



Laboratorio

Android: microfono

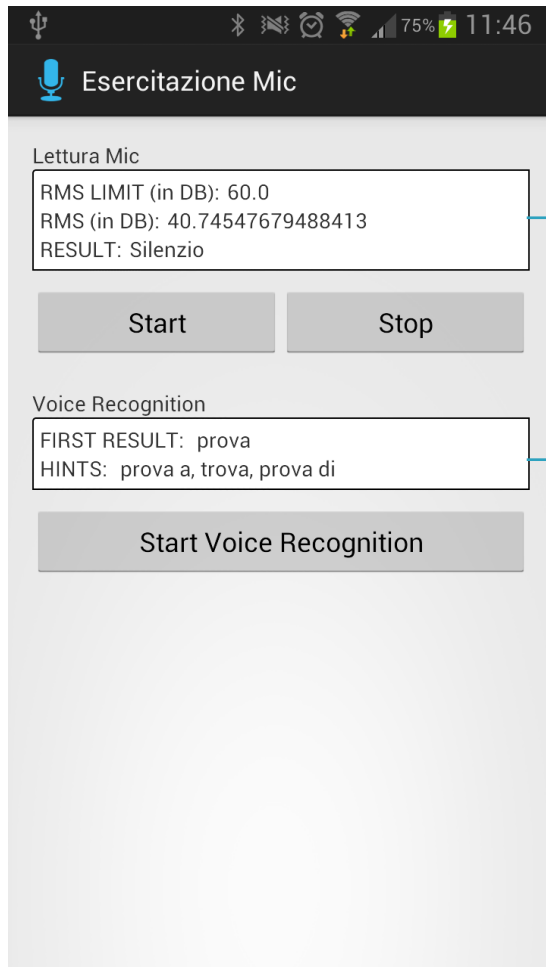
Corso di Smart City e Tecnologie Mobili

Università di Bologna

Dipartimento di Informatica — Scienza e Ingegneria

Luca Calderoni, Andrea Cirri, Dario Maio

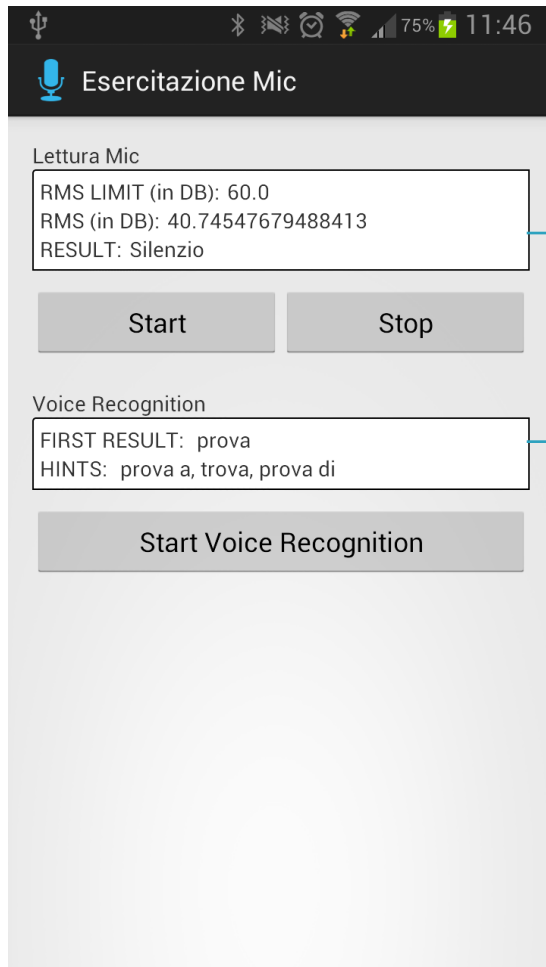
Obiettivi



Visualizzare il valore in Decibel relativo al segnale percepito dal microfono. Il valore deve essere calcolato come RMS (*Round Mean Square*) dei valori presenti in un singolo intervallo di campionamento (buffer di acquisizione del segnale). Il campo RESULT deve indicare la presenza di silenzio o rumore rispetto alla soglia RMS LIMIT.

Visualizzare il risultato di una operazione di riconoscimento vocale operata su una parola o una frase pronunciata al microfono. *First result* rappresenta il risultato più attendibile mentre *hints* è una lista di altri possibili suggerimenti.

Aggiornamento dell'interfaccia



```
public void setRMSValues(final double value);
```

```
public void setActivityResult(List<String> list);
```

Per stampare a video i valori inferiti dal sensore si possono utilizzare i metodi sopra citati, già implementati nel progetto scheletro.

Misurazione del segnale

In acustica si utilizza il Decibel Sound Pressure Level, denotato con dB_{SPL} , per indicare il livello di pressione sonora; vale:

$$SPL = 10 \times \log_{10} \left(\frac{p^2}{u^2} \right) = 20 \times \log_{10} \left(\frac{p}{u} \right)$$

dove u è la pressione sonora corrispondente alla soglia di udibilità ($20 \times 10^{-6} Pa$).

dB_{SPL}	Sorgente	dB_{SPL}	Sorgente
300	Eruzione del Krakatoa nel 1883	90	Urlo, fischietto
250	All'interno di un tornado	80	Camion pesante a 1 m
180	Razzo al decollo	70	Aspirapolvere a 1 m, radio ad alto volume
140	Colpo di pistola a 1 m	60	Ufficio rumoroso, radio, conversazione
130	Soglia del dolore	50	Ambiente domestico, teatro a 10 m
125	Aereo al decollo a 50 m	40	Quartiere abitato, di notte
120	Sirena, Auto di Formula 1 in pista	30	Sussurri a 1 m
110	Motosega a 1 m	20	Respiro umano
100	Discoteca, concerto rock	0	Soglia dell'udibile

Audio Record

Alla pressione del pulsante «Start» deve iniziare il campionamento del segnale rilevato dal microfono. A tal fine si utilizza l'oggetto *AudioRecord* che permette di interagire col microfono:

- **Int CHANNEL_CONFIGURATION**: (stereo, mono, solo canale destro ecc..) utilizzare **CHANNEL_IN_DEFAULT**.
- **Int SAMPLE_RATE**: Frequenza di campionamento. L'unico valore il cui supporto per il sistema Android è garantito da ogni produttore è **44100**.
- **Int ENCODING**: Codifica del valore campionato, ad esempio **ENCODING_PCM_16BIT** indica 16 bit a Pulse Code Modulation.

```
public void start(View v) {  
  
    bufferSize = AudioRecord.getMinBufferSize(SAMPLE_RATE, CHANNEL_CONFIGURATION, ENCODING);  
    recorder = new AudioRecord(AudioSource.MIC, SAMPLE_RATE, CHANNEL_CONFIGURATION, ENCODING,  
        bufferSize);  
}
```

La funzione *AudioRecord.getMinBufferSize* permette di conoscere la dimensione minima del buffer da allocare per poter salvare i dati relativi ad un campionamento.

Audio Record: lifecycle

Dopo la creazione dell'oggetto *AudioRecord* è possibile acquisire il controllo del sensore microfono con la primitiva *StartRecording* (si noti che da questo momento il sensore non sarà utilizzabile da altre applicazioni, incluse le funzioni telefoniche, fino ad esplicito rilascio della risorsa).

Il processo di lettura dei dati campionati va eseguito in un *thread* diverso da quello dell'*activity* principale in quanto costituisce operazione sincrona o bloccante.

Quando l'applicazione passa in secondo piano è opportuno prevedere il rilascio della risorsa e di conseguenza la distruzione del *thread* relativo.

```
recorder.startRecording();
short[] audioData = new short[bufferSize];
recordingThread{
    while(isRecording){ ... recorder.read(audioData, 0, bufferSize); ... }
}
recordingThread.start();

//fermo il campionamento del segnale e rilascio la risorsa microfono
isRecording = false;
recorder.stop();
recorder.release();
```

Calcolo del RMS

Il valore RMS va calcolato ad ogni campionamento. Una volta letto il buffer di campionamento occorre scorrere tutti i valori rilevati e sommare il loro quadrato.

Poi si esegue:

```
RMS = Math.sqrt(RMS / audioData.length);
```

Infine si trasforma il valore in Sound Pressure Level:

```
RMSdB = 20 * Math.log10(rms / earThreshold);
```

Dove *earThreshold* è il valore che rappresenta la soglia di udibilità.

Per completare la procedura è necessario applicare un fattore di correzione.

Calibrazione

A causa delle diverse caratteristiche dei sensori montati negli smartphone, si rende necessario impostare un valore per correggere il segnale acquisito.

Questo valore, variabile da microfono a microfono, dovrebbe essere calcolato con l'ausilio di strumentazione adeguata. Ad esempio, su un Samsung Galaxy S3, la correzione da applicare varia tra i -80 e i -100 dB_{SPL} rispetto al valore riportato dal sensore.

Ai fini dell'esercitazione sarà opportuno fare dei test con il dispositivo in dotazione per impostare il valore più adeguato. La costante *MIC_CALIBRATION* deve correggere quanto restituito dal sensore in modo che il valore risultante si attesti a circa 40 dB_{SPL} in presenza del rumore di fondo circostante.

La costante *RMS_THRESHOLD* indica la soglia da superare per rilevare rumore e si può impostare su un valore a piacere (ad esempio 60 dB_{SPL}).

```
setRMSValue(RMSdB - MIC_CALIBRATION);
```


Consigli per l'implementazione

L'acquisizione del sensore deve essere collegata al pulsante «Start» presente nell'applicazione scheletro.

Il rilascio del sensore deve essere collegato al pulsante «Stop» ma deve anche essere applicato nel caso l'applicazione passi in secondo piano.

```
//Permessi di accesso:  
//Nei sistemi Android, alcune azioni sono possibili solo se il Manifest dell'applicazione esplicita la  
//richiesta di accesso alla risorsa opportuna.  
//Ad esempio, per utilizzare il microfono e registrare il segnale sonoro è necessario che il Manifest  
//contenga la specifica:  
  
<uses-permission android:name="android.permission.RECORD_AUDIO" />
```

Implementazione: sound pressure level

Una volta calcolato il RMS a seguito di campionamento è possibile settare il valore sulla Text View preposta. Se il valore supera la soglia del rumore impostata, il box diventa rosso.

```
RMS = ...  
  
//visualizzazione nell'interfaccia grafica su TextView  
setRMSValues(RMS);
```

Implementazione: riconoscimento vocale

Per il riconoscimento vocale si può utilizzare un servizio di Google inc.

Occorre lanciare un intent ad una activity con il costrutto *startActivityForResult*. Se non specificato diversamente, il riconoscimento viene fatto con la lingua di sistema.

Una volta elaborato il segnale vocale l'activity risponderà con la lista di risultati calcolati (che vanno raccolti dal metodo *onActivityResult*).

```
Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL, RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Test di Activity Recognition");
startActivityForResult(intent, REQUEST_CODE);

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

    if (requestCode == REQUEST_CODE && resultCode == RESULT_OK) {
        ArrayList<String> matches = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
        setActivityRecognitionResult(matches);
    }

    super.onActivityResult(requestCode, resultCode, data);
}
```