

- Elettronica Open Source - <https://it.emcelettronica.com> -

## Costruiamo un Voice Shield per far parlare Arduino [Progetto Completo Open Source]

Posted By *Adriano Gandolfo* On 7 agosto 2013 @ 5:00 In Arduino | [18 Comments](#)

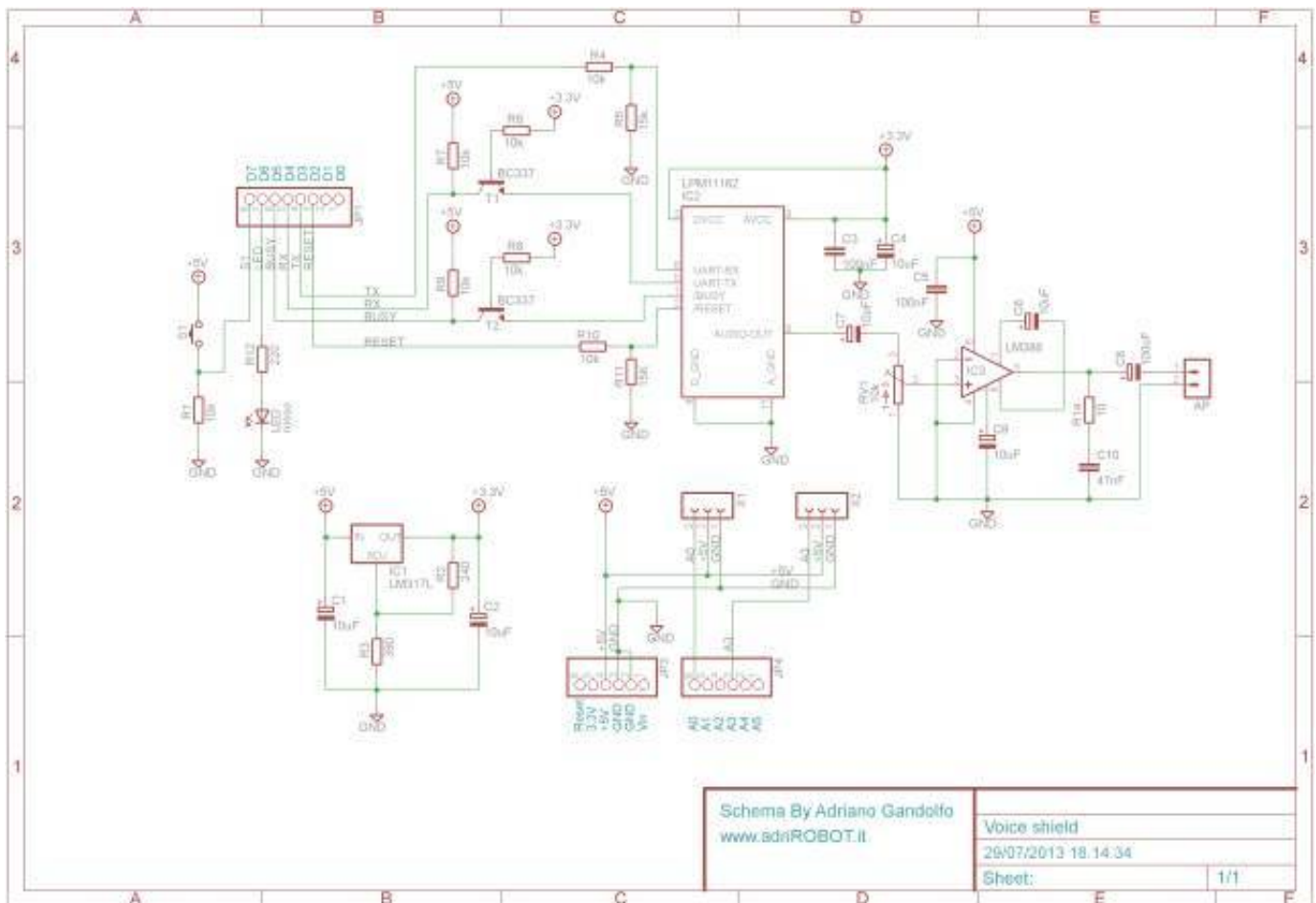
---



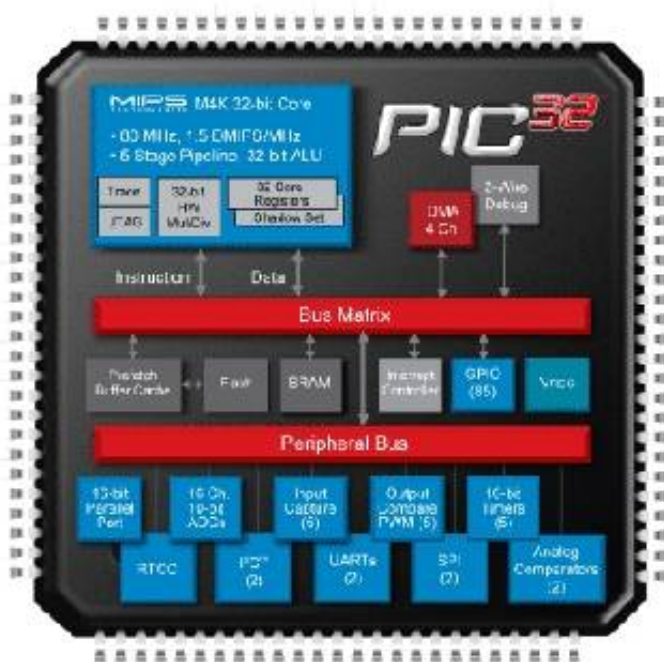
***Perché leggere un valore di tensione o una temperatura su un display quando possiamo conoscere il valore misurato grazie alla voce della nostra scheda? Vedremo insieme, con questo articolo, come dare una voce ad Arduino.***

Il progetto proposto permette di dare la parola ad Arduino <sup>[1]</sup>; sarà così facile, per esempio, sentire il valore letto tramite una porta analogica attraverso l'altoparlante connesso alla scheda. Il tutto è possibile utilizzando uno speciale shield su cui è installato un modulo sonoro **LPM11162** con cui è possibile memorizzare e riprodurre file audio in formato wave. Lo shield misura 43x59 mm e, su di esso, sono presenti uno stadio alimentatore, un'interfaccia per l'adattamento delle tensioni di comunicazione tra il processore Arduino e il modulo LPM11162, uno stadio amplificatore e, infine, una sezione sensori e una di I/O.

### Schema elettrico



Cuore dello shield è il Modulo **LPM1162**, che si presenta come un piccolo circuito (28x20 mm) dotato di due file da sei pin. Sul modulo sono presenti pochi componenti tra cui il processore, un PIC32MX320F, e una memoria flash da 2Mbyte che permette di memorizzare un massimo di 95 secondi di registrazione in formato wave 11 KHz-16bit mono (tempo massimo per tutti i file), mentre il numero massimo di file è di 512 e non c'è limitazione alle dimensioni di ciascun file.



Per gestire il modulo sono necessari pochi e semplici comandi seriali che vengono inviati

tramite le quattro linee RX, TX, Busy, Reset.

Per maggiori informazioni e per l'acquisto del modulo vedere il sito del produttore **LP Elettronica** [2]

## Descrizione del circuito

Per la descrizione del circuito partiamo dallo stadio di alimentazione del modulo **LPM11162** che necessita della tensione di 3,3 V, senza utilizzare quella fornita da Arduino.

Nel nostro caso, si utilizza la tensione derivata da quella a +5V che è ridotta utilizzando un classico regolatore **LM317LZ** (corrente massima fornita 100mA), il valore della tensione è impostato tramite le due resistenze da 240Ω e 390Ω.

Dato che il modulo LPM11162 funziona con una tensione di alimentazione di +3,3 V, anche le comunicazioni seriali e i segnali di reset e busy sono e devono essere a questa tensione. Visto che Arduino ha una logica a +5V, è necessario adattare i livelli uscenti ed entranti dal modulo. Per quelli entranti (RX - Reset), è sufficiente un partitore resistivo 10K/15K che adatta la tensione d'uscita 5 V di Arduino ai 3,3 V con cui il modulo LPM11162 è alimentato. Per quelli uscenti, si sono utilizzati due transistor **BC337** che traslano il livello d'uscita 3,3 V del modulo audio ai 5 V di Arduino.

Per amplificare il segnale audio analogico in uscita dal pin 9 del modulo LPM11162, è utilizzato un integrato amplificatore, in questo caso un **LM386**. Questo circuito integrato che si presenta in un contenitore plastico ad 8 pin (DIP8), è molto diffuso perché con pochi componenti esterni è in grado di pilotare direttamente un piccolo altoparlante: con 5 V di alimentazione riesce a fornire una potenza di circa 300 mW su un altoparlante da 8 Ohm.

Per quanto riguarda il guadagno, questo è impostato a 200 tramite il condensatore C6 da 10 μF connesso ai pin 1 ed 8.

Il condensatore, infatti, influisce su questo valore dal momento che esso funge da bypass per il resistore e la componente resistiva in gioco è quella della impedenza complessa che varia come  $1/C$ .

La regolazione del volume è ottenuta con il trimmer RV1 da 10 kOhm, il condensatore da 47 nF e la resistenza da 10 Ω in serie sull'uscita dell'amplificatore, sono necessari per rendere stabile l'amplificatore come indicato nel Datasheet. In serie all'altoparlante è presente il condensatore di disaccoppiamento C9 che costituisce un blocco per la componente in continua evitando che la medesima venga inviata all'altoparlante.

Per il collegamento dei sensori sono presenti due connettori, **X1** connesso all'ingresso Analogico A0, e **X2** connesso all'ingresso A3.

Abbiamo poi altre due possibilità di interagire con la scheda, rappresentati da **un pulsante** connesso alla porta digitale D7 e da **un led** dotato di resistenza limitatrice connesso alla porta digitale D6.

## Connettori per sensori

Ai connettori X1 e X2, è possibile collegare sensori con uscita analogica come un sensore di temperatura LM35; questo si presenta con tre terminali: uno per l'alimentazione, uno di massa e uno per l'uscita della tensione proporzionale alla temperatura rilevata che è pari a 10 mV per ogni grado centigrado, ed è calibrato in gradi Celsius. Per trasformare la tensione letta si può utilizzare la formula:

$$\text{temp} = (5.0 * \text{analogRead}(\text{tempPin}) * 100.0) / 1024$$

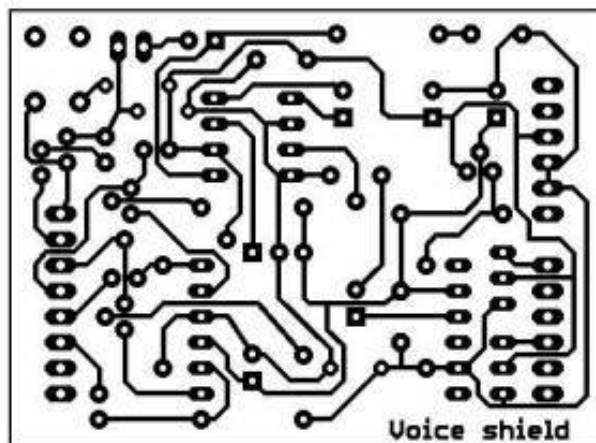
Si consideri che, con un campionamento a 10 bit, Arduino ha una risoluzione in tensione di circa 5 mV, e considerando che **LM35** fornisce 10 mV per ogni grado centigrado, la massima precisione che si può ottenere, è di mezzo grado.

Altro tipo di dispositivo collegabile al connettore è un potenziometro; si possono utilizzare indifferentemente i potenziometri in contenitore rotativo o lineare.



Anche in questo caso per la lettura della porta analogica, si userà il comando **analogRead()** che converte la tensione d'ingresso da 0 a 5 volt, in un valore digitale. Ruotando l'albero del potenziometro, si cambia la quantità di resistenza su entrambi i lati del pin centrale del potenziometro. Questo cambia la resistenza relativa tra il pin centrale e i pin esterni, dando una tensione diversa all'ingresso analogico. Quando l'albero è girato completamente in una direzione, non vi è resistenza tra il pin centrale e il pin collegato a GND. La tensione al pin centrale è quindi 0 volt. Quando l'albero è girato tutto nella direzione opposta, non vi è resistenza tra il pin centrale e il pin collegato a +5 volt. La tensione al pin centrale è quindi di 5 volt. Perciò analogRead() restituirà un numero tra 0 e 1023 che è proporzionale alla quantità di tensione applicata al pin analogico.

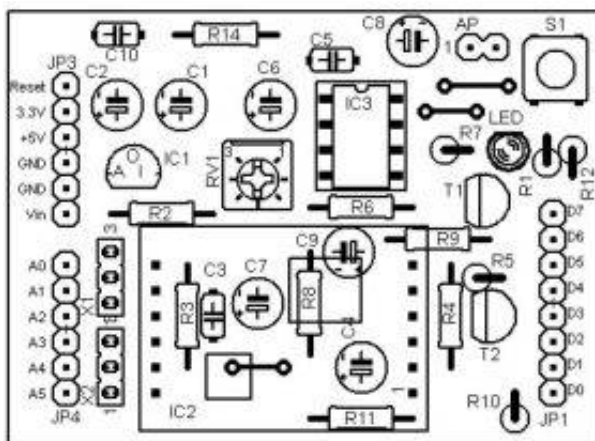
## Circuito stampato



L'immagine vista riporta la traccia del circuito stampato, questo è del tipo monofaccia e sono previsti solamente tre ponticelli.

### Piano di montaggio

Di seguito invece è visibile l'immagine del piano di montaggio.



### Lista Parti

## Part list Voice shield con LMP11162

n°	Q.tà	Riferimento	Descrizione	Part
1	6	C1,C2,C4,C6,C7,C9	Condensatore	10 µF elettrolitico
2	2	C3,C5	Condensatore	100 nF multistrato
3	1	C8	Condensatore	100 µF elettrolitico
4	1	C10	Condensatore	47nF poliestere
5	1	IC1	Regolatore di tensione	LM317L
6	1	IC2	Modulo audio	LPM11162
7	1	IC3	Amplificatore	LM386
8	2	T1,T2	Transistor	BC337
9	1	LED	Led	Led rosso Ø 3mm
10	7	R1,R4,R6,R7,R8,R9,R10	Resistenza	¼W 10kΩ
11	1	R2	Resistenza	¼W 240Ω
12	1	R3	Resistenza	¼W 390Ω
13	2	R5,R11	Resistenza	¼W 15kΩ
14	1	R12	Resistenza	¼W 220 Ω
15	1	R14	Resistenza	¼W 10 Ω
16	1	RV1	Trimmer	10kΩ
17	1	S1	Pulsante	Pulsante CS
18	2	X1,X2	Pinstrip	maschio 1x3
19	1	JP1	Pintrip	femmina 1x08
20	2	JP3,JP4	Pintrip	femmina 1x06
21	1	AP	Pintrip	maschio 1x2

## Realizzazione pratica

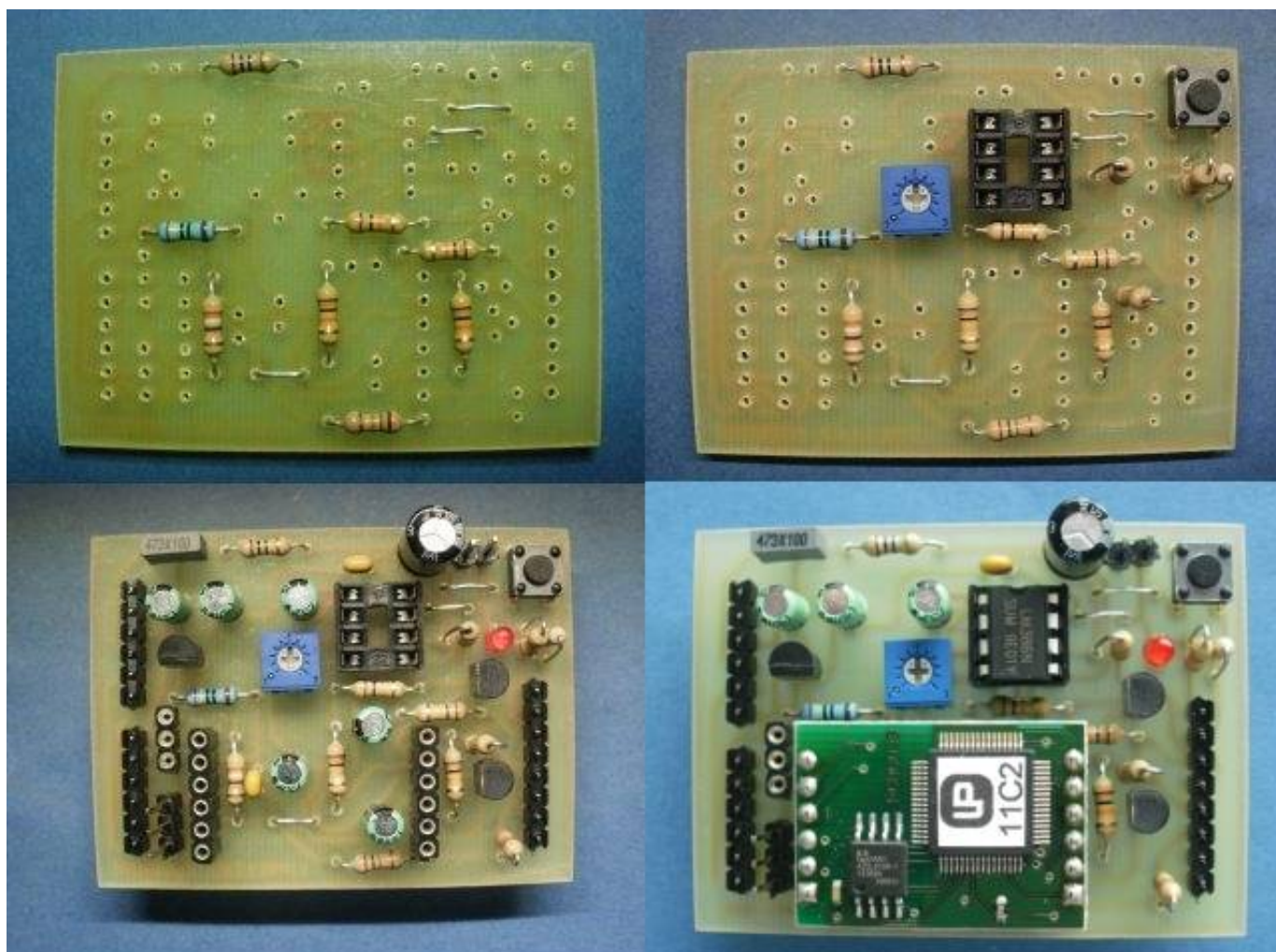
Per la costruzione della scheda, si procederà iniziando dalla realizzazione del circuito stampato, la traccia è scaricabile in scala 1:1 mediante il link (in fondo all'articolo).

Occorre ricordare che la stampa dovrà essere fatta deselezionando la funzione che adatta il foglio alla pagina. Per la sua realizzazione, si utilizzerà una basetta in vetronite (monofaccia) di dimensioni 43x59 mm, il metodo potrà essere quello della fotoincisione o del trasferimento termico utilizzando i cosiddetti fogli blu (**PRESS-N-PELL**), in questo caso ricordo che l'immagine delle tracce del circuito dovrà essere speculare. Una volta inciso il rame, si verificherà in controluce o mediante l'utilizzo di un multimetro, che non vi siano cortocircuiti soprattutto tra le piste più vicine. Si passerà quindi alla foratura della stessa, utilizzando principalmente una punta da 0,8 mm, mentre, se ne utilizzerà una da 1 mm per le pin strip. In seguito, si potrà passare al posizionamento e alla saldatura dei componenti seguendo lo schema di montaggio visto prima.

Per la saldatura, si utilizzerà un piccolo saldatore a punta fine, della potenza di circa 25 – 30 W. S'inizierà dai vari ponticelli, continuando con le resistenze; si potrà, quindi, procedere con il pulsante, lo zoccolo dell'integrato, i condensatori. Si concluderà con le pin strip e i connettori.

Terminata la saldatura, si potranno inserire gli integrati IC2 (Modulo LPM1162) e IC3

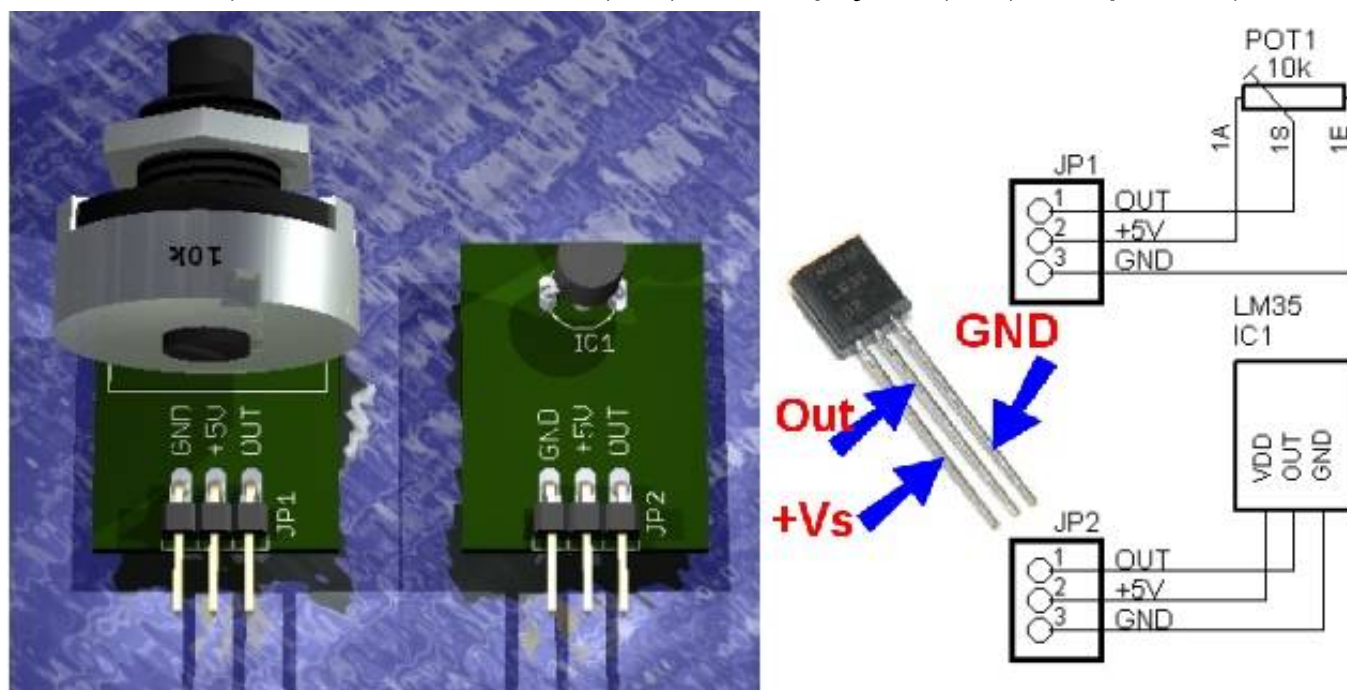
(LM386) nell'apposito zoccolo facendo attenzione alla tacca di riferimento.



## Sensori

Come detto sopra, alla scheda possono essere connessi vari sensori, la cui uscita sia analogica con un range di tensione compreso tra 0 e +5V.

Nella figura sotto, sono visibili due esempi: un potenziometro (utilizzato poi nel test) e un sensore di temperatura LM35.



## Programma ToolKit

Per trasferire i file wav nella memoria del modulo LPM1162, è disponibile presso il sito del produttore, uno speciale programma chiamato **LPM1162 ToolKit**:<sup>[3]</sup> con quest'applicazione per PC l'uso del modulo diventa semplice.

Con questo programma è possibile:

Convertire i file wave nel formato mono-11KHz-16bit.

Programmare i file wave sul LPM1162

Riprodurre i file wave programmati con regolazione del volume.

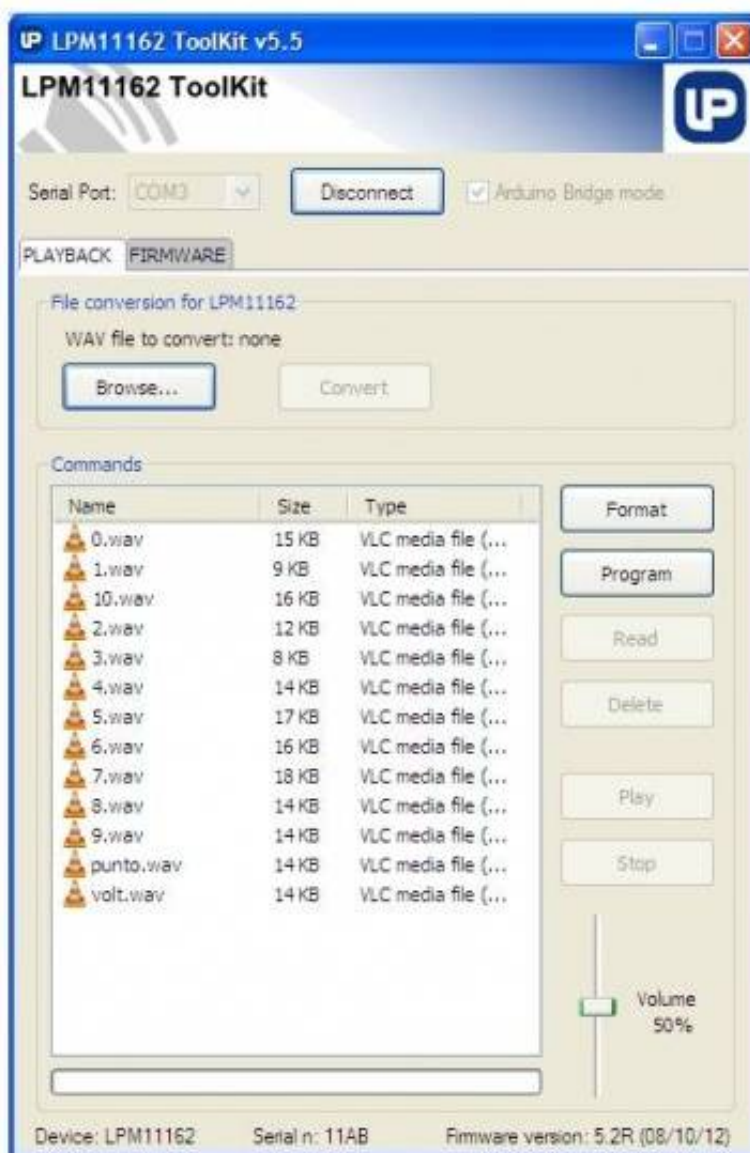
Verificare la versione di firmware dei moduli LPM1162.

Aggiornare il firmware all'ultima versione Firmware.

Per la programmazione tramite la scheda Arduino, è disponibile la **modalità Bridge**:

Arduino diventa programmatore per il modulo audio LPM1162 e non sono necessari dispositivi esterni. Quindi il modulo LPM1162 è collegato ad Arduino che a sua volta è collegato al PC tramite una COM virtuale (connessione USB).





## Libreria Arduino

Per la gestione del modulo, è disponibile, sempre presso il sito del produttore, la **LPM1162 Arduino Library** <sup>[4]</sup> che è una libreria open-source per il controllo dei moduli audio LPM1162 con Arduino.

La libreria fornisce alcune semplici funzioni e invia comandi seriali al modulo audio per riprodurre i file programmati. Sono disponibili le seguenti funzioni:

### **begin(baudrate)**

Configura la libreria e l'hardware per comunicare con il modulo audio LPM1162 con il baudrate selezionato, tipicamente 9600.

### **end()**

All'opposto di `begin()` rilascia interamente le risorse hardware e software impegnate per la comunicazione con il modulo audio LPM11162.

### **play(fileName)**

Inizia la riproduzione del file WAV il cui nome è specificato come parametro della funzione.

### **stop()**

Interrompe la riproduzione del file WAV che è in corso.

### **volume(volume)**

Regola il livello del volume di riproduzione audio passando un numero tra 0 e 100 come parametro per indicare il volume.

### **isBusy()**

Legge lo stato del pin /BUSY del modulo LPM11162. Questa informazione è utile per sapere se la riproduzione di un file WAV è ancora in corso.

### **reset(resetLevel)**

Pilota direttamente il pin /RESET del modulo LPM11162.

### **synch()**

Verifica la presenza del collegamento con il Toolkit per la programmazione dei file WAV sul modulo LPM11162.

### **bridge()**

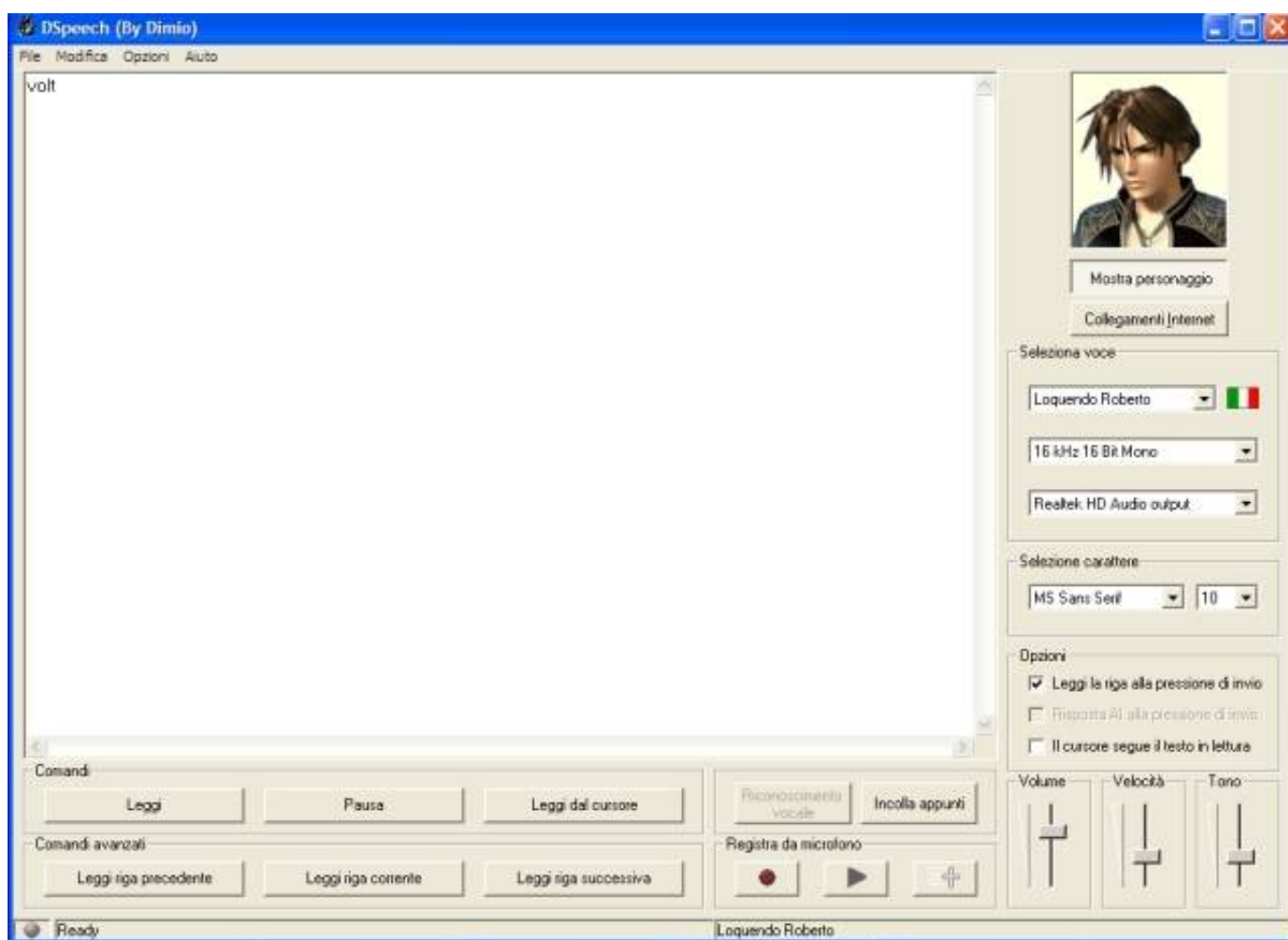
Con questa funzione la libreria Arduino entra in modalità Bridge.

## **Programma per creazione file wav**

Se, oltre ai numeri e alle parole presenti nel programma demo, si vogliono registrare altri numeri o parole da pronunciare, si può utilizzare un qualsiasi programma di registrazione audio, ma utilizzando un programma di **TTS** cioè Text To Speech tutto si semplifica poiché è sufficiente scrivere quello che vogliamo far pronunciare e il programma, utilizzando la sintesi vocale che è la tecnica per la riproduzione artificiale della voce umana, creerà per voi il file che potrà essere salvato in formato wav. Quello che consiglio si chiama **DSpeech** è stato scritto da Dimitrios Coutsoumbas (Dimio). Il programma che tra l'altro è freeware, oltre alle funzioni TTS, possiede le funzionalità di **ASR** (Automatic Speech Recognition) integrate. E' cioè, in grado di leggere ad alta voce il testo scritto e di scegliere le frasi da pronunciare a seconda delle risposte vocali dell'utente.

Non ha bisogno di essere installato e occupa poca memoria. Permette di salvare l'output sotto forma di un file Wav, Mp3, Aac, Wma o Ogg.

Per scaricarlo e per le istruzioni di come installarlo vedere il [seguito link](#). [5]



## Programma di prova

Segue un esempio di programma, in questo caso è letto il valore di tensione variato tramite un potenziometro collegato al connettore X2. Il valore di tensione è poi “pronunciato” tramite l’altoparlante connesso all’uscita. La procedura di caricamento è la seguente.

Lanciare l’IDE e trasferire il programma nella memoria di Arduino. A caricamento terminato, lanciare il programma ToolKit e utilizzando la funzione bridge trasferire nella memoria del modulo LPM11162 i file wav presenti nel file zip allegato a quest’articolo, essi sono riferiti ai numeri, la parola “punto” e “volt”. Sconnettere il programma toolkit e resettare la scheda Arduino. A questo punto ruotando l’alberino del potenziometro, si udrà dall’altoparlante il valore di tensione letta

```
#include <SoftwareSerial.h>

#include <LPM11162.h>

#define LPM11162_RX      3 //Arduino TX (out)

#define LPM11162_TX      4 //Arduino RX (in)

#define LPM11162_RESET  2 //(out)

#define LPM11162_BUSY    5 //(in)

#define pin_tensione    A3 // Pin analogico lettura

LPM11162 audio(LPM11162_RX, LPM11162_TX, LPM11162_RESET, LPM11162_BUSY);

int pinval = 0;

int oldpinval =0;

void setup()

{
```

```
Serial.begin(9600);

if ( audio.synch() )

{

    audio.bridge();

}

audio.begin(9600);

}

void loop()

{

    pinval = analogRead(pin_tensione); // Legge il pin analogico

    if(pinval != oldpinval)

    {

        oldpinval = pinval;

        SayVolts(pinval); // Converte in una tensione e dice il valore

    }

    delay (2000);

}

void SayVolts(int volts)

{

    int BigNumber = 0;
```

```
int SmallNumber = 0;

BigNumber = abs(volts/204); // Converte il valore analogico in tensione

Serial.println (BigNumber);

SmallNumber = volts - abs(BigNumber*204); // Converte i decimali ad una sola
cifra

SmallNumber = abs(204-SmallNumber)/2;

SmallNumber = abs(100-SmallNumber)/10;

Serial.println (SmallNumber);

if (BigNumber != 0)

{

String intero= String (String (BigNumber) + ".wav");

char frase_array[20];

intero.toCharArray( frase_array, sizeof(frase_array) );

audio.play(frase_array);

while( audio.isBusy() );

delay (1000);

}

audio.play("punto.wav"); // Dice "punto"

while( audio.isBusy() );

delay (1000);
```

```
String decimale = String (String (SmallNumber) + ".wav");

char frase_array[20];

decimale.toCharArray( frase_array, sizeof(frase_array) );

audio.play(frase_array);

while( audio.isBusy() );

delay (1000);

audio.play("volt.wav");          // Dice "volt"

while( audio.isBusy() );

delay (1000);

}
```

## Conclusioni

Il programma presentato è molto semplice, ma è possibile crearne altri, per esempio uno che leggendo la tensione in uscita dal sensore di LM35, converte il valore letto e fa pronunciare quest'ultimo tramite l'altoparlante.

### Documentazione completa Open Source

La documentazione completa è disponibile, a questo link: [voice shield arduino.zip](#) [6]

Schema elettrico PDF

Layout PDF

Circuito stampato PDF

Schema elettrico e layout sorgente EAGLE

Sketch per Arduino

File wav da memorizzare sul Modulo LPM11162.

---

Article printed from Elettronica Open Source: <https://it.emcelettronica.com>

URL to article: <https://it.emcelettronica.com/costruiamo-voice-shield-far-parlare-arduino-progetto-completo-open-source>

URLs in this post:

[1] Arduino: <https://it.emcelettronica.com/arduino>

[2] **LP Elettronica**: <http://www.lpelettronica.it/Modulo-audio-LPM11162.html>

[3] **LPM11162 ToolKit**;: <http://www.lpelettronica.it/LPM11162-Toolkit.html>

[4] **LPM11162 Arduino Library** : <http://www.lpelettronica.it/LPM11162-Arduino-Library.html>

[5] seguente link.: <http://www.guidami.info/2010/02/dspeech-il-pc-parla-il-testo-che-legge.html>

[6] voice\_shield\_arduino.zip: [https://it.emcelettronica.com/wp-content/uploads/files/voice\\_schild\\_arduino.zip](https://it.emcelettronica.com/wp-content/uploads/files/voice_schild_arduino.zip)

Copyright © 2017 Elettronica Open Source. All rights reserved.