

GStreamerの応用例… 倍速プレーヤを作る

松原 克弥

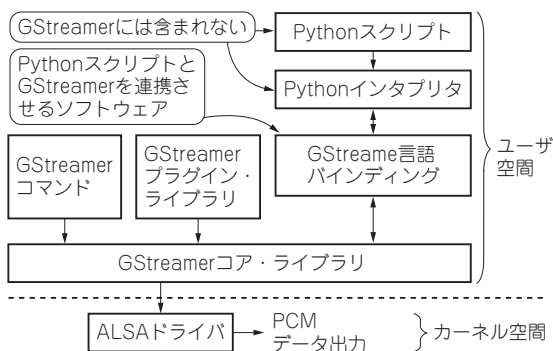


図1 コマンドラインを使わずともPythonスクリプトからGStreamerを実行できる

音声や動画などのマルチメディア・プレーヤを開発するためのライブラリ群GStreamerは、コマンドラインから実行するだけではなく、Pythonなどのスクリプト言語と連携しても実行できます。さらにPythonと連携できるソフトウェアを呼び出してGStreamerにはないGUIの機能も追加で実装できます。本章では、GStreamerとPythonの連携の例として倍速プレーヤを制作します。図1にソフトウェアの構成を示します。ハードウェア構成は第10章の図Aと同じです

● 再生速度を上げるだけではうまく倍速再生できない

speedプラグインは、再生速度を変更するプラグインです。図2に示すコマンドで実行します。

```
ubuntu@arm:~$ gst-launch-1.0 -v filesrc location=sample.m4a !
qt demux ! decodebin ! speed speed=2.0 ! alsasink device="hw:1,0" 2倍速で再生 ← 1行で記述
```

図2 speedプラグインで2倍速再生してみる

speedプラグインは、パイプラインに列挙された各エレメントへ再生速度の変更を指示できる。ただし、再生速度に比例して音程も上下する

```
ubuntu@arm:~$ gst-launch-1.0 -v filesrc location=sample.m4a !
qt demux ! decodebin ! audioconvert ! pitch pitch=2.0 !
audioconvert ! alsasink device="hw:1,0" 音程調整つきで2倍速再生 ← $より後を1行で記述
```

フォーマット変換のためにpitchの前後にaudioconvertプラグインが必要、この処理が高負荷となる

図3 pitchプラグインを使うと音程が変化しないで倍速再生できるが負荷が高い

pitchプラグインの入出力仕様に合わせてデータのフォーマット変換が必要。変換処理の負荷が高くなるのでBBBでは音飛びや再生できない場合もある

speedパラメータの値が1より大きい場合は早回し、1未満の値の場合はスロー再生になります。実際に再生してみるとわかりますが、再生速度に比例して音程も上下してしまい、再生速度を変更するだけでは聞き苦しい音になってしまいます。

● 音程も調整できるプラグインもあるがCPUへの負荷が高い

pitchプラグインは、再生速度を変更するだけでなく、再生速度に合わせて音程も調整できます。図3に示すコマンドで実行します。

ただし、pitchプラグイン入出力の対応が浮動小数点フォーマット(F32LE)のみであるため、audioconvertを使ったフォーマット変換が必要です。フォーマット変換は負荷が高く、しかも、フォーマット変換の処理も倍速で行う必要があるため、BeagleBone Blackのプロセッサでは、処理が間に合わなくなる可能性が高いです。

GStreamerは、システム・クロックに合わせてUSB-DACに出力を行うため、処理が間に合わなかったデータはドロップ(破棄)され、音飛びや無音状態が発生します。speedプラグインとpitchプラグインを使わず、音程の変わらない倍速再生を実現することはできないのでしょうか。

● 再生速度から必要に応じて音程の調整を自動的に行うプラグインを使う

scaletempoプラグインは、再生速度に合わせて