**Audio si MIDI in Linux**

**Serban Alexandru Constantin**

 **432A**

**Cuprins :**

1 . Semnale Audio si Linux-audio . ( Prezentarea stratulurilor de la hardware la software in Linux ) .

 2. Midi in Linux –( Transformarea de semnal midi in sunet , conectarea hardware a unui dispozitiv midi , utilizarea unor softuril sequencer pentru compozitie ) .

 3. Bibliografie .

Nota : Datorita complexitatii subiectului am decis sa prezint introducerea in midi sub forma unei brosuri cu ‘FAQ’ sau intrebari frecvent puse . Aceasta este pusa la dispozitie prin intermediul unei anexe ‘ Intromidi’ . Avand in vedere parcurgerea ei sau insusirea cunostintelor MIDI din alte surse , am prezentat in continuare subiectul strict legat de Linux .

Observatie : Diferente de formatare a textului pot aparea in functie de software-ul folosit pentru vizualizare .

1. **Semnale Audio si Linux-audio**

Multa lume considera ca exista o problema cu partea audio din Linux, și că nu funcționează întotdeauna. Problema consta in dificultatea acesteia. Acest lucru devine repede evident cand incercam sa vedem relațiile dintre tehnologiile implicate în transportul audio de la un fișier de muzică la difuzoare.

Dacă am desena modelul OSI folosit pentru a descrie cadrul de rețea care conectează aparatul la orice altă mașină din rețea, vom putea găsi straturi clare, fiecare cu propriul domeniu de procese și funcționalitate. Există foarte puține suprapuneri în straturi, și cu siguranță nu se găsesc procesele de utilizator final în stratul șapte încurcate cu impulsurile electrice ale fluxurile de biți prime în stratul unul.

Cu toate acestea, acest lucru este exact ceea ce se poate întâmpla cu framework-ul audio Linux. Nu avem chiar un nivel inferior clar definit, cu mai multe tehnologii audio avand de-a face cu kernel-ul și hardware-ul independent.

Open Sound Protocol , de exemplu, fi găsit la nivel de kernel comunicand cu hardware-ul direct, acum este un strat de compatibilitate care sta pe partea de sus a ALSA. ALSA în sine are un nivel stiva nucleu și un API mai mare pentru programatori, driver de mixare și proprietăți hardware cu capacitatea de a reda un sunet sau un codec MP3. Când cele mai multe distribuții au PulseAudio și GStreamer pe partea de sus, totul devine complicat si instabil.

****

Aici este o vedere simplificata a straturilor audio utilizate de obicei în Linux. Cel mai profund strat, aproape de hardware-ul este:

**ALSA :**

Intrări: PulseAudio, Jack, GStreamer, Xine, SDL, ESD

Iesiri: Hardware, OSS

Când vine vorba de Linux- audio modern , la început este arhitectura avansată de sunet Linux, sau ALSA. Acest lucru se conectează la nucleul Linux și oferă o funcționalitate pentru restul sistemului. Dar este, de asemenea, mult mai ambițioasa decât un kernel obișnuit , poate asigura compatibilitatea cu alte straturi, poate crea un API pentru programatori și să lucreze la o latență scăzută și stabilă, care poate concura cu ASIO și echivalentele CoreAudio pe Windows și platform OS X.

Este mai ușor să ne gândim la ALSA ca stratul driver de de sunet al Linux. Hardware-ul audio are nevoie de un modul de kernel corespunzător, precedat de snd\_, și acest lucru trebuie să fie încărcat și rulat pentru ca ceva să se întâmple. De aceea avem nevoie de un driver de kernel ALSA pentru ca orice sunet să fie auzit. Din fericire, cele mai multe distributii vor configura automat dispozitivele dvs. și modulele.

ALSA este responsabil pentru traducerea capacităților hardware-ului audio într-un software API pe care de sistemul il foloseste pentru a manipula sunetul. Acesta a fost conceput pentru a aborda multe din neajunsurile OSS (și multe alte drivere de sunet la acel timp),cel mai notabil fiind ca o singura aplicatie putea accesa partea hardware la un moment dat . Acesta este motivul pentru care o componentă software în ALSA trebuie să gestionează cererile audio și sa înțeleaga capacitățile hardware-ului.

Dacă vrem să ne jucam în timp ce ascultam muzică, ALSA trebuie să fie în măsură să ia atât de aceste fluxuri audio și sa le mixeze împreună în software-ul, sau sa foloseasca un mixer hardware de pe placa de sunet în același sens. ALSA poate administra, de asemenea, până la opt dispozitive audio și uneori accesa funcționalitatea MIDI pe hardware-ul, deși acest lucru depinde de specificațiile hardware audio .

****

**O imagine cu mixerul ALSA.**

Modul in care ALSA difera de modulele kernel tipice sau de driverele dispozitivelor consta in posibilitatea de configurare din partea utilizatorului Acum ,complexitatea în Linux audio începe să apară, deoarece puteți modifica aproape toata configuratia ALSA prin crearea propriului fișier de configurare - de la modul în care fluxurilee audio sunt mixate și prin ce iesiri parasesc sistemul, la rata de eșantionare , efecte bit profunzime și în timp real.

Transparența relative a ALSA , eficiența și flexibilitatea au ajutat să-l standardizeze pentru Linux- audio, si sa devina stratul perin prin care aproape fiecare cadru audio trebuie să treacă in scopul de a comunica cu hardware-ul audio .

**PulseAudio :**

Intrări: GStreamer, Xine, ALSA

Iesiri: ALSA, Jack, ESD, OSS

Insa , dacă ne gândim că lucrurile se vor obține mai ușor în condițiile de siguranță ALSA ne înșelam . ALSA acoperă cele mai multe probleme in a adduce semnalul in si din masina de calcul , dar urmeaza un alt strat de complexitate. Acesta este domeniul PulseAudio - o încercare de a reduce decalajul dintre capacitățile hardware și software, mașini locale și la distanță, precum și conținutul de fluxuri audio. PulseAudio face pentru retele ce face ALSA pentru mai multe plăci de sunet, și a devenit un fel de standard, în multe distributii Linux datorita flexibilitatii sale.

Ca și în cazul ALSA, această flexibilitate aduce complexitate, dar problema apare la PulseAudio, deoarece nu este mai orientat spre utilizator. Acest lucru înseamnă că utilizatorii normali au mai multe sanse de a s se prinde in panzele sale .

La prima vedere, PulseAudio nu pare să adauge ceva nou la Linux-audio, motiv pentru care se confruntă cu atât de mult ostilitate. Nu simplifica ceea ce avem deja sau face sunetul mai robust, dar , de fapt ,adaugă mai multe caracteristici importante. Este, de asemenea,’ catch-all layer ‘ pentru aplicații audio Linux, indiferent de capacitățile lor individuale sau specificațiile hardware-ului.

****

**PulseAudio este puternic, dar de multe ori ridiculizat pentru ca face audio Linux și mai complicat.**

În cazul în care toate aplicațiile ar utiliza PulseAudio, lucrurile ar fi simple. Dezvoltatorii nu ar avea nevoie să isi faca griji privind complexitatea altor sisteme, deoarece PulseAudio aduce compatibilitate cross-platform. Spre deosebire de ALSA, PulseAudio poate rula pe mai multe sisteme de operare, inclusiv alte platforme POSIX și Microsoft Windows. Acest lucru înseamnă că, dacă vă construiți o aplicație care să utilizeze PulseAudio, mai degrabă decât ALSA, portarea pe o platformă diferită ar trebuie să fie ușor de facut .

Dar există o relație simbiotică între ALSA și PulseAudio pentru că, pe sistemele Linux, acesta din urmă are nevoie de ALSA pentru a supraviețui. PulseAudio se configurează ca un dispozitiv virtual conectat la ALSA, la fel ca orice altă piesă de hardware. Acest lucru face ca PulseAudio sa semene mai mult cu Jack, pentru că stă între ALSA și desktop, conducand fluxuri de date înainte și înapoi transparent. El are, de asemenea, propria sa terminologie. ‘Sinks’, de exemplu, reprezinta destinațiile finale. Acestea ar putea fi o altă mașină de pe rețea sau ieșirile audio de pe placa de sunet . Partile din PulseAudio care umplu aceste ‘sinks’ sunt numite "surse" - de obicei aplicații audio-generatoare de pe sistemul dumneavoastră, intrari audio de la placa de sunet, sau un flux audio rețea trimise de la un alt aparat PulseAudio.

**GStreamer :**

Intrări: Phonon

Intrari: ALSA, PulseAudio, Jack, ESD

Incepand cu GStreamer, Linux-audio începe să arate chiar mai confuz. Acest lucru se datorează faptului că, la fel ca PulseAudio, GStreamer nu pare să adauge ceva nou pentru mixare . Este un alt cadru de lucru multimedia și a câștigat un numar rezonabil de dezvoltatori în anii de dinainte de PulseAudio, mai ales pe desktop Gnome. Este una din puținele metode de a instala și de a folosi codecuri proprietare cu ușurință pe desktop Linux. Este, de asemenea, cadrul audio de alegere pentru dezvoltatori GTK, și puteți găsi chiar și o versiune de manipulare audio pe Palm Pre.

 GStreamer este unic, deoarece nu este proiectat exclusiv pentru audio - suporta mai multe formaturi media de streaming, inclusiv video, prin utilizarea de plugin-uri.

Redare MP3, de exemplu, este în mod normal adăugata la sistemul dvs printr-o descărcare de codec suplimentar, care se atașează ca un plugin GStreamer. Comercialul Fluendo MP3 player , unul dintre codec-urile autorizate în mod oficial disponibile pentru Linux, este furnizat ca un plug-in GStreamer, așa cum sunt si alte codecuri proprietare, inclusiv MPEG-2, H.264 și MPEG.

**Jack :**

Intrări: PulseAudio, GStreamer, ALSA,

Iesiri : OSS, FFADO, ALSA

În ciuda avantajelor de configurații deschise, cum ar fi PulseAudio, toate acestea conduc semnalul audio intre aplicatii cu prezumtia ca acesta va ajunge direct la iesire . Jack este stratul de mijloc - echivalentul audio al apelurilor de procedură la distanță în programare, permițând ca aplicațiile audio să fie construite dintr-o varietate de componente.

Cel mai bun exemplu este un studio de inregistrari virtual, unde o aplicatie este responsabila cu obtinerea de semnal audio , iar o alta aplicatie cu efectele ; înainte de a trimite în cele din urmă fluxul rezultat printr-un procesor de mastering , semnalul trebuie pregătit pentru lansare. Un studio de înregistrare real, s-ar putea folosi de o rețea de cabluri, pentru a construi aceste legături. Jack face la fel și în software.

Jack este un acronim pentru "Jack Audio Connection Kit". Este construit pentru a fi cu latență redusă, ceea ce înseamnă că nu există nici o procesare necorespunzătoare e care ar putea împiedica progresul audio. Dar, pentru ca Jack să fi util, o aplicație audio trebuie să fie conceputa în mod special să se ocupe de conexiuni Jack. Ca urmare, nu este un simplu înlocuitor pentru place de ALSA și PulseAudio, și trebuie să fi rulat pe partea de sus de un alt sistem care va genera un sunet și oferă intrări fizice.



Cu Jack, vă puteți conecta ieșirea audio de la aplicații la intrarea audio de la alte aplicatii - la fel ca într-un studio de inregistrari reale.

Cu cele mai multe aplicații Jack-compatibile, suntem liberi să rutam audio și intrări în funcție de cum dorim. Am putea lua ieșirea de la VLC, de exemplu, și sa o conducem direct în Audacity pentru a înregistra fluxul pe care acesta s ail poate reda inapoi.Sau am putea să-l trimitem prin JackRack, o aplicație care ne permite să construim un turn de efecte în timp real, inclusiv delay , ping, vocodare voluptoase.

Această versatilitate este fantastica pentru stațiile de lucru audio digitale. Ardour folosește Jack pentru conexiuni interne și externe, de exemplu, și procesorul Jamin pentru masterizare poate fi utilizat numai ca parte a unui lanț de procese Jack. Este echivalentul controlului deplin asupra modului în care este setat fir cu fir studioul dumneavoastră. Punerea sa în aplicare a avut atât de mult succes pe desktop Linux, incat Jack a fost implementat similar si pe alte sisteme de operare.

**FFADO :**

Intrări: Jack

Iesiri: hardware audio

În lumea audio profesionala și semi-profesionala, multe interfețe audio se conectează la mașină gazdă folosind un port FireWire. Această abordare are multe avantaje. FireWire este rapid și dispozitivele pot fi direct alimentate. Multe laptopuri și mașini de birou au porturi FireWire, fără nici o modificare în continuare, iar standardul este stabil și in cea mai mare parte matur.

Dar, spre deosebire de USB, în cazul în care există un standard pentru manipularea audio fără drivere suplimentare, interfețe audio FireWire au nevoie de propriile drivere. Complexitatea protocolului FireWire înseamnă ca acesta nu poate folosi cu ușurință o interfață ALSA, deci are nevoie de propriul lui strat. Inițial, acest lucru a scăzut la un proiect numit FreeBOB. Aceasta a profitat de faptul că mai multe dispozitive audio FireWire s-au bazat pe același hardware.

FFADO este succesorul FreeBOB, și deschide platforma driver la multe alte tipuri de interfata audio FireWire. Versiunea 2 a fost lansat la sfârșitul anului 2009, și include suport pentru mai multe unități de sunet produse de Alesis, Apogee, ART, CME, Echo, Edirol, Focusrite, M-Audio, Mackie, Terratec.

O altă caracteristică eleganta, în FFADO este că unele caracteristici DSP ce tin de hardware au fost integrate în driver, complet cu un mixer grafic pentru controlul echilibrului diferitelor intrări și ieșiri. Acest lucru este diferit de mixer-ul ALSA, pentru că înseamnă fluxuri audio ce pot fi controlate pe hardware-ul de la zero latenta, este exact ceea ce avem nevoie, pentru înregistrarea live.

**Xine:**

Intrări: Phonon

Iesiri: PulseAudio, ALSA, ESD

Am început să intram în nișa geologica a Linux-audio. Xine este ceea ce a mai rămas după multe alte straturi audio ce au fost eliminate . Cei mai mulți utilizatori vor recunoaste numele dupa capacitatile extraordinare de redare DVD si media player , pe care cele mai multe distribuții le includ încă in pachet, în ciuda vârstei sale; asta defininind cheia pentru longevitatea Xine .

Când Xine a fost creat, dezvoltatorii l-au divizat într-o bibliotecă de back-end care sa se ocupe de media, și o aplicație front-end pentru interacțiunea cu utilizatorul. Este biblioteca cea care a persistat, datorită capacității sale de a reda numeroase extensii, inclusiv AVI, Matroska și Ogg, și zeci de formate pe care le conțin, cum ar fi AAC, FLAC, MP3, Vorbis și WMA. Ea face acest lucru prin valorificarea competențelor numeroase alte bibliotecii. Ca urmare, Xine poate acționa ca un cadru catch-all pentru dezvoltatorii care doresc să ofere cea mai bună gamă de compatibilitati , fără griji cu privire la legalitatea de codecuri proprietare și brevete.

Xine poate vorbi cu ALSA și PulseAudio pentru producția sa, și există încă multe aplicații care pot vorbi cu xine direct. Cele mai populare sunt front-end gxine și Totem, dar Xine este, de asemenea, implicit back-end pentru Phonon KDE, astfel încât il puteți găsi in orice, de la Amarok la Kaffeine.

**Phonon :**

Intrări: aplicatii Qt si KDE

Iesiri: GStreamer, Xine

Phonon a fost conceput pentru a face viața mai ușoară pentru dezvoltatori și utilizatori prin eliminarea unora dintre complexitățile sistemului. A început viața ca un alt nivel de abstractizare audio pentru aplicatii KDE 4, dar a fost considerat o idee bună .

Acest lucru a avut avantaje pentru dezvoltatorii de aplicatii cross-platform. Ea a făcut posibilă pentru a scrie un music player pe Linux cu Qt și pur și simplu recompilarea pentru OS X și Windows, fără griji despre modul in care muzica ar fi redat, capacitățile de hardware de sunet folosite, sau modul în care sistemul de operare destinație ar ocupa audio. Acest lucru a fost făcut în mod automat de către Qt și Phonon, trecand audio la CoreAudio API pe OS X, de exemplu, sau DirectSound pe Windows. Pe platforma Linux (și spre deosebire de versiunea KDE originală a Phonon), Phonon Qt trece audio la GStreamer, mai ales pentru transparenta codecului sau .

2 . Midi in Linux .

Pentru expunerea subiectului Midi in Linux am decis sa exemplific pasii necesari configurarii partii audio si midi . Avand in vedere multitudinea de pasi si de cunostinte ce sunt necesare , putem intelege de ce nu multa lume opteaza in productia muzicala pentru Linux . De asemenea pentru a intelege urmatoarea expunere este nevoie de cunostinte MIDI , ce pot fi insusite parcurgand anexa ‘ Intromidi ‘ .

**Grup audio** :

Software-ul audio are nevoie de a rula la o prioritate mai mare și cu memorie blocată, astfel încât să nu se interschimbe cu hard disk-ul. Pentru a da unui utilizator aceasta putere, vom crea un grup "audio", vom da grupului unele privilegii speciale, apoi vom adauga utilizatorul în acel grup.

În cazul în care sistemul dumneavoastră nu are un grup "audio", încercați acest lucru pentru a va asigura:

sudo groupadd audio

Sunt șanse bune sa vedeti un mesaj care indică faptul că grupul "audio" există deja. Asta e bine. Apoi, vom da grupul audio unele privilegii ridicate. Pentru aceasta avem nevoie sa facem unele modificări la fișierul / etc / security / limits.conf. Editează acest fișier cu editorul dvs. favorit. Eu folosesc, de obicei, Nano:

sudo nano /etc/security/limits.conf

Acum, adăugați următoarele linii la sfârșitul fișierului limits.conf:

@audio - rtprio 100

@audio - nice -10

@audio - memlock 429612

Aceste linii dau grupului audio abilitatea de a ridica prioritate pana la -10 (cea mai mare este de -20), iar prioritatea în timp real la 100 (cea mai mare este de 99). Toate software-ele audio necesită abilitatea de a ridica prioritate astfel încât sincronizarile sa fie conforme. Transformand patrimile in doimi nu este un lucru de dorit in muzica.

Linia memlock permite unui membru al grupului audio sa reserve 430MB de memorie. Fluidsynth-ului , de exemplu, îi place să aibă 430MB de memorie pentru a lucra și va emite mesaje de eroare, dacă nu-I va lua. Pentru alte configurații, ar putea fi capabil sa reduca acest număr.

Acum, avem nevoie sa ne adăugam la grupul audio. Putem utiliza gpasswd pentru a face acest lucru. Presupunând că numele dvs. de utilizator se întâmplă să fie "Alex":

sudo gpasswd -a alex audio

Această schimbare nu va avea efect imediat. Trebuie să dati logout apoi conectați-l din nou. Utilizați comanda "groups" pentru a vedea dacă au fost adăugate cu succes la grupul audio.

**Fluidsynth :**

Fluidsynth este un sintetizator software sau "softsynth". Poate transforma datele MIDI in sunete prin utilizarea unui "SoundFont". Pe o distribuție bazata apt (Ubuntu, Debian, Mint ...), puteți face următoarele pentru a obține fluidsynth și sa instalati un SoundFont:

sudo apt-get install fluidsynth
sudo apt-get install fluid-soundfont-gm

Alte distributii ar trebui să aibă la dispoziție pachete similare. Pentru a reda un fișier MIDI numit "song.mid":

fluidsynth --audio-driver=alsa /usr/share/sounds/sf2/FluidR3\_GM.sf2 song.mid

To stop fluidsynth, type "quit" at its ">" prompt. We'll need to stop fluidsynth for the next section.

**Aplaymidi**

În loc sa sa avem fluidsynth ce reda un fișier MIDI, putem, de asemenea, utiliza fluidsynth pentru a face muzica din datele MIDI ce vin de la alte programe. Pentru a testa acest lucru, vom folosi programul aplaymidi care face parte din pachetul "alsa-utils" În distributiile bazate apt :

udo apt-get install alsa-utils

Acum vom folosii fluidsynth ca un server. Acest lucru înseamnă că va rula și aștepta ca alte programe să se conecteze la acesta și sa trimita datele MIDI.

fluidsynth --server --audio-driver=alsa /usr/share/sounds/sf2/FluidR3\_GM.sf2

Acum va trebui sa conectam aplaymidi cu fluidsynth :

aplaymidi –l // vom afla la ce porturi sunt conectate acestea ;

Port Client name Port name

 14:0 Midi Through Midi Through Port-0

128:0 FLUID Synth (2825) Synth input port (2825:0)

Fluidsynth este pe portul 128:0 . Vom folosi aceasta informative pentru a instiinta aplaymidi sa trimita date catre acest port :

aplaymidi -p 128:0 song.mid

**Tastatura MIDI virtuala :**

In cazul în care nu aveți o tastatură MIDI fizica, puteți utiliza una virtuala. Pentru acest exemplu , vom folosi vmpk, Virtual MIDI Piano Keyboard. Pentru a instala și rula:

sudo apt-get install vmpk
vmpk &

Nu va merge până nu o vom conecta la fluidsynth. Vom folosi comanda "aconnect" pentru a face acest lucru. În primul rând, avem nevoie pentru a verificape ce porturi sunt MIDI fluidsynth și vmpk .

"aconnect-i" ne va arăta MIDI porturile de "intrare":

$ aconnect -i

client 0: 'System' [type=kernel]

 0 'Timer '

 1 'Announce '

client 14: 'Midi Through' [type=kernel]

 0 'Midi Through Port-0'

client 129: 'VMPK Output' [type=user]

 0 'VMPK Output '

Din aceasta putem vedea că "VMPK out" este pe 129:0. Notă : "0" pe linia "VMPK out" este 0 după două puncte. În continuare vom folosi "aconnect-o", pentru a ne asigura ca fluidsynth functioneaza normal :

$ aconnect -o

client 14: 'Midi Through' [type=kernel]

 0 'Midi Through Port-0'

client 128: 'FLUID Synth (3206)' [type=user]

 0 'Synth input port (3206:0)'

client 130: 'VMPK Input' [type=user]

 0 'VMPK Input '

Fluidsynth este la 128 . Acum le vom conecta pe cele 2 :

aconnect 129:0 128:0

Acum vom auzi pian cand vom apasa tastele vmpk .

**Configurarea unui dispozitiv MIDI hardware :**

Există mai multe controlere MIDI pe piata , ce se conecteaza prin USB. Eu voi testa Akai LPK25. Când am conectat și m-am uitat la dispozitivele mele MIDI, am putut vedea următoarele:

$ aconnect -i

client 0: 'System' [type=kernel]

 0 'Timer '

 1 'Announce '

client 14: 'Midi Through' [type=kernel]

 0 'Midi Through Port-0'

client 20: 'LPK25' [type=kernel]

 0 'LPK25 MIDI 1 '

Deci 20:0 este LPK25. Il pot conecta la fluidsynth (care se întâmplă să fie la 129:0 pentru mine acum), folosind patchage sau aconnect:

aconnect 20:0 129:0

**MIDI Sequencers :**

După ce ne-am dat seama cum să conectam toate componentele hardware, MIDI și softsynths, putem instala și folosi un sequencer MIDI pentru compoziția de muzică. Două dintre cele mai bune sunt Rosegarden și hidrogen.

**Rosegarden**

Rosegarden este un sequencer MIDI, care oferă înregistrare și redare multi-track, împreună cu editare notelor.

Rosegarden are nevoie de un pic de configurare pentru a-l folosi . Uneori, aceasta preia lucrurile in mod automat, dar alte ori, va trebui să se meargă la Studio> Administrare Dispozitive MIDI aici vom fi capabili să se conectla Rosegarden la synth-uri , tastaturi și alte dispozitive MIDI. Iată un screenshot de Rosegarden creat pentru fluidsynth și Akai mea LPK25:



**Hydrogen**

Hydrogen este un sequencer-tobe. Are in componenta sa un soft-synth ceea ce inseamna ca poate fi folosit direct .



1. Bibliografie

<http://sourceforge.net/apps>

[http://www.tldp.org](http://www.tldp.org/)

<http://tedfelix.com/linux>

<http://linux-sound.org/>

<http://xed.ch/help/midi.html>

<http://en.wikibooks.org/wiki/Configuring_Sound_on_Linux>

<https://wiki.archlinux.org/index.php/Sound_system>

<http://www.alsa-project.org/main/index.php/Main_Page>

<http://en.wikipedia.org/wiki/Advanced_Linux_Sound_Architecture>

<http://en.wikipedia.org/wiki/List_of_Linux_audio_software>

# The MIDI Manual: A Practical Guide to MIDI in the Project Studio - [David Miles Huber](http://www.google.ro/search?tbo=p&tbm=bks&q=inauthor:%22David+Miles+Huber%22)

[**Electronic Music and MIDI Projects**](http://www.amazon.com/dp/1870775244?tag=midiworld&camp=14573&creative=327641&linkCode=as1&creativeASIN=1870775244&adid=0A70XAJFRQ1WW6A3RYZS&&ref-refURL=http%3A%2F%2Fwww.classicalmidiconnection.com%2Fbooks.htm)**-R. A. Penfold**