

## Historische Dokumentation zur APP-Entwicklung

Der Autor hatte in der Vergangenheit eine Vielzahl von Auswerteprogrammen in verschiedenen Programmierwelten erstellt. Die Studierenden konnten auf diese Software im großen Stil nicht zugreifen, weil die relativ teure Messtechnikausrüstung nicht in größerer Stückzahl verfügbar war. Die APPs waren hier die Lösung, denn die Smartphones verfügten über eine gewisse Sensorik. Man musste sie nur ausnutzen.

In den Jahren 2014 (April 2014) bis 2017 (November 2017) wurden im "Google Play Store" neun verschiedene APPs veröffentlicht. Neben drei reinen APPs, die ohne Sensoren auskommen, nutzten 4 APPs das Vorhandensein des Mikrofons und zwei APPs die geräteinternen Beschleunigungssensoren.

Wegen zu zeitraubender Aktualisierungsanforderungen wurde die Weiterentwicklung auf Eis gelegt. Dadurch sind die APPs Zug um Zug in den Jahren bis 2023 aus dem "Google Play Store" verschwunden.

Übersicht über die APPs, die hiermit zur Verfügung gestellt werden:

control_loop_2017.apk	benötigt keine Sensoren
vibration_tamper_2017.apk	benötigt keine Sensoren
vibration_plate_2017.apk	benötigt keine Sensoren
acoustic_frequencies_2017.apk	Mikrofon, Berechtigung erforderlich
vibration_analysis_2017.apk	die drei Beschleunigungssensoren, keine Berechtigung erforderlich

Weitere 4 APPs aus dem Zeitraum waren kostenpflichtig. Sie sind jetzt nicht so einfach verfügbar, weil sie im GOOGLE PLAY STORE wieder angeboten werden sollen.

Weiter unten werden alle neun APPs vorgestellt.

Des Weiteren ist jeweils eine aktuelle BETA-Version in KOTLIN (Frühjahr 2026) zum  
vibration-tamper → a\_spring\_26\_1  
und  
control-loop → a\_spring\_26\_9  
vorhanden.

Durch Nachfragen und Drängen früherer Nutzer ist das Vorhaben entstanden, diese Aktivitäten wieder aufzunehmen. Damals verlangte Google eine Homepage, wo Datenschutzerklärungen etc. zu finden sind. Des Weiteren wurde eine Gewerbeanmeldung verlangt. Die Gewerbeanmeldung erfolgte unter dem Namen "KröberApps" und ist seitdem vorhanden. Eine eigene Homepage gibt es seit dem März 2026. Ihre Adresse lautet [www.kroeberapps.de](http://www.kroeberapps.de).

Was findet man auf dieser Homepage? Hier sind derzeit alle meine gestellten schriftlichen Hochschulprüfungen seit der Jahrhundert- bzw. Jahrtausendwende zu finden.

Die neun APPs sollen folgen. Im ersten Schritt ist geplant, dass die drei APPs, die ohne Sensoren auskommen, dort eingestellt werden (Ziel: GOOGLE PLAY STORE).

Danach folgen die APPs um die Nachhallzeit zu messen.

Es gibt viele Ideen, wie man die APPs mit noch mehr Funktionalität ausstatten kann. Zu nennen ist zum Beispiel das Abspeichern von Messwerten, um diese später mit einer Tabellenkalkulation weiter auszuwerten. Dann wäre beispielhaft zu nennen: Geschwindigkeitserkennung von vorbeifahrenden Kraftfahrzeugen, Approximationstools aus der Numerischen Mathematik, Finite-Elemente-Programm zur Berechnung von Eigenfrequenzen.

Spezialgebiet:

Für APPs mit Beschleunigungssensoren sind bei höherer Abtastrate viele Anwendungen möglich, die vor 10 Jahren nicht funktioniert haben. In meiner aktiven Zeit als Ingenieur habe ich Software entwickelt, mit

der man während der Verdichtung aufgrund von Beschleunigungsmessungen auf den Verdichtungszustand des Bodens schließen kann. Wenn es gelingt, die Messaufgabe anstatt mit aufwändiger Gerätetechnik auf einer Maschine durch eine APP auf einem Smartphone zu realisieren, ist der hier angestrebte Wiedereinstieg in diese Programmieretechnik voll gerechtfertigt. Hier sei auf drei nennenswerte Publikationen zur Thematik verwiesen, die unter [www.kroeberapps.de](http://www.kroeberapps.de) unter Diverses zu finden sind.

Wurden die APPs vor einem Jahrzehnt durchgängig in JAVA programmiert, empfiehlt GOOGLE für Neuentwicklungen die Programmiersprache KOTLIN. Der Aufwand ist in jedem Fall immens, wie weiter unten noch gezeigt werden soll. Ein Unterschied auf dem ersten Blick fällt auf, dass in JAVA eine Befehlszeile mit einem Semikolon abgeschlossen wird und bei KOTLIN nicht.

Zunächst bestand im Jahr 2026 das Vorhaben für eine APP zur Regelungstechnik bis Ende des SS2026 eine Fassung in KOTLIN zu erstellen. Das hat aus zeitlichen Gründen nicht komplett geklappt. Es gibt eine Fassung unter dem Namen "a\_spring\_26\_9\_app-debug.apk". Insbesondere die Startseite ist wirklich noch unfertig (Layoutausgestaltung). Es fehlen auch noch viereckige und runde Icons, die "Tags" sind noch nicht alle da usw., aber man kann damit Bode-Diagramme erstellen und eine Simulation für einen Sollwertsprung erzeugen. Der Regler kann durch Wischaktionen angepasst werden. Dazu gibt es weiter unten noch Erläuterungen. Die frühere Version dazu lautet "control\_loop\_2026.apk". Hier ist/war die Startseite natürlich fertig.

Eine APP zur Visualisierung eines Vibrationsstumpfers gibt es eine KOTLIN-Version "a\_spring\_26\_1\_app-debug.apk" und eine ältere JAVA-Version "vibration\_tamper\_2026.apk". Diese sind "äußerlich" eigentlich fertig. Hier sind lediglich noch kleine redaktionelle Anpassungen erforderlich. Um den Unterschied in JAVA und KOTLIN zu demonstrieren sind die Program-Listings des Vibrationsstumpfers aus der "Java-Zeit" und aus der jetzigen "Kotlin-Zeit" beigelegt. Man erkennt, dass im Detail doch erhebliche Änderungen erforderlich sind. Es ist das Hauptprogramm und das Unterprogramm hintereinander jeweils in eine Datei kopiert. Hier kann der Leser gerne Detailvergleiche vornehmen. Bei der KOTLIN-Fassung sind bei der APP-Anwendung weitere Funktionalitäten aktiviert. Diese Funktionalitäten werden aktiviert, wenn man diagonal von links unten nach rechts oben wischt. Dabei bewegt sich ein Ball wie bei einem Billardspiel (war eine Vorstufe der APP, aus dem Internet) und es wird der Wischweg, den ein Finger erzeugt, sozusagen als Spur dargestellt.

Wenn man die fünf APPs aus der Zeit um 2017 installiert, sind einige kleinere Hürden zu überwinden, die einem im Allgemeinen nicht begegnen. Zum Beispiel: Diese App wurde für eine ältere Android-Version entwickelt und bietet keinen aktuellen Datenschutz (dafür kann ich natürlich keine Gewähr zur Verfügung stellen). Unter "Weitere Details ..." kann man die Installation aber trotzdem ausführen. Es folgen im weiteren Verlauf noch weitere Schritte, die es "sonst nicht gibt". Aber nach der Installation funktionieren die APPs einwandfrei.

Die APP "acoustic\_frequencies" benötigt die Erlaubnis, dass RECORD AUDIO zugelassen werden muss. Das Abfragen des Beschleunigungsaufnehmers (vibration analysis) benötigt interessanterweise keine Berechtigung.

Die neueren Apps "a\_spring\_26\_1" (ist der Vibrationsstumpfer) und a\_spring\_26\_9 (ist die Regelungstechnik-App) wird bei der Installation ein App-Scan empfohlen. Wird dieser ausgeführt erscheint die Meldung "diese App ist anscheinend sicher". Diese Meldung zeigt: Android lässt sich immer noch eine Tür offen, aber die Installation verläuft letztlich unproblematischer.

Als weiteres Beispiel zum Unterschied zwischen JAVA und KOTLIN sind die Programmauszüge dargestellt, wie und wo auf dem Bildschirm die Wischaktionen zur Veränderung der Verstärkung von  $K_p$  (linker Bildschirmteil) und  $T_n$  (rechter Bildschirmteil) erfolgt. Dabei ist jeweils eine Vertikale Wischaktion (nach oben oder unten) erforderlich. Es steht in den Listings noch mehr, das muss man sich einfach wegdenken.

## Java-Beispiel zur Wischaktion (aus knapp vor 2017):

```
public boolean onTouch(View vv, MotionEvent event) {
    int action = event.getAction();
    int downx, downy, upx, upy;

    if(action == MotionEvent.ACTION_DOWN) {
        downx = (int) event.getX();
        downy = (int) event.getY();
        Up0.downx=downx;
        Up0.downy=downy;
    }

    if(action == MotionEvent.ACTION_UP) {
        upx = (int) event.getX();
        upy = (int) event.getY();
        Up0.upx=upx;
        Up0.upy=upy;
        int idraw=0;

        if((Up0.downx<(int)(0.4*Up0.b))&&(Up0.upx<(int)(0.4*Up0.b))&&((Up0.downy-Up0.upy)>(int)(0.1*Up0.h))){
            Up0.Kp=1.1*Up0.Kp;
            if(Up0.Kp>99.99)Up0.Kp=99.99;
            //startActivity(new Intent(this,Seite3.class));
            //finish();
            Toast.makeText(getApplicationContext(), text: "Kp +",Toast.LENGTH_SHORT).show();
            idraw=1;
        }
        if((Up0.downx<(int)(0.4*Up0.b))&&(Up0.upx<(int)(0.4*Up0.b))&&((Up0.downy-Up0.upy)<-(int)(0.1*Up0.h))){
            Up0.Kp=Up0.Kp/1.1;
            if(Up0.Kp<0.01)Up0.Kp=0.01;
            //startActivity(new Intent(this,Seite3.class));
            //finish();
            Toast.makeText(getApplicationContext(), text: "Kp -",Toast.LENGTH_SHORT).show();
            idraw=1;
        }
        if((Up0.downx>(int)(0.6*Up0.b))&&(Up0.upx>(int)(0.6*Up0.b))&&((Up0.downy-Up0.upy)>(int)(0.1*Up0.h))){
            Up0.Tn=1.1*Up0.Tn;
            if(Up0.Tn>99.99)Up0.Tn=99.99;
            //startActivity(new Intent(this,Seite3.class));
            //finish();
            Toast.makeText(getApplicationContext(), text: "Tn +",Toast.LENGTH_SHORT).show();
            idraw=1;
        }
        if((Up0.downx>(int)(0.6*Up0.b))&&(Up0.upx>(int)(0.6*Up0.b))&&((Up0.downy-Up0.upy)<-(int)(0.1*Up0.h))){
            Up0.Tn=Up0.Tn/1.1;
            if(Up0.Tn<0.1)Up0.Tn=0.1;
            //startActivity(new Intent(this,Seite3.class));
            //finish();
            Toast.makeText(getApplicationContext(), text: "Tn -",Toast.LENGTH_SHORT).show();
            idraw=1;
        }
        if(idraw==1){
            // *****
            my();
            // *****
        }
    } // if(action == MotionEvent.ACTION_UP)
    return true;
}
```

Kotlin-Beispiel zur Wischaktion (aus 2026), leider mit schwarzem Standard-Hintergrund von Android-Studio Panda 3. Es wurden Hardcopies erzeugt, um die gute farbliche Ausgestaltung des Editors zu zeigen.

```
override fun onTouchEvent(event: MotionEvent): Boolean {
    when (event.action) {
        MotionEvent.ACTION_DOWN -> {
            xdown = event!!.x
            ydown = event!!.y
            idrawline = 0
            return true
        }

        MotionEvent.ACTION_UP -> {
            xup = event!!.x
            yup = event!!.y
            idrawline = 1
            iwisch = 1
        }
    }
}
```

```
if ((xdown < 0.4f * b) && (xup < 0.4f * b) && (yup - ydown) > 0.1f * h) {
    Kp_ = 0.9f * Kp_
}
if ((xdown < 0.4f * b) && (xup < 0.4f * b) && (yup - ydown) < -0.1f * h) {
    Kp_ = 1.1f * Kp_
}
if ((xdown > 0.6f * b) && (xup > 0.6f * b) && (yup - ydown) > 0.1f * h) {
    Tn_ = 0.9f * Tn_
}
if ((xdown > 0.6f * b) && (xup > 0.6f * b) && (yup - ydown) < -0.1f * h) {
    Tn_ = 1.1f * Tn_
}
```

```
if ((xup - xdown) > 0.75 * b) {
    if ((-yup + ydown) > 0.75 * h) {
        iall = 1
    }
}
return true
}
}

invalidate()
return super.onTouchEvent(event)
}
```

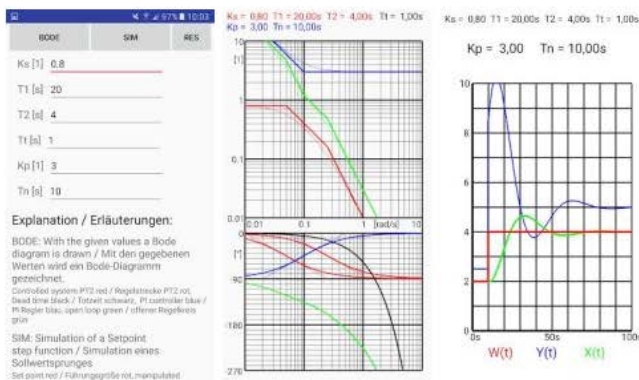
Unter der Internetquelle <https://droidinformer.org/music-audio/reverberation-time/>, welches ich gewissermaßen als ein Museum über Vergangenes ansehe, findet man Informationen über meine alten Apps, die im Folgenden kurz vorgestellt werden. Neben Hardcopies aus dem Internet sind dann noch weitere Informationen angegeben, die die Funktionalität weitergehender erläutern.

Home • Educational • control loop

control loop  
By Kröber

Package: com.kroeber.control\_loop Downloads: 1K Size: 1.2 MB Requires Android: 2.2 and up Updated: Sep 28, 2016

★★★★★ 9 Latest version: 1.0 Age: Everyone Download



In der App wird eine PT2-Strecke mit Totzeit mit einem PI-Regler geregelt. Mit den Parametern wird ein Bode-Diagramm erstellt und alternativ wird eine Simulation im Zeitbereich durchgeführt.

### Erläuterungen:

**BODE:** Mit den gegebenen Werten wird ein Bode-Diagramm gezeichnet.  
Regelstrecke PT2 rot, Totzeit schwarz, PI Regler blau, offener Regelkreis grün

**SIM:** Simulation eines Sollwertsprunges  
Führungsgröße rot, Stellgröße blau, Regelgröße grün  
Wischen in der linken Bildschirmhälfte nach oben erhöht Kp um den Faktor 1,1  
Wischen in der linken Bildschirmhälfte nach unten verringert Kp um den Faktor 1,1  
Wischen in der rechten Bildschirmhälfte nach oben erhöht Tn um den Faktor 1,1  
Wischen in der rechten Bildschirmhälfte nach unten verringert Tn um den Faktor 1,1  
Diese Veränderungen wirken nur innerhalb der Simulation.

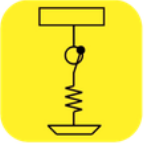
**RES:** Rücksetzen auf die Ausgangswerte

Generelle Begrenzungen bei Berechnungen:  
 $T1_{min} = T2_{min} = Tn_{min} = 0,1s$   $Tt_{max} = 9,9s$

Zusätzliche Begrenzungen bei der Simulation:  
 $0,1 \leq Kp \leq 99,99$   $0,1s \leq Tn \leq 99,99s$   $Ks \geq 0,2$

Home • Educational • vibration tamper


[Share](#)
[Tweet](#)



# vibration tamper

By Kröber

**Package:**  
com.kroeber.vibration\_tamper

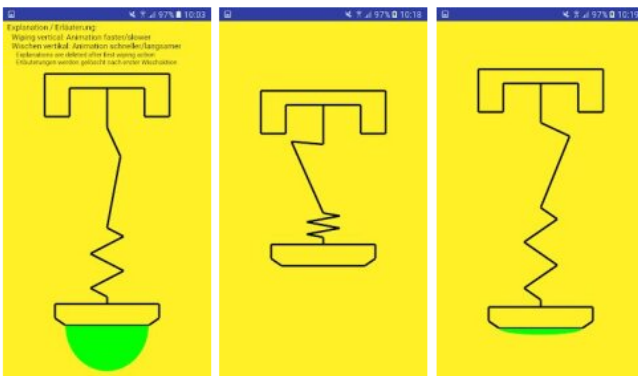
 **Downloads:**  
100

**Size:**  
1.2 MB

**Requires Android:**  
2.2 and up

**Updated:**  
Sep 28, 2016

★★★★★ 1
Latest version: 1.0
Age: Everyone
Download




## Vibration Tamper

Die APP zeigt in einer Animation das Bewegungsverhalten eines Vibrationsstampfers beim Betrieb auf einem auf festen Untergrund. Durch vertikale Wischaktionen kann die Geschwindigkeit der Animation verändert werden.

Das Zusammenwirken der einzelnen Massen und Federn, angefacht durch den Kurbeltrieb, ist letztlich ein maschinendynamisches und technisches Wunderwerk.

Home • Educational • vibration plate

Share Tweet



# vibration plate

By Kröber

Package: com.kroeber.vibration\_plate

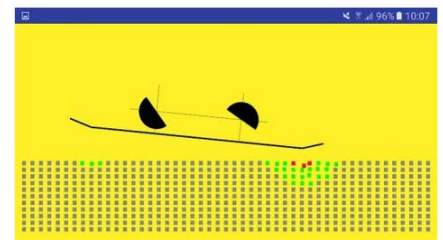
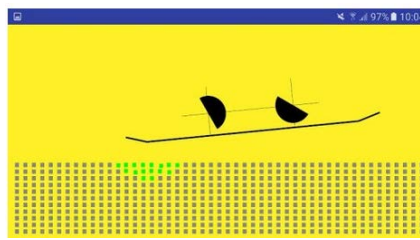
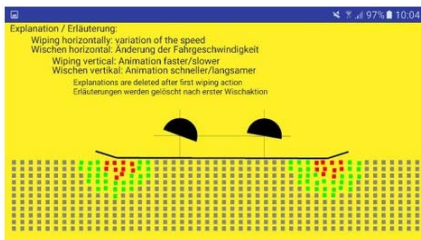
Downloads: 500

Size: 1.2 MB

Requires Android: 2.2 and up

Updated: Sep 28, 2016

★★★★☆ 6 Latest version: 1.0 Age: Everyone Download



Die APP zeigt in einer Animation das Bewegungsverhalten einer Vibrationsplatte beim Betrieb auf einem festen Untergrund. Durch horizontale Wischaktionen kann die Fahrgeschwindigkeit (Vorwärts und Rückwärts) variiert werden. Durch vertikale Wischaktionen kann die Geschwindigkeit der Animation verändert werden.

Die Idee, nur durch Ändern der Phasenlage von zwei Unwuchten zueinander, eine Vorwärts- und Rückwärts geschwindigkeit zu erzeugen, ist eine bemerkenswerte Erfindung.

Home • Music & Audio • acoustic frequencies

Share Tweet

Hz  
es  
dB

# acoustic frequencies

By Kröber

Package: com.kroeber.acoustic\_frequencies

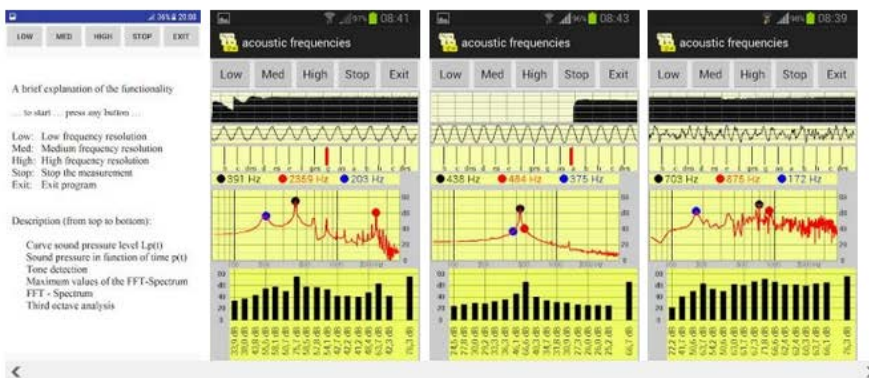
Downloads: 50K

Size: 1.3 MB

Requires Android: 2.2 and up

Updated: Feb 27, 2017

★★★★☆ 206 Latest version: 1.1 Age: Everyone Download



Es war meine erste APP, die im GOOGLE PLAY STORE verfügbar war (20.04.2014)

Kurzerläuterung der Funktionalität

... zum Starten ... einen Button drücken ...

- Low: Kleine Frequenzauflösung
- Med: Mittlere Frequenzauflösung
- High: Hohe Frequenzauflösung
- Stop: Stoppen der Messung
- Exit: Programm beenden

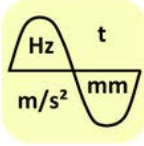
Darstellung (von oben nach unten):

- Schrieb Schalldruckpegel  $L_p(t)$
- Schalldruck in Funktion der Zeit  $p(t)$
- Tonerkennung
- Maximalwerte im FFT - Spektrum
- FFT - Spektrum
- Terzpegelanalyse

Hinweis: In der NICHT-PRO-VERSION ist sind die Buttons Med und High nicht verfügbar. Ferner wird der A-bewertete Pegel nicht angezeigt.

Home • Tools • vibration analysis


[Share](#)
[Tweet](#)



# vibration analysis

By Kröber

**Package:**  
com.kroeber.vibration\_analysis

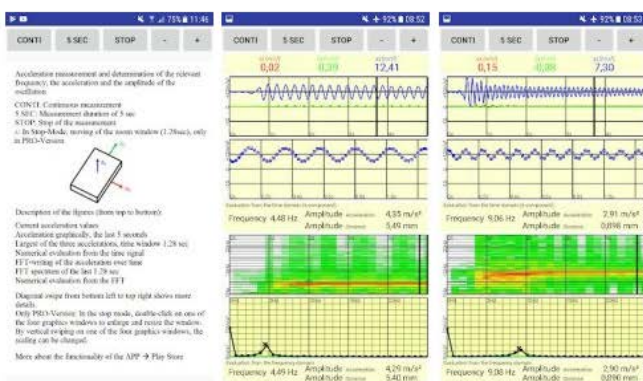
 **Downloads:**  
10K

**Size:**  
1.3 MB

**Requires Android:**  
4.1 and up

**Updated:**  
Nov 19, 2017

★★★★☆ 93
 Latest version: 1.0
Age: Everyone
Download



## Beschleunigungsmessung und Ermittlung der maßgeblichen Frequenz, der Schwingbeschleunigung und der Schwingamplitude

Erläuterung der oben angeordneten Buttons

CONTI: Kontinuierliche Messung

5 SEC: Feste Messdauer von 5 sec, danach automatisch STOP

STOP: Stop der Messung, es werden die letzten 5 sec dargestellt

-: Im Stop-Modus Zoomfenster (1,28sec) nach links verschieben, nur bei PRO-Version

+: Im Stop-Modus Zoomfenster (1,28sec) nach rechts verschieben, nur bei PRO-Version

Unter den Buttons sind sieben Bereiche angeordnet.

Oberer Bereich "Beschleunigung, numerisch":

Die Beschleunigungen werden als Zahlenwert angezeigt. Die Abtastfrequenz der meisten Smartphones beträgt 50 Hz. Weicht die tatsächliche Abtastfrequenz um  $\pm 1$  Hz ab, dann erscheint die tatsächliche Abtastfrequenz (blinkend). Verschiedene Zeitangaben müssen dann entsprechend korrigiert werden. Die numerisch angezeigten Messwerte sind automatisch korrigiert.

Bei einer Wischaktion "von links unten nach rechts oben" geht die App in einen Diagnose-Modus. Dann erscheinen die tatsächliche Abtastfrequenz, die Standardabweichung der Abtastfrequenz und ein interner Zähler.

Darunter liegender Bereich "Beschleunigungen über Zeit":

Es zeigt die Beschleunigungen der letzten 5 Sekunden über der Zeit. Nach Betätigen von STOP wird ein Fenster von 1,28 sec gekennzeichnet (vertikale schwarze Balken).

bei PRO-Version: das Fenster kann durch  $\pm$  verschoben werden. Nach einem Doppel-Tipp auf den Bereich erscheint er doppelt so groß, nach einem weiteren Doppel-Tipp wird er wieder verkleinert. Wischen nach oben vergrößert, Wischen nach unten verkleinert den Abbildungsmaßstab.

Darunter liegender Bereich "Größe der drei Beschleunigungskomponente über Zeit":

Es zeigt die Beschleunigungskomponente mit der maximalen Beschleunigungsänderung. Dargestellt wird das Zoomfenster von 1,28 sec. Es eignet sich besonders bei Schwingfrequenzen von 1 Hz bis 15 Hz. Je sinusförmiger das Zeitsignal, desto eindeutiger sind die ermittelten Werte. Im Diagnose-Modus sind Triggerpunkte und Sinusapproximationen sichtbar.

Bei PRO-Version: Nach einem Doppel-Tipp auf den Bereich erscheint er doppelt so groß, nach einem weiteren Doppel-Tipp wird er wieder verkleinert. Wischen nach oben vergrößert, Wischen nach unten verkleinert den Abbildungsmaßstab.

Darunter liegender Bereich "Zeitauswertung, numerisch":

Hier werden Auswertungen aus dem Zeitbereich numerisch angezeigt. Dies sind die Frequenz, die Beschleunigungsamplitude und die Wegamplitude der Schwingung. Im Diagnose-Modus wird die Anzahl der ermittelten Schwingungen angezeigt.  $S_x$  ist deren Standardabweichung. Je größer  $n$  und kleiner  $S_x$ , desto sicherer sind die ermittelten Werte. Bei der Beschleunigungsamplitude wird auch eine Standardabweichung angegeben. Die Berechnung erfolgt mit der Minimierung der Fehlerquadrate.

Darunter liegender Bereich "FFT über Zeit":

Hier wird ein Schrieb des Beschleunigungsspektrums über der Zeit dargestellt. Dargestellt werden die letzten 5 Sekunden. Das Zoom-Fenster ist markiert.

Bei PRO-Version: nach einem Doppel-Tipp auf den Bereich erscheint er doppelt so groß, nach einem weiteren Doppel-Tipp wird er wieder verkleinert. Wischen nach oben vergrößert, Wischen nach unten verkleinert den Intensitätsmaßstab.

Darunter liegender Bereich "FFT, graphisch":

Es zeigt das FFT-Spektrum innerhalb des Zoom-Fensters. Die schwarze Kurve ist die resultierende Beschleunigung aus allen drei Komponenten.

Bei PRO-Version: Nach einem Doppel-Tipp auf der Bereich erscheint er doppelt so groß, nach einem weiteren Doppel-Tipp wird er wieder verkleinert. Wischen nach oben vergrößert, Wischen nach unten verkleinert den Abbildungsmaßstab.

Unten liegender Bereich "FFT, numerisch":

Es handelt sich um die numerische Ausgabe des FFT-Spektrums innerhalb des Zoom-Fensters. Das Maximum der FFT ist durch ein Kreuz gekennzeichnet. Frequenz, Maximalwert der Beschleunigung und Amplitude werden ausgegeben.

**Es folgen nun noch vier APPs, die nur mit einem geringen Kostenbeitrag zu erwerben waren.**



# pro acoustic frequencies

By Kröber

Package:  
com.kroeber.pro\_acoustic\_frequenci...



Downloads:  
500

Size:  
1.3 MB

Requires Android:  
2.2 and up

Updated:  
Nov 19, 2017



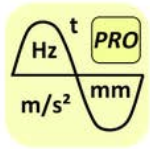
Latest version: 1.2

Age: Everyone

Download



Gegenüber der Standard-Version ist hier die mittlere und höheren Auflösung möglich. Es gab auch einmal zusätzlich eine ULTRA-hohe-Auflösung. Die angezeigten A-bewerteten Pegel sind/waren für ein Galaxy S5 kalibriert.



# vibration analysis pro

By Kröber

Package:  
com.kroerber.vibration\_analysis\_pro

Downloads:  
100

Size:  
1.3 MB

Requires Android:  
4.1 and up

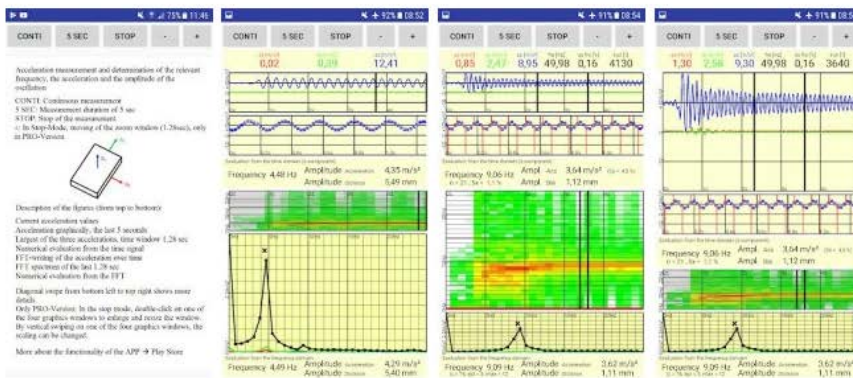
Updated:  
Nov 19, 2017



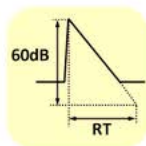
Latest version: 1.0

Age: Everyone

Download



Besonders hervorzuheben sind hier die Zoom-Möglichkeiten und das Zurück-Scrollen über der Zeit. In der Nicht-Pro-Version ist nur die Messzeit am Ende des Beobachtungszeitraums ausgewertet/angezeigt.



# reverberation time

By Kröber

Package:

com.kroeber.reverberation\_time



Downloads:

1K

Size:

1.3 MB

Requires Android:

2.2 and up

Updated:

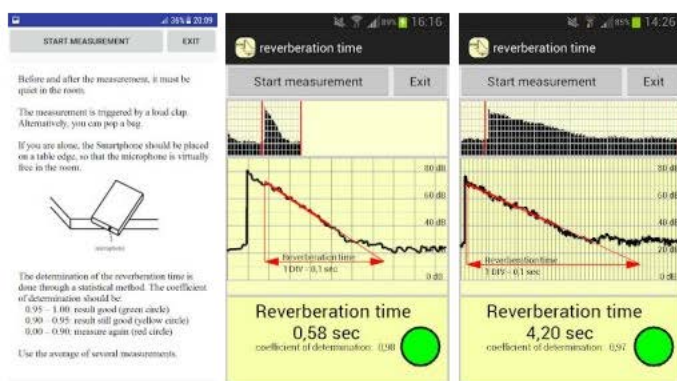
Feb 27, 2017



Latest version: 1.1

Age: Everyone

Download



Vor und während der Messung muss es leise im Raum sein.

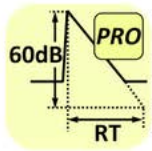
Die Messung wird getriggert durch ein lautes Klatschen oder das Zerplatzen einer Papiertüte.

Wenn Sie alleine sind, sollte das Smartphone auf eine Tischkante platziert werden, so dass das Mikrofon praktisch frei im Raum ist.

Die Bestimmung der Nachhallzeit basiert auf einem statistischen Auswerteverfahren. Das Bestimmtheitsmaß sollte sein:

- 0.95 – 1.00: statistisch sicher (grüner Kreis)
- 0.90 – 0.95: statistisch noch sicher (gelber Kreis)
- 0.00 – 0.90: Messung wiederholen (roter Kreis)

Verwenden Sie den Mittelwert aus mehreren Messungen.



# reverberation time pro

By Kröber

**Package:**

com.kroerber.reverberation\_time\_pro

**Downloads:**

500

**Size:**

1.3 MB

**Requires Android:**

2.2 and up

**Updated:**

Nov 19, 2017



Latest version: 1.2

Age: Everyone

Download



Bei einer erfolgreichen statistischen Auswertung ist die graphische Balkenanzeige (im unteren Bereich) grün, bei nicht erfolgreicher Auswertung rot. Zur Differenzierung werden abgestufte Farbkombinationen verwendet. Der numerisch angezeigte Mittelwert beinhaltet die Frequenzen von 250 Hz bis 5000 Hz. Im Mittelwert werden nur die Messwerte mit grünem Balken berücksichtigt. Diese gelten als hinreichend statistisch sicher. Mit dem Button – und dem Button + kann man die Abklingkurven aller Frequenzen einsehen.

### Besonderheit der Pro-Version:

Bei einer erfolgreichen statistischen Auswertung ist die graphische Balkenanzeige grün, bei nicht erfolgreicher Auswertung rot. Zur Differenzierung werden auch abgestufte Farbkombinationen verwendet. Der numerisch angezeigte Mittelwert beinhaltet die Frequenzen von 250 Hz bis 5000 Hz. Im Mittelwert werden nur die Messwerte mit grünem Balken berücksichtigt. Diese gelten als hinreichend statistisch sicher.

Mit dem Button – und dem Button + kann man die Abklingkurven aller Frequenzen einsehen.

Der Zahlenwert für die Nachhallzeit, der ganz rechts steht, ist mit Z (Z für Zero) bezeichnet. Hier findet keine Frequenzanalyse oder Frequenzbewertung statt. Dieser Wert wird bei der Standardversion der APP angezeigt.

Allgemeines:

Mit dieser APP kann die Nachhallzeit eines Raumes gemessen werden.

Die Nachhallzeit ist eine akustische Kenngröße für die Halligkeit eines Raumes. Zur Messung wird der Abfall des Schalldruckpegels nach einem kurzen lauten Geräusch (Zerplatzen einer Papiertüte oder Händeklatschen) gemessen. Die Zeit, die Schall braucht um 60 dB abzuklingen, wird als Nachhallzeit bezeichnet.

Bei der Zuordnung "80 dB = Laut" und "20 dB = Leise" kann man auch Folgendes sagen: Die Nachhallzeit ist die Zeit, die vergeht, bis man ein zuvor lautes Geräusch nicht mehr wahrnimmt.

In einem Unterrichtsraum in der Schule oder einem Aufenthaltsraum in einer Kindertagesstätte soll die Nachhallzeit ca. 0,6 Sekunden betragen. Der anzustrebende Wert ist leicht vom Volumen des Raumes abhängig:  $180 \text{ m}^3 = 0,55 \text{ s}$  /  $360 \text{ m}^3 = 0,65 \text{ s}$ .

Der Wert von 0,6 Sekunden gilt für einen besetzten Raum, in dem sich Schüler bzw. Kinder befinden. Durch die Absorption der Personen ist die Nachhallzeit eines leeren Raumes höher. Für einen leeren Raum gilt ein anzustrebender Wert von 0,75 Sekunden anstatt 0,6 Sekunden.

Für hörgeschädigte Kinder oder auch Erwachsene mit einem Tinnitus können von den Sollnachhallzeiten nochmals 20 % abgezogen werden.

Zum Vergleich: In einem Raum für Musiknutzung soll die Nachhallzeit 1,1 Sekunden betragen, in Kirchen liegt der Wert bei über 2 Sekunden.

Die Nachhallzeit ist zu lang:

In einem halligen Raum erklingt jedes Wort lauter. Ferner klingt ein zuvor gesprochenes Wort noch nach, wenn die Folgeworte gesprochen werden. Sie fallen sich buchstäblich ins Wort. Der Sprechende versucht zur besseren Sprachverständlichkeit lauter zu sprechen und dadurch wird es noch lauter im Raum.

Die Nachhallzeit ist zu kurz:

Kinder, die frontal angesprochen werden, verstehen das Gesprochene gut. Durch die zu hohe Schallabsorption im Raum sind die Worte im seitlichen und besonders im rückwärtigen Bereich aber nicht mehr ausreichend zu hören. Sie fühlen sich "nicht angesprochen". Dies kann durch Anheben der Stimme ausgeglichen werden, führt aber zur Heiserkeit des Sprechenden.

Das Messen der Nachhallzeit:

Vor und während der Messung muss es leise im Raum sein. Die Messung wird getriggert durch ein lautes Klatschen oder das Zerplatzen einer Papiertüte. Die ermittelten Werte werden unmittelbar angezeigt. Zur Auswertung soll der Mittelwert von 3-5 Messungen verwendet werden.