Typ	**Train Simulator 2013 - Tutorial**
Autor	**Benjamin Ebrecht**
Version	1.0
Datum	17.04.2013
Kontakt	**ebrecht@trainteamberlin.de**

1. Vorwort

Lieber Railworksfan,

in diesem Dokument soll die Nutzung des 3D Byte Soundsystems für eigene Szenarien erläutert werden.
Beim Soundsystem handelt es sich um eine externe Audiobibliothek, die über Lua-Skripte angesteuert werden kann. Sie dient der komfortablen Einbindung von (Stations)Ansagen in eigenen Szenarien. Diese Audio-Bibliothek wird von Fahrzeugbauern in die Fahrzeugskripte direkt in deren Skripte verlinkt und mit dem entsprechenden Railworksfahrzeug installiert. Für den Szenarienersteller ist nur noch die Anwendung der Bibliothek zu erlernen. Dies soll mit diesem Tutorial erläutert werden.
Zum Zeitpunkt der Veröffentlichung dieses Dokuments verfügen folgende Fahrzeuge über das 3d Byte Soundsystem:

BR143 PlusPack (BR143 Railworks-Standard, Köln-Düsseldorf Addonlok, Köblitz-Addonloks)
BR294 PlusPack (BR294 Railworks-Standard)

Es ist geplant, auch die Fahrzeuge des **BR101 PlusPacks** sowie des **BR151-PlusPacks** in künftigen Updates mit den Bibliotheken auszuliefern.
Weiterhin werden künftige Fahrzeugneukontruktionen über die Audiobibliothek verfügen können.

TrainTeamBerlin
Berlin, im April 2013

2. Inhalt

2. Tutorial

An dieser Stelle soll zunächst die Funktionsweise des 3D Byte Soundsystems in aller Kürze erläutert werden.
Jedes Railworksfahrzeug verfügt über ein Fahrzeugskript. Mit dem Einbau des 3D Byte Soundsystems ist es nun möglich, einer externen dll-Datei Befehle wie "Spiele eine bestimmte Tondatei ab" zu übergeben, ohne dass der Spieler dafür etwas tun müsste.
Weiterhin ist das Fahrzeugskript in der Lage, Nachrichten von Signalen (die ebenfalls über Skripte gesteuert werden), sogenannte Messages, zu empfangen und zu verarbeiten. Die Taktik des Systems ist es nun, bestimmte (szenarienspezifische) Konfigurationssignale im Szenario zu verbauen und dem Soundsystem beim Überfahren des Signals einen Audiobefehl zu übergeben.
Um das System einfach und flexibel zu gestalten, wurden sogenannte Playlists eingeführt. Es handelt sich um simpel editierbare xml-Dateien, in denen alle für ein Szenario benötigten Tondateien verlinkt sind und einen Index zugewiesen bekommen.
Will man nun ein Szenario mit automatisch abzuspielenden Ansagen ausstatten, so sind folgende Schritte auszuführen:

→ (1) Tondateien erstellen
→ (2) Playlist erstellen und Tondateien darin verlinken
→ (3) Konfigurationssignal für diese Playlist erstellen
→ (4) Konfigurationssignal im Szenarioeditor verbauen

Diese Schritte werden im folgenden detailliert erläutert. Sofern Probleme auftreten, kann die im Archiv gelieferte Asset-Ordnerstruktur ins Hauptverzeichnis kopiert und zu Schritt 4 gesprungen werden.

2.1. Tondateien erstellen

Zunächst werden die Tondateien benötigt, die später abgespielt werden sollen. Auf welchem Wege Sie diese erstellen, bleibt ihrer Kreativität überlassen. Wichtig ist jedoch, dass sie die Dateien entweder im

Wav-Format (.wav) oder im
Ogg-Vorbis-Format (.ogg)

abspeichern. Die Dateien müssen nun an einem bestimmten Ort abgelegt werden. Das Grundverzeichnis hierfür ist

.. \railworks\Assets\TrainTeamBerlin\SoundSystem\TrainTeamBerlin\PIS\Files

Zusatzinformation: Der Pfad ..\Soundsystem\TrainTeamBerlin\ ist das Grundverzeichnis für die Tondateien und Playlists. Das Unterverzeichnis PIS steht für "Passenger Information System" und entspricht einem von mehreren möglichen Slots des Systems. Für unsere Anwendung ist das nicht von Belang, wir werden ausschließlich den PIS-Slot und damit auch ausschließlich den PIS-Ordner nutzen.

Da dieser Ordner von allen verschiedenen Szenarienerstellern genutzt wird, empfiehlt es sich, eigene Unterordner zu erstellen, um Kollisionen von Dateien verschiedener Nutzer zu vermeiden. Die vorgeschlagene Ordnerstruktur ist

..
\railworks\Assets\TrainTeamBerlin\SoundSystem\TrainTeamBerlin\PIS\Files\<Nutzer>\<Scenario>

Für dieses Tutorial gehen wir davon aus, dass wir unsere Tondateien (zwei Samples zum Testen sind im Archiv enthalten) im Nutzerordner "TTB-Tutorial" für unser Szenario "3DByteFirstTest" nutzen. Also werden die Dateien im Ordner

.. \railworks\Assets\TrainTeamBerlin\SoundSystem\TrainTeamBerlin\PIS\Files\TTB-Tutorial\TTB_TUT_Test

gespeichert.

2.2. Playlist erstellen

Als nächstes ist eine Playlist im xml-Format zu erstellen. Im folgenden wird eine minimal konfigurierte Playlist gezeigt, die für die meisten Anwendungen ausreichend ist:

```
<?xml version="1.0" encoding="utf-8"?>
<SoundSystem>
  <Sounds>
    <Sound index="1" main="TTB-Tutorial/TTB_TUT_Test /TTB_02_01.ogg" />
    <Sound index="2" main="TTB-Tutorial/TTB_TUT_Test /TTB_02_02_Augs_Haunstetter.ogg"/>
  </Sounds>
</SoundSystem>
```

Auch mit wenigen xml-Grundkenntnissen wird die einfache Dateistruktur schnell deutlich. Innerhalb des "Sounds"-Tags wird für jede Tondatei ein "Sound"-Tag erstellt, der die Eigenschaften "index" und "main" enthält.
Der Index kann aus Zahlen und/oder Buchstaben bestehen und wird später aus dem Konfigurationssignal ausgelesen, um zu steuern, welche Datei der Playlist abgespielt werden sollte. Demnach müssen eindeutige Indizes verwendet werden. Es empfiehlt sich, keine Sonderzeichen zu verwenden. Der Einfachheit halber können die Ansagen mit Zahlen ihrer Abspielreihenfolge nach eingetragen werden.
In "main" ist der Pfad zur Tondatei anzugeben. Ausgehend davon, dass wir eine Playlist für den PIS-Slot erstellen, ist der Pfad relativ zum Verzeichnis

.. \railworks\Assets\TrainTeamBerlin\SoundSystem\TrainTeamBerlin\PIS

anzugeben. Entsprechend sind die von uns angelegten Unterordner TTB-Tutorial/3DByteFirstTest noch zusätzlich zum Dateinamen anzugeben. Dies ist im Beispielcode (oben) bereits richtig geschehen.

Damit ist die Playlist fertig gestellt. Sie wird im xml-Format im Ordner des jeweiligen Slots (richtig, bei uns PIS) abgelegt. Der Übersichtlichkeit halber erstellen wir wieder einen Unterordner mit unserem Entwicklernamen, in diesem Tutorial "TTB-Tutorial". Nennen wir die Datei TTB_TUT_Test.xml. Demnach wird sie als

.. \railworks\Assets\TrainTeamBerlin\SoundSystem\TrainTeamBerlin\PIS\TTB-Tutorial\TTB_TUT_Test.xml

abgespeichert.

2.3. Konfigurationssignal für die Playlist erstellen

Nun fehlt nur noch ein Konfigurationssignal für unsere Playlist, welches bei Überfahrt mit der Lok über das Signal entsprechende Anweisungen zum Abspielen der Tondateien auslöst. Dieses Signal muss mit Hilfe des BluePrint-Editors erstellt werden. Im Ordner "Material" dieses Archivs liegen zwei Unterordner bei.

1. Der Unterordner "Meshes" enthält die nötigen Quelldaten eines 3D-Modells für unser Konfigurationssignal. Dieser muss nur einmal ins Source-Verzeichnis kopiert und kann für sämtliche Konfigurationssignale genutzt werden. Auch darf er in kompilierter Version als Teil von Szenarien-Ansagenpaketen kostenlos veröffentlicht werden. Kopieren Sie diesen Ordner nach

..\railworks\Source\TrainTeamBerlin\SoundSystem\TrainTeamBerlin\EventSignals

2. Es sind im Materialordner zwei Dateien " TTB_TUT_Test .lua" und " TTB_TUT_Test .xml" enthalten. Sie bilden eine Einheit und sind ein Signalblueprint (.xml) samt zugehörigem Signalskript (.lua). Diese Dateien sind bereits vorbereitet und müssen nur an wenigen Stellen angepasst werden. Auch sie dürfen in kompilierter Version als Teil von Szenarien-Ansagenpaketen kostenlos veröffentlicht werden.

Passen wir das Blueprint nun entsprechend an. Als erstes Kopieren Sie beide Dateien in den Ordner

..\railworks\Source\TrainTeamBerlin\SoundSystem\TrainTeamBerlin\EventSignals\<Nutzer>

In unserem Falle also:

..\railworks\Source\TrainTeamBerlin\SoundSystem\TrainTeamBerlin\EventSignals\TTB-Tutorial

Als erstes wird das lua-Skript angepasst. Öffnen Sie dieses mit einem einfachen Texteditor. In Zeile 17 finden Sie den Eintrag

```
tmp[1] = "6687 TTB-Tutorial/TTB_TUT_Test 0 " .. signalID;
```

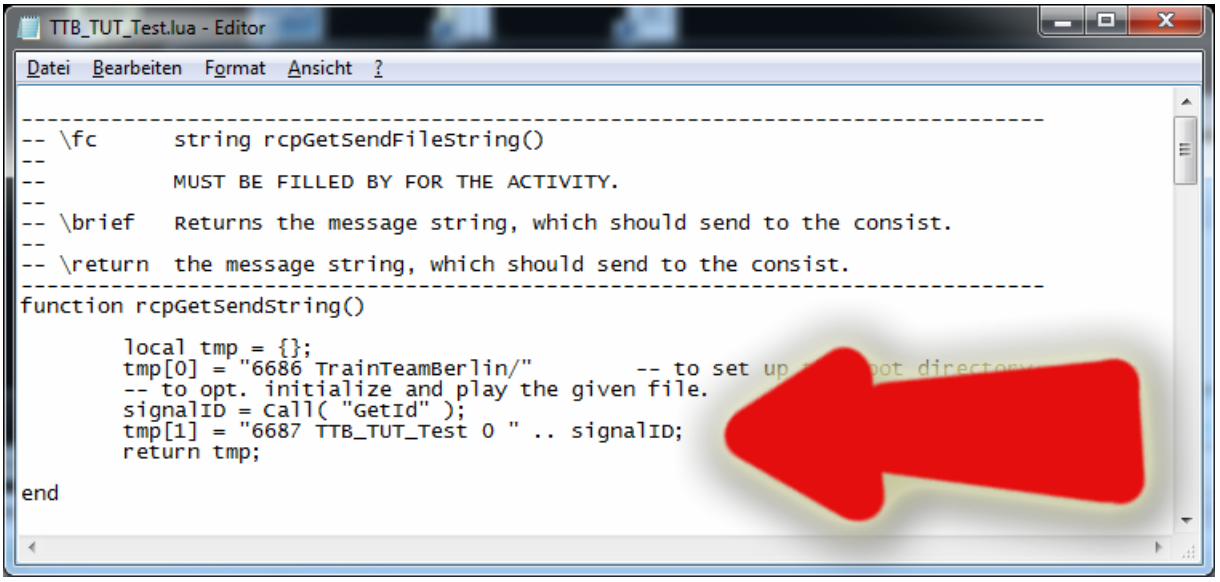
Der Eintrag hat die Struktur

"**Befehlsnummer**[Leerzeichen]**RelativPfadZurPlaylist**[Leerzeichen]**Slot**[Leerzeichen]**IndexInPlaylist**"

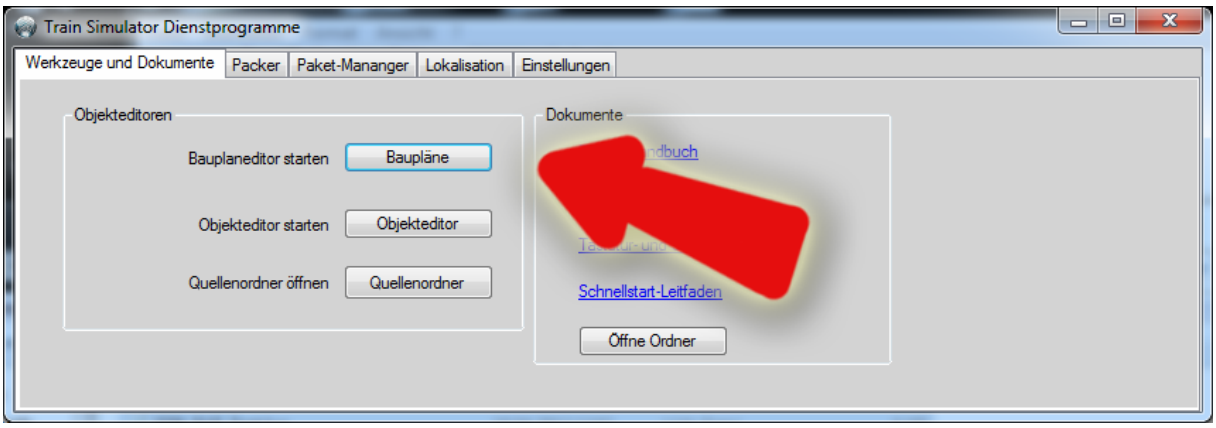
Hiermit werden die an das Fahrzeug übergebenen, durch Leerzeichen getrennten (Teil)Befehle definiert. Die **6687** ist eine Befehlfsnummer und darf **nicht geändert** werden. Es folgt die Angabe der Playlist relativ zum Pfad

..\railworks\Assets\TrainTeamBerlin\SoundSystem\TrainTeamBerlin\PIS

ohne Dateiendung! Demnach muss für dieses Tutorial **TTB-Tutorial/TTB_TUT_Test** eingetragen werden. Die nachfolgende Null gibt den benutzten Slot an - in unserem Fall bleibt dies der PIS-Slot, der durch die 0 repräsentiert wird. Weiterhin folgt die Angabe des Indizes der abzuspielenden Datei aus der angegebenen Playlist. Hier wird der Name des Signals, den wir im folgenden Schritt des Tutorials eingeben werden, ausgelesen und an die Lok übergeben. Insofern ist auch dieser Eintrag nicht zu ändern.



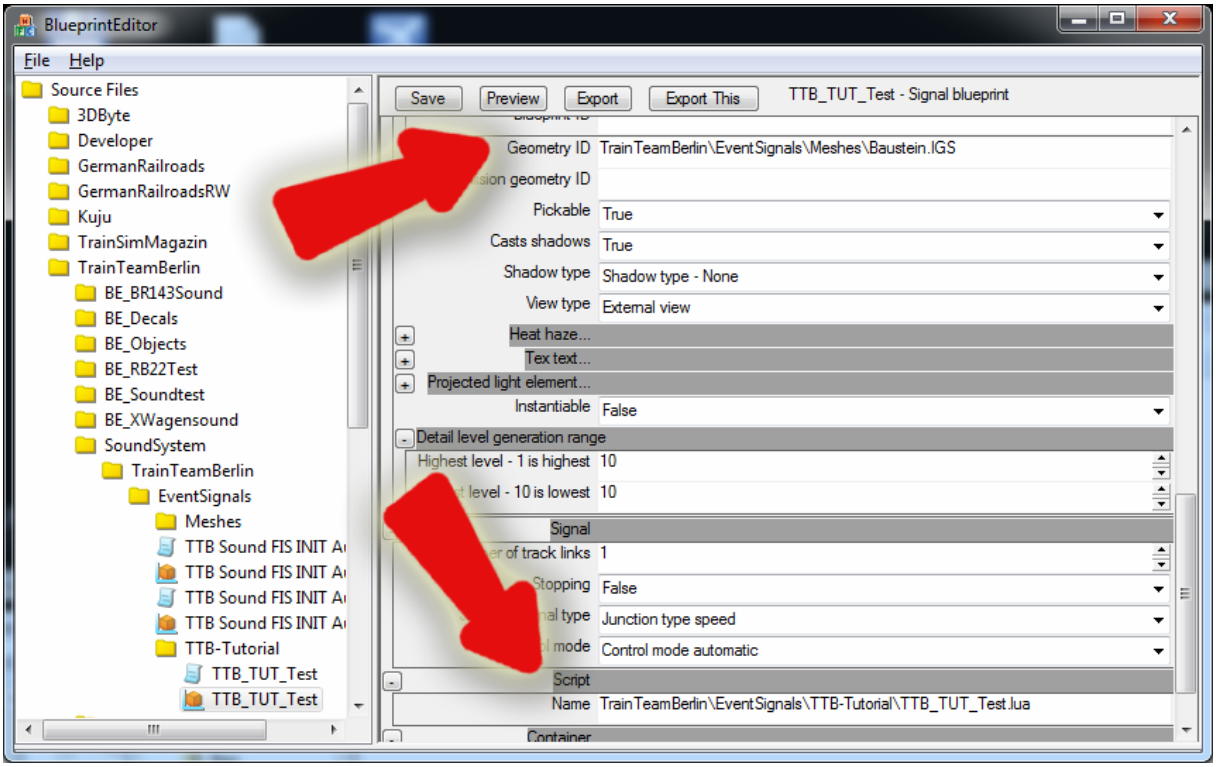
Nun öffnen Sie den Railworks BluePrint-Editor. Dazu öffnen Sie die Utilities.exe im Railworks-Hauptverzeichnis und wählen im Reiter "Werkzeuge und Dokumente" (ist standardmäßig bereits gewählt) "Baupläne" aus.



Es öffnet sich der BluePrint Editor. Navigieren Sie sich im Dateiexplorer auf der linken Seite in das Verzeichnis

..\railworks\Source\TrainTeamBerlin\SoundSystem\TrainTeamBerlin\EventSignals\TTB-Tutorial

Ihnen werden beide Dateien angezeigt - .lua und .xml. Öffnen Sie die xml-Datei (das Blueprint wird mit einem orangen Würfel im Icon angezeigt). Auf der rechten Seite des Blueprinteditors erscheint eine strukturierte Wiedergabe der Datei.



Im oberen Bereich der Datei ist zweimal der Name des Blueprints einzutragen. Dies ist bereits geschehen. Für eigene Konfigurationssignale sind die Namenseinträge entsprechend anzupassen, sie geben an, unter welchem Namen das Konfigurationssignal im Szenarioeditor zu finden ist.

Weiter unten finden Sie den Eintrag "Geometry ID". Hier wird ein 3D-Modell an das Signal gebunden. Wenn Sie den Schritt 2.3.1 richtig ausgeführt haben, stimmt auch hier der Pfad bereits überein und Sie müssen an diesem Eintrag nichts ändern - auch nicht, wenn Sie später eigene Szenarien mit Ansagen ausstatten.

Entscheidend ist das verlinkte Skript (vgl. Abbildung). Dieses muss für jedes Szenario neu verlinkt werden, da wir (wie bereits weiter oben geschehen) im lua-Skript die szenariospezifisch geladene Playlist definieren müssen. Der Eintrag

TrainTeamBerlin\EventSignals\TTB-Tutorial\TTB_TUT_Test.lua

stimmt auch hier bereits für das von uns ins Source-Verzeichnis kopierte lua-Skript überein. Klicken Sie nun oben auf den Button Save, danach auf Export. Wird in der Konsole am unteren Bildschrimrand kein Fehler und "Successful build" angezeigt, haben Sie auch diesen Schritt des Tutorials erfolgreich abgeschlossen.

Hinweis: Das Konfigurationssignal samt zugehörigem Script ist nicht an diese Ordnerstruktur gebunden, das trifft nur auf die Ansagendateien und die xml-Playlist zu. Erfahrene Railworksnutzer, die sich bereits eine eigene Ersteller-Produktordnerstruktur im Source-Verzeichnis angelegt haben, können das Signalblueprint und das Lua-Skript also auch in ihrem "persönlichen" Ordner abspeichern und exportieren.

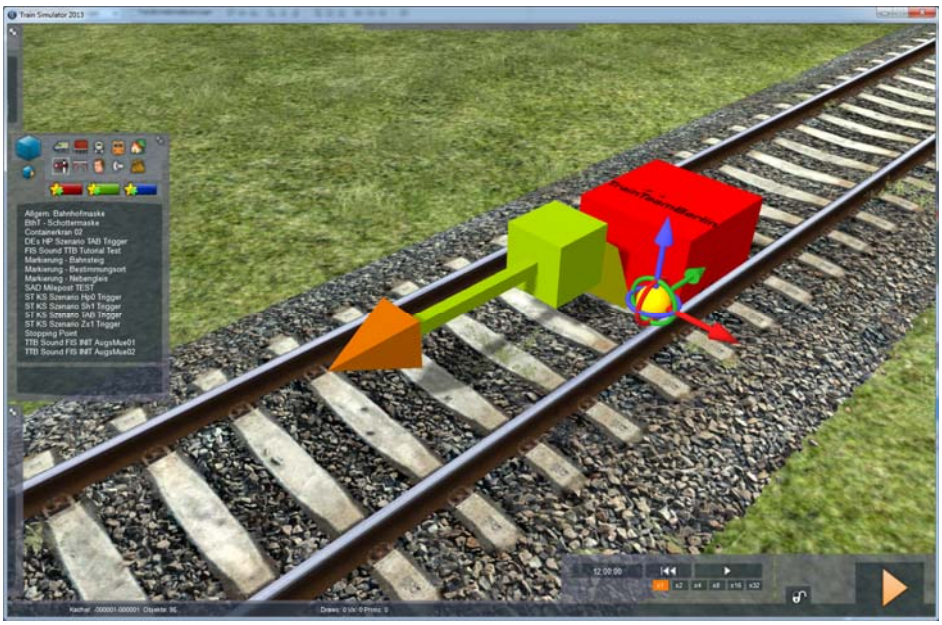
2.4. Konfigurationssignal im Szenarioeditor verbauen

Nun gilt es, die erstellten Ansagen und zugehörigen Konfigurationsdateien in einem konkreten Szenario zu verbauen. Wir setzen an dieser Stelle Kenntnisse in der Szenarienerstellung voraus.

Öffnen Sie nun den Railworks Szenario-Editor und erstellen Sie ein neues Szenario auf einer beliebigen Strecke. Platzieren Sie ein Fahrzeug, das mit dem 3DByte-Soundsystem ausgestattet ist und weisen Sie ihm einen Fahrer zu und markieren es ggf. als Spielerfahrzeug.

Nun muss im Objektgruppenfilter das Konfigurationssignal freigeschaltet werden. Dazu ist das Paket TrainTeamBerlin → SoundSystem zu aktivieren. Das Signal ist nun im Objektauswahlbereich "Gleisinfrastruktur" unter dem Namen "FIS Sound TTB Tutorial Test" zu finden.

Platzieren Sie das Signal mit einem linken Mausklick auf dem gewünschten Gleis im Fahrweg der platzierten Lokomotive. Es erscheint ein weißer Würfel mit TTB-Logo. Klicken Sie nun nochmals in der Nähe des Würfels auf das Gleis, es erscheint ein Pfeilsymbol, der Signallink - beim Überfahren des Signallinks wird die im Skript definierte Nachricht an das Fahrzeug gesandt. Dies geschieht jedoch nur, wenn der Link in Pfeilrichtung überfahren wird. Sollte der Pfeil entgegengesetzt der gewünschten Richtung zeigen, klicken Sie diesen mit gleichzeitig gedrückter Umschalttaste nochmals an, um die Ausrichtung zu wechseln.



Nun klicken Sie doppelt auf den Würfel mit dem TTB-Logo. Am rechten Bildschirmrand erscheint ein Eigenschaftenfenster, welches das zweigeteilte Namensfeld enthält. Geben Sie im rechten Textfeld die Nummer der Datei an, die an diesem Punkt abgespielt werden soll. Sie muss dem Index-Eintrag in der xml-Playlist entsprechen und wird ohne führende Nullen eingetragen.
Wenn Sie nochmals auf den Würfel klicken und das Fadenkreuz erscheint, können Sie diesen verschieben und unter der Erde verstecken.
Platzieren Sie nun noch ein zweites Konfigurationssignal, in dem Sie die zweite Datei der Playlist abspielen lassen. Danach sind die Änderungen des Szenarios nur noch zu speichern und dieses abzuspielen. Bei der Überfahrt über die Konfigurationssignale werden die jeweiligen Ansagen abgespielt.