

Date voce ai vostri progetti

Posted By [Adriano Gandolfo](#) On 2 marzo 2017 @ 1:00 In [Arduino](#) | [4 Comments](#)



Si ha l'esigenza a volte, nelle proprie applicazioni elettroniche, di avere oltre che un segnale visivo dato da un led, di un segnale sonoro, ma non il semplice suono fornito da un buzzer ma un vero e proprio messaggio. In quest'articolo vedremo invece l'utilizzo del modulo WTV020-SD, poco costoso, che conserva le informazioni su una memoria MicroSD analoga a quelle presenti nei moderni smartphone. Questo modulo con pochi componenti esterni, permette di riprodurre suoni (o voci) memorizzati.

Descrizione del modulo

Il modulo WTV020-SD, si trova in due versioni (Figura 1): **WTV020-SD-20S** o **WTV020-SD-16P**.



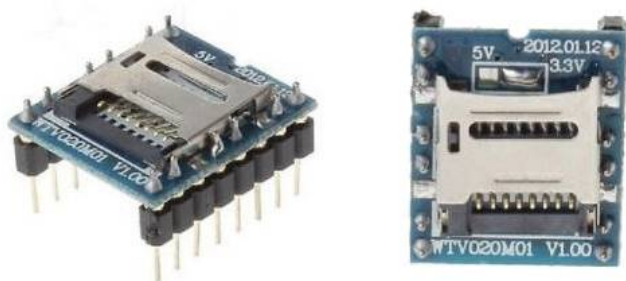
[1]

Figura 1: Differenti versioni del modulo

Il primo usa schede SD standard ed ha un connettore bifilare a passo ridotto, mentre il secondo, più comune, usa una MicroSD e presenta un package DIL a 16 piedini.

Questi moduli sono facilmente reperibili in rete e hanno un costo molto contenuto, intorno ai 5 Euro.

Il modello che sarà analizzato è il WTV020-SD-16P (Figura 2), che come abbiamo detto, consente di leggere file audio da una scheda microSD e di riprodurli con un semplice altoparlante.

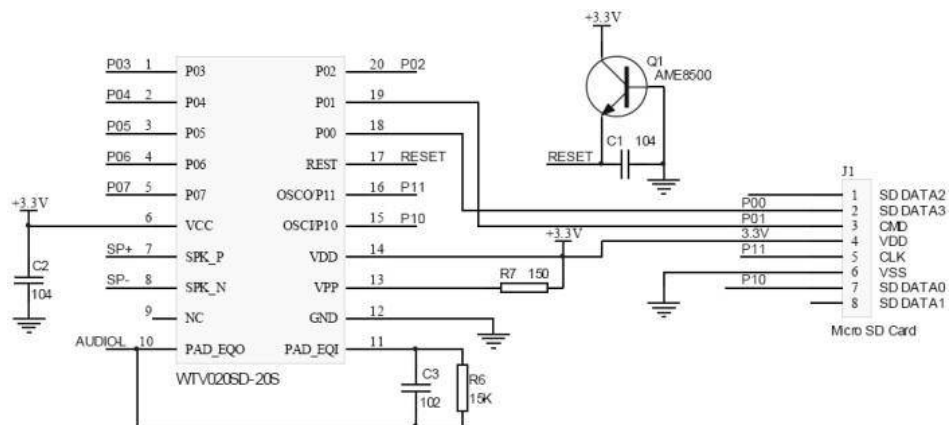


[2]

Figura 2: Modulo WTV020-SD-16P

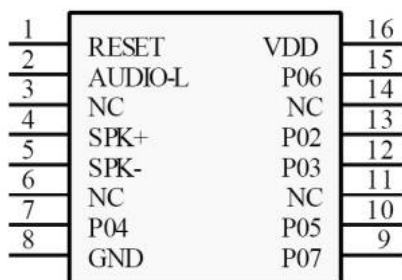
Il modulo, può essere utilizzato **in modo autonomo con dei pulsanti esterni** oppure associato a un controllore per la gestione della comunicazione seriale su 2 linee di input/output, utilizzando per esempio [Arduino](#) ^[3]. Il modulo è molto compatto poiché misura solo 20x18 mm e il suo peso è di soli 5 gr, la sua impostazione "a circuito integrato" permette l'installazione su zoccolo o su una millefori.

Il cuore del circuito del modulo (Figura 3) è l'integrato WTV020SD-20S prodotto dalla Waytronic Electronics Co.,Ltd, nella figura 4 è riportata la piedinatura del modulo.



[4]

Figura 3: Circuito elettrico del modulo



PIN	SYS.	DESCRIPTION	FUNCTION
1	RESET	RESET	Reset pin
2	AUDIO-L	DAC+	DAC audio output(+) to amplifier
3	NC	NC	NC
4	SP+	PWM+	PWM audio output to speaker
5	SP-	PWM-	PWM audio output to speaker
6	NC	NC	NC
7	P04	K3/A2/CLK	Key /CLK in two line serial
8	GND	GND	Address pin
9	P07	K5/A4/SBT	Key
10	P05	K4/A3/DI	Key /DI in two line serial
11	NC	NC	NC
12	P03	K2/A1	Key
13	P02	K1/A0	Key
14	NC	NC	NC
15	P06	BUSY	BUSY pin
16	VDD	VDD	Power input

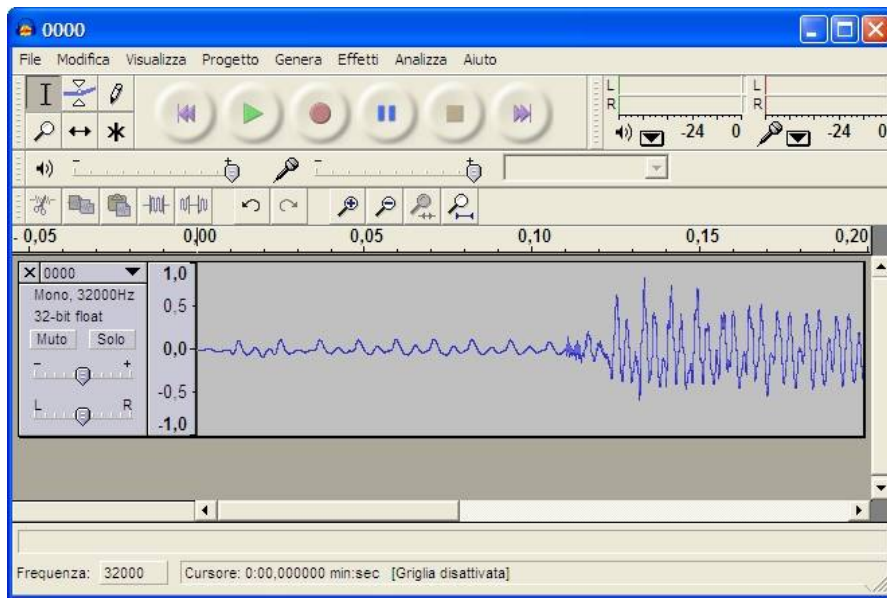
[5]

Figura 4: Piedinatura del modulo

I file audio

Per riprodurre i brani, è necessario registrare dei file audio compatibili con il modulo e caricarli nella scheda microSD, questo perché il nostro modulo non è in grado di riprodurre tutti i formati audio ma solo files con estensione .ad4

I file con estensione .ad4 compatibili con il modulo, sono file a 4 bit con frequenze di campionamento da 6 kHz fino a 32 kHz, codificati in **ADPCM** (**A**daptive **D**ifferential **P**ulse **C**ode **M**odulation) che è un'estensione della codifica digitale PCM di base, usata largamente nel campionamento, quantizzazione e codifica di segnali audio, soprattutto in campo di telefonia fissa. **Per realizzare i file audio si potrà utilizzare uno dei tanti sound editor** che si trovano gratuitamente su Internet, come **Audacity** [6] (Figura 5) che ha il vantaggio di essere multiplatforma.

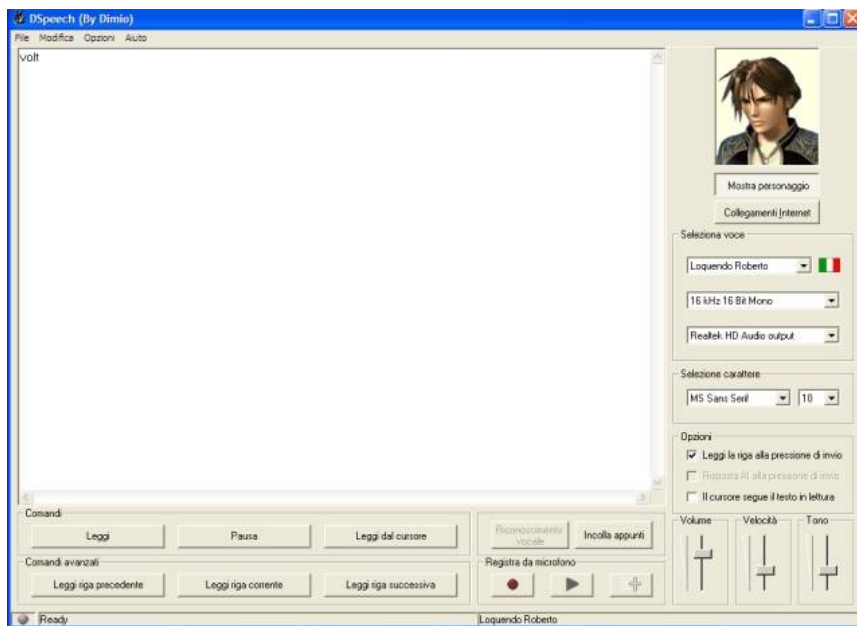


[7]

Figura 5: Il programma Audacity

Per creare file contenenti parole, è più indicato utilizzare un programma di **TTS** cioè **Text To Speech** [8]; tutto si semplifica poiché è sufficiente scrivere quello che vogliamo far pronunciare e il programma, utilizzando la sintesi vocale, creerà per voi il file che potrà essere salvato.

Il programma che consiglio, si chiama **DSpeech** [9] (Figura 6), è freeware ed è stato scritto da Dimitrios Coutsoumbas (Dimio). Non ha bisogno di essere installato e occupa poca memoria.



[10]

Figura 6: Il programma DSpeech

Permette di salvare l'output sotto forma di un file Wav, Mp3, Aac, Wma o Ogg. Sarà possibile partire da file audio già pronti con estensioni come **MP3 o WAV**.

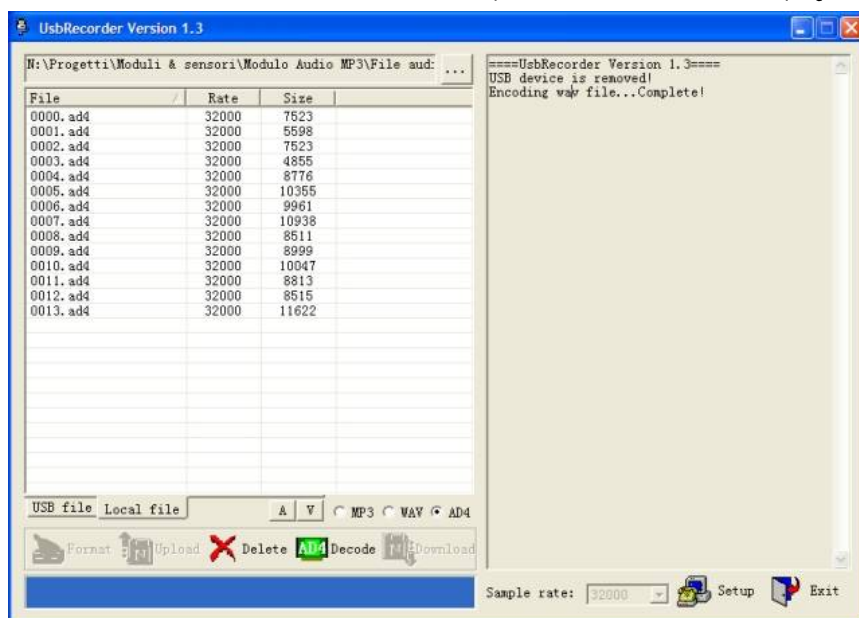
Per ottenere la migliore qualità audio .ad4, è meglio rispettare però, queste caratteristiche di partenza:

Il suono, se stereo, dovrebbe essere convertito in mono.

La frequenza di campionamento deve essere impostata a 32000Hz.

La profondità di bit deve essere impostata a 16

Si tratta ora di convertire il nostro file nel formato *.ad4 (Figura 7), per realizzare questo tipo di file esistono in rete un paio di applicazioni dedicate che sono liberamente scaricabili: [4D-SOMO-Tool](#) [11], oppure [AD4CONVERTER](#) [12].



[13]

Figura 7: Il programma UsbRecorder

Bisogna ricordare che i files generati, devono avere il nome nel formato XXXX.ad4, dove le X stanno ad indicare delle cifre in numerazione decimale, da 0000 a 0511; per cui sulla microSD dovremo avere qualcosa del genere:

0000.ad4
 0001.ad4
 0002.ad4
 0003.ad4

 0511.ad4.

L'ordine dei files sarà quello che avremo in riproduzione quando invieremo il comando "next".

L'installazione e l'utilizzo dei programmi è molto semplice, una volta evidenziati i file da convertire, basterà premere il tasto **AD4 Decode** e in poco tempo i file saranno convertiti.

La scheda di memoria SD

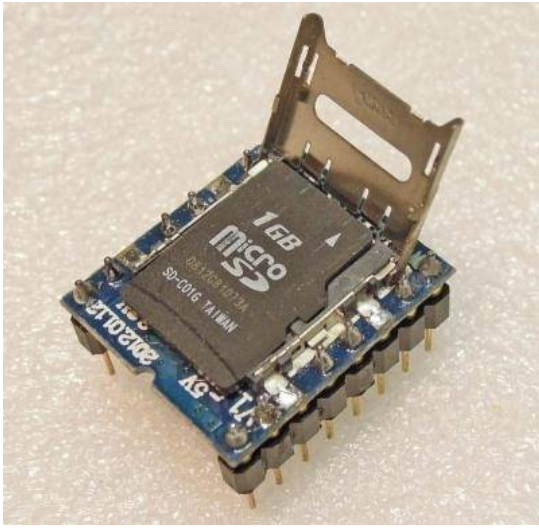
Come prima cosa, dovremo procurarci una scheda MicroSD in quanto questa non viene fornita col modulo. Non tutte le schede sono compatibili con il modulo, in particolare NON sono compatibili le schede SDHC (Figura 8).



[14]

Figura 8: Tipi di scheda di memoria SD

La scheda va poi formattata in formato FAT32, altri tipi di file system non sono riconosciuti dal modulo. Ora che la microSD è pronta, dobbiamo inserirla nello slot del modulo (Figura 9), che non è del tipo a molla, ma va prima aperto facendo slittare indietro il coperchio metallico e poi aprendolo come uno sportello, lo slot previene la fuoriuscita accidentale della MicroSD.



[15]

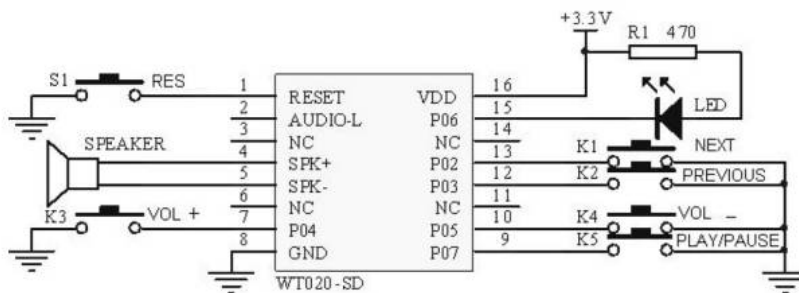
Figura 9: Inserimento della memoria MicroSD all'interno del modulo

Utilizzo del modulo

Se leggiamo il [datasheet](#) ^[16], troviamo che il modulo può lavorare in svariate modalità: MP3, Key (3 group of voice), Key (5 group of voice), Loop play, Two line serial mode; non tutte sono disponibili in quanto risultano opzionali, e vanno ordinate al fabbricante, ovviamente in grandi quantità. Per quanto riguarda i moduli commercializzati, questi prevedono due modalità: **MP3** e **Two line serial mode**.

Modalità MP3 (stand-alone)

In questa modalità, il modulo non ha bisogno di alcun micro per poter riprodurre i files audio e risulta completamente autonomo. E' sufficiente collegare i pulsanti per il controllo, un piccolo altoparlante, un led e una resistenza (Figura 10).



[17]

Figura 10: Schema di base (tratto dal datasheet)

I piedini utilizzabili in questa modalità sono descritti in tabella 1.

Pin	Funzione	Descrizione
1	Reset	Con un pulsante collegato verso massa riporta a 0 il contatore dei brani.
2	Audio-L	Uscita audio analogica a basso livello.
4-5	SPK+ e SPK-	Uscita PWM, a questi due pin e' possibile collegare direttamente un piccolo altoparlante da 8 Ohm/0,5W
7	Vol+	Collegando un pulsante collegato verso massa possiamo aumentare il volume da 0 (mute) a 7
8	massa	Negativo dell'alimentazione
9	Play/Pause	Collegando un pulsante verso massa possiamo riprodurre o interrompere il brano corrente
10	Vol -	Collegando un pulsante collegato verso massa possiamo diminuire il volume da 7 a 0 (mute).
12	Previous	Con un pulsante collegato verso massa possiamo passare al brano precedente
13	Next	Con un pulsante collegato verso massa possiamo passare al brano successivo

15	Busy	A questo pin possiamo collegare un LED e relativa resistenza da 470Ω per indicare quando un brano è in riproduzione
16	+3,3 V	Non applicate i 5 Volt al modulo pena la distruzione della scheda MicroSD.
3,6,11,14	NC	Pin non utilizzati

Tabella 1: Pin WT020-SD

Per la realizzazione del circuito (Figura 11 e 12) sono necessari i seguenti componenti:

Una breadboard.

Il modulo WTV020-SD completo di memoria SD.

Un alimentatore da breadboard con uscita 3,3V.

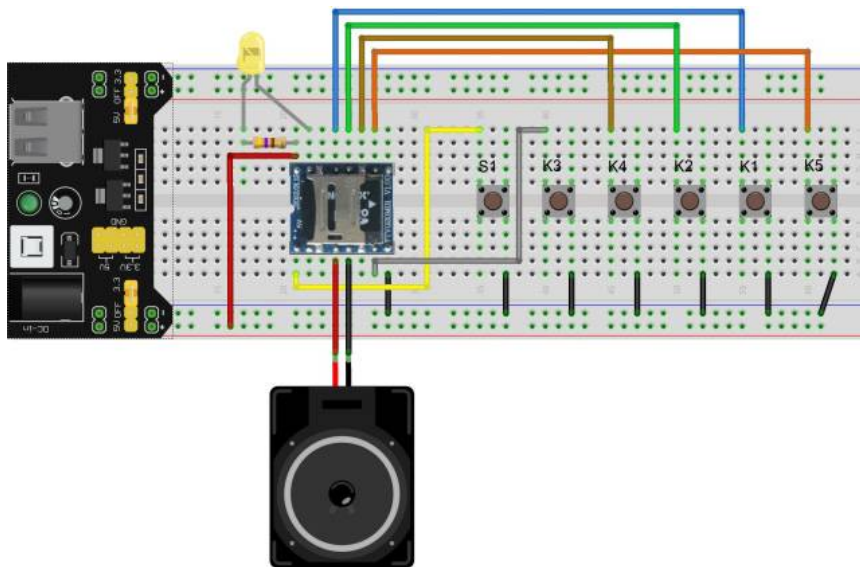
Un altoparlante 8 ohm – 0,5W.

6 pulsanti NA da circuito stampato.

N° 1 led giallo.

1 resistenza da 470 ohm.

Cavi vari.

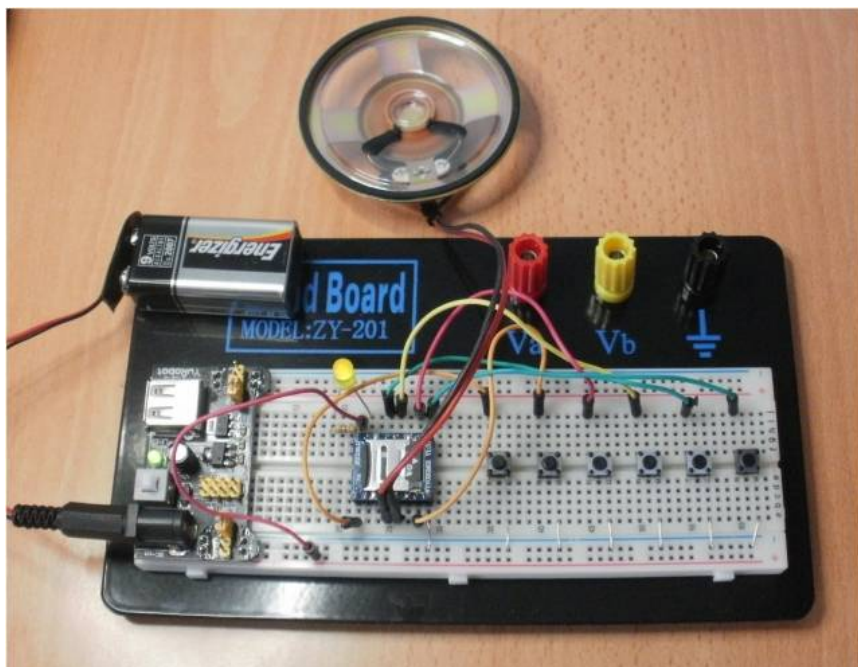


[18]

Figura 11: Schema dei collegamenti

Istruzioni d'uso:

- K1 - permette di mandare in esecuzione il brano seguente a quello in esecuzione.
 - K2 - permette di mandare in esecuzione il brano precedente a quello in esecuzione.
 - K3 - Aumenta il volume.
 - K4 - Diminuisce il volume.
 - K5 - Manda in esecuzione il brano o lo mette in pausa.
 - S1 - Resetta il modulo in caso di blocchi.
- L'accensione del led segnala che il brano è in esecuzione.



[19]

Figura 12: Il prototipo

Modalità Two line serial mode tramite Arduino

In questa modalità, è possibile effettuare la gestione del modulo tramite comandi inviati da Arduino, che è facilitata dalla disponibilità di una libreria freeware in grado di controllare con pochissimi comandi questo modulo. **Per utilizzare questa libreria occorre scaricare il file [Wtv020sd16p](#)** [20], poi decomprimerlo e copiare la cartella risultante, nella cartella Library dell'IDE Arduino. Una volta riavviato l'ambiente di programmazione di Arduino, troveremo anche la libreria chiamata **Wtv020sd16p** e negli esempi, uno sketch completo di tutti i comandi utilizzabili.

Utilizzo della libreria

Per utilizzare la libreria occorrerà il comando **#include**:

```
#include <Wtv020sd16p.h>
```

Successivamente si assegnerà il numero dei pin utilizzati per le varie funzioni :

```
int resetPin = 2; // Reset pin 2.
int clockPin = 3; // Clock pin 3.
int dataPin = 4; // Data pin 4.
int busyPin = 5; // Busy pin 5.
```

Quindi si potrà creare l'istanza:

```
Wtv020sd16p wtv020sd16p (resetPin, clockPin, dataPin, busyPin);
```

Sono disponibili i seguenti comandi:

wtv020sd16p.reset(): esegue il reset del modulo

wtv020sd16p.setVolume(X): imposta il volume dell'uscita collegata all'altoparlante, dove X è un valore tra 0 e 7.

wtv020sd16p.asyncPlayVoice(X): manda in esecuzione il file relativo, dove X è un valore tra 0 e 511.

wtv020sd16p.pauseVoice(): arresta la riproduzione del brano, o riprende il brano dopo una pausa.

wtv020sd16p.mute(): interrompe il suono durante la riproduzione.

wtv020sd16p.unmute(): riattiva il suono durante la riproduzione

wtv020sd16p.stopVoice(): interrompe la riproduzione del file corrente.

Realizzazione di un voltmetro parlante

Per testare il nostro modulo, proveremo la Modalità Two line serial mode tramite Arduino.

La nostra applicazione permetterà di misurare il valore di una tensione continua, il cui valore è variato tramite il potenziometro collegato alla porta analogica A0. Il valore di tensione è poi "pronunciato" tramite l'altoparlante connesso all'uscita. Per la realizzazione del circuito (tabella 1) sono necessari (figura 13-15) i seguenti componenti:

Una breadboard.

Il modulo WTV020-SD completo di memoria SD.

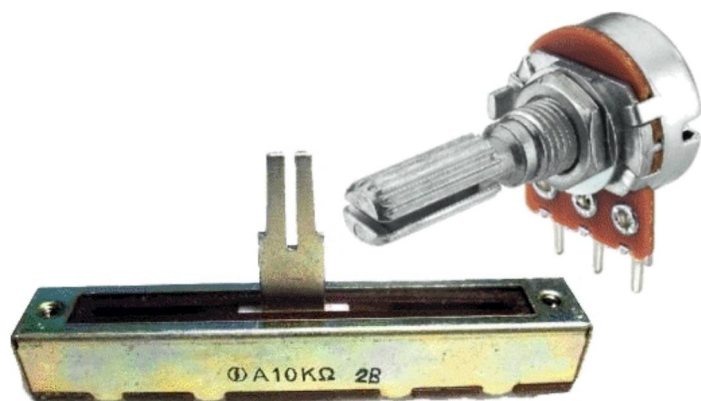
Una scheda [Arduino UNO](#). [3]

Un altoparlante 8 ohm - 0,5W.

Un potenziometro del valore di 10 kΩ (si possono utilizzare indifferentemente i potenziometri in contenitore rotativo o lineare - Figura 13).

Questo sarà collegato in modo che un lato sia connesso al pin +5V, il lato opposto al pin GND e il pin centrale al pin d'ingresso analogico A0.

Cavi vari.

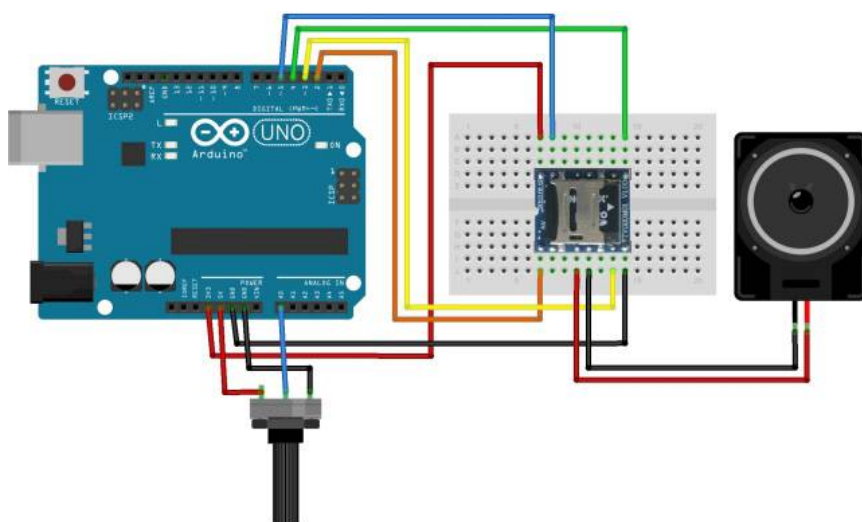


[21]

Figura 13: Tipologie di potenziometri - Lineare e rotativo

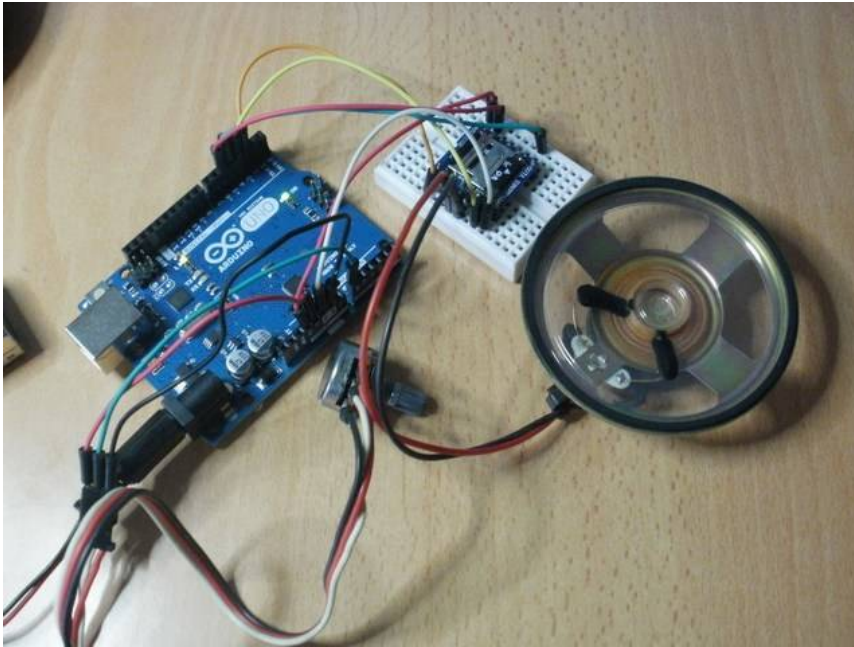
Pin Arduino	Pin Wtv020	Funzione	Descrizione
2	1	Reset	Reset del modulo
3	7	P04/Clock	CLOCK linea seriale
4	10	P05/Data	DATA linea seriale
5	15	P06 /Busy	BUSY
GND	8	massa	Negativo dell'alimentazione del modulo e uno dei pin laterali del potenziometro
-	4	SPK+	Polo positivo altoparlante
-	5	SPK-	Polo negativo altoparlante
3,3V	16	+3,3 Volt	Alimentazione modulo
5V		+5,0 Volt	Uno dei pin esterni del potenziometro
A0		Ingresso analogico	Pin centrale del potenziometro per la lettura della tensione

Tabella 1: Collegamenti



[22]

Figura 14: Schema dei collegamenti



[23]

Figura 15: Prototipo in fase di test

Il Programma

Segue il listato del programma, che può essere scaricato dal [seguente link](#) [24], la principale funzione utilizzata è 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, fornendo 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 completamente nella direzione opposta, non vi è resistenza tra il pin centrale e quello collegato a +5 volt.

La tensione al pin centrale è quindi di 5 volt. Dato che il convertitore analogico interno dell'Arduino UNO ha una risoluzione di 10 bit, **analogRead()** restituirà un numero compreso tra 0 e 1023, che è proporzionale alla quantità di tensione applicata al pin analogico.

Il programma convertirà il numero letto e farà in modo da richiamare i vari file presenti sulla scheda SD pronunciando il valore di tensione misurato.

Per fare questo sulla memoria SD sono salvati i file che contengono la pronuncia dei numeri da zero a nove, la parola punto e la parola Volt. Per il comando del modulo, sono utilizzate 4 porte digitali e l'ingresso analogico A0.

La procedura di caricamento del programma è la seguente:

trasferire nella memoria MicroSD i files audio, che possono essere scaricati dal seguente link [File audio](#) [25];
 eseguire l'IDE e trasferire il programma nella memoria di Arduino, verificando l'esistenza dell'apposita libreria;
 a questo punto, ruotando il potenziometro, sarà possibile ascoltare il valore della tensione pronunciato attraverso l'altoparlante.
 Di seguito il listato:

```
#include <Wtv020sd16p.h>
int resetPin = 2; // pin connesso a reset pin 1.
int clockPin = 3; // pin connesso a clock pin 7.
int dataPin = 4; // pin connesso a data pin 10.
int busyPin = 5; // pin connesso a busy pin 15.
#define pin_tensione A0 // Pin analogico lettura
Wtv020sd16p wtv020sd16p(resetPin, clockPin, dataPin, busyPin);
int pinval = 0;
int oldpinval = 0;
void setup()
{
  //Initializes the module.
  wtv020sd16p.reset();
  delay(1000);
  wtv020sd16p.setVolume(7);
}
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
  Serial.println (BigNumber); // analogico in tensione
  SmallNumber = volts - abs(BigNumber * 204); // Converte i decimali
  SmallNumber = abs(204 - SmallNumber) / 2; // ad una sola cifra
  SmallNumber = abs(100 - SmallNumber) / 10;
  Serial.println (SmallNumber);
  if (BigNumber != 0)
  {
    wtv020sd16p.asyncPlayVoice (BigNumber);
    delay (1000);
  }
  wtv020sd16p.asyncPlayVoice (12); // Pronuncia "Punto"
  delay (1000);
  wtv020sd16p.asyncPlayVoice (SmallNumber);
  delay (1000);
  wtv020sd16p.asyncPlayVoice (11); // Pronuncia "Volt"
  delay (1000);
}
```

Nel Video 1 è possibile vedere il funzionamento dell'intero prototipo.

Tester parlante con Arduino e modulo WTV020-SD-16P



Video 1: Funzionamento del prototipo

Conclusioni

Le potenzialità di questo modulo sono decisamente infinite. Si può andare dal campo automotive (**allarme antifurto, sensore di parcheggio, sistemi di navigazione GPS** ^[26], **chiusura centralizzata**), al campo della domotica, allarmi antifurto, strumenti medici, elettrodomestici (lavastoviglie, lavatrici, forni a microonde), dispositivi di intrattenimento o come aiuto all'apprendimento come audiolibri, dispositivi inerenti i mezzi di trasporto come caselli, parcheggi. Di seguito alcune idee:

Sicurezza: si può collegare ad Arduino un sensore di presenza che, attivandosi, faccia emettere un messaggio di avvertimento di tipo vocale.

Pubblicità: utilizzando un sensore sonar, è possibile rilevare la presenza di una persona davanti ad una vetrina, trasmettendo un messaggio pubblicitario.

Istruzione: in una mostra fotografica di animali, davanti ad ogni foto, si potrebbe inserire un tasto alla cui pressione il sistema potrebbe trasmettere il verso dell'animale raffigurato.

E voi, quale applicazione avete in mente ?

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

URL to article: <https://it.emcelettronica.com/date-voce-ai-vostri-progetti>

URLs in this post:

- [1] Image: <https://it.emcelettronica.com/wp-content/uploads/2016/05/wtv020sd-modelli.jpg>
- [2] Image: https://it.emcelettronica.com/wp-content/uploads/2016/05/wtv020sd-16p_foto.jpg
- [3] Arduino: <https://it.emcelettronica.com/category/arduino>
- [4] Image: <https://it.emcelettronica.com/wp-content/uploads/2016/05/schema.jpg>
- [5] Image: https://it.emcelettronica.com/wp-content/uploads/2016/05/wtv020sd-16p_pin1.jpg
- [6] Audacity: <http://www.audacityteam.org/>
- [7] Image: <https://it.emcelettronica.com/wp-content/uploads/2016/05/Audacity.jpg>
- [8] Text To Speech: <https://it.emcelettronica.com/speech-recognition-cosa-si-nasconde-dietro-il-riconoscimento-vocale>
- [9] DSpeech: <http://dimio.altervista.org/ita/>
- [10] Image: https://it.emcelettronica.com/wp-content/uploads/2016/05/arduino_voce_DSpeech.jpg
- [11] 4D-SOMO-Tool: <https://it.emcelettronica.com/wp-content/uploads/2016/05/4D-SOMO-Tool.zip>
- [12] AD4CONVERTER: <https://it.emcelettronica.com/wp-content/uploads/2016/05/AD4CONVERTER.zip>
- [13] Image: <https://it.emcelettronica.com/wp-content/uploads/2016/05/usbrecorder.jpg>
- [14] Image: https://it.emcelettronica.com/wp-content/uploads/2016/05/wtv020sd-16p_memorie.jpg
- [15] Image: https://it.emcelettronica.com/wp-content/uploads/2016/05/wtv020sd-16p_slot.jpg
- [16] datasheet: <https://it.emcelettronica.com/wp-content/uploads/2016/05/WTV020-SD.pdf>
- [17] Image: https://it.emcelettronica.com/wp-content/uploads/2016/05/wtv020sd-schema_stand_alone.jpg
- [18] Image: https://it.emcelettronica.com/wp-content/uploads/2016/05/MODULO_MP3_tasti_bb1.jpg
- [19] Image: https://it.emcelettronica.com/wp-content/uploads/2016/05/wtv020sd-16p_collegamenti1.jpg
- [20] Wtv020sd16p: <https://it.emcelettronica.com/wp-content/uploads/2016/05/Wtv020sd16p.zip>
- [21] Image: https://it.emcelettronica.com/wp-content/uploads/2016/05/arduino_voce_potenziometri.gif
- [22] Image: https://it.emcelettronica.com/wp-content/uploads/2016/05/MODULO_MP3_Arduino_Jettura_bb1.jpg
- [23] Image: https://it.emcelettronica.com/wp-content/uploads/2016/05/wtv020sd-16p_arduino-parlante1.jpg
- [24] seguente link: https://it.emcelettronica.com/wp-content/uploads/2016/05/voltmetro_parlante.zip
- [25] File audio: <https://it.emcelettronica.com/wp-content/uploads/2016/05/File-audio.zip>
- [26] GPS: <https://it.emcelettronica.com/?s=gps>

Copyright © 2017 Elettronica Open Source. All rights reserved.