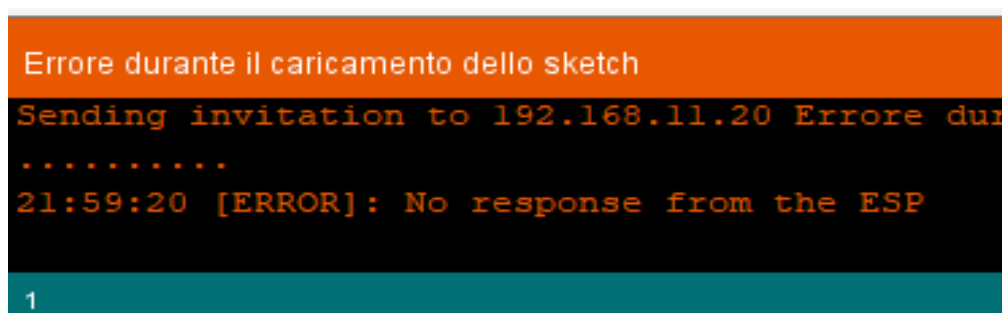
Premere il tasto di esecuzione standard dell'IDE Arduino, che ricompilerà lo sketch e programmerà il lotModule via Wifi



In caso di successo si leggerà "Caricamento completato"



In caso di errore (lotModule non predisposto al FUOTA o altri errori di comunicazione), si leggerà "Errore durante il caricamento dello sketch"

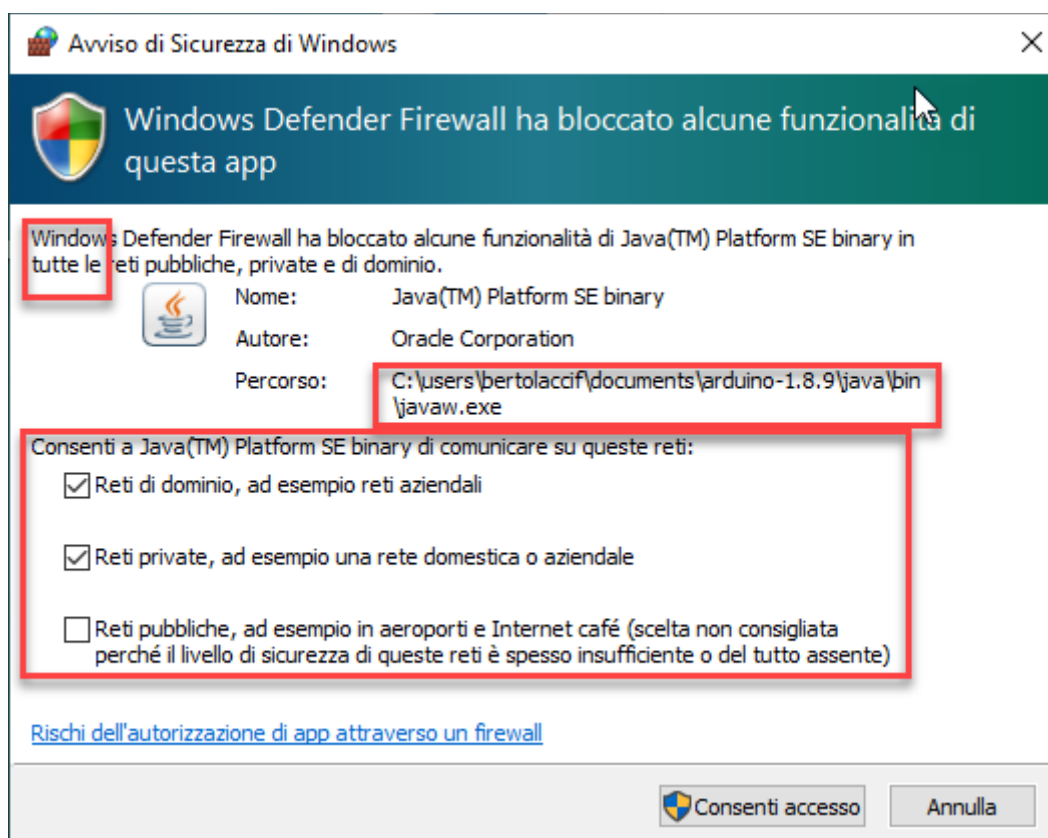


FUOTA - Rete e Firewall

L'IDE Arduino è in grado di connettersi al IoTModule per eseguire la procedura FUOTA, solo se il PC dove è eseguito l'IDE e lo IoTModule si trovano sulla stessa rete.

Ovvero devono essere sullo stesso dominio di Broadcast (non importa se il PC è connesso con cavo Ethernet o via WiFi).

Inoltre, le regole di firewall di Windows relative all'IDE Arduino devono essere state attivate. Esse sono proposte al primo avvio dell'IDE in questo modo:



Connettere lotHAL e lotModule su reti distinte

L'applicazione lotHAL riconosce automaticamente tutti gli lotModule solo se il PC dove è eseguito e gli lotModule sono configurati e connessi ad una stessa rete, per esempio se il loro IP appartiene alla rete 192.168.1.0/24 (ovvero con maschera 255.255.255.0) e quindi 192.168.1.5, 192.168.1.6, 192.168.1.7 e così via.

Con poco sforzo è anche possibile far comunicare lotHAL e lotModule su reti diverse. L'unico vincolo è che la configurazione dei lotModule che non appartengono alla stessa rete del PC siano stati configurati con IP statico oppure con l'equivalente reservation DHCP.

Ipotizziamo che il PC dove viene eseguita l'applicazione lotHAL faccia parte della rete "pippo" 192.168.1.0/24 e parte dei lotModule siano sulla stessa rete e altri sulla rete "pluto" 192.168.99.0/24 e le due reti siano raggiungibili a vicenda tramite routing.

Gli lotModule sulla rete "pippo" saranno riconosciuti automaticamente, mentre quelli collocati sulla rete "pluto", per essere riconosciuti, dovranno essere elencati puntualmente nel file CommOptions.txt (consultare gli esempi nel file), usando la direttiva WanIpInfo:

WanIpInfo <indirizzo IP lotModule> <porta UDP lotModule> <timeout ms>

Valorizzando i parametri, in questo modo:

WanIpInfo 192.168.99.5 49153 1000

WanIpInfo 192.168.99.6 49153 1000

WanIpInfo 192.168.99.7 49153 1000

Sfruttando la direttiva descritta è altresì possibile connettere gli lotModule attraverso una VPN o, caso estremo, direttamente via internet; in quest'ultimo caso i singoli lotModule sono esposti a rischi di malfunzionamenti dovuti all'esposizione diretta della loro porta UDP su internet.

Inoltre è richiesta anche competenza nella configurazione delle regole di NAT e di firewall dei router connessi ad Internet e partecipanti al collegamento, ciò va oltre lo scopo di questa documentazione.

Ecco un esempio di configurazione che permette di connettere tre lotModule via internet, senza VPN, collegati direttamente alla rete di un router e raggiunti attraverso NAT puntuali su tre porte UDP distinte:

WanIpInfo <ip statico pubblico del router> 49153 1000

WanIpInfo <ip statico pubblico del router> 49154 1000

WanIpInfo <ip statico pubblico del router> 49155 1000

Sostituire <ip statico pubblico del router> con l'effettivo IP pubblico assegnato dall'Internet provider.

Rete locale o Internet

Internet è più vulnerabile rispetto al WiFi e alla rete locale cablata. Per aumentare la sicurezza in Internet si potrebbe utilizzare una VPN. In ogni caso i maggiori pericoli non vengono dai pirati informatici, ma da interruzioni della comunicazione, errori umani o guasti. Quindi, come già scritto, non utilizzate mai il nostro sistema per controllare apparecchiature pericolose o essenziali.

Configurazione di rete

Configuration Multi SSID

IOTMODULE_STATIC_IP : deprecato, deve essere rimosso dalla configurazione

IOTMODULE_WIFI_MODE : già utilizzato come valore numerico, ora è una stringa di una sola parola. Definisce la modalità di connessione principale del modulo. Può essere STATION o SOFTAP

IOTMODULE_WIFI_ASSOC : nuova stringa di una sola parola.

Definisce la modalità wifi associata ad una singola "colonna" di configurazione. Può essere STATION o SOFTAP

IOTMODULE_IP_MODE : nuova stringa di più parole

Definisce la modalità di configurazione del livello IP. Può essere DHCP o STATICO

IOTMODULE_IP_IP : nuova stringa di più parole

Definisce l'indirizzo del modulo IPv4. Vedi esempi

IOTMODULE_IP_NETWORK : nuova stringa multi parola

Definisce la maschera di rete IPv4. Vedi esempi

IOTMODULE_IP_GATEWAY : nuova stringa di più parole

Definisce l'indirizzo del gateway di rete IPv4. Vedi esempi

IOTMODULE_IP_DNS : nuova stringa di più parole

Definisce l'indirizzo IPv4 del server DNS. Vedi esempi

IOTMODULE_WIFI_SSID : già utilizzato come stringa di una sola parola, ora è una stringa di più parole

Definisce il nome della rete locale wireless

IOTMODULE_WIFI_PASSWORD : già utilizzato come stringa di una sola parola, ora è una stringa di più parole

Definisce la password di rete locale wireless

IOTMODULE_SPACE_CHAR : nuovo singolo personaggio

Definisce un singolo carattere per la sostituzione dello spazio nelle parole SSID e password

IOTMODULE_UDP_PORT_NUMBER 49153

Il numero di porta UDP predefinito per la comunicazione IoT HAL è 49153. Per utilizzare un numero di porta diverso, rimuovere il commento e adattare la riga seguente. E imposta la stessa porta anche in IoT HAL.

IOTMODULE_WIFI_CHANNEL 1

Se imposti il modulo come SoftAP, con questa opzione puoi scegliere il canale WiFi (da 1 a 13).

Come convertire i parametri di configurazione nel file lotModule.h dalla v.182 alla v.191

Esempio 1 (station, dhcp)

v.182:

```
#define IOTMODULE_WIFI_MODE 0
#define IOTMODULE_WIFI_SSID "ssid1"
#define IOTMODULE_WIFI_PASSWORD "pwd1"
```

v.191:

```
#define IOTMODULE_WIFI_MODE "STATION"
#define IOTMODULE_WIFI_ASSOC " STATION"
#define IOTMODULE_WIFI_SSID " ssid1 "
#define IOTMODULE_WIFI_PASSWORD " pwd1 "
#define IOTMODULE_IP_MODE " DHCP "
#define IOTMODULE_IP_IP " - "
#define IOTMODULE_IP_NETWORK " - "
#define IOTMODULE_IP_GATEWAY " - "
#define IOTMODULE_IP_DNS " - "
```

Esempio 2 (station, static)

v.182:

```
#define IOTMODULE_WIFI_MODE 0
#define IOTMODULE_WIFI_SSID "ssid1"
#define IOTMODULE_WIFI_PASSWORD "pwd1"
#define IOTMODULE_STATIC_IP "192.168.11.1 192.168.11.254 255.255.255.0 192.168.11.254"
                                (module IP) (gateway IP) (net mask) (DNS IP)
```

v.191:

```
#define IOTMODULE_WIFI_MODE "STATION"
#define IOTMODULE_WIFI_ASSOC " STATION "
#define IOTMODULE_WIFI_SSID " ssid1 "
#define IOTMODULE_WIFI_PASSWORD " pwd1 "
#define IOTMODULE_IP_MODE " STATIC "
#define IOTMODULE_IP_IP " 192.168.11.1 "
#define IOTMODULE_IP_NETWORK "255.255.255.0 "
#define IOTMODULE_IP_GATEWAY "192.168.11.254 "
#define IOTMODULE_IP_DNS "192.168.11.254 "
```

Esempio 3 (softap)

v.182:

```
#define IOTMODULE_WIFI_MODE 1
#define IOTMODULE_WIFI_CHANNEL 1
#define IOTMODULE_WIFI_MODE_PIN -1
#define IOTMODULE_WIFI_SSID "ssid1"
#define IOTMODULE_WIFI_PASSWORD "pwd1"
```

v.191:

```
#define IOTMODULE_WIFI_MODE "SOFTAP"
```

```
#define IOTMODULE_WIFI_CHANNEL 1
#define IOTMODULE_WIFI_MODE_PIN -1
#define IOTMODULE_WIFI_ASSOC "SOFTAP"
#define IOTMODULE_WIFI_SSID "ssid1"
#define IOTMODULE_WIFI_PASSWORD "pwd1"
#define IOTMODULE_IP_MODE "DHCP"
#define IOTMODULE_IP_IP " - "
#define IOTMODULE_IP_NETWORK " - "
#define IOTMODULE_IP_GATEWAY " - "
#define IOTMODULE_IP_DNS " - "
```

Esempio 4 (station con softap opzionale, dhcp)

v.182:

```
#define IOTMODULE_WIFI_MODE 0
#define IOTMODULE_WIFI_CHANNEL 1
#define IOTMODULE_WIFI_MODE_PIN 33
#define IOTMODULE_WIFI_SSID "ssid1"
#define IOTMODULE_WIFI_PASSWORD "pwd1"
```

v.191:

Ora è possibile avere due coppie distinte di SSID/password.

```
#define IOTMODULE_WIFI_MODE "STATION"
#define IOTMODULE_WIFI_CHANNEL 1
#define IOTMODULE_WIFI_MODE_PIN 33
#define IOTMODULE_WIFI_ASSOC "STATION SOFTAP"
#define IOTMODULE_WIFI_SSID "ssid1 ssid2"
#define IOTMODULE_WIFI_PASSWORD "pwd1 pwd2"
#define IOTMODULE_IP_MODE "DHCP DHCP"
#define IOTMODULE_IP_IP " - - "
#define IOTMODULE_IP_NETWORK " - - "
#define IOTMODULE_IP_GATEWAY " - - "
#define IOTMODULE_IP_DNS " - - "
```

Nuove possibili configurazioni disponibili con v.191

Esempio 5 (multi station, dhcp)

```
#define IOTMODULE_WIFI_MODE "STATION"
#define IOTMODULE_WIFI_ASSOC " STATION STATION STATION STATION STATION STATION"
#define IOTMODULE_WIFI_SSID " ssid1 ssid2 ssid3 ssid4 ssid5 ssid6 "
#define IOTMODULE_WIFI_PASSWORD " pwd1 pwd2 pwd3 pwd4 pwd5 pwd6 "
#define IOTMODULE_IP_MODE " DHCP DHCP DHCP DHCP DHCP DHCP "
#define IOTMODULE_IP_IP " - - - - - - "
#define IOTMODULE_IP_NETWORK " - - - - - - "
#define IOTMODULE_IP_GATEWAY " - - - - - - "
#define IOTMODULE_IP_DNS " - - - - - - "
```

Esempio 6 (multi station, mixed dhcp + static)

```
#define IOTMODULE_WIFI_MODE "STATION"
#define IOTMODULE_WIFI_ASSOC " STATION STATION STATION STATION STATION STATION STATION STATION STATION "
#define IOTMODULE_WIFI_SSID " ssid1 ssid2 ssid3 ssid4 ssid5 ssid6 ssid7 ssid8 "
#define IOTMODULE_WIFI_PASSWORD " pwd1 pwd2 pwd3 pwd4 pwd5 pwd6 pwd7 pwd8 "
#define IOTMODULE_IP_MODE " DHCP DHCP DHCP DHCP DHCP DHCP DHCP STATIC STATIC "
#define IOTMODULE_IP_IP " - - - - - - - 192.168.11.1 172.20.1.1 "
#define IOTMODULE_IP_NETWORK " - - - - - - - 255.255.255.0 255.255.0.0 "
#define IOTMODULE_IP_GATEWAY " - - - - - - - 192.168.11.254 172.20.1.254 "
#define IOTMODULE_IP_DNS " - - - - - - - 192.168.11.254 172.20.1.254 "
```

Esempio 7 (multi station, mixed dhcp + static, optional softap)

Un livello basso sul pin 33 all'avvio seleziona le modalità della stazione, un livello alto seleziona la modalità SofAP.

```
#define IOTMODULE_WIFI_MODE "STATION"
#define IOTMODULE_WIFI_CHANNEL 1
#define IOTMODULE_WIFI_MODE_PIN 33
#define IOTMODULE_WIFI_ASSOC " STATION STATION STATION STATION STATION STATION STATION STATION STATION SOFTAP "
#define IOTMODULE_WIFI_SSID " ssid1 ssid2 ssid3 ssid4 ssid5 ssid6 ssid7 ssid8 ssid9 "
#define IOTMODULE_WIFI_PASSWORD " pwd1 pwd2 pwd3 pwd4 pwd5 pwd6 pwd7 pwd8 pwd9 "
#define IOTMODULE_IP_MODE " DHCP DHCP DHCP DHCP DHCP DHCP DHCP STATIC STATIC DHCP "
#define IOTMODULE_IP_IP " - - - - - - - 192.168.11.1 172.20.1.1 - "
#define IOTMODULE_IP_NETWORK " - - - - - - - 255.255.255.0 255.255.0.0 - "
#define IOTMODULE_IP_GATEWAY " - - - - - - - 192.168.11.254 172.20.1.254 - "
#define IOTMODULE_IP_DNS " - - - - - - - 192.168.11.254 172.20.1.254 - "
```

Esempio 8 (sostituzione di spazi)

I caratteri spazi sono già utilizzati come separatori tra le parole in stringhe di più parole; se è necessario inserire uno o più spazi nell'SSID e/o nella password, è necessario identificare un carattere non ancora utilizzato nella coppia di credenziali.

Questo particolare carattere deve essere specificato nella direttiva dedicata e utilizzato al posto degli spazi nelle parole di configurazione; questi caratteri verranno automaticamente sostituiti da spazi dal firmware durante l'analisi della configurazione, come puoi vedere in questo esempio.

```
#define IOTMODULE_WIFI_MODE "STATION"
#define IOTMODULE_SPACE_CHAR '&'
#define IOTMODULE_WIFI_ASSOC " STATION "
#define IOTMODULE_WIFI_SSID " SSID&WITH&spaces "
#define IOTMODULE_WIFI_PASSWORD " password&with&SPACES "
#define IOTMODULE_IP_MODE " DHCP "
#define IOTMODULE_IP_IP " - "
#define IOTMODULE_IP_NETWORK " - "
#define IOTMODULE_IP_GATEWAY " - "
#define IOTMODULE_IP_DNS " - "
```

Si prega di notare l'uso di virgolette singole nella definizione dei caratteri.
In questo caso le parole SSID e password effettive verranno internamente convertite da:

"SSID&WITH&spaces" and "password&with&SPACES"

to:

"SSID WITH spaces" and "password with SPACES"

Esempio 9 (come evitare configurazioni errate)

Per evitare configurazioni errate, nelle colonne dove si utilizza la modalità DHCP, è obbligatorio inserire un segnaposto (qualsiasi carattere o parola che si vuole) come falso valore di valori di configurazione non necessari (IP, NETWORK, GATEWAY e DNS); nell'esempio seguente i quattro segnaposto necessari (area rossa, ultime 5 colonne) sono stati omessi. Questo è sbagliato, dopo la fase di analisi della configurazione durante l'avvio, solo la prima colonna con configurazione IP STATICO verrà presa in carico dal firmware.

```
#define IOTMODULE_WIFI_MODE "STATION"
#define IOTMODULE_WIFI_ASSOC " STATION STATION STATION STATION STATION STATION"
#define IOTMODULE_WIFI_SSID " ssid1 ssid2 ssid3 ssid4 ssid5 ssid6 "
#define IOTMODULE_WIFI_PASSWORD " pwd1 pwd2 pwd3 pwd4 pwd5 pwd6 "
#define IOTMODULE_IP_MODE " STATIC DHCP DHCP DHCP DHCP DHCP "
#define IOTMODULE_IP_IP " 192.168.11.1 "
#define IOTMODULE_IP_NETWORK " 255.255.255.0 "
#define IOTMODULE_IP_GATEWAY " 192.168.11.254 "
#define IOTMODULE_IP_DNS " 192.168.11.254 "
```

Questo è un esempio di configurazione corretta, il carattere "-" è stato utilizzato come segnaposto del valore:

```
#define IOTMODULE_WIFI_MODE "STATION"
#define IOTMODULE_WIFI_ASSOC " STATION STATION STATION STATION STATION STATION"
#define IOTMODULE_WIFI_SSID " ssid1 ssid2 ssid3 ssid4 ssid5 ssid6 "
#define IOTMODULE_WIFI_PASSWORD " pwd1 pwd2 pwd3 pwd4 pwd5 pwd6 "
#define IOTMODULE_IP_MODE " STATIC DHCP DHCP DHCP DHCP DHCP "
#define IOTMODULE_IP_IP " 192.168.11.1 - - - - - "
#define IOTMODULE_IP_NETWORK " 255.255.255.0 - - - - - "
#define IOTMODULE_IP_GATEWAY " 192.168.11.254 - - - - - "
#define IOTMODULE_IP_DNS " 192.168.11.254 - - - - - "
```


Altri miglioramenti

- Le parole chiave STATION, SOFTAP, DHCP, STATIC non fanno distinzione tra maiuscole e minuscole, possono essere utilizzate tutte maiuscole, minuscole o in un mix.
- La modalità di configurazione IP di SoftAP è disponibile esclusivamente in DHCP; La modalità STATICA è stata rimossa.
- I messaggi del registro di debug, opzionalmente configurabili e disponibili nella finestra del monitor seriale nell'IDE di Arduino, sono stati migliorati ed elencano ora gli SSID disponibili e quello utilizzato
- Nuova piedinatura per le schede ESP32-TTGO T7 v1.0 e ESP32-TTGO T7 v1.3.
È stato introdotto il pinout di tipo 3 e deve essere utilizzato con il modulo ESP32-TTGO T7 v1.3 perché i pin 18 e 19 sono incrociati dal modulo ESP32-TTGO T7 v1.0.
- Correzione su GPIO 36, 37 e 39.
GPIO 36 e 39 non sono stati gestiti correttamente e lo stato di uno interferiva con l'altro. Corretti anche i possibili tipi di pin utilizzabili sul GPIO 37
- Compatibilità con la libreria Arduino ESP32 v 1.0.5
La libreria Arduino ESP32 v.1.0.5, basata su ESP-IDF v3.3, disponibile dal 23/02/2021, introduce una modifica a una definizione del tipo di dati che impedisce la compilazione del progetto
- Revisione PINOUT di ESP32-TTGO T7 v1.3
Rimossi GPIO7 e GPIO11, non utilizzabili in quanto già collegati a chip flash esterno, sostituiti da GPIO16 e GPIO17
- Nuovo parametro per la potenza wifi TX: `IOTMODULE_WIFI_TX_POWER_INDEX`
I valori possibili sono:
10 = 19,5 dBm (predefinito, utilizzato dalla precedente revisione del firmware)
9 = 19 dBm
8 = 18,5 dBm
7 = 17 dBm
6 = 15 dBm
5 = 13 dBm
4 = 11 dBm
3 = 8,5 dBm
2 = 7 dBm
1 = 5 dBm
0 = 2 dBm
Ad esempio, per impostare la potenza TX massima a 13 dBm:
`#define IOTMODULE_WIFI_TX_POWER_INDEX 5`
- Ottimizzazione della velocità di connessione wifi se è definita una sola colonna di configurazione
L'operazione di scansione della rete wifi nel caso di configurazione multi SSID introduce un piccolo ritardo nella connessione. Se non è richiesta la configurazione multi SSID, ovvero se nella configurazione è presente una sola colonna per i parametri SSID, la modifica introdotta annulla questo ritardo velocizzando al limite la connessione.

Compatibilità con Arduino IDE

La compilazione del firmware della versione v.191 e la funzionalità FUOTA sono state testate con successo con Arduino IDE v.1.8.10 e v.1.8.19; non sono stati ancora testati con Arduino IDE v.2.0.