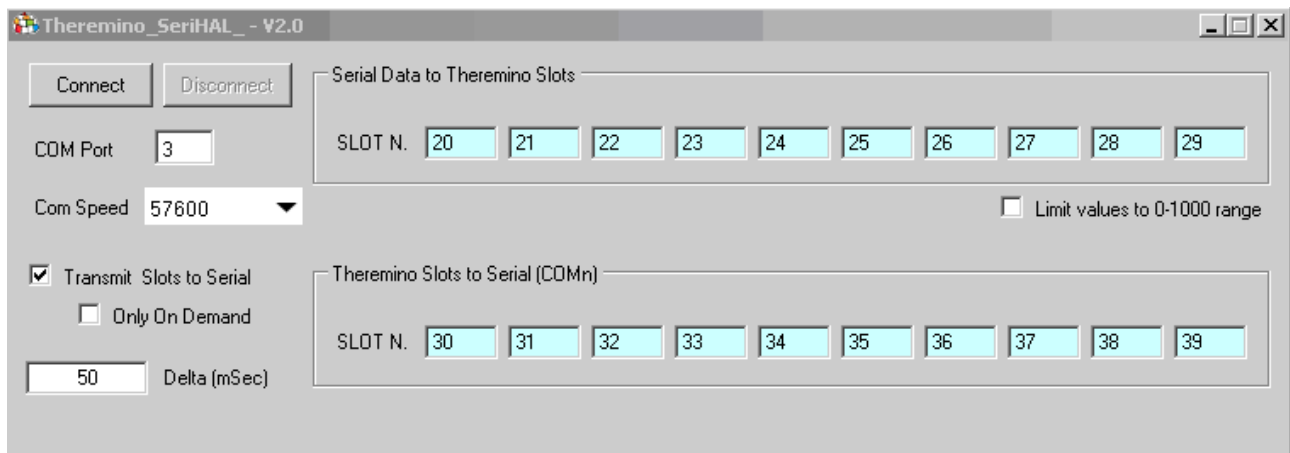


Theremino_SeriHal_V2.2

Theremino_SeriHal è un bridge fra gli Slot del sistema Theremino ed una porta seriale fisica (RS232 o USB) o virtuale (serve un emulatore di cavo seriale virtuale). Con Theremino_SeriHal è possibile leggere e/o scrivere 10 slot scambiando il loro contenuto con l'interfaccia seriale connessa. I dati potranno poi essere ricevuti e/o inviati ad altre applicazioni del mondo thereminico, ad esempio SlotViewer, SignalScope, SDR, Logger o scambiati con un un MasterV5 ed altro ancora.



Gli slot da scrivere e leggere sono configurati nel file **Theremino_SeriHAL_INI.txt** e sono caricati all'avvio

```
COM_Speed=      57600
COM_Port=        3
SLOT_N1=         20
.....
SLOT_N10=        29
LimitValues=     False
TransmitSlots=   True
OnlyOnDemand=    True
SLOT_Out1=       30
....
SLOT_Out10=      39
DELTA_TX=        50
```

I parametri della comunicazione e delle altre opzioni sono salvati/caricati automaticamente dopo il primo avvio.

Sono disponibili tre modalità di scambio dei dati fra gli slot e l'interfaccia seriale

1) ricezione di dati dalla seriale e scrittura di un massimo di 10 slot.

Questa modalità è sempre attiva ed il formato della stringa da ricevere è il seguente

DatoNumerico1;DatoNumerico2;.....;DatoNumericoN <LF>

(N <= 10, senza spazi in mezzo e senza "punto e virgola" alla fine)

I dati vengono scritti in sequenza secondo la configurazione degli slot

Il DatoNumerico è un numero decimale 123456.789 le cui singole cifre sono caratteri ASCII.

esempi:

- la stringa alfanumerica **123.45;-78;450.3<LF>** scrive 123.45 nel 1° slot , -78 nel 2° e 450.3 nel 3°
- la stringa vuota **<LF>** scrive 0 nel primo slot

E' implementata una utile funzione di "troncamento" dei dati numerici in input qualora si desideri troncare questi valori al range thereminico "classico" tra 0 e 1000: questa funzione potrebbe essere utile ad esempio per la conversione di numeri derivanti da conversioni analogico/digitali a 10 bit di precisione, con numeri che possono arrivare a 1024.... I più esperti potranno modificare il programma e fare in modo che i dati in output vengano riscaldati su altro "range" di valori anziché operare un semplice troncamento.

2) trasmissione continua

questa modalità si abilita se è spuntata la casella "Transmit Slot On Serial". Vengono letti i dieci slot configurati ed il contenuto è inviato continuamente sulla seriale con il formato

DatoNumerico1;DatoNumerico2;.....;DatoNumerico10 <LF>

Il DatoNumerico è un numero decimale 123456.789 le cui singole cifre sono caratteri ascii. Il tempo di ripetizione dell'invio può variare fra 2 e 10.000 msec

3) trasmissione on demand

questa modalità si abilita se sono spuntate entrambe le caselle "Transmit Slot On Serial" ed "Only on Demand". E' alternativa alla trasmissione continua ed il formato dei dati inviati sulla seriale è il medesimo. La trasmissione si attiva quando si riceve un carattere <LF> ad esempio

- a) ricezione di una stringa **123.45;-78;1000;0.23;-345.67<LF>** che scrive valori nei primi 5 slot
- b) ricezione di una stringa vuota <LF> che scrive uno zero nel primo slot

Consigli ed accortezze

Per connettere Theremino_SeriHal a programmi esterni (Processing, Octave, Python, terminali seriali, ecc) occorre installare sul PC un emulatore di cavo seriale virtuale che crea interfacce seriali virtuali alle quali si connettono i vari moduli software.

Questi sono alcuni indirizzi, ma se ne possono trovare altri

<http://www.eterlogic.com/Products.VSPE.html> (free a 32 bit e 25\$ per 64 bit)

<http://com0com.sourceforge.net/> (free)

<https://freevirtualserialports.com/> (free basic)

<https://www.hhdsoftware.com/virtual-serial-ports> (55\$ + 20\$ per update a vita)

<https://www.eltima.com/products/vspdxp/> (140\$!!!)

Le porte virtuali debbono essere create (una tantum) prima dell'avvio di Theremino_SeriHal e di Processing o altri software per permetterne la connessione senza problemi o errori. Per le prove è stato usato VSPE di eterlogic configurando la porta virtuale come "Connector" poichè in tal modo può essere aperta contemporaneamente da due applicazioni (SeriHAL e Processing)

Theremino_SeriHal riceve ed invia i dati sotto forma di stringhe ascii: è compito del codice dei software collegati gestire la conversione fra i dati numerici e le stringhe ascii (in entrata ed in uscita) tramite parsing, conversioni e formattazioni.

E' meglio lavorare a baudrates non eccessivamente alti per non appesantire la comunicazione e la conversione dei dati, soprattutto se si usa la ricezione continua.

Occorre configurare tutte le applicazioni: in particolare occorre prestare attenzione al fatto che gli slot siano indirizzati correttamente, evitando sovrascritture e verificando che SlotViewer (se usato) sia correttamente impostato per gestirli e visualizzarli. E' fondamentale configurare correttamente il file **Theremino_SeriHAL_v2_12_INI.txt** se l'avvio è automatico tramite uno script.

Tutta la serie di programmi può essere fatta partire automaticamente con il semplice ed utilissimo

script **Start_ALL_Theremino_Apps.vbs** che si trova sempre sul sito theremino
<http://www.theremino.com/downloads/foundations#starter>

E' importante sottolineare che lo script NON viene editato/modificato: occorre solo inserirlo in una cartella che contiene tutte le sotto-cartelle dei programmi che devono avviarsi. I programmi che si avviano automaticamente devono avere il nome con formato "**Theremino_nnn.exe**".

Ad esempio nella cartella principale sono presenti 6 sottocartelle secondo il seguente schema:

...\CartellaPrincipale\Start_ALL_Theremino_Apps.vbs

...\CartellaPrincipale_1_HAL

...\CartellaPrincipale_2_SlotViewer

...\CartellaPrincipale_3_Logger

...\CartellaPrincipale_4_SeriHAL

...\CartellaPrincipale_5_SignalScope

...\CartellaPrincipale_6_Helper

All'interno di ciascuna cartella c'è il programma corrispondente (ad es. nella prima cartella c'è Theremino_HAL.exe). Lo script avvierà i programmi dell'ordine delle sottocartelle, il cui nome inizia con un numero progressivo

Il programma **Theremino_Helper.exe** (presente nel sito) consente invece di chiudere tutte le applicazioni con nome "**Theremino_xxxx**"

<http://www.theremino.com/downloads/uncategorized#helper>

=====
Test
=====

L'applicazione è stata provata con successo su interfacce seriali fisiche e virtuali connesse a schede HW ed a Processing (altri software come Python ed Octave richiedono un plugin/libreria aggiuntivo per gestire la trasmissione seriale)

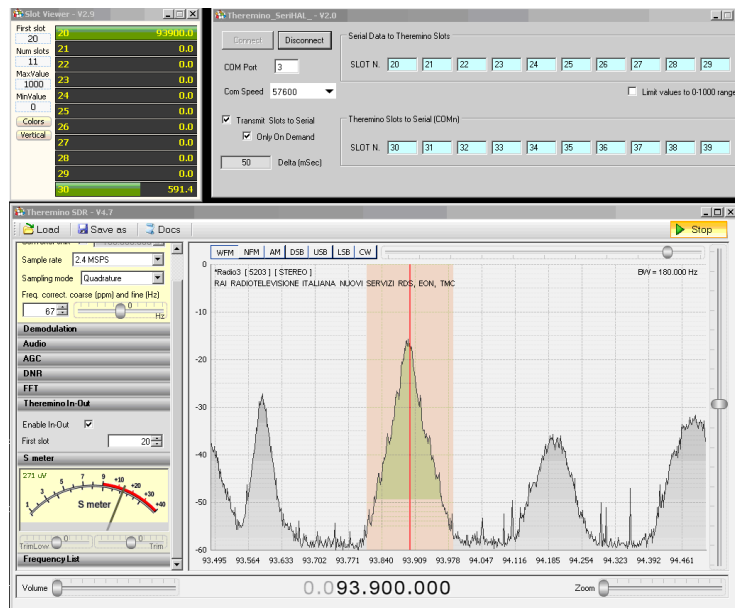
1) la scheda a microprocessore con un sensore a protocollo I2C (sensore di pressione BMP180)
acquisisce i dati di temperatura e pressione e li invia alla porta seriale.

HAL -> SlotViewer -> Theremino_Logger -> Theremino_SeriHal

2) Arduino UNO con encoder rotativo e display LCD

ArduinoUno/usb -> Theremino_SeriHal -> Theremino_SDR -> dongle RTL8320

La lettura degli slot è del tipo on demand



- arduino imposta la frequenza di 93.900 Mhz (RADIO RAI 3) e la invia a Theremino_SDR scrivendo il primo slot. La funzione Serial.print() formatta la variabile in una stringa ascii
- Theremino_SeriHal legge 10 slot e li invia ad arduino, uno slot contiene la frequenza ed un altro il livello del segnale sintonizzato
- arduino riceve i dati ascii, separa i campi, li converte in numeri con la funzioni atol()
- arduino visualizza frequenza e livello del segnale su un display lcd
- arduino tramite un encoder modifica la frequenza a passi di +/-100 kHz per sintonizzare tutte le stazioni FM
- arduino e Theremino_SDR sono sempre sincronizzati con la stessa frequenza di sintonizzazione

3) Processing comanda due servocomandi

mousePC-> Processing-> Theremino_SeriHal -> HAL-> usb/masterV5 -> servi pan&tilt

- Processing legge la posizione X-Y del mouse, la converte nel range 0-1000 e la invia sulla seriale
- Theremino_SeriHal la scrive in due slot (es 1-2)
- Theremino-HAL è configurato con due slot (es 1-2) di tipo servo16
- il masterV5 riceve i dati e muove i due servocomandi sui connettori 1 e 2
- i servi pan&tilt si muovono in sincronia col mouse

nota: la libreria Serial di Processing ha una istruzione molto spartana **nomeporta.write(char)** che scrive sulla seriale **un solo carattere** alla volta. Non c'è alcuna formattazione interna quindi per inviare il numero 12345.67 formattato come stringa ascii occorre scomporre il numero in singole cifre, poi sommare 48 a ciascuna ed infine scriverle in sequenza sulla seriale. Occorre gestire anche il punto decimale (ascii 59).

4) Processing varia il colore ad una finestra in base alla temperatura letta da un LM35

LM35-> master V5-> HAL -> Theremino_SeriHal -> Processing

- un LM35 è collegato al pin 6 del Master V5 ed HAL scrive lo slot 6
- Theremino-SeriHAL legge 10 slot ed il 1° è lo slot 6
- Processing riceve la stringa, ne fa il parsing ed estrae il primo dato
- Processing varia il colore di una finestra in funzione della temperatura del LM35

Le catene possono essere ulteriormente ampliate aggiungendo l'utilissimo nuovo visualizzatore di segnali Theremino_SignalScope che consente di analizzare in dettaglio cosa succede in due slot selezionabili a piacimento: per l'utilizzo di SignalScope si rimanda alla sua pagina specifica nel sito.

appendice 1: codici degli esempi di test

codice arduino per esempio 2)

```
/*
gestione di thereminoSDR (frequenza & livello segnale)
tramite bridge SeriHAL V2.11 e slot Theremino memory mapped

routine di parsing dei dati in arrivo copiata da
http://www.cplusplus.com/reference/cstring/strtok/

Theremino serial:
Com = arduino - speed 57600
trasmit slot on serial ON
only on demand ON
slot write 20 - xx al slot read 30 - xx
Theremino sdr:
i/o first slot= 20 (20=frequenza 30 = signal)
----
arduino
- pin2,3 encoder
- pin 4,5,6,7,8,9 display LCD
- pin 11 pulsante
- pin 13 led
- pin 10,12 free

all'avvio pigiare il bottone dopo la connessione di serihal
ruotando l'encoder si varia la frequenza di sintonizzazione
pigiando il pulsante si aggiorna la lettura del livello di segnale
*/

#include <LiquidCrystal.h>
#include <rotary.h> // adeguare il codice alla libreria usata
#define build "> ardu-SDR 100 <"
#define N 10 // numero slot ricevuti
const float f_start= 93900.0; // Khz iniziali
const float f_step=100.0; // KHz step
byte P2=2;
byte P3=3;
byte led=13;
byte but=11;
float slot[11]; // vettore dei dati per slot
char str[127]; // vettore caratteri ricevuti
char inChar; // singolo carattere ricevuto
float SDRlevel,SDRfreq;
int i,j,kk,kk1; // contatori
LiquidCrystal lcd(9, 8, 7, 6, 5, 4);
Rotary r = Rotary(2,3); // pin 2,3 encoder
volatile int stato; // ritorno encoder
//-----
void encoder() { // routine di interrupt encoder
  unsigned char result = r.process(); // 16 DIR_CW 32 DIR_CWW
  if (result){ stato=((result*2)-48)/(-16);} // +1 CW -1 CWW
}
//-----

void setup() {
  pinMode (led,OUTPUT);
  pinMode (but, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(P2), encoder, CHANGE);
  attachInterrupt(digitalPinToInterrupt(P3), encoder, CHANGE);
  Serial.begin (57600);
  i=1; // inizializza posizione vettore ricezione
```

```

lcd.begin(16, 2);
lcd.setCursor(0, 0); lcd.print(build);          // display versione
lcd.setCursor(0, 1); lcd.print("*****");
delay(2000);

while(digitalRead(but)){ // attende la connessione di serialHAL
  lcd.setCursor(0, 0); lcd.print("CONNECT SERIALHAL");
  lcd.setCursor(0, 1); lcd.print(" PUSH BUTTON ");
  digitalWrite(led,!digitalRead(led));
  delay(200);
  digitalWrite(led,LOW);
}

SDRfreq=f_start;
// scrive 1°slot serialHAL (frequenza) e riceve 10 slot
Serial.print(SDRfreq,0); Serial.println();
}
//-----

void loop() {
  if (!digitalRead(but)) {single_read(); } // lettura manuale
  if(Serial.available()) {leggi_slot();} // dati ricevuti
  if(stato) {scrivi_slot(); } // encoder ruotato
  // fai altro
}
//-----

void leggi_slot(){ // riceve i 10 slot
while (Serial.available() > 0) { // dati presenti
  inChar = Serial.read(); // legge carattere

  if (inChar != '\n') { // attesa terminatore
    str[i] =inChar;
    i=i+1;
  }

  else { // parsing
    char * pch;
    pch = strtok (str,";"); //elenco delimitatori parsing <;>
    j=1;
    while (pch != NULL) {
      slot[j]=atof(pch); // converte stringa in float
      j=j+1;
      pch = strtok (NULL, ";");
    }

    SDRlevel=slot[1]; // 1° slot rx configurato in serialHAL
    lcd.setCursor(0, 0); lcd.print("freq: ");
    lcd.setCursor(6, 0); lcd.print(SDRfreq,3); // display frequenza
    lcd.setCursor(0, 1); lcd.print("level: ");
    lcd.setCursor(7, 1); lcd.print(SDRlevel,0); // display livello

    i=0; // svuota vettore lettura e vettore slot
    for (int k=0;k<50;k++){str[k]="";}
    for (int k=0;k<=N;k++){slot[k]=0;}
  } // close else
} // close while
} //close function
//-----

void scrivi_slot(){
  if(stato==1) SDRfreq+=f_step; // CW incrementa frequenza
  if(stato==-1) SDRfreq-=f_step; // CWW decrementa frequenza
  stato=0; // azzera stato encoder
}

```

```

    Serial.print(SDRfreq,0);    Serial.println();
}
//-----

void single_read(){
    Serial.print(SDRfreq,0); Serial.println();
    delay(200);
}
//-----

```

codice per processing per esempio 3)

```

/*
theremino serihal.ini
COMport COM7
speed 57600
slot read -
slot write 1,2
trasmit slot to serial OFF
only on demand OFF
limit value OFF
---
master V5
pin 1, 2 as servo_16 -> pan&tilt
maxvalue 1000
tempo max 2000
tempo min 1000
---
processing
    invia sulla seriale la posizione X,Y del mouse all'interno del quadrato
*/

import processing.serial.*;
Serial myPort;

void setup(){
    size(200, 200);                // disegna un quadrato 200x200
    printArray(Serial.list());      // stampa la lista seriali disponibili
    String portName = Serial.list()[1]; // <-- aggiornare in base alla stampa
    myPort = new Serial(this, portName, 57600); // configura la porta seriale
}
//-----
void draw() {
    background(100);                // colora il quadrato di grigio
    int X=(mouseX)*5;                // normalizza a 1000
    int Y=(mouseY)*5;
    formattal(Y);                    // formatta e scrive Y
    myPort.write(59);                // scrive il separatore <;>
    formattal(X);                    // formatta e scrive X
    myPort.write(10);                // scrive il terminatore <LF>
    delay(100);
}
//-----
void formattal(int a){
    int appo=a;
    byte[] cifra=new byte [10];
    int n=int(log(a)/log(10))+1;      // calcola il numero di cifre
    for (int i = 1; i <= n; i = i+1){

```

```
    cifra[i]=byte(appo%10);
    appo=appo/10;
}
for (int i = n; i > 0; i = i-1){
    myPort.write(char(cifra[i]+48)); // converte cifra in ascii
}
//-----
```


codice arduino per esempio 4)

```
/*
theremino serihal.ini
  COMport COM7
  speed 57600
  slot read 6
  slot write 20
  trasmit slot to serial ON
  only on demand ON/OFF
  limit value OFF
---
master V5
  pin 6 as adc8 -> LM35 10mV/°C
  max 3300
  min 0

*/

import processing.serial.*;
Serial myPort; // Create object from Serial class

int lf = 10; // Linefeed in ASCII
String myString = null;
char sep1=59; // ->>;<<-

void setup() {
  colorMode(HSB, 255, 255, 255);
  size(200, 200);
  background(100);
  printArray(Serial.list());
  String portName = Serial.list()[1];
  myPort = new Serial(this, portName, 9600);
  myPort.clear();
  myString = myPort.readStringUntil(lf);
  myString = null;
}

void draw(){
  while (myPort.available() > 0) { //riceve i dati
    myString = myPort.readStringUntil(lf); // legge la stringa
    if (myString != null) {
      println(myString);
      float nums[] = float(split(myString, sep1)); //parsing
      for(int k=0;k<=9;k++){print (nums[k]); print (" - ");}
      println();println();
      int colore=int(nums[0]); // converte il primo dato
      print("slot: "); println(colore);
      colore /=10;
      print("gradi: "); println(colore);
      colore *=6;
    }
  }
}
```

```
    print("gradi x 6=HUE: "); println(colore);
    println("-----");
    background(colore,255,200); //HSB  // setta il colore-hue MAX 360
  }
}
//-----

void mousePressed() {
  myPort.write(10); // richiede una lettura on demand
}
//-----
```