

Štefan Kiss

Do útrob MIDI

Základy systému MIDI

pre všetkých tých, ktorí chcú MIDI porozumieť

Cieľom tohto seriálu je priblížiť základy problematiky MIDI všetkým tým, ktorí sa o MIDI len zaujímajú, ale aj tým, ktorí pri svojej práci s hudbou využívajú počítač s midi rozhraním a software Notedit. Tento program obsahuje okrem iných možností aj nástroj Event editor, ktorým je možné veľmi dôkladne preskúmať obsah midi súborov a upraviť iba požadované hodnoty, pričom po uložení ostane súbor takmer identický s originálom. Akákoľvek úprava midi súborov cez Event editory si však vyžaduje znalosť MIDI, pretože tu nepracujeme s notami, ale číslami a príkazmi. Vymazať nesprávny riadok, alebo vložiť nesprávnu hodnotu by mohlo mať nepríjemné následky. Ak vás však problematika MIDI zaujíma a chcete príkazom a číslam porozumieť, preštudujte si dôkladne tento seriál a potom sa už nemusíte ničoho báť.

Podotýkam, že tento seriál si kladie za cieľ priblížiť iba základy MIDI v rozsahu, aký je potrebný pre bežného užívateľa Noteditu a jeho Event editoru. Preto sú aj informácie v tomto seriáli podávané tak, aby zodpovedali rozhraniu, ktoré prináša program Notedit. Technickí znalci mi preto odpustia občasné termínové nepresnosti, ktoré sú úplne zámerné.

Snažil som sa tento seriál napísať tak, aby aj zdanlivo náročným veciam bolo možné porozumieť. MIDI sa dá pochopiť, ak je problematika správne vysvetlená. Predkladám vám preto tento seriál v nádeji, že práve túto požiadavku u vás do bodky naplní.

Obsah

Obsah.....	2
1. Čo je midi a ako funguje	4
Čo je midi	4
Ako to funguje?.....	5
Čo je sekvencer	6
Midi prehrávače.....	6
Prečo nahrávať do midi?	7
Midi kanály	7
2. Číselné sústavy	9
Číselná sústava	9
Šestnástková – hexadecimálna číselná sústava	10
Dvojková - binárna sústava	11
Ďalšie číselné sústavy.....	12
Prevod z dvojkovej na desiatkovú sústavu.....	12
Prevodná tabuľka čísel desiatkovej, dvojkovej a šestnástkovej sústavy.....	13
3. Bity a bajty	19
Bajty	19
Bity	19
Bity a bajty v midi	21
Stavové bajty	21
Dátové bajty	21
4. Midi hodiny, čas a tempo	23
Čas a notový zápis	23
Čas a midi	23
Midi hodiny	24
Tempo.....	24
Midi tiky	25
5. Štruktúra a možné typy midi súboru	28
Štruktúra midi súboru.....	28
Tracky.....	28
Typy midi súborov	29
6. Hlavička midi súboru	30
7. Štruktúra a typy midi správ	31
Čo je midi udalosť	31
Štruktúra midi správy	31
Typy midi správ	32
Kanálové správy	32
Nekanálové správy	33
Systémové správy.....	33
Meta správy	33
SysEx správy	33
8. Kanálové správy	34
Číslovanie nôt.....	34
\$9K – Note on – zapnutie noty.....	34
\$8K – Note off – vypnutie noty	35
\$AK - Polyphonic aftertouch – individuálna tlaková citlivosť	36
\$BK – Controllers – Controll mode change - - kontroléry	36

\$CK – Program change – zmena programu	37
\$DK – Chanel aftertouch – spoločná tlaková citlivosť	37
Dvojbajtové správy – MSB a LSB	38
\$EK – Pitchband – ohýbanie tónu	39
Tabuľka kanálových správ	40
Tabuľka tónov a ich midi čísel	40
Tabuľka 1:	40
Tabuľka 2:	41
9. Kontroléry	42
Čo sú kontroléry	42
1. Kontinuálne kontroléry – potenciometre (0-63).....	42
2. Spínače (64-79)	43
3. Modulačné kontroléry (80-95)	43
4. Špeciálne kontroléry (96-119).....	44
5. Špeciálne riadiace povely (120-123).....	45
6. Režimové povely (124-127).....	45
Tabuľka kontrolérov	46
10. Voľba zvukov	48
Norma GM	48
Zvukové banky	49
Ako v midi zvoliť zvuk	49
Tabuľka zvukových rodín podľa normy GM	51
Tabuľka zvukov v banke podľa normy GM.....	51
11. Systémové správy	55
Čo sú systémové správy	55
1. Zvláštne systémové správy – SysEx	55
2. Systémové správy reálneho času	55
3. Spoločné systémové správy	56
Tabuľka systémových správ	57
12. Meta správy	58
Čo sú meta správy	58
1. Textové meta správy	58
2. Číselné správy	59
3. Ostatné meta správy	60
Tabuľka meta správ	60
Tabuľka taktov a ich midi označení	61
13. Systémové exkluzívne správy – SysEx	62
14. Midi normy a režimy	64
GM norma – General midi	64
GS norma – General standard	64
XG norma – Extended GM – Rozšírené GM	64
GM2 norma – General midi 2	65
Vzájomná kompatibilita a resety	65

1. Čo je midi a ako funguje

Čo je midi

Predstavte si veľký symfonický, alebo hoci aj malý sláčikový orchester, skrátka hudobné teleso, kde hrá viac hudobníkov. Každý má svoj hudobný nástroj a v jednej chvíli môže hrať iba na ňom. Ťažko by huslista hral na husliach i klarinete naraz. Práve preto si orchestre, kde má hrať mnoho nástrojov, vyžaduje aj mnoho hudobníkov. Keď však prišli na svet elektronické hudobné nástroje, vznikla myšlienka, že tu by azda mohol jeden hráč hrať aj na viacerých nástrojoch. A tak začali viaceré firmy, vyrábajúce elektronické hudobné nástroje, experimentovať. Dielo sa z časti aj vydarilo. Stačilo káblom prepojiť napríklad dva syntetizátory a ako náhle hudobník stlačil klávesu na jednom z nich, ozval sa tón z oboch súčasne. Takto sa mohli využiť viaceré zvuky naraz. Hudobník si na jednom nástroji nastavil klavír, na druhom sláčiky a hraním na klaviatúre vznikol krásny sýty zvuk. Povieme si, že to pre nás nie je nič nové, veď my dnes bez problémov nastavíme klavír a sláky spolu na jedinom nástroji. To síce áno, ale ja hovorím teraz o polovici minulého storočia, kedy takéto možnosti neboli a nástroje boli maximálne jednoduché. Avšak práve ich vzájomným prepojením sa už vtedy dala dosiahnuť väčšia sýtosť zvukov. Hudobník hral na jednom z nástrojov a na jeho ruky reagovali všetky pripojené nástroje a to nie len tým, ako hudobník hral, ale aj ako menil hlasitosť, pridával či uberal vibráciu alebo moduláciu, ako menil zvuky. Skrátka všetko, čo hudobník urobil na jednom nástroji, sa vykonalo na všetkých ostatných. Tak vlastne vznikol systém midi – teda spôsob, ako prepojiť hudobné nástroje a navzájom ich riadiť. Midi je skratka z anglického Musical Instruments Digital Interface, čo v preklade znamená Digitálne rozhranie hudobných nástrojov a prvé pokusy o takéto prepojenie dali základ systému midi.

Ale bol tu jeden závažný problém. Prepojiť sa dali iba tie nástroje, ktoré si vzájomne rozumeli. Každý úkon hudobníka, teda stlačenie klávesy, zmena hlasitosti, zmena zvuku atď. musel byť zakódovaný do nejakej správy, do nejakého príkazu, ktorý sa káblom okamžite odoslal do pripojeného nástroja a ten musel prijatú správu zasa správne rozšifrovať. Ak boli nástroje vzájomne kompatibilné, urobil druhý nástroj presne to, čo mal. Ak ale kompatibilné neboli, neurobil druhý nástroj nič, alebo urobil dačo úplne iné. Hudobník práve stlačil sustainový pedál a z druhého nástroja zaručala trúba plnou hlasitosťou. Áno, aj to sa mohlo ľahko stať, ak si prijímací nástroj zle prečítal prijatý príkaz. Žiaľ, bolo veľmi málo navzájom kompatibilných nástrojov. Vlastne by sa dalo povedať, že kompatibilné boli iba nástroje jednej značky, pretože každý výrobca si vymyslel iný spôsob šifrovania správ. A tak správa, ktorá u jednej značky znamenala „stlač pedál sustain“, u inej značky znamenala „trúb čo najhlasnejšie ako sa len dá“. Nástroje rôznych výrobcov sa tak veľmi kombinovať nedali.

Prišiel však rok 1981 a v New Yorku sa stretli výrobcovia najznámejších značiek elektronických hudobných nástrojov, aby vypracovali a neskôr prijali spoločný spôsob šifrovania muzikantských správ pre hudobné nástroje. Vznikla tak norma, ktorá dostala názov General midi – skrátene GM. Ako ale zistíme neskôr, GM nie je jediný štandard, no bol prvý a dodnes je považovaný za základný. Tým, že ho výrobcovia začali používať vo svojich nástrojoch teda nastala doba, kedy takmer všetky hudobné nástroje sú schopné medzi sebou komunikovať.

Ako to funguje?

Predstavme si dva moderné elektronické hudobné nástroje – nazývajte ich pre zjednodušenie ľudovým pojmom „klávesy“. Ak chceme tieto dva naše klávesy prepojiť cez midi, uistíme sa, že oba midi podporujú, teda majú na svojej zadnej stene prípojku midi. Ak sme si nekúpili tie najlacnejšie nástroje, je veľmi pravdepodobné, že midi majú. Na midi prepojenie sa používa midi kábel, ktorý je na oboch stranách zakončený okrúhlym päťkolíkovým konektorom. Na zadnej stene nástroja sú zasa dve alebo tri kruhové prípojky s označením out a in (v prípade že sú tri, je tretia označená ako thru). Midi je systém jednosmernej komunikácie, príkazy preto pôjdu iba jedným smerom. Pri prepájaní našich dvoch klávesov si teda musíme vybrať, odkiaľ kam majú príkazy ísť. Jeden nástroj, ten na ktorom budeme hrať, bude teda riadiaci (nazvime ho master) a ten druhý, ktorý bude iba poslúchať, prijímať správy a vykonávať ich bude podriadený (nazvime ho slave). Kábel na nástroji master pripojíme do konektora out (z anglického von), pretože tadiaľ správy vychádzajú a na nástroji slave ho pripojíme do konektora in (z anglického dnu, do), lebo tade nástroj správy prijíma. Keď teraz na nástroji master stlačíme niektorú klávesu – napríklad C1 -, nástroj master začne hrať tón C1, ale zároveň okamžite vyšle midi správu do pripojeného slave nástroja, že bola stlačená klávesa C1. Nástroj slave teda tiež zapne notu C1 a dokonca notu bude hrať rovnakou silou. Ak pritlačíme k prvej aj nejakú inú notu – napríklad E1 -, zasa sa odošle správa „zapni notu E1 takou a takou silou“ a nota sa ozve z oboch nástrojov. Ak teraz obe klávesy naraz pustíme, vyšlú sa rýchlo za sebou dve správy, ktoré noty vypnú aj na pripojenom slave: „vypni C1“ a „vypni E1“ a je ticho. Toto isté sa deje, keď pritlačíme pedál, keď prepneme zvuk, keď otočíme kolieskom hlasitosti, alebo keď spustíme automatický doprovod. Každý náš úkon sa okamžite prevedie na nejakú číselnú správu, ktorú si nástroj slave správne prečíta a urobí to isté.

Ak som povedal, že midi prepojenie je jednosmerné, stále to platí, no napriek tomu môžeme u dvoch klávesov dosiahnuť aj obojsmerné prepojenie. Urobíme to jednoducho tak, že vezmeme dva káble. Jeden zapojíme v nástroji master do konektora out a v nástroji slave do konektora in a ten druhý naopak. Teraz budú oba nástroje reagovať na seba navzájom podľa toho, ktorý práve do konektora in dostane nejakú správu.

Pre zjednodušenie hovorím stále o dvoch klávesových nástrojoch. Midi sa však dnes už rozšírilo natoľko, že pomocou neho možno prepojiť nielen klávesové nástroje, ale aj gitary, bicie, počítače, sekvencery, ba dokonca aj audio techniku, video techniku alebo osvetlenie. Midi správy umožňujú niesť v podstate akúkoľvek informáciu a dôležité je len to, aby jej prijímací prístroj správne porozumel. Ak by sme teda mali reflektor, ktorý sa na určitú midi správu zapne a zasa na inú vypne, nebol by pre jedného klaviristu problém hrať a pritom v určitom okamihu poslať správu reflektoru, aby stlmil svetlo, alebo aby naopak, zasvietil naplno. Rovnako nie je problém k midi zariadeniu pripojiť počítač, ktorý bude tieto povely posielat' automaticky a tým dokonca hudobníkovi ušetrí čas i prácu. Práve pre nesmierne široké možnosti využitia midi nemôžeme myslieť iba na klávesové nástroje a vlastne akýkoľvek prístroj, ktorý dokáže pracovať s midi musíme nazývať dôsledne midi zariadenie. Ja však v tejto i ďalších častiach seriálu ostanem pri predstave dvoch klávesov alebo klávesov a počítača, aby sme sa držali vo svete, ktorý je pre väčšinu z nás reálny – vlastný počítač a klávesový nástroj.

Čo je sekvencer

Ľudia si vždy túžili uľahčiť prácu a prehrávanie hudby nejako zautomatizovať. Možno za tým bola lenivosť, možno túžba zachovať peknú melódiu. Tak už v dávnej minulosti nachádzame prístroje, ktoré zaznamenávali hudbu. Spomeňme si na hracie skrinky či trhové verklíky. Boli to vlastne prvé midi prehrávače, pretože neobsahovali záznam hudby, ale nôt.

Sekvencer je prístroj, ktorý dokáže prijímať midi správy a ukladať si ich do pamäte, alebo uložiť do súboru. Vznikne tak zoznam príkazov, ktoré ak sekvencer zasa pošle späť do midi zariadenia, - teda do našich kláves -, dôjde k zopakovaniu toho, čo sme predtým my sami urobili. Sekvencer vytvorí súbor s príponou mid, ktorý ľudovo nazývame midi súbor.

Midi súbor si však nemôžeme pliesť so súbormi MP3 rovnako ako si nemôžeme nahrávanie do sekvencera zameniť s nahrávaním na CD alebo magnetofón. Porovnávať tieto dva spôsoby je nemožné rovnako, ako sa nedá porovnať CD Michaela Jacksona s notovými predlohami jeho piesní. Ak si kúpite CD s Jacksonovými piesňami, budete lahodne počúvať zvuk jeho kapely, vychutnávať jeho hlas, či jeho zbory. Ak si kúpite knihu s notami jeho piesní, máte iba základ pre vlastnú interpretáciu. Potrebujete hudobné nástroje, potrebujete muzikantov, zboristov aj spevákov, aby ste si kúpené noty pretvorili na hudbu. A ak to všetko aj zoženiete a premeníte noty na zvuk, nebude to Jacksonova hudba a už vôbec nie jeho spev. Budú to tie isté tóny, tá istá melódia i akordy, no bude to iný zvuk. To isté sa deje pri MP3 a mid. MP3 je nahrávka zvuku. Nahrali ste do nej nejaký konkrétny zvuk. Mid je súbor, ktorý obsahuje iba príkazy, aké noty treba stlačiť. Potrebujete nejaký hudobný prehrávač, ktorý Vám noty prehrá a to ešte stále nestačí. Dôležité je, akými zvukmi bude Váš prehrávač vybavený. Midi súbor obsahuje totiž len príkaz, že sa má zapnúť napríklad nota C1. Ak tento príkaz pošlete do drahého a kvalitného klávesového nástroja, ozve sa zrejme krásny čistý tón. Ak ten istý príkaz pošlete do midi prehrávača, ktorý máte vo svojom počítači, ozve sa pravdepodobne zvuk, ktorý síce bude znieť ako C1, ale muzikantsky si ho veľmi neužijete.

Ale späť k sekvenceru. Sekvencer je teda zariadenie, ktoré z pripojeného midi zariadenia (napríklad klávesov), prijíma midi správy a nikam ich neposiela, nijako s nimi nepracuje, len ich ukladá pekne jednu za druhou do dlhého zoznamu. Každú správu pritom obohatí o jeden dôležitý údaj a to je čas. Hudba sa totiž odohráva v čase a ak by sekvencer po skončení Vašej hry poslal všetky príkazy, ktoré prijal, v jednom okamihu späť do Vášho klávesu, ozval by sa jeden veľký tresk a potom veľké ticho. Sekvencer musí správy prijímať jednu po druhej a zaznamenávať ich časový odstup. V tom istom čase ich potom aj posiela späť. Teda ak ste hrali rýchlu stupnicu, sekvencer si presne označí, v ktorej mikrosekunde ste ktorý tón stlačili a pustili. Takto je možné hudbu zaznamenávať a opäť prehrávať.

Sekvencer pritom nemusí byť samostatný prístroj. Ak vo Vašom klávesovom nástroji máte funkcie pre midi záznam Vašej hudby, tak vlastne máte v klávesách sekvencer. Ako sekvencer môže rovnako dobre poslúžiť aj počítač, ktorý je vybavený vhodným programom ako je napríklad Notedit. Takýto program môže prijímať správy z Vašich klávesov a teda vykonáva úlohu sekvencera.

Midi prehrávače

Načo by sme zaznamenávali našu hudbu do midi, ak by sme ju nemohli následne prehrať. Prehrávače midi sú také zariadenia, ktoré dokážu prehrať súbory typu mid. Sú to samotné klávesové nástroje, ktoré zväčša dovoľujú prehrať súbor mid z USB kľúča alebo počítača (v takom prípade ho nemusíme pripájať cez konektory midi ale môžeme použiť USB, ak je ním kláves vybavený). Ako midi prehrávač však môžeme použiť aj počítač,

pretože dnes je už každý počítač vybavený základnou sadou zvukov a teda pri pokuse prehrať súbor midi sa ozve aký-taký zvuk. Prehrávačmi súborov midi sú de facto aj mobilné telefóny, pretože aj tie využívajú súbory midi na prehrávanie zvonení. Sekvencery a prehrávače musíme teda od seba odlišovať, pretože hoci každý sekvencer je aj prehrávač, nie každý prehrávač je aj sekvencer.

Prečo nahrávať do midi?

Možno sa Vám už vynorila v mysli otázka, prečo nahrávať skladby do midi, keď si pokojne môžeme urobiť aj hudobný záznam našej skladby priamo do MP3. Odpoveď je nasledovná: do MP3 môžeme nahrávať vtedy, ak chceme svoje dielo iba prehrať a nepotrebujeme s ním robiť nič iné. Do midi zasa budeme nahrávať vtedy, ak chceme našu skladbu nejako upravovať, obohacovať alebo ju šíriť v inej než zvukovej podobe. Áno, nepomýlil som sa. Keďže súbory midi sú vlastne zoznamami príkazov o notách, existuje veľké množstvo programov, ktoré dokážu tieto súbory pretvoriť na notový zápis. Našu skladbu tak môžeme zobrazit' ako noty a vytlačiť ju. Môžeme ju však aj obohacovať tým, že k už nahratým partom budeme hrať ďalšie a ďalšie, čím si sami vytvoríme zvuk celej kapely. Midi nám tiež umožňuje opravovať chyby, ktoré sme urobili, teda si po vytvorení surovej nahrávky môžeme skladbu prejsť a opraviť všetky zlé údery, vyvážiť hlasitosť či pridať iné efekty. Midi súbor nám skrátka umožňuje našu skladbu plne dotvárať s tým, že keď ju už máme hotovú a s výsledkom sme nadmieru spokojní, nič nám nebráni finálnu verziu si prehrať a jej zvuk zachytiť do MP3 súboru, ktorý už potom šírimo ako definitívne dielo.

Je tu ale aj ďalší spôsob využitia. Ako som spomenul vyššie, možnosti midi sú nesmierne široké a ak máme skladbu, vytvorenú v midi pomocou nejakého midi sekvencera, môžeme ju obohatiť o rôzne midi príkazy, ktoré sa prejavio akusticky aj opticky. Napríklad ak hráme v hudobnej kapele, môžeme si v midi pripraviť hudobný podklad – najlepšie s použitím rytmických nástrojov. Na koncerte spustíme z prehrávača midi skladbu, takže nám budú automaticky zniet' bicie nástroje – prípadne aj ďalšie, ktoré sú v midi súbore nahraté. Midi prehrávač bude posilať po celý čas rôznym pripojeným zariadeniam príkazy v presne stanovenom čase, takže na koncerte môžeme docieľiť, že kým z reproduktorov znejú nástroje, spevákovi sa na displeji premieta text piesne, zvukár dostáva v presne stanovených časoch pripomienky čo má robiť, v presne určený čas sa upraví svetlo, alebo sa ozve nejaký efekt – to všetko môže prísť z jediného midi súboru, do ktorého my s úsmevom na tvári hráme svoj part, alebo sa len tak tvárime, nakoľko aj naše vlastné party máme vopred uložené v midi súbore. Nuž áno, aj to je možné. Za pomoci midi súboru a kvalitného prehrávača sa môže muzikantom stať navonok aj ten, kto bez techniky ťažko zahrá stupnicu. No ale aké je potom potešenie z hry? Midi má našu hudbu obohacovať, nie ju nahrádzať. Ja osobne mám tiež mnoho podkladov pripravených v midi, no vlastné party (spravidla klavírne) hrám vždy naživo.

Midi kanály

Nakoniec si ešte povedzme niečo o midi kanáloch. Je ich 16 a sú to akési pomyselné dráhy, v ktorých prúdia noty a midi príkazy. Načo je to dobré? Predstavme si, že máme ku klávesu pripojený sekvencer a nahrávame naše dokonalé dielo, skomponované za bezsenných nocí. Začneme klavírnou pasážou, do ktorej primiešame sláky. Stlačíme teda tlačidlo pre

voľbu klavíra a potom tlačidlo pre voľbu slákov. Na našom nástroji sa zvuky premiešajú, a do sekvencera odídu dva príkazy „Nastav klavír“ a „Nastav sláky“. Potom už len hráme a hráme, kým neprídeme k pasáži, kde chceme klavír zmeniť na saxofón. Stlačíme tlačidlo, zvuk sa prepne a do sekvencera odíde správa „Zapni saxofón“. Skladba zdarne skončí, vypneme nahrávanie a výsledok si prehráme. Naš kláves dostane prvé dva príkazy: „Zapni klavír“ a „Zapni sláky“. Čo sa stane? Na jednu mikrosekundu sa nám zapne klavír a vzápätí sa prepne na sláky. Zvuky sa nepremiešajú ale zostane nastavený ten posledný a všetky noty, ktoré sme tak predcízne stláčali, budú hrať ako sláky. A keď príde ten veľký okamih a náš kláves dostane príkaz „Zapni saxofón“, všetky sláky sa prepnú na saxofón a kvílenie našich kláves nám po chvíli začne trhať uši. Čo sa vlastne stalo?

Takto by sa zachoval každý nástroj, ak by neexistovalo rozdelenie na kanály. Našťastie v midi prepojení existuje 16 kanálov a tak každá nota, alebo úkon, ktorý urobíme, sa udeje len na tom určitom kanáli. Ak teda zapneme nástroj i sekvencer a zvolíme klavír, odíde do sekvencera správa „Zapni klavír na kanáli 1“. Stlačíme nejaké to premiešavacie tlačidlo na našom nástroji a zvolíme sláky. Slákom sa priradí kanál 2 a do sekvencera odíde príkaz „Zapni sláky na kanáli 2“. Každá jedna nota potom odchádza aj s určením kanála. Ak teda my zahráme prvý tón a z nášho nástroja zazneje ako zmes klavíra a slákov, odídu do sekvencera dva príkazy rýchlo za sebou „Zapni notu C1 na kanáli 1“ a „Zapni notu C1 na kanáli 2“. Takto je každá nota posielaná ako dva príkazy a to zaručuje, že pri spätnom prehrávaní sa bude prehrávať tiež každá nota zvlášť, pričom na kanáli 1 bude znieť klavír a na kanáli 2 sláky.

Zjednodušene by sa dalo povedať, že v midi má každý hudobný nástroj vyhradený svoj vlastný kanál. Ak teda chceme skomponovať a nahráť skladbu, v ktorej bude hrať naraz 5 nástrojov (bicie, basa, klavír, sláky a trubka), vyhradíme každému nástroju jeden kanál. Tak zaručíme, že noty sa navzájom nepremiešajú a každá zmena sa udeje len na tom danom kanáli.

Nezáleží na tom, ktorý kanál ktorému nástroju priradíme. Všeobecne sa však bicie nástroje umiestňujú do kanála 10, pomocné perkusie do kanála 9, basová gitara do kanála 11 a melódia, ak ju budeme tiež nahrávať, do kanála 4. V praxi sa ale stretne aj s inými použitiami kanálov a je preto vždy dôležité v midisúbore, ktorý sme sami nevytvorili, sa najprv zorientovať.

V tomto úvodnom článku sme nazreli do sveta midi. V nasledovných častiach sa budeme podrobne venovať midi príkazom a ich štruktúre, aby sme za použitia vhodného počítačového programu akým je napríklad Notedit, mohli naše vlastné ale aj získané midi súbory otvárať a upravovať. No skôr, ako sa vrhneme na jednotlivé tajomstvá midi súborov a príkazov, zastavme sa ešte pri číselných sústavách. Ich ovládanie je pre prácu s midi nevyhnutné, preto Vám odporúčam nasledovnú kapitolku si dôkladne naštudovať.

2. Číselné sústavy

Asi málokto z nás prísne rozlišuje medzi pojmami číslo a číslica. Pripomeňme si preto, aký je medzi týmito dvoma pojmami rozdiel.

Číslica je grafický symbol - znak, ktorý vyjadruje nejakú číselnú hodnotu. V našich končinách sú takýmito číslicami nasledovné symboly: 0, 1, 2, 3, 4, 5, 6, 7, 8 a 9. Nemýlite sa, skutočne ide o čísla 0 až 9. No týchto prvých 10 čísel nazývame zároveň aj číslicami, pretože sú to čísla, ktoré vyjadríme jediným znakom – jedinou cifrou.

Číslo je zasa číselná hodnota, ktorá je vyjadrená jednou alebo viacerými číslicami. Teda napríklad na vyjadrenie čísla sto potrebujeme až tri číslice: 1 0 a 0. Ak ich napíšeme pekne vedľa seba, vznikne nám číslo 100.

Číselná sústava

Všimnime si, že jednotlivé číslice – cifry – neukladáme vedľa seba len tak, bez ladu a skladu. Práve naopak. Máme na to premyslený systém.

Začíname jednou číslicou – cifrou, ktorú napíšeme nie na ľavý kraj riadku, ale naopak, na pravý. Všetky ďalšie cifry budeme pridávať pred túto prvú smerom doľava a ich poradie budeme nazývať pozíciami, pričom rátať ich budeme vždy zprava. Ak sme teda napísali prvú cifru na pravý kraj riadku, využili sme pozíciu 1.

Pre vyjadrenie prvých 10 číselných hodnôt nám vystačí našich 10 cifier 0 až 9. Potom sa nám už cifry minú a my k našej prvej cifre pridáme druhú a to naľavo od tej prvej, teda na pozíciu 2. Aj to však nie hocijakú, ale tú najnižšiu po nule, pretože nula je číslica, ktorá je ako keby vždy prítomná, ak ju nepíšeme. Ak sme teda úplne vpravo – na pozícii 1 - už minuli všetky cifry, pridáme na pozíciu 2 cifru 1 a celý cyklus sa úplne vpravo opakuje. Keď sa znovu cifry minú, číslicu na pozícii 2 o jednu zvýšime a cyklus sa opakuje. Takto pokračujeme stále dokola, kým sa neminú všetky cifry aj na pozícii 2. Vtedy zvýšime tretiu pozíciu z fiktívnej nuly na jednotku a cyklus môže znova pokračovať najprv na pozícii 1 a potom na pozícii 2. Takto môžeme pokračovať do nekonečna, pričom po každom desiatom čísle zvýšime hodnotu cifry, ktorá stojí naľavo od tej, ktorá naposledy prebehla cyklom a celý proces začne od začiatku zprava.

Ukážme si to na príklade (čísllice 0 v našom príklade slúžia iba na to, aby sme videli pozície cifier. Inak ich bežne nepíšeme):

0001 (= 1)

0002 (= 2) až

0009 (= 9). Minuli sa cifry, preto zvýšime najbližšiu ľavú cifru

0010 (= 10) Teraz zvyšujeme prvú cifru vpravo až po

0019 (= 19) Na pozícii 1 zprava sa minuli cifry, preto zvýšime pozíciu 2

0020 (=20) a pokračujeme až kým sa aj na pozícii 2 neminú cifry:

0021

0029

0031

0039

0041

0099 (= 99) Teraz už musíme pridať pozíciu 3:

0100 (= 100) a zasa zvyšujeme od pozície 1:

0101 (= 101) až po

0109 (= 109) kedy zvýšime pozíciu 2, lebo na pozícii 1 sa cifry minuli
0110 (= 110) Ak sa minuli aj cifry na pozícii 3
0999 (= 999), pridáme pozíciu 4
1000 (= 1000) a zasa zvyšujeme od pozície 1
1001 (= 1001).

Tento systém práce kombinovania čísel sa nazýva číselná sústava. Ak sme mu porozumeli, máme vyhraté.

Všimnime si, že jedným znakom môžeme vyjadriť 10 čísel (0 až 9 – aj nula je číslo) a najbližšiu ľavú pozíciu zvyšujeme vždy po prebehnutí cyklu, ktorý má 10 krokov, nakoľko práve toľko nám to dovoľí náš počet cifier. Z týchto dôvodov číselnú sústavu, ktorú sme si takto predstavili, nazývame desiatková (decimálna) a je to bežne používaná číselná sústava.

Šestnástková – hexadecimálna číselná sústava

Desiatková číselná sústava nie je jediná. Pre technické účely bola vyvinutá šestnástková – často nazývaná aj hexadecimálna, správne však sedecimálna - číselná sústava. Ak sme porozumeli fungovaniu desiatkovej sústavy, ľahko použijeme aj hexadecimálnu číselnú sústavu. Princíp fungovania je totiž ten istý. Výhodou tejto sústavy vo výpočtovej technike je, že dokáže menej ciframi vyjadriť väčšie číselné hodnoty, pretože – ako vyplýva z názvu – hodnoty najbližšej pozície sa budú zvyšovať až po 16-tich krokoch, čo v konečnom dôsledku urobí dosť podstatný rozdiel. Teraz sa logicky spýtate, ako je možné vyjadriť jednou cifrou až 16 čísel, keď v našej abecede máme iba 10 číselných znakov? Správna a logická otázka. Tvorcovia šestnástkovej sústavy si totiž požičali aj písmená A B C D E a F, ktorými tiež vyjadrujú číselnú hodnotu. Znak 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F budú teda v tejto sústave vyjadrovať prvých 16 čísel.

Siahnime po našom osvedčenom príklade a ukážme si, ako to bude fungovať (nuly sú tu opäť iba na zobrazenie neviditeľných – nulových – pozícií a číslo v zátvorke vyjadruje hodnotu zapísaného čísla v desiatkovej – teda našej ľudskej reči):

\$0001 (= 1)

\$0002 (= 2) až

\$000F (= 15). Minuli sa cifry, preto zvýšime najbližšiu ľavú cifru

\$0010 (= 16) Teraz zvyšujeme prvú cifru vpravo až po

\$001F (= 31) Na pozícii 1 zprava sa minuli cifry, preto zvýšime pozíciu 2

\$0020 (=32) a pokračujeme až kým sa aj na pozícii 2 neminú cifry:

\$0021

\$002F

\$0031

\$008F

\$00A1

\$00FF (= 255) Teraz už musíme pridať pozíciu 3:

\$0100 (= 256) a zasa zvyšujeme od pozície 1:

\$0101 (= 257) až po

\$010F (= 271) kedy zvýšime pozíciu 2, lebo na pozícii 1 sa cifry minuli

\$0110 (= 272) Ak sa minuli aj cifry na pozícii 3

\$0FFF (= 4095), pridáme pozíciu 4

\$1000 (= 4096) a zasa zvyšujeme od pozície 1

\$1001 (= 4097).

Všimnite si, že čísla, ktoré v šestnástkovej sústave potrebujú iba tri znaky, potrebujú v desiatkovej sústave viac znakov. Práve to je prednosť šestnástkovej sústavy, používa sa aj v midi a je dobré, ak chápeme rozdiel medzi číslami desiatkovej a šestnástkovej sústavy.

Aby sa ale čísla oboch týchto sústav neplietli, sú čísla šestnástkovej sústavy označované znakom \$ pred samotným číslom, alebo písmenkom h tesne za ním. Ak teda v ďalších článkoch zazriete číslo \$90 alebo 90h, nie je to naše bežné číslo 90, ale hexadecimálne číslo, ktorého hodnota je v desiatkovej sústave 144.

Ak Vás zaujíma, prečo sa v midi využívajú čísla šestnástkovej a nie našej bežnej desiatkovej sústavy, prečítajte si nasledovnú kapitolku o bitoch a bajtoch.

Dvojková - binárna sústava

Ak sme už porozumeli číselným sústavám a vieme s nimi pracovať, bola by škoda nespomenúť sústavu, ktorá je v technickom svete najpoužívanejšou, aj keď málokedy prenikne do ľudského sveta. Je to dvojková – binárna číselná sústava. Ako už jej názov napovedá, pracuje iba s dvoma ciframi – číselnými znakmi, pomocou ktorých vyjadruje hodnoty. Tými znakmi sú číslica 0 a 1. Je to tak preto, lebo počítače, kalkulačky, mobily a všetky ostatné technické zariadenia dokážu vo svojej podstate pracovať iba s nulami a jednotkami. Do svojich pamätí, na diskety, pevné disky alebo pamäťové karty zapisujú iba nuly a jednotky. Svoje pamäte si rozdelia na malinké chlieviky, ktoré magnetická hlavička pri zápise postupne prechádza a buď do daného chlievika vloží magnetický náboj (vtedy je hodnota chlievika 1), alebo ho tam nevloží (vtedy je hodnota chlievika 0). To je vlastne všetko, čo počítač dokáže. Ak by sme po takom pevnom disku alebo pamäťovej karte prešli silným magnetom, ktorý by všetky chlieviky zmagnetizoval rovnakou magnetickou energiou, došlo by k tomu, že by sa hodnoty všetkých chlievikov zrazu rovnali jednotke a teda by boli naše informácie stratené. (Poznámka pre expertov: uvedomujem si, že sa vyjadrujem veľmi zjednodušene a teda nepresne, ale mojím cieľom je teraz iba priblíženie samotnej podstaty fungovania počítača).

Nakoľko počítače do svojich chlievikov a šuflíčkov vedia ukladať iba kladné alebo záporné magnetické hodnoty – v našej reči – vedia zapísať iba nuly a jednotky – musí byť každé číslo, každý znak, každá informácia prevedená do dvojkovej sústavy. Práve preto je dvojková – binárna sústava všade tam, kde je prítomné nejaké technické zariadenie.

Ako dvojková sústava funguje? Presne tak, ako všetky doterajšie s tým rozdielom, že máme k dispozícii iba dva číselné znaky a teda budeme zvyšovať pozície oveľa skôr. Táto sústava si preto vyžaduje pre zápis čísla omnoho viac pozícií. No veď skúsme a použijeme pritom náš osvedčený príklad (nuly naľavo od použitých pozícií opäť slúžia len na ilustráciu voľných miest a číslo v zátvorke je dané číslo v desiatkovej teda našej sústave):

%0001 (= 1) Ešte nikde nič a už sa minuli cifry. Zvýšime teda pozíciu 2

%0010 (= 2) až

%0011 (= 3). Minuli sa cifry aj na pozícii 2, preto zvýšime pozíciu 3

%0100 (= 4) Teraz zvyšujeme prvú cifru vpravo

%0101 (= 5) Na pozícii 1 zprava sa minuli cifry, preto zvýšime pozíciu 2

%0110 (=6) a pokračujeme až kým sa aj na pozícii 2 neminú cifry:

%0111 (= 7) Teraz už musíme pridať pozíciu 4:

%1000 (= 8) a zasa zvyšujeme od pozície 1

%1001 (= 9).

Vidíme, že kým v desiatkovej sústave sme zápisom 1000 vyjadrili číslo s hodnotou 1000 a v šestnástkovej sústave sme tým istým zápisom vyjadrili až 4096, v dvojkovej sústave sme sa dostali sotva k číslu 8.

Zápis čísel v dvojkovej sústave sa bežne v textoch a zápisoch, určených pre ľudí nevyskytuje, no nie je zlé o tejto sústave vedieť a v prípade potreby správne rozlúštiť zmes jednotiek a núl. Taktiež nám znalosť tejto sústavy pomôže lepšie pochopiť prácu počítača a systém ukladania informácií do pamätí a na disky. V nasledujúcej kapitole budeme hovoriť práve o chlievikoch, do ktorých si počítače ukladajú informácie a budeme pri tom pracovať práve s dvojkovou číselnou sústavou.

Aby sa zápisy čísel v dvojkovej číselnej sústave neplietli s inými zápsmi, označujú sa čísla dvojkovej sústavy znakom %. Ak teda uvidíte niekde číslo %10, je to naša stará známa dvojková vyjadrená v dvojkovej číselnej sústave. Ak zasa uvidíte \$10, (alebo 10h), je to naša stará známa šestnástka. A jedine zápis 10 bez akéhokoľvek iného označenia je nám dôverne známou desiatkou. (Všimnite si súvislosť medzi hodnotou zápisu 10 v danej sústave a názvom sústavy.)

Ďalšie číselné sústavy

Keď sme si už predstavili systém fungovania číselnej sústavy, vedeli by ste vytvoriť napríklad osmičkovú číselnú sústavu? Skúste ako domácu úlohu napísať prvých 100 čísel v tejto sústave. Mimochodom, osmičková číselná sústava naozaj existuje a používa sa práve aj v midi.

Prevod z dvojkovej na desiatkovú sústavu

Niekedy sa môže stať, že sa stretne so zápisom čísla v dvojkovej sústave a budeme potrebovať jeho prevod do desiatkovej sústavy. Nebude na škodu, ak si osvojíme zaručený fígel, pomocou ktorého túto úlohu ľahko zvládneme aj bez prevodnej tabuľky. Stačí ak si zapamätáme, že hodnota každej cifry v čísle je vyjadrená mocninou čísla 2, ktoré je umocnené číslom svojej pozície, pričom pozície sa číslujú od nuly. Inak povedané: Zrátame si počet cifier v našom binárnom čísle (je ich napríklad 5) a potom si na pomocný papier vedľa seba rozpíšeme mocniny dvojky od 0 po 4 takto: 1, 2, 4, 8, 16. Čo sme to vlastne napísali? Iba výsledky nasledovných zápisov: 2 na nultú, 2 na prvú, dva na druhú, dva na tretiu a dva na štvrtú. Ak by bolo cifier viac, mohli by sme pokračovať – stále dva na vyššiu a vyššiu. Teraz si všetky tie pozície, ktoré v našom čísle predstavujú nulu škrtneme a výsledné čísla, ktoré sme neškrtnli zrátame.

Príklad:

Máme dvojkové číslo

1010011

Toto dvojkové číslo má 7 cifier, preto si urobíme rozpis mocniny 2 na 0 až 2 na 6:

1, 2, 4, 8, 16, 32, 64

Teraz pôjdeme pozíciu za pozíciou (pozor, v pôvodnom čísle začíname z pravej strany)

a každú pozíciu, ktorá v pôvodnom čísle predstavuje 0, prepíšeme tiež na 0. Dostaneme tak 1, 2, 0, 0, 16, 0, 64

Napokon čísla, ktoré nám zostali spočítame. Výsledok je 83. Takto jednoducho sme previedli dvojkové číslo na desiatkové.

Prevodná tabuľka čísel desiatkovej, dvojkovej a šesťnástkovej sústavy

Na záver tejto vyčerpávajúcej kapitoly prikladám kompletnú prevodnú tabuľku čísel desiatkovej, dvojkovej a šesťnástkovej sústavy. Môžete ju jednoducho použiť, ak budete v budúcnosti potrebovať previesť jedno číslo na iné, alebo v nej môžete jednoducho sledovať, ako sa číslo v danej sústave mení a odlišuje od zápisu v ostatných sústavách.

Desiatkové číslo	Šesťnástkové číslo	Dvojkové číslo
0	\$0	%0
1	\$1	%1
2	\$2	%10
3	\$3	%11
4	\$4	%100
5	\$5	%101
6	\$6	%110
7	\$7	%111
8	\$8	%1000
9	\$9	%1001
10	\$A	%1010
11	\$B	%1011
12	\$C	%1100
13	\$D	%1101
14	\$E	%1110
15	\$F	%1111
16	\$10	%10000
17	\$11	%10001
18	\$12	%10010
19	\$13	%10011
20	\$14	%10100
21	\$15	%10101
22	\$16	%10110
23	\$17	%10111
24	\$18	%11000
25	\$19	%11001
26	\$1A	%11010
27	\$1B	%11011
28	\$1C	%11100
29	\$1D	%11101
30	\$1E	%11110
31	\$1F	%11111
32	\$20	%100000
33	\$21	%100001
34	\$22	%100010

35	\$23	%100011
36	\$24	%100100
37	\$25	%100101
38	\$26	%100110
39	\$27	%100111
40	\$28	%101000
41	\$29	%101001
42	\$2A	%101010
43	\$2B	%101011
44	\$2C	%101100
45	\$2D	%101101
46	\$2E	%101110
47	\$2F	%101111
48	\$30	%110000
49	\$31	%110001
50	\$32	%110010
51	\$33	%110011
52	\$34	%110100
53	\$35	%110101
54	\$36	%110110
55	\$37	%110111
56	\$38	%111000
57	\$39	%111001
58	\$3A	%111010
59	\$3B	%111011
60	\$3C	%111100
61	\$3D	%111101
62	\$3E	%111110
63	\$3F	%111111
64	\$40	%1000000
65	\$41	%1000001
66	\$42	%1000010
67	\$43	%1000011
68	\$44	%1000100
69	\$45	%1000101
70	\$46	%1000110
71	\$47	%1000111
72	\$48	%1001000
73	\$49	%1001001
74	\$4A	%1001010
75	\$4B	%1001011
76	\$4C	%1001100
77	\$4D	%1001101
78	\$4E	%1001110
79	\$4F	%1001111
80	\$50	%1010000
81	\$51	%1010001
82	\$52	%1010010

83	\$53	%1010011
84	\$54	%1010100
85	\$55	%1010101
86	\$56	%1010110
87	\$57	%1010111
88	\$58	%1011000
89	\$59	%1011001
90	\$5A	%1011010
91	\$5B	%1011011
92	\$5C	%1011100
93	\$5D	%1011101
94	\$5E	%1011110
95	\$5F	%1011111
96	\$60	%1100000
97	\$61	%1100001
98	\$62	%1100010
99	\$63	%1100011
100	\$64	%1100100
101	\$65	%1100101
102	\$66	%1100110
103	\$67	%1100111
104	\$68	%1101000
105	\$69	%1101001
106	\$6A	%1101010
107	\$6B	%1101011
108	\$6C	%1101100
109	\$6D	%1101101
110	\$6E	%1101110
111	\$6F	%1101111
112	\$70	%1110000
113	\$71	%1110001
114	\$72	%1110010
115	\$73	%1110011
116	\$74	%1110100
117	\$75	%1110101
118	\$76	%1110110
119	\$77	%1110111
120	\$78	%1111000
121	\$79	%1111001
122	\$7A	%1111010
123	\$7B	%1111011
124	\$7C	%1111100
125	\$7D	%1111101
126	\$7E	%1111110
127	\$7F	%1111111
128	\$80	%10000000
129	\$81	%10000001
130	\$82	%10000010

131	\$83	%10000011
132	\$84	%10000100
133	\$85	%10000101
134	\$86	%10000110
135	\$87	%10000111
136	\$88	%10001000
137	\$89	%10001001
138	\$8A	%10001010
139	\$8B	%10001011
140	\$8C	%10001100
141	\$8D	%10001101
142	\$8E	%10001110
143	\$8F	%10001111
144	\$90	%10010000
145	\$91	%10010001
146	\$92	%10010010
147	\$93	%10010011
148	\$94	%10010100
149	\$95	%10010101
150	\$96	%10010110
151	\$97	%10010111
152	\$98	%10011000
153	\$99	%10011001
154	\$9A	%10011010
155	\$9B	%10011011
156	\$9C	%10011100
157	\$9D	%10011101
158	\$9E	%10011110
159	\$9F	%10011111
160	\$A0	%10100000
161	\$A1	%10100001
162	\$A2	%10100010
163	\$A3	%10100011
164	\$A4	%10100100
165	\$A5	%10100101
166	\$A6	%10100110
167	\$A7	%10100111
168	\$A8	%10101000
169	\$A9	%10101001
170	\$AA	%10101010
171	\$AB	%10101011
172	\$AC	%10101100
173	\$AD	%10101101
174	\$AE	%10101110
175	\$AF	%10101111
176	\$B0	%10110000
177	\$B1	%10110001
178	\$B2	%10110010

179	\$B3	%10110011
180	\$B4	%10110100
181	\$B5	%10110101
182	\$B6	%10110110
183	\$B7	%10110111
184	\$B8	%10111000
185	\$B9	%10111001
186	\$BA	%10111010
187	\$BB	%10111011
188	\$BC	%10111100
189	\$BD	%10111101
190	\$BE	%10111110
191	\$BF	%10111111
192	\$C0	%11000000
193	\$C1	%11000001
194	\$C2	%11000010
195	\$C3	%11000011
196	\$C4	%11000100
197	\$C5	%11000101
198	\$C6	%11000110
199	\$C7	%11000111
200	\$C8	%11001000
201	\$C9	%11001001
202	\$CA	%11001010
203	\$CB	%11001011
204	\$CC	%11001100
205	\$CD	%11001101
206	\$CE	%11001110
207	\$CF	%11001111
208	\$D0	%11010000
209	\$D1	%11010001
210	\$D2	%11010010
211	\$D3	%11010011
212	\$D4	%11010100
213	\$D5	%11010101
214	\$D6	%11010110
215	\$D7	%11010111
216	\$D8	%11011000
217	\$D9	%11011001
218	\$DA	%11011010
219	\$DB	%11011011
220	\$DC	%11011100
221	\$DD	%11011101
222	\$DE	%11011110
223	\$DF	%11011111
224	\$E0	%11100000
225	\$E1	%11100001
226	\$E2	%11100010

227	\$E3	%11100011
228	\$E4	%11100100
229	\$E5	%11100101
230	\$E6	%11100110
231	\$E7	%11100111
232	\$E8	%11101000
233	\$E9	%11101001
234	\$EA	%11101010
235	\$EB	%11101011
236	\$EC	%11101100
237	\$ED	%11101101
238	\$EE	%11101110
239	\$EF	%11101111
240	\$F0	%11110000
241	\$F1	%11110001
242	\$F2	%11110010
243	\$F3	%11110011
244	\$F4	%11110100
245	\$F5	%11110101
246	\$F6	%11110110
247	\$F7	%11110111
248	\$F8	%11111000
249	\$F9	%11111001
250	\$FA	%11111010
251	\$FB	%11111011
252	\$FC	%11111100
253	\$FD	%11111101
254	\$FE	%11111110
255	\$FF	%11111111

3. Bity a bajty

Mnoho ľudí má nočnú moru z týchto dvoch pojmov a myslí si, že im nikdy neporozumie. Som však presvedčený, že to nie je až také zložité a hoci pre prácu s miči túto kapitolku nevyhnutne nepotrebujete a ak teda vyslovene nechcete, môžete ju preskočiť, bude dobré, ak si ju aspoň prečítate.

Skúsme si teda úplne jednoducho vysvetliť, čo sú bajty a bity.

Bajty

Anglické byte – v slovenčine nazývaný bajt – je najmenšia jednotka počítačovej pamäte. V anglickom označení sa píše s tvrdým y a označuje sa veľkým písmenom B.

Každá elektronická pamäť v počítači, mobile, fotoaparáte alebo v akomkoľvek inom zariadení je vlastne miesto, kam si počítač ukladá dáta a to v bajtoch. Bajty si môžeme predstaviť ako tehličky, ktoré počítač ukladá vedľa seba, aby vybudoval v pamäti dielo, ktoré sme my vytvorili. Povedané inak: ak si chceme postaviť dom, potrebujeme najprv prázdny priestor, na ktorý začneme ukladať tehly – pekne jednu za druhou. Čím viac tehál použijeme, tým bude náš dom väčší, vyšší alebo krajší. Ak máme veľa priestoru, môžeme postaviť palác, ak máme málo priestoru, možno sa zmôžeme iba na psiu búdu.

A teraz si predstavme, že ideme podobné dielo budovať v počítači. Napríklad ideme napísať list kamarátovi. Otvoríme si prázdny dokument a ťukáme písmenko za písmenkom. Počítač si každé naše písmenko musí uložiť do pamäte a to tak, že každé písmenko zaberie jeden bajt. Ak napíšeme 10 znakov (zámerne nehovorím písmeniak, pretože aj medzery, bodky a čiarky sú znaky), počítač už uložil vedľa seba 10 tehličiek. Ak sme napísali už 1000 znakov, použil tehličiek 1000. Ak sme napísali sto strán textu, použil presne toľko tehličiek, koľko znakov sme do textu vložili. Môžu to byť tisíce, státisíce, dokonca i milióny tehličiek. Kto by to rátal pri takých obrovských množstvách? Počítač nám však napriek tomu presný počet použitých bajtov povie, ale neukáže nám ich v miliónových číslach, ale trochu inteligentnejšie – v kilobajtoch, megabajtoch a gigabajtoch.

Na prevod týchto jednotiek existuje jednoduchý vzorec:

1000 bajtov tvorí 1 kilobajt (jeho značkou je KB – pozor! obe písmená veľké)

1000 kilobajtov (= 1000000 bajtov) tvorí 1 megabajt (jeho značkou je MB – pozor! obe písmená veľké)

1000 megabajtov (= 1000000 kilobajtov) tvorí 1 gigabajt (jeho značkou je GB – pozor! Obe písmená veľké)

Aby ste sa presvedčili, že je to naozaj tak, urobte jednoduchý pokus: otvorte vo svojom počítači Poznámkový blok (nie Word ani iný textový editor) a napíšte jedno písmenko. Súbor uložte a pozrite sa na jeho veľkosť. Počítač by Vám mal oznámiť, že veľkosť súboru je 1 bajt.

Počítač môže do pamäte alebo na pevný disk ukladať toľko bajtov, koľko to pamäť dovolí. Práve preto sú všetky CD, DVD, pevné disky či pamäťové karty označené kapacitou a to buď v MB, GB alebo dnes už dokonca v TB (TB sú terabajty – 1 TB = 1000 GB).

Bity

Anglické bite – v slovenčine nazývaný bit – je najmenšia jednotka počítačovej pamäte v rámci bajtu. V anglickom označení sa píše s mäkkým i a označuje sa malým písmenom b.

Ak sme v predošlom odseku ukladali na seba tehličky z bajtov, teraz sa pozrieme do jednej tehličky. Tá je rozdelená na 8 priehradiek a to sú práve bity. Každá jedna tehlička má teda 8 priehradiek – bitov, do ktorých počítač dokáže zapísať jednotky alebo nuly. Všetky elektronické zariadenia zapisujú dáta v dvojkovej číselnej sústave, ktorá sa skladá z jednotiek a núl (pozri predošlú kapitolu 2 o číselných sústavách) a práve bity v tehličke sú tie miesta, kam sa jednotky a nuly zapíšu.

Predstavme si nasledovnú situáciu: otvoríme v počítači Poznámkový blok a stlačíme klávesu A. Na obrazovke sa objaví písmeno „A“ a toto písmenko sa zapíše do jedného bajtu v pamäti. Keďže my už vieme, že počítač nedokáže zapísať do bajtov texty, iba čísla, musí byť písmenko A najprv prevedené na čísla. Na to existujú kódovacie tabuľky, ktoré určujú, aké číslo bude ktorému písmenu priradené. Systém Windows v našich zemepisných končinách používa kódovaciu tabuľku Windows 1250, ktorá hovorí, že veľkému písmenu A zodpovedá číslo 65 a malému písmenu a číslo 97. Ak sme teda napísali veľké A, to sa podľa kódovacej tabuľky prevedie na číslo 65, ktoré je v dvojkovej sústave vyjadrené ako %1000001. Toto číslo sme v dvojkovej sústave teraz vyjadrili použitím siedmych cifier, ale nakoľko naše číslo bude o chvíľu zapísané do prázdneho bajtu, potrebujeme ho mať v tvare osemmiestneho čísla, pretože 8 je bitov v jednom bajte.

V dvojkovej sústave – rovnako ako v desiatkovej – začíname vždy písať čísla zprava a teda ak naše číslo zaberie menší počet cifier než 8, zvyšné nuly dopíšeme vľavo. Číslo 65 v dvojkovej sústave teda bude %01000001. A to už máme 8 cifier a každá bude zapísaná do jedného bitu v prázdnej tehličke menom bajt. Tak sme úspešne zapísali do pamäte písmeno A.

Čo sa stane, keď otvoríme súbor, ktorý obsahuje iba naše písmenko A? Počítač začne prezerat' tehličky - bajty – jeden za druhým. V našom prípade nájde jeden bajt s 8 bitmi, ktoré sú zaplnené takto: %01000001. Tieto nuly a jednotky v našej desiatkovej sústave tvoria číslo 65, preto počítač pozrie do kódovacej tabuľky aby zistil, aký znak má zobrazit'. Keďže tomuto číslu prislúcha písmeno A, na obrazovke sa objaví veľké A.

Ak sme porozumeli, ako počítač ukladá a načítava informácie, bude nám už jasné, aké dôležité je mať správnu kódovaciu tabuľku. Pri každom zápise a načítaní dát sa jednotky a nuly zo súboru vyhľadávajú v danej tabuľke, aby sa zobrazil znak, ktorý mal autor súboru na mysli. Nesprávna kódovacia tabuľka môže narobiť nemalé problémy. Kódovacích tabuliek je viac, rôzne operačné systémy používajú vlastné tabuľky a dokonca aj v rámci systému Windows je tabuliek viac – pre každý svetový región iná. Tak sa celkom ľahko stane, že v slovenskom Windows napíšete text, uložíte ho, no po otvorení v nemeckom či talianskom Windows budú vyzerat' niektoré znaky úplne inak. My však už poznáme riešenie takej situácie – nastaviť takú kódovaciu tabuľku, akú použil autor súboru pri jeho tvorbe, alebo použiť konvertovací program, ktorý dokáže znaky v samotnom súbore nahradiť, ak mu povieme, z akej tabuľky do akej má zámeny vykonať.

Celú problematiku bitov a bajtov teda môžeme zhrnúť takto: bajt je základná stavebná jednotka pamäte. Každý bajt má 8 bitov, na uloženie jednotiek a núl. Tu si musíme uvedomiť, že tento systém vznikol v dobe, keď počítače boli maximálne jednoduché a slúžili hlavne na prácu s textami a číslami. Vtedy úplne stačilo, ak mal jeden bajt 8 bitov. Tento konštantne daný počet cifier je totiž obmedzujúci a prísne nám hovorí, že do jedného bajtu môžeme zapísať iba čísla v rozpätí 0 až 255 – áno, presne toľko čísel totiž vieme vyjadriť 8 ciframi v dvojkovej sústave. Pre číslo 256 by sme už potrebovali deviatu cifru a jeden bajt by nám nestačil. Vytvárajúca sa technika však aj toto priniesla. Počítače sa začali využívať aj na spracovávanie zvuku, obrazu, videa a rôznych iných náročných úloh, takže 256 kombinácií pre jednu informáciu nestačilo. Napríklad vyjadriť úroveň digitálneho zvuku alebo farebného spektra sa proste 8 ciframi nedá. Preto novšie systémy začali pre zápis používať 16, potom 32

a dnes už dokonca 64 bitov. Áno, presne ako tušíte, bajt aj dnes má len oných 8 bitov, preto ak počítač pracuje s obrazom či videom, musí si pre zápis jednej informácie vziať aj 4 alebo 8 bajtov. (Pri 8 bajtoch má k dispozícii už 8x8 teda 64 bitov, čo je podstatne viac kombinácií ako pri 8 bitoch)

Bity a bajty v midi

Vráťme sa však k hudbe. Aj v midi sú bajty a bity využívané na prenos a ukladanie hudobných informácií. Midi informácie sú navyše dosť stručné na to, aby im stačilo 8 bitov, takže mnohé midi zariadenia aj dnes pracujú ako 8 bitové. Pri ďalšom skúmaní sveta midi si preto odmyslíme dnešné 32 a 64 bitové systémy a pod každým bajtom si predstavíme jednoduchých 8 bitov.

Ak hudobný nástroj alebo počítač posielajú midi príkaz, spravidla vyšle niekoľko bajtov s číselnými údajmi, ktoré, ak im chceme porozumieť, musíme podľa príslušnej tabuľky rozšifrovať. Posielané bajty môžeme rozdeliť na dva druhy: sú to stavové a dátové bajty.

Stavové bajty

Stavové bajty určujú, aký úkon sa má v zariadení vykonať a na akom kanáli. Je to tá najdôležitejšia informácia, preto sa aj nazýva „status“.

Stavové bajty majú vždy na prvom bite (prvý zľava) číslo 1. Podľa toho zariadenie vie, že prichádza stavová informácia. Táto skutočnosť sa výrazne podpisuje pod množstvo kombinácií, ktoré do bajtu môžeme zapísať. Ak si totiž predstavíme, že okrem prvého bitu sú všetky ostatné bity nuly, získame číslo %10000000, čo je v desiatkovej sústave 128 a v šestnástkovej \$80. Do stavového bajtu teda nedokážeme zapísať menšie číslo ako 128 (nedovoľuje nám to jednotka na prvom bite.) Maximálne číslo, ktoré do stavového bajtu vieme zapísať je 255 (dvojkovo %11111111 a šestnástkovo \$FF). Stavový bajt teda vždy bude obsahovať číslo v rozsahu 128 až 255 (šestnástkovo \$80 až \$FF).

Stavový bajt je výhodné čítať ako šestnástkové a nie desiatkové číslo. Stavový bajt totiž obsahuje dve zakódované informácie, vyjadrené jednou cifrou, ktorú uvidíme práve pri prevode stavového bajtu do šestnástkového čísla. V takom prípade totiž prvá cifra určuje udalosť, ktorú má zariadenie vykonať a druhá cifra určuje kanál, na ktorom sa má úkon vykonať. Kanály sa pritom číslujú od 0 do 15. Tu je odpoveď aj na otázku, prečo má midi práve 16 kanálov. Nuž je to preto, lebo šestnástková sústava umožnila jednou cifrou vyjadriť práve čísla v rozsahu 0 až 15.

Ako teda budeme čítať stavové bajty? Prevedieme si ich na číslo šestnástkovej sústavy. Dostaneme tak napríklad číslo \$90. Prvá cifra (\$9) nám prezradí, že bude zapnutá nota (príkaz \$9 je práve note on – zapnutie noty). Druhá cifra (\$0) zasa prezradí, že sa to udeje na prvom kanáli. Príkaz \$98 zasa znamená zapnutie noty na siedmom kanáli. Príkaz \$9F znamená zapnutie noty na šestnástom kanáli. Hľa, aké jednoduché. Stačí len poznať význam prvej cifry (môže byť iba v rozsahu \$8 až \$F) a presne vieme, čo daný príkaz vykoná. Zoznam týchto príkazov nájdete neskôr (od kapitoly 7).

Dátové bajty

Dátové bajty nasledujú hneď za stavovými a bližšie určujú príkaz, zadaný v stavovom bajte.

Dátové bajty majú vždy na prvom bite (prvý zľava) číslo 0. Podľa toho zariadenie vie, že prichádza dátová informácia. Aj tu sa táto skutočnosť výrazne podpisuje pod množstvo kombinácií, ktoré do bajtu môžeme zapísať. Ak má byť prvý bit zľava vždy 0, máme k dispozícii iba rozsah %00000000 (čo je desiatková 0) až %01111111 (čo je desiatkových 127). Dátový bajt teda bude mať vždy hodnotu v rozsahu 0 až 127.

Keď sme v predošlom odseku pomocou príkazu \$90 zapínali notu na prvom kanáli, určíte ste si v duchu položili otázku: ktorá nota to vlastne bude zapnutá? O tom stavové bajty nehovoria, no keďže táto informácia je veľmi dôležitá, prináša ju hneď najbližší dátový bajt. Všetky noty sú očíslované a rozsah klaviatúry od subkontra A po C5 je 21 až 108. Stačí ak si zapamätáme, že naše stredoeurópske C1 má hodnotu 60 (ostatné tóny si ľahko vypočítame). Ak teda midi zariadenie dostane stavový bajt \$90 a hneď za ním dátový bajt s hodnotou 60, je to už úplne jasná informácia: „zapni na kanáli 1 notu C1“. Výsledkom bude tón C1, ktorý sa ozve z nášho klávesu alebo počítača. Tento tón bude znieť až dovtedy, kým zariadenie nevypneme, neodpojíme od elektriny, alebo – čo je inteligentnejšie – kým nepošleme príkaz na vypnutie noty. Noty vypína príkaz \$8, preto na vypnutie noty by sme do zariadenia poslali stavový bajt \$80 a hneď za ním dátový bajt 60. Z povedaného vyplýva, že dátové bajty je jednoduchšie čítať ako desiatkové – nie šestnástkové čísla.

Dátové bajty samozrejme nevyjadrujú vždy iba výšku noty. Je to logické, pretože každý príkaz potrebuje doplňujúcu informáciu iného druhu. Kým príkaz pre zapnutie či vypnutie noty (\$9 a \$8) potrebuje v dátovom bajte dodať informáciu o tom, ktorá nota má byť zapnutá či vypnutá, príkaz pre zmenu nástroja (\$C) potrebuje v dátovom bajte určiť, ktorý nástroj má byť zvolený. Dátový bajt je teda doplňujúca rozširujúca informácia pre predošlý stavový bajt, pričom jeden stavový bajt môže potrebovať aj viac dátových bajtov.

Nahliadli sme už pod pokrievku midi a načrtli sme štruktúru správ. V neskorších kapitolách (od kapitoly 7) si každý stavový príkaz konkrétne preberieme a povieme, koľko dátových bajtov potrebuje a čo v nich má byť vyjadrené.

4. Midi hodiny, čas a tempo

Hudba je úzko spojená s časom, pretože sa odohráva v plynúcom čase a bez tejto veličiny by snáď ani nebola možná.

Čas a notový zápis

Už v klasickom notovom zápise sa stretávame s pokusmi zachytiť do notových značiek nejaké tie časové údaje. Sú to jednak informácie o dĺžkach nôt, ktoré môžu byť krátke i dlhé, ale sú to tiež taktové čiary, ktoré nám hovoria niečo o časovom zaradení noty.

Ak máme notový zápis a ideme hrať skladbu podľa neho, potrebujeme nejakým spôsobom odpočítavať čas. Základnou jednotkou tu bude doba, ktorú v prípade potreby delíme na polovicu, štvrtinu či osminku doby. V zásade však počítame iba doby podľa predpísaného taktu (v jednom takte 2, 4, 6 alebo aj 8 dôb). Počítať pritom môžeme voľne v duchu, alebo pomocou metronómu, ktorý si k našej muzike pustíme. Nech to však robíme tak alebo onak, stále budeme iba počítat doby od 1 po X a v notách sa budeme orientovať pomocou taktových čiar. Ak máme napríklad skladbu v štvorštvrt'ovom takte a na začiatku skladby 3 prázdne takty, najprv trikrát napočítame 4 doby a potom začneme hrať. To je všetka naša časová orientácia v notovom zápise. Nepracujeme tu s minútami, sekundami ani inou časovou veličinou, počítame len doby.

Rovnako otvorená je pri hraní z nôt otázka tempa. V notách je tempo určené buď vôbec nijako, alebo slovným označením, ktoré nám povie, či máme hrať skladbu pomaly, rýchlo, rozvláčne alebo rezko. O tom, že pojmy ako rýchlo alebo veselo sú relatívne a každý teda bude hrať inak snáď ani netreba hovoriť.

Čas a midi

Úplne iná je situácia v midi. V midi neexistujú takty, ba dokonca ani celé, štvrt'ové či osminové noty. Celá skladba, zapísaná a uložená v midi, je iba zmesou príkazov, ktoré sú však presne umiestnené na časovej osy a to s presnosťou na mikrosekundy.

Ak hudobník stlačí na hudobnom nástroji klávesu a drží ju napríklad jednu dobu, vytvorí sa dve midi udalosti – jedna pre zapnutie a druhá pre vypnutie noty. Medzi nimi je časový odstup, ktorý je určený s presnosťou na mikrosekundy a je len veľmi malá pravdepodobnosť, že to bude presne jedna štvrt'ová doba. Uchom rozdiel vôbec nepostrehneme, ale pri pohľade do notového zápisu, ktorý nám z našich zahratých tónov vytvorí počítač, bude jasné, že hráme veľmi nepresne. Uvidíme, že niektoré noty začínajú trochu skôr, iné trochu neskôr. Jedny trvajú o nejakú tú mikrosekundu viac ako by mali, iné sú zasa o kus kratšie. Voľným hraním a zachytávaním nôt do počítača alebo sekvenceru tak získame maximálne neúhľadný zápis, ktorý pri spätnom prehrávaní nemusí znieť zle, ale pri pokuse vytlačiť ho, nám dá ešte kus práce upraviť ho do publikovateľnej podoby. Tento problém sa väčšinou snažia riešiť aj samotné hudobné programy procesom, ktorý sa nazýva kvantizácia.

Midi hodiny

Keďže – ako sme si povedali – midi a čas sú úzko späté, musí každé zariadenie, ktoré má nahrávať a ukladať midi príkazy alebo ich prehrávať, obsahovať midi hodiny – anglicky nazývané spravidla ako midi clock. Nejde pritom o hodiny v klasickom zmysle, teda o hodiny, ktoré by ukazovali koľko je hodín, ale ide o hodiny, ktoré s presnosťou na mikrosekundy merajú čas. Ak spustíme nahrávanie novej skladby alebo prehrávanie už existujúcej, spustia sa midi hodiny od hodnoty 0 a tikajú a tikajú. Pri nahrávaní je každý prijatý príkaz označený časovým údajom a uložený. Nahrávka tak obsahuje nie len samotné príkazy, ale aj informáciu, v akom čase príkaz prišiel. Pri spätnom prehrávaní sú tieto príkazy znova posielané v tých istých časových odstupoch, ako prišli. Napríklad obyčajné nahratie jedného tónu si už vyžaduje odmeranie času. Ak spustíme nahrávanie, spustia sa hodiny od času 0. My najprv nič nerobíme, až po chvíli stlačíme tón C1. Midi sekvencer prijme príkaz pre zapnutie noty a zistí, že od spustenia nahrávania ubehlo 3000 mikrosekúnd. Prvý príkaz bude teda doplnený o časový údaj. My po chvíli tón pustíme a náš nástroj pošle sekvenceru príkaz pre vypnutie noty. Sekvencer zistí, že od začiatku nahrávania prešlo už 3500 mikrosekúnd a tak znova vybaví prijatý príkaz časovým údajom a uloží ho. Ak potom zastavíme nahrávanie a naše dielo si prehráme, bude najprv opäť chvíľka ticha a prvý príkaz pre zapnutie tónu sa odošle až vtedy, keď midi hodiny dosiahnu hodnotu 3000. Tón bude znieť až dovtedy, kým midi hodiny nedosiahnu hodnotu 3500.

Midi hodiny sa nachádzajú v každom počítači, ktorý prehráva midi, v každom sekvenceri, ktorý nahráva a prehráva midi skladby, ba aj v našich hudobných nástrojoch, ktoré sú vybavené sekvencermi alebo automatickými doprovodmi. Áno, aj taká banálna vec ako automatické bicie si vyžaduje prítomnosť midi hodín. Aj búchanie bicích sa predsa odohráva v čase a v nástroji musí existovať zariadenie, ktoré stráži, kedy má ktorý tón zaznieť.

Napokon si ešte k téme midi hodiny povedzme, že ak by sme prepojili viac hudobných nástrojov dokopy, musia vždy tikať iba jedni midi hodiny. Práve preto by každý hudobný nástroj mal mať možnosť svoje midi hodiny vypnúť. Ak vypneme na našom klávesovom nástroji midi hodiny, bude to v praxi znamenať, že sa nám viac nepodarí spustiť automatický doprovod ani midi skladbu. Po stlačení play či start sa proste nestane nič. Nástroj má vypnuté hodiny, no zároveň je pripravený prijímať časové povely zo svojho vstupného midi in konektora. Ak teda prepojíme napríklad dva hudobné nástroje midi káblom, na jednom z nich (riadiacom master nástroji) ponecháme zapnuté midi hodiny a na druhom (podriadenom slave) ich vypneme. Teraz môžeme na oboch nástrojoch spustiť automatické doprovody a hrať. Oba nástroje budú hrať úplne synchronne, pretože budú riadené jednými midi hodinami. Práve toto by sa nám nepodarilo dosiahnuť, ak by sme ponechali midi hodiny u oboch nástrojov zapnuté. Veľmi skoro by sme sa mohli časovo rozísť a vôbec – už len predstava, že by sa nám podarilo spustiť na dvoch nástrojoch doprovodné štýly úplne naraz je veľmi absurdná.

Tempo

S časom a hudbou veľmi úzko súvisí otázka tempa. Kým v minulosti sa tempo vyjadrovalo iba slovne (pomaly, stredne rýchlo, rezko a pod.), dnes ho vo svete digitálnej techniky vyjadrujeme presnými číslami. Jednotkou tempa je BPM (beats per minute) a určuje počet štvrt'ových dôb za jednu minútu. Za štandardné tempo sa považuje 120 BPM čo znamená, že za jednu minútu pri tomto tempe odznie 120 štvrt'ových nôt.

Hodnota BPM je však určená pre ľudí a ich predstavu o hudbe v čase. V midi sa tempo vyjadruje v mikrosekundách za jednu štvrt'ovú notu (**μspb – microsecond per beat**). Pripomeňme si, že jedna sekunda má 1000000 mikrosekúnd. Jedna minúta má teda 60 000 000 mikrosekúnd. Počet mikrosekúnd na jednu štvrt'ovú notu vyrátame veľmi jednoducho vzorcom $60\,000\,000 / \text{BPM}$. V našom prípade BPM 120 by $60\,000\,000 / 120$ bolo rovných 500 000 a dostali sme sa k štandardnému midi tempu, ktoré bude v midi zapísané ako 500000 (mikrosekundy za štvrt'ovú dobu - **μspb**), ale nám sa na display nástroja zobrazí ako 120 (BPM – štvrt'ových nôt za minútu).

Je paradoxné, že ak budeme na nástroji tempo uberať, bude číslo BPM klesať, ale číslo **μspb stúpať**. **Je to tým, že čím menej nôt za minútu má zaznieť, tým je každá dlhšia a teda trvá viac mikrosekúnd.**

Pre úplnosť uvádzam najčastejšie používané tempá s ich približnými hodnotami BPM

Veľmi pomalé tempá (40 – 60 BPM): Lentissimo/Adagissimo, Grave, Largo, Lento, Adagio, Larghetto

Pomalé tempá (cca. 60 BPM): Andante, andante con moto, Andantino, Sostenuto, Comodo, Maestoso

Stredné tempá (80 – 90 BPM): Allegro moderato, Allegretto, Animato

Rýchle tempá (120 – 130 BPM): Allegro, allegro con brio, allegro con moto, allegro molto, Allegro assai

Veľmi rýchle tempá (cca. 160 BPM): Allegro vivace, Vivo/vivace

Veľmi rýchle tempá (170 – 210 BPM): Presto/molto presto/molto vivace, Prestissimo

Midi tiky

Vráťme sa však k problematike nahrávania skladieb do midi súborov. Práve tu sa totiž odohráva proces, ktorý nás teraz zaujíma a to je časová pečiatka. Pri reálnom hraní na dvoch nástrojoch naraz v midi systéme časový údaj nie je dôležitý. Príkaz sa jednoducho vykoná ihneď – teda stlačenie klávesy na jednom nástroji má za následok, že zaznie okamžitý tón na

oboch nástrojoch. Ak ale skladbu nahrávame, musí byť každý príkaz vybavený časovou pečiatkou.

Aby to ale nebolo až také jednoduché, používa sa v midi zvláštna časová jednotka a to sú midi tiky. Pripomeňme si, že čas meriame na sekundy (s), ktorých je v jednej minúte rovných 60. Každá sekunda sa však delí na menšie a menšie jednotky takto:

Milisekunda (značka **ms**) je tisícina sekundy, $1 \text{ s} = 1000 \text{ ms}$

Mikrosekunda (značka **μs**) je miliontina sekundy, $1 \text{ s} = 1\,000\,000 \text{ μs}$

Nanosekunda (značka **ns**) je miliardtina sekundy, $1 \text{ s} = 10^9 \text{ ns}$

Pikosekunda (značka **ps**) je biliontina sekundy, $1 \text{ s} = 10^{12} \text{ ps}$

Femtosekunda (značka **fs**) je tisícina biliontina sekundy, $1 \text{ s} = 10^{15} \text{ fs}$

Attosekunda (značka **as**) je miliontina biliontina sekundy, $1 \text{ s} = 10^{18} \text{ as}$.

Nie, nebojte sa, nebudeme rozoberať sekundy až na attosekundy. Budeme ich však premieňať na midi tiky. Základným problémom je totiž tempo. Midi sekvencer všetky príkazy, ktoré dostane, vybaví časovou značkou, no nie v milisekundách ani mikrosekundách, pretože pri takomto značení by bolo možné späťne skladbu prehrať iba v tom tempe, v akom bola zaznamenaná. Použijeme náš príklad z predošlých odsekov: ak spustíme nahrávanie a po troch sekundách stlačíme tón C, sekvencer zistí, že uplynulo 3000 milisekúnd a príkaz uloží. Ak by časová pečiatka obsahovala presne číslo 3000, myslené tým počet milisekúnd, pri spätnom prehrávaní by sa opäť tón ozval po troch sekundách – ani skôr ani neskôr. My však chceme meniť tempo skladby, chceme ju zrýchľovať a spomaľovať. Preto bola vyvinutá zvláštna jednotka midi tiky, ktorá má relatívnu dĺžku a tá sa vyráta pre každú situáciu či skladbu zvlášť zo zadaného tempa a z počtu midi tikov na jednu štvrtú notu.

Tempo a počet midi tikov na štvrtú dobu sú teda dve základné informácie, ktoré vždy potrebujeme. Tempo skladby by malo byť v každej skladbe uložené – ak tak nie je, prehrávač použije štandardné tempo 120 BPM (500 000 μspb). Počet midi tikov na štvrtú dobu je povinný údaj, ktorý musí byť uložený v každom midi súbore a každý sekvencer ho môže mať nastavený inak. Z tohto údaju sa dozvieme, aká dlhá je v danom súbore štvrtá nota. Ak napríklad počet tikov na štvrtú dobu je 380, budeme časový odstup 380 medzi zapnutím a vypnutím noty chápať ako jednu štvrtú notu.

Ak midi prehrávač získa dva uvedené údaje (tempo a počet tikov za štvrtú dobu), vyráta z nich dĺžku jedného tiku v mikrosekundách. Pri nahrávaní a prehrávaní bude potom vždy prevádzať reálny čas na tieto tiky. My sa teda nemusíme vôbec starať o skutočný čas, o sekundy, minúty či mikrosekundy, ale budeme pracovať iba s midi tikmi.

Napokon si však ešte povedzme poslednú dôležitú vec – midi sekvencer neukladá nikdy do súboru počet midi tikov, ktoré uplynuli od začiatku nahrávania, ale počet midi tikov, ktoré uplynuli od poslednej udalosti. Časová pečiatka pri jednotlivých udalostiach – budeme ju nazývať čas delta – teda určuje odstup jednotlivých udalostí od seba a tento odstup bude zachovaný aj pri prehrávaní skladby. Tu sa práve prejavuje výhoda zavedenej veličiny midi tikov, pretože ak zvýšime alebo znížime tempo skladby, zmení sa dĺžka midi tikov, ale odstup príkazov ostane rovnaký. Ak máme teda jeden príkaz, ktorý zapne notu a hneď za ním iný, ktorý ju vypne, je dôležitý práve odstup týchto dvoch príkazov v midi tikoch. Dajme tomu, že tento odstup je 1000. Ak počet midi tikov za štvrtú dobu je 500, potom z uvedeného odstupu 1000 tikov hneď vieme, že pôjde o pólóvu notu. Ak nastavíme rýchle tempo, každý midi tik bude krátky, odstup 1000 tikov uplynie rýchlo a oná nota sa nám bude zdať kratšia. Ak tempo znížime, dĺžka jedného tikov stúpne a odstup 1000 sa nám bude javiť ako dlhá nota.

Pre úplnosť si teraz ešte uveďme, ako by vyzeral zápis jedného štvrt'ového kvintakordu do midi súboru, ak počet tikov na štvrt'ovú notu je 500 (prvé číslo je čas delta – časová pečiatka, ktorú príkaz získal pri prijatí v sekvenceri):

0 – zapni notu C1
0 – zapni notu E1
0 – zapni notu G1
500 – vypni notu C1
0 – vypni notu E1
0 – vypni notu G1

Možno Vás prekvapil veľký počet núl v čase delta. Uvedomme si, že čas delta neuvádza skutočný počet tikov od začiatku skladby, ale od poslednej udalosti. Prvou udalosťou je zapnutie noty C1, ktorá sa má zapnúť v čase 0 – teda zapne sa hneď po spustení prehrávania. Ďalšie dve udalosti však majú tiež čas delta 0, teda budú nasledovať okamžite za sebou (druhá udalosť hneď po prvej a tretia hneď po druhej). To v praxi spôsobí, že tri tóny zaznejú naraz. Potom sa 500 tikov neudeje nič a po oných 500 tikoch prídu zasa tri príkazy tesne za sebou a noty C, E a G vypnú. Z uvedeného odstupu 500 navyše vieme vyčítať, že akord bude znieť jednu dobu.

Ak ste po prvom prečítaní tejto kapitoly neporozumeli ako funguje tempo a ako sa v midi tikoch označujú midi udalosti, odporúčam Vám preštudovať si ju ešte raz. Správne pracovať s časom delta v midi súboroch je veľmi dôležité. Práve editáciou týchto časov môžeme neraz vykonať úpravy, ktoré by nám inak zabrali veľa času. Napríklad posunúť celú skladbu alebo jej úsek o pár dôb či taktov sem alebo tam.

Na precvičenie ponúkam ešte dva príklady midi zápisu (v oboch vychádzame z počtu tikov na štvrt'ku 500):

0 – zapni notu C1
0 – zapni notu E1
0 – zapni notu G1
500 – vypni notu C1
500 – vypni notu E1
500 – vypni notu G1

(Správne riešenie: hneď po spustení prehrávania zaznie kvintakord C E G, ale noty sa budú vypínať postupne po štvrt'ových dobách, takže jednu dobu bude znieť celý akord, po prvej dobe sa vypne C1, po ďalšej E1 a po ďalšej G1)

2000 – zapni notu C1
500 – zapni notu E1
500 – zapni notu G1
500 – vypni notu C1
0 – vypni notu E1
0 – vypni notu G1

(Správne riešenie: prvé štyri doby bude ticho. Až v druhom takte sa ozve nota C1, o ďalšiu dobu sa pridá E1 a o ďalšiu dobu sa pridá G1. Spolu ako akord budú znieť jednu dobu a potom sa všetky naraz vypnú.)

5. Štruktúra a možné typy midi súboru

Po náročných kapitolách o číselných sústavách, bajtoch, bitoch a midi tikoch si dnes trochu vydýchame a pozrieme sa na to, ako vyzerá midi súbor.

Budeme pritom hovoriť o súboroch, ktoré označujeme ako SMFF (Standard midi file format – formát štandardných midi súborov) alebo skráteno SMF (Standard midi file – štandardný midi súbor). Hudbu, výpisy pamätí hudobných nástrojov a rôzne sekvencie je možné zapisovať do rôznych typov súborov, ale my budeme v celom tomto seriáli hovoriť práve o SMF súboroch, ktoré majú príponu mid a sú prehrateľné v počítačoch, midi zariadeniach i mobilných telefónoch.

Štruktúra midi súboru

Každý midi súbor má hlavičku a tracky. Kým v hlavičke sa nachádza iba zopár potrebných údajov, v trackoch sa nachádzajú konkrétne midi príkazy.

Tracky

Viete ako sa v hudobnom štúdiu nahráva CD? Štúdiové magnetofóny majú viac stôp a dokážu teda nahrávať viac nástrojov – každý do inej stopy, ktoré potom naraz prehrávajú. V praxi to vyzerá tak, že bubeník hrá svoj part a ten sa nahráva do jednej stopy. Potom sa táto stopa prehráva a súčasne sa do inej nahráva to, čo hrá basgitarista. Potom sa nástroja chopí klávesista, technik mu do sluchátok pustí prvé dve nahraté stopy a klavirista počúva svojich kolegov z pásu a pritom hrá svoj part. Tento sa nahrá do tretej stopy a tak sa pokračuje, kým každý hudobník a spevák nenahrá a nenaspieval to svoje – každý do samostatnej stopy. Vznikla tak nahrávka, pri ktorej je v každej stope uložený zvuk iného nástroja a nasleduje konečná fáza celej práce a tou je mixovanie – zvukový majster jednotlivé stopy ponastavuje tak, aby to celé znelo dobre. Každý nástroj môže trochu upraviť, pridať alebo ubrať jeho hlasitosť, polohovať ho do pravého či ľavého kanála, alebo naň aplikovať nejaký zvukový efekt. Keď je výsledok uspokojivý, všetky stopy sa v tomto nastavení zmiešajú do jednej výslednej nahrávky, ktorá sa potom uloží do počítača alebo napáli na CD.

Takto isto môže fungovať aj midi súbor. Môže obsahovať viac trackov – stôp – v ktorých sú uložené jednotlivé príkazy. Bežne je zaužívané, že každý nástroj je uložený v samostatnom tracku. Je to z časti logické, pretože – ako sme hovorili už v prvej kapitole – každý nástroj potrebuje svoj vlastný kanál a mnohé programy teda ukladajú každý kanál do samostatného tracku. Nemožno však povedať, že koľko je v skladbe použitých kanálov, toľko bude mať výsledný súbor trackov. Track totiž nemusí obsahovať len noty. Môžu to byť pokojne aj príkazy, ktoré obsluhujú zvukové efekty, vokalizér, alebo premietajú text piesne. Nedá sa vyvodiť žiadny štandard, podľa ktorého by sme určili, koľko trackov midi súbor má mať. Dôležité však je, aby sme rozumeli pojmu track a vedeli si predstaviť, že track je jednoducho zoznam midi príkazov. Tiež je dôležité uvedomiť si, že všetky tracky sa prehrávajú naraz. Ak sú teda v trackoch uložené jednotlivé noty, budú znieť súčasne ako jeden orchester. Ak sú v tracku nenotové príkazy, vykonajú sa vtedy, keď podľa midi hodín nastane ich čas. Každý príkaz má totiž svoj čas delta (pozri predošlú kapitolu) a tak sa môže pokojne stať, že v midi súbore budeme mať 10 trackov, v ktorých budú noty pre 10 nástrojov.

Tie budú prehrávať skladbu, ktorá bude trvať 20 minút. A v jedenástom tracku môžeme mať jediný príkaz, ktorý bude mať čas delta 20 minút a prvú sekundu (samozrejme vyjadrenú v midi tikoch) a keď daný čas nastane, vypíše nám na displej nápis „Konečne koniec“.

Počet príkazov v tracku teda nie je nijako daný ani obmedzený, počet trackov v skladbe rovnako nie. Predsa však aj v tomto existuje určitý východiskový systém, ktorý súvisí s typom midi súboru.

Typy midi súborov

Rozoznávame 3 typy midi súborov, ktoré sú číslované ako 0, 1 a 2.

Midi súbor typu 0 má hlavičku a jeden jediný track. Všetky príkazy sú zmiešané do jedného tracku a je jedno, či sú to príkazy notové alebo nenotové. Dokonca to môžu byť príkazy pre rôzne kanály, takže prítomnosť jedného tracku nás nijako hudobne neobmedzuje a do súboru môžeme zapísať celú skladbu.

Midi súbor typu 1 má hlavičku a viac trackov, pričom prvý track obsahuje okrem samotných dát aj definičné údaje o skladbe ako sú tempo, takt, stupnica, prípadne autor a podobne (aj údaje o takte či tempe sú totiž uložené ako obyčajné príkazy, ktoré po načítaní nastaví prehrávač na požadované hodnoty.). Tieto sa už v ďalších trackoch neuvádzajú. Ostatné tracky už obsahujú len ľubovoľné dáta – notové príkazy, nenotové príkazy, texty piesní a podobne. Všetky sa však prehrávajú súčasne od začiatku do konca – lineárne prehrávanie.

Midi súbory typu 2 obsahujú hlavičku a viac trackov, pričom zámerom vytvorenia tohto typu bolo poskytnúť priestor pre zápis blokových sekvencií pre takzvané patternové prehrávanie. Ako príklad nám môže poslúžiť doprovodný štýl hudobného nástroja, ktorý je rozdelený do viacerých častí (a, B, intro, záver a podobne). Každá časť by bola uložená v samostatnom tracku, pričom track by obsahoval celú sekvenciu danej časti – teda kompletnú hudbu, aj keď vo viacerých kanáloch. Keďže jednotlivé časti štýlu môžu mať rôzne nastavenia tempa či taktu, v tomto type súboru by každý track mal obsahovať tieto definičné údaje. Tracky by sa pri načítaní neprehrávali súčasne (ako v type 1) ale v poradí, ako by si to zariadenie vyžadovalo. Hovoríme o tom však iba teoreticky, pretože v praxi sa tento typ veľmi neujal a dokonca aj súbory štýlov sú zapisované do súborov typu 0, pričom jednotlivé časti (patterns) sú zoradené v jedinom tracku za sebou, oddelené špeciálnymi príkazmi.

Ako vidíme, jednotlivé typy súborov sa líšia iba usporiadaním dát, nie obsahom. Najviac rozšíreným typom je typ 0 a potom typ 1. Do oboch môžeme zapísať kompletne skladby, no zároveň musíme dbať na to, aby naše midi zariadenie podporovalo daný typ súborov. Ak teda máme napríklad klávesový nástroj staršieho dáta alebo lacnejšej kategórie môže sa stať, že úplne platný a korektný midi súbor neprehráme iba preto, že je napríklad typu 1, ktorý náš nástroj nepodporuje. Myslím si však, že dnes je už takých zariadení málo a oba typy (0 i 1) sú veľmi rozšírené a bohato podporované.

6. Hlavička midi súboru

Všetky typy SMF – teda štandardných midi súborov – obsahujú na svojom začiatku hlavičku, ktorá nám prezradí tri dôležité údaje:

- Typ súboru – hovorí, akého typu súbor je. Možné hodnoty sú 0, 1 alebo 2. Bližšie pozri kapitolu 5

- Počet trackov – hovorí, koľko trackov sa v súbore nachádza. Ak je predošlý údaj (typ súboru) 0, počet trackov je vždy 1. Ak je predošlý údaj 1, môže byť trackov viac. Bližšie pozri kapitolu 5.

- Počet midi tikov na štvrt'ovú dobu – dôležitý údaj, ktorý hovorí, koľko tikov trvá jedna štvrt'ová nota. Tento údaj je dôležitý jednak pre správne zobrazenie tónov v notovom zápise, ale aj pre správne vyrátanie a nastavenie tempa skladby. Bližšie pozri kapitolu 4.

Uvedené tri údaje nájdeme v hlavičke každého typu SMF midi súboru. Program Notedit nám ich zobrazí vo svojom nástroji Event editor, ak daný súbor načítame a zobrazíme si obrazovku General. Bližšie pozri seriál Event editor na domovskom webe programu Notedit.

7. Štruktúra a typy midi správ

Potom čo sme sa oboznámili s princípom fungovania midi, nahliadli sme do midi súboru a sme pripravení pracovať s desiatkovými aj šestnástkovými číslami, poďme sa pozrieť na to, čo je v celom systéme midi najdôležitejšie – na konkrétne midi správy.

Čo je midi udalosť

Anglickým slovom event (udalosť) alebo message (správa) sa označuje konkrétny príkaz, ktorý je vyslaný z jedného zariadenia do druhého, je uložený do midi súboru alebo naopak, zo súboru prečítaný a odoslaný. Slovo príkaz nie je celkom výstižné, pretože hoci príkaz prikazuje nástroju vykonať nejakú úlohu (zapnúť notu, pridať hlasitosť alebo zmeniť zvuk nástroja), nie je to dosiahnuté jednou inštrukciou. Nástroj musí prijať konkrétny príkaz k činnosti, ale musí prijať aj ďalšie doplnujúce informácie. Nestačí teda iba príkaz „zapni notu“ alebo „pridaj hlasitosť“, ale musí byť povedané aj akú notu, na akú intenzitu pridať hlasitosť a podobne. Príkaz je teda zložený z viacerých informácií a celý tento komplex je vhodnejšie a výstižnejšie označiť slovíčkom správa (message) alebo udalosť (event). Midi nástroj teda dostane správu, ktorá mu poskytne komplexnú inštrukciu – čo urobiť, aký parameter zmeniť a na akú hodnotu. V tejto chvíli preto definitívne opustíme pojem „príkaz“ a budeme hovoriť o správach alebo udalostiach. Je však dobré pamätať si aj anglické ekvivalenty týchto slov, pretože sa aj v hudobných nástrojoch, dokumentácii alebo v hudobných programoch stretne s pojmami midi message alebo midi event. Ak potom v nejakom programe objavíme nástroj, pomenovaný ako event editor budeme vedieť, že je to nástroj, ktorý nám dovoľuje editovať – upravovať – midi udalosti.

Ako sme si už v predošlých kapitolách povedali, midi správy sú v súboroch uložené v trackoch.

Štruktúra midi správy

Každá midi správa je zložená z viacerých bajtov. My však teraz opustíme technickú rovinu bajtov a bitov a predstavíme si štruktúru midi správy tak, ako s ňou pracuje program Notedit. Veď napokon, tento seriál je určený práve tým, ktorí chcú pracovať s programom Notedit a tak nám experti odpustia, že sa v nasledovných riadkoch dopustíme malých technologických a pojmových nepresností, ktoré sú úplne zámerné s cieľom čo najjednoduchšie priblížiť užívateľom problematiku midi správ.

Teda začnime ešte raz: Každá midi správa je zložená zo šiestich častí, pričom záleží od typu správy, ktorá časť nesie informáciu. V určitom type správ môžu byť obsadené iba prvé políčka a ostatné ostávajú prázdne, v inom type zasa je to presne naopak.

Konkrétne časti midi správy sú:

1. *Čas delta* – prvý údaj, ktorým musí byť vybavená každá správa. Tento údaj stojí na prvom mieste, pretože sa do zariadenia neposielajú. Slúži iba prehrávaču, aby vedel, kedy má danú správu poslať. Čas delta je vyjadrený v midi tikoch (pozri kapitolu číslo 4) a neurčuje čas, ktorý uplynul od začiatku skladby ale čas, ktorý uplynul od predošlej udalosti. Poradie

udalostí v tracku je teda dôležité a iba zmenou času delty ho nemôžeme zmeniť. Ak potrebujeme zmeniť poradie správ, musíme ich fyzicky preusporiadať.

2. *Status* – najdôležitejšia informácia - stavový bajt, ktorý určuje, čo sa má v midi zariadení vykonať. Obsahuje informáciu o udalosti a kanáli, na ktorom sa táto má vykonať. Status čítame spravidla ako hexadecimálne číslo, pretože to nám umožňuje z jedného čísla dešifrovať oba údaje. Status môže byť číslo v rozsahu \$80 až \$FF, pričom prvá číslica určuje udalosť a druhá číslica určuje kanál (kanály sa číslujú od nuly).

3. *Hodnota 1* – dátový bajt, ktorý dopĺňuje status. Čítame ho spravidla ako klasické desiatkové číslo. Môže mať rozsah 0 až 127.

4. *Hodnota 2* – dátový bajt, ktorý dopĺňuje status. Čítame ho spravidla ako klasické desiatkové číslo. Môže mať rozsah 0 až 127. Nie každý status však vyžaduje dva dopĺňujúce dátové bajty. Ak je v položke status vyjadrený príkaz, ktorý vyžaduje iba jednu dopĺňujúcu informáciu, bude mať druhý dátový bajt Hodnota 2 hodnotu 0. Bližšie o dátových bajtoch Hodnota 1 a Hodnota 2 pozri v kapitole 8.

5. – *Meta* – dátový bajt, ktorý má väčšinou hodnotu 0. Hodnotu vyššiu než 0 má len vtedy, ak status hovorí, že ide o takzvanú meta správu (pozri nižšie typy správ). V takom prípade hodnota meta určuje, o aký druh správy pôjde. Konkrétne meta správy pozri v kapitole 12.

6. *SysEx* – číselný alebo textový reťazec, ktorý je väčšinou prázdny. Použitý je iba vtedy, ak status hovorí, že ide o SysEx alebo meta správu – bližšie pozri nižšie odsek o typoch správ.

Ako som už naznačil, nie v každej správe sú využité všetky položky. Dôležitá je vždy položka status, ktorá určuje, akého typu správa je. Konkrétny typ správy potom určí, ktoré políčka budú použité a ktoré nie. Nepoužité číselné položky budú mať hodnotu 0 a nepoužitá položka SysEx bude predstavovať prázdny reťazec.

Typy midi správ

Midi správy môžeme rozdeliť v prvom rade na dve veľké skupiny a to sú kanálové a nekanálové správy.

Kanálové správy

Kanálové správy sú správy, ktoré sa vykonajú len na konkrétnom kanáli a neovplyvnia zvyšok skladby. Sem patria správy ako zapni notu, vypni notu, zmeň nástroj, nastav kontrolér a podobne.

Kanálové správy sú tie, ktoré majú status v rozsahu \$80 až \$EF. Keďže vieme, že druhá číslica v stave je kanál, je pri kanálových správach práve táto druhá číslica tá, ktorá určí, na ktorom kanáli sa udalosť vykoná (kanály číslujeme od nuly). Prvá číslica predstavuje

konkrétnu udalosť. Z uvedeného teda vidíme, že v midi máme iba 7 kanálových udalostí (\$8, \$9, \$A, \$B, \$C, \$D a \$E). Konkrétne si ich vymenujeme v kapitole 8.

Nekanálové správy

Nekanálové správy sú správy, ktoré sa vykonajú pre celú skladbu. Sú to tie, ktoré majú status v rozsahu \$F0 až \$FF. V tomto prípade druhá číslica statusu nebude vyjadrovať kanál, ale druh príkazu, nakoľko prvé číslo je obsadené pre samotnú informáciu, že ide o nekanálovú udalosť.

Nekanálové správy môžeme rozdeliť do troch väčších skupín:

Systémové správy

Systémové správy sú tie, ktoré majú status v rozsahu \$F1 až \$FF – teda vlastne všetky okrem \$F0. Systémové správy sú správy, ktoré riadia midi zariadenie (napríklad reset zariadenia, synchronizácia a podobne). Konkrétne si ich vymenujeme v kapitole 11.

Meta správy

Meta správy sú tie, ktoré majú status \$Ff, ale položku meta majú inú než 0. Práve položka meta určuje, že ide o meta správu a hovorí, o akú správu ide. Meta správy sú správy, ktoré väčšinou priamo neovplyvňujú znenie skladby (s výnimkou tempa a taktu, čo sú tiež meta správy), ale prinášajú rôzne zobrazované alebo nastavovacie informácie (stupnica, text piesne, rôzne zobrazované poznámky a podobne). Samotná informácia, ktorá sa má zobraziť alebo nastaviť, je uložená v poslednom políčku správy – SysEx. Konkrétne meta správy si vymenujeme v kapitole 12.

SysEx správy

Sú to správy, ktoré majú status \$F0. Uložené sú v políčku SysEx avšak nie sú tvorené textom, ale sériou hexadecimálnych čísel. Tieto správy nie sú všeobecné a je na výrobcovi každého zariadenia, aký úkon tej ktorej správe priradil. Pri používaní týchto správ sme nútení nazeráť do dokumentácie nášho midi zariadenia a úkon, ktorý po vyslaní tejto správy vykoná naše zariadenie takmer naisto nevykoná iné zariadenie. SysEx správy sú teda určené len pre konkrétne midi zariadenie a preto sa aj nazývajú exkluzívne systémové správy – skrátene SysEx (system exclusive messages). Viac o nich pohovoríme v kapitole 13.

8. Kanálové správy

Skôr ako si začneme menovať konkrétne kanálové správy, povedzme si najprv niečo o číslovaní nôt. Väčšina kanálových správ totiž bude obsahovať informáciu o konkrétnej note a táto bude samozrejme vyjadrená číslom. Ako sa teda v midi číslujú noty?

Číslovanie nôt

Noty sú v midi číslované od 0 po 127, pričom ale najnižšie pozície nie sú obsadené. Naše bežné C1 predstavuje hodnota 60 a z tejto polohy si ostatné tóny ľahko vyrátame. Klasický klavírny rozsah od subkontra A po C5 by sme v midi vyjadрили číslami 23 až 108.

Pritom ale mám stále na mysli klasický hudobný nástroj so štandardným ladením. Ak totiž pošleme nástroju príkaz "zapni notu 60", nezaručuje to, že v nástroji sa skutočne ozve C1. Ak takýto príkaz pošleme nástroju, ktorý je naladený inak, ozve sa tón, zodpovedajúci príslušnému ladeniu. To sú však menej bežné situácie a spomínam to tu len preto, aby sme pochopili podstatu, totiž, že midi správa je len nositeľkou požiadavky, aby sa v nástroji zapol tón s číslom 60. Čo sa nástroju skutočne vyderie z úst je už plne v jeho kompetencii a záleží to od toho, čo má k číslu 60 priradené. My to zo strany "posielateľov" midi správ nemôžeme nijako ovplyvniť a zmenu môžeme dosiahnuť iba konfiguráciou nástroja, ak nám to on samozrejme umožňuje. Tu len pre zaujímavosť povedzme, že skutočne existujú nástroje, ktoré majú programovateľnú možnosť priradiť k jednotlivým číslam rozličné tóny a tak by sme si teoreticky mohli zostaviť vlastné divoké stupnice. Na čo by to ale bolo dobré? V ďalších podkapitolách teda vychádzajme z toho, že máme pred sebou klasicky naladený nástroj a tón 60 je skutočne C1.

Majme však na pamäti ešte aj to, že v škále 0 až 127 nepracujeme iba s tónmi. Kým napríklad pri zvolenom zvuku klavíra predstavuje za normálnych okolností číslo 60 tón C1 a číslo 62 tón D1 (ako ste iste postrehli, číslo 61 predstavuje poltón Cis), pri zvolení sady bicích sa pod tými istými číslami môžu ukrývať údery bubnov, paličiek, metličiek alebo činelov. Aj bicie sú totiž z technického hľadiska noty ako každé iné. Stačí, ak si na svojom nástroji zvolíte nejakú sadu bicích a zahráte si stupnicu. Rovnako pri zvolení sady efektov sa pod číslom 60 môže ukrývať klaksón auta a pod číslom 62 šumenie mora. Technicky je to však stále to isté, ako keby sme pri tejto zvolenej sade stlačili na nástroji klávesu C1.

Zjednodušene povedané: zaslaním príkazu s číslom tónu 60 vždy dosiahneme to isté, ako keby sme stlačili na nástroji klávesu C1.

V závere tejto kapitoly vám ponúkam kompletnú tabuľku tónov s ich midi označením pri štandardnom ladení.

A teraz sa už pozrime na konkrétne kanálové správy. Ako už bolo neraz povedané, je ich 7 a vykonávajú sa iba na jednom konkrétnom kanáli.

\$9K – Note on – zapnutie noty

Hoci tento príkaz nie je v poradí prvý, ale až druhý, zámerne som zmenil poradie, pretože je logickejšie notu najprv zapnúť a až potom ju vypnúť. Venujme sa teda najprv v poradí druhému príkazu \$9, ktorý je v angličtine nazývaný note on a v slovenčine ho môžeme

nazvať "zapnutie noty". Správa s týmto príkazom musí vždy obsahovať stavový bajt (status) a dva dátové bajty.

Stavový bajt – status – má vždy hodnotu v rozsahu \$90-\$9F, pričom prvá cifra hovorí, že sa má zapnúť nota a druhá udáva kanál, na ktorom sa to má stať. Kanály sú číslované od 0 po 15, takže príkaz \$90 zapne notu na prvom a príkaz \$9F na šestnástom kanáli.

Prvý dátový bajt – hodnota 1 – obsahuje číslo noty, ktorá sa má zapnúť – číselná hodnota môže byť v rozsahu 0 – 127.

Druhý dátový bajt – hodnota 2 – určuje, akou silou má byť nota zapnutá. Táto vlastnosť sa v angličtine označuje slovíčkom *velocity* a ide vlastne o silu úderu na kláves. Moderné nástroje majú dynamické klávesnice a tak nie je jedno, akou silou tón udrieme. Silnejší úder má za následok hlasnejší tón, ale aj výraznejšiu farbu tónu, kým jemnejší úder vygeneruje tichší a mäkší tón. Túto silu úderu (*velocity*) môžeme v midi určiť práve druhým dátovým bajtom, ktorého číselná hodnota môže byť v rozsahu 0 – 127, pričom 0 je najtichší a teda vlastne nepočuteľný tón a 127 je najsilnejší tón.

Príklady:

\$90 60 127 – zapne notu C1 najvyššou silou na kanáli 1

\$93 72 70 – zapne notu C2 strednou silou na kanáli 2;

\$9A 50 0 – zapne notu malé D nulovou silou na jedenástom kanáli – pozor, tón pre jeho nulovú silu nebude počuť.

Ak nástroj obdrží správu typu \$9, zapne notu a drží ju zapnutú, kým opäť tú istú notu nevypneme. V praxi to však neznamená, že nota bude celý čas znieť. V prípade klavíra alebo gitary bude tón znieť tak dlho, ako je to nastavené v jeho zvukovej obálke, teda ako je pre tón tohto typu prirodzené (klavír udrie a doznie, gitara brnkne a doznie), slák či harmonika bude znieť nepretržite. Musíme si však uvedomiť, že v oboch prípadoch – či už tón znie alebo nie – ak nebol vypnutý príkazom pre vypnutie noty, zaberá miesto v polyfónii nástroja, čiže aj keď ho sluchom nepočujeme, nástroj s ním ráta ako s hrajúcim. Každú zapnutú notu je teda korektné vypnúť.

\$8K – Note off – vypnutie noty

Teraz sa dostávame vlastne k prvému kanálovému príkazu, ktorým je príkaz \$8 – anglicky nazývaný *note off*, v slovenčine "vypnutie noty". Ako ste si už iste domysleli, tento príkaz vypne notu, ktorú sme predtým zapli. Rovnako ako *note on* aj *note off* musí mať stavový bajt a dva dátové bajty.

stavový bajt môže byť v rozsahu \$80 až \$8F, pričom druhá cifra určuje kanál.

Prvý dátový bajt určuje notu, ktorá má byť vypnutá – číselná hodnota môže byť v rozsahu 0 – 127.

Druhý dátový bajt vyjadruje rýchlosť vypnutia, čo znie síce podivne, no v praxi je tento bajt vždy s hodnotou 0.

Kompletná správa *note off* teda vyzerá takto: \$80 60 0 – vypne notu C1 na prvom kanáli.

Pre zaujímavosť ešte dodám, že v praxi sa môžeme stretnúť aj s tým, že niektoré nástroje príkaz *note off* vôbec nepoužívajú a nahrádzajú ho príkazom *note on*, ktorý má na druhom dátovom bajte hodnotu 0. V praxi je to vlastne zapnutie noty s nulovou hlasitosťou, no keďže z logického hľadiska zapnúť notu, ktorá neznie, je blbosť, majú dnešné nástroje schopnosť pochopiť správu *note on* s druhým dátovým bajtom 0 aj ako *note off*.

\$AK - Polyphonic aftertouch – individuálna tlaková citlivosť

Niektoré nástroje, ak stlačíte klávesu iba miernou silou a ozve sa tón, ktorý je relatívne tichý a mäkký, majú schopnosť po dotlačení klávesy tento tón ďalej spracovať. Ide teda vlastne o stlačenie tónu na dvakrát, pričom každá klávesa reaguje zvlášť. Preto hovoríme o individuálnej tlakovej citlivosti – reakcii na dotlačenie jednej konkrétnej klávesy. V midi túto vlastnosť vyvoláme príkazom \$A, ktorý má jeden stavový a dva dátové bajty.

Stavový bajt môže mať rozsah \$A0 až \$AF, pričom druhá cifra určuje kanál.

Prvý stavový bajt logicky určuje notu, ktorej sa má efekt týkať – rozsah 0 – 127.

Druhý stavový bajt určuje silu, aká má byť na dotlačenie klávesy použitá.

Táto vlastnosť je však relatívne málo rozšírená a osobne som ešte nemal v rukách midi zariadenie, ktoré by na midi správy tohto typu korektne reagovalo. Z tohto dôvodu sa ani v midiskladbách so správami polyphonic aftertouch veľmi nestretáme.

\$BK – Controllers – Controll mode change - - kontroléry

Kontroléry – toto slovo ste už určite v súvislosti s midi počuli. Mnohí ľudia slovo poznajú, no nie celkom rozumejú jeho významu. Pritom to nie je až také zložité. Všetky doterajšie midi správy ovplyvňovali noty, teda robili za nás to, čo by sme my robili na klaviatúre nástroja. Kontroléry za nás vykonávajú to, čo by sme my urobili rukou na ovládacom paneli. Kontroléry sú teda príkazy, ktoré otáčajú kolieskami nastavení, alebo zapínajú či vypínajú spínače.

Zaiste vám je jasné, že ovládať všetky vlastnosti nástroja sa jednou správou nedá. Preto pri midi správach typu kontrolér vlastne ide o celú skupinu ovládačov. Midi správa typu kontrolér má pritom takú istú štruktúru, ako všetky doterajšie správy – musí obsahovať jeden stavový a dva dátové bajty.

Stavový bajt je vždy v rozsahu \$B0 až \$BF, pričom prvá cifra nástroju hovorí, že ide o kontrolér a druhá označuje kanál, na ktorom sa má zmena vykonať.

Prvý dátový bajt určuje číslo kontroléru. Toto je veľmi dôležité, pretože práve týmto číslom je nástroju povedané, ktorú vlastnosť treba zmeniť. Číselná hodnota môže byť v rozsahu 0 až 127 a každému číslu zodpovedá jedna vlastnosť: napríklad kontrolér číslo 7 zmení hlasitosť, číslo 10 stereo polohu, číslo 67 sustain. Kompletný zoznam kontrolérov si preberieme v kapitole 9.

Druhý dátový bajt obsahuje hodnotu, ktorá sa má nastaviť vlastnosti, určenej v prvom dátovom bajte. Ide tiež o rozsah 0 až 127, pričom v prípade potenciometrov môže každé číslo určovať inú hladinu nastavenia, v prípade spínačov však spravidla čísla od 0 po 63 predstavujú polohu off (teda vypnutie) a čísla v rozsahu 64 až 127 polohu on (teda zapnuté).

Príklady:

\$B0 7 127 – nastaví hlasitosť prvého kanála na maximum

\$B0 7 0 – nastaví hlasitosť prvého kanála na minimum

\$B3 10 10 – nastaví stereo polohu štvrtého kanála takmer celkom doľava

\$B8 67 127 – zapne sustainový pedál na kanáli 9

Problematiku kontrolérov si podrobnejšie rozoberieme v kapitole 9. Nateraz stačí ak si zapamätáme, že kontroléry sú riadiace správy, ktoré neovplyvňujú samotné tóny, ale

nastavenia nástroja a že ich do nástroja vysielame ako klasickú kanálovú správu s prvým stavovým bajtom \$B.

\$CK – Program change – zmena programu

Aký program to ideme meniť? Možno vás prekvapí, že midi terminológia pod pojmom program (skratka prg.) rozumie konkrétny zvuk vášho nástroja. Ak máte v nástroji napríklad 389 zvukov, v midi povieme, že máte 389 programov, ktoré počas skladby potrebujeme zapínať a meniť. Ak v midi skladbe nie je zahrnutý žiadny príkaz pre zmenu programu, bude nástroj hrať zvukom klavíra, pretože tento zodpovedá programovému číslu 0.

Prvé elektronické nástroje mali iba niekoľko zvukov, preto sa tvorcovia prvých noriem domnievali, že stačí, ak bude na voľbu nástroja použitá jedna kanálová správa, ktorá bude mať (tak ako to už v midi chodí) dátový bajt v rozsahu 0 až 127, pričom každé číslo bude predstavovať jeden konkrétny zvuk. Dlhú dobu to aj tak naozaj fungovalo a na zvolenie zvuku stačilo nástroju poslať jedinou správou typu Program change, ktorá mala iba jeden stavový a jeden dátový bajt. Všimnite si, že doteraz mali všetky kanálové správy dva stavové bajty, až teraz prvýkrát narážame na správu, ktorej stačí jeden stavový bajt. Druhý je prázdny (resp. môže obsahovať nulu, ale nástrojom je ignorovaný).

Stavový bajt správy Program change má hodnotu v rozsahu \$C0 až \$CF, pričom prvá cifra nástroju oznamuje, že treba zmeniť zvuk a druhá cifra určuje, na ktorom kanáli sa to má udiť.

Prvý dátový bajt je v číselnom rozsahu 0 až 127 a určuje, ktorý zvuk má byť zapnutý. Usporiadanie zvukov je pritom štandardizované tak, že v sade zvukov začínajú najprv klavírové zvuky, potom nasledujú zvuky elektrických klavírov a tak ďalej, až po zvuky rôznych efektov, ako sú dažde, potlesky alebo helikoptéry. Každý nástroj by mal vo svojej dokumentácii obsahovať tabuľku všetkých zvukov, pri ktorých je uvedené aj ich číslo.

S týmto všetkým by sme pohodlne vystačili, keby nástroje aj dnes mali iba 127 zvukov. Nástroje však dnes majú už viac ako tisíc zvukov, preto sa aj systém ich voľby musel rozšíriť. Podrobne si o tom povieme v kapitole 10. Nateraz stačí ak si zapamätáte, že konkrétny program volíme správou typu \$C, ktorá má so sebou iba jeden dátový bajt s číslom zvuku.

\$DK – Chanel aftertouch – spoločná tlaková citlivosť

Už sme hovorili o individuálnej tlakovej citlivosti, teda o tom, že ak na niektorých nástrojoch stlačíte akord iba miernou silou a potom klávesu dotlačíte, môže nástroj na toto dotlačenie (teda doslovne povedané následné stlačenie – aftertouch) zareagovať dodatočným efektom, ktorý aplikuje na zvuk jedného konkrétneho tónu, ktorý bol stlačený. V midi sme tento efekt vyvolávali správou typu \$A – polyphonic aftertouch.

Teraz je reč o tom istom, akurát že po dotlačení klávesy nebude spracovaný iba jeden konkrétny tón, ale všetky tóny toho istého zvuku – odborne povedané toho istého kanála. Preto sa aj správa, o ktorej hovoríme, nazýva Chanel aftertouch – teda kanálová tlaková citlivosť, alebo – ako sa bežnejšie hovorí – spoločná tlaková citlivosť. Správa tohto typu má iba stavový bajt a jeden dátový bajt. Logicky totiž odpadá určenie konkrétnej noty, ktoré bolo potrebné pri individuálnej tlakovej citlivosti.

Stavový bajt môže mať rozsah \$D0 až \$DF, pričom druhá cifra určuje kanál.

Prvý dátový bajt určuje silu, aká má byť na dotlačenie použitá.

Rovnako ako v prípade individuálnej tlakovej citlivosti, aj spoločná tlaková citlivosť je relatívne málo implementovaná a tak je dosť možné, že ak takúto správu aj pošlete, váš nástroj na ňu nijako nezareaguje.

Dvojbajtové správy – MSB a LSB

Všetky doterajšie správy boli jednobajtové. Možno si teraz povieť, ako je to možné, keď každá obsahovala minimálne jeden stavový a jeden dátový bajt. To áno. Doterajšie správy síce obsahovali dva alebo až tri bajty, ale každý bajt niesol jednu samostatnú informáciu, napríklad typ príkazu, číslo noty, úroveň nastavenia potenciometru. To všetko sú hodnoty, ktoré sa dali pohodlne zašifrovať do jedného bajtu, pretože na to stačil rozsah 0 až 127. Ako sme si totiž povedali v kapitole 3, midi bajty majú prvý bit vždy obsadený, takže namiesto plných 8 bitov máme iba 7 bitov na to, aby sme vyjadrili čo potrebujeme. To nás obmedzuje len na 128 kombinácií. Sú však prípady, kedy 128 možností nestačí. Preto tvorcovia midi vymysleli takzvané dvojbajtové správy. To znamená, že do nástroja prídu dva bajty, ktorých hodnoty sa ale zlúčia a vznikne jedna konečná hodnota. Ak totiž vezmeme 7 bitov z jedného a 7 bitov z druhého bajtu, máme k dispozícii už 14 bitov, do ktorých môžeme zapísať číslo v rozsahu 0 až 16383. Zaiste uznáte, že to je poriadny rozdiel.

Ak teda posielame dvojbajtovú správu, budeme číselnú hodnotu rozkladať na dve časti. Tieto nazývame skratkami MSB a LSB. Ak teda uvidíte niekde v midi dokumentácii alebo tabuľke skratky MSB a LSB, vnímajte ich vždy tak, že sú to dve čísla, ktoré musia byť nástroju poslané spolu, pretože až ako celok – po zlúčení – vytvorí jednu konečnú hodnotu.

Podotýkam, že pod pojmom zlúčenie nemám na mysli matematické sčítanie. Teda ak MSB je 100 a LSB je tiež 100, výsledok nebude 200 (teda súčet) ani 10000 (súčin). Musíme stále pamätať na to, že pracujeme s bajtmi a bitmi. Preto ak chcete porozumieť tomu, ako funguje MSB a LSB, preštudujte si dôkladne kapitolu 2, kde sa píše o číselných sústavách. Potom vezmite číslo MSB (v našom príklade 100) a zapíšte si ho v dvojkovej sústave. Dostanete číslo %1100100. Potom vezmite číslo LSB (v našom príklade takisto 100) a pripíšte ho za prvé MSB číslo, ktoré ste si zapísali v dvojkovej sústave. Dostanete opäť číslo %1100100. A práve teraz nastáva ten okamih zlúčenia – obe tie dlhé desiatkové čísla máme zapísané vedľa seba (MSB je vždy prvé) a všetkých 14 cifier prečítame spolu ako jedno dlhokánske binárne číslo %11001001100100. Ak si trúfate, môžete sa pokúsiť previesť toto binárne číslo na naše klasické desiatkové číslo. (Metóda ako to urobiť je opísaná v kapitole 2). Pre túto chvíľu vám pomôžem a prezradím, že toto dlhokánske dvojkové číslo je v desiatkovej sústave 12900.

Aby sme si to vedeli lepšie predstaviť, prirovnám to k posielaniu SMS správ. Predstavte si, že by ste kamarátovi posielali SMS, ale technika by bola tak v plienkach, že do jednej SMS by sa zmestili iba 4 znaky. Slová ako AHOJ UZ Som DOMA by ste v pohode poslali, pretože každé by sa vám zmestilo do jednej správy. Potom by ste sa ale chceli kamoša spýtať, či nepríde k Vám. Slovo NEPRIDES už do jednej správy proste nenapracete. Preto by ste si ju museli rozdeliť na časť MSB (začiatok) a LSB (koniec). Najprv by ste poslali časť MSB NEPR a potom časť LSB IDEs. Váš kamoš by však pochopil, že ide o jedno slovo, spojil by si tie dva nezmysly do celku a pochopil by, že ho voláte k sebe. Rovnako ak potrebujeme midi nástroju poslať číslo väčšie ako 128, musíme ho rozdeliť na časť MSB a LSB, pričom po ich odoslaní si ich nástroj zlúči a pochopí ako jednu informáciu. Keďže na poslanie jedného čísla potrebujeme v tomto prípade dva bajty, hovoríme o dvojbajtových správach.

Takýmto typom správy je posledná kanálová správa Pitchband.

\$EK – Pitchband – ohýbanie tónu

Koliesko pre ohýbanie tónu pozná asi každý hráč. Ak si malé dieťa sadne k nášmu vysoko profesionálnemu nástroju, hoci nevie zahrať ani "Jedna druhej riekla", môžeme mať istotu, že sa dobre zabaví s kolieskom pre ohýbanie tónu. Správa, o ktorej je práve reč, by podľa midi normy mala vyvolávať práve tento efekt a to s možnosťou veľmi malých skokov, pretože jednotlivé skoky nevyjadrujeme číslami 0 až 127, ale číslami 0 až 16383. Správa Pitchband má jeden stavový a dva dátové bajty.

Stavový bajt je v rozsahu \$E0 až \$EF, pričom prvá cifra hovorí, že ide o pitchband a druhá určuje kanál.

Dva dátové bajty predstavujú MSB a LSB časť hodnoty s tým, že v tejto správe sú zadávané opačne, teda najprv LSB a až potom MSB. Nástroj prijme našu správu, a až keď po časti LSB prijme aj de facto začiatok (MSB), spojí si obe časti a preráta na výslednú hodnotu. Pre nás to v praxi znamená, že najprv začneme pridávať hodnoty v prvom dátovom bajte a až potom v druhom. Ak si spomínate na kapitolu o číselných sústavách, tak toto je vlastne niečo podobné, ako keby sme pracovali so stodvadsaťosmičkovou číselnou sústavou. Ak majú oba bajty hodnotu 0, je výsledná hodnota 0. Ak prvý bajt dosiahne hodnotu 127 a druhý je stále 0, je to spolu stále 127. No keď hodnotu na prvom dátovom bajte opäť vrátime na 0 a druhý zvýšime na 1, bude to už vlastne 128. Inak povedané, pridávaním hodnoty o 1 na prvom dátovom bajte (LSB) zvyšujeme celkovú hodnotu o 1, ale ak budeme pridávať hodnoty po jednom na druhom dátovom bajte, zvyšujeme výslednú hodnotu každým krokom o 128. Vo chvíli, keď budú oba dátové bajty mať hodnotu 127, bude spoločný výsledok 16384. Možno sa to na prvý pohľad zdá zložité, no môžete si zapamätať jednoduchý vzorec: výslednú hodnotu ľahko vyrátame tak, že MSB (teda druhý dátový bajt) vynásobíme 128 a k výsledku prirátame hodnotu LSB (prvého dátového bajtu). Napríklad, ak nastavíme oba bajty na hodnotu 30, výsledná hodnota je 3871. Midi norma pritom hovorí, že kľudovej polohe kolieska ohýbania tónu zodpovedá hodnota 2000, teda správa s dátovými bajtmi 80 15.

Pre upresnenie však podotkneme aj to, že nie všetci výrobcovia implementovali tento príkaz dôsledne a preto sa často stáva, že na ohýbanie tónu stačí aj rozsah 0 až 127, a teda obsadzujeme iba prvý dátový bajt a druhý nechávame prázdny. Je vhodné tento príkaz metódou pokus/omyl otestovať a používať podľa toho, ako je implementovaný vo vašom nástroji.

Tabuľka kanálových správ

Prebrali sme si podrobne kanálové správy. Pre zopakovanie vám ponúkame súhrnnú tabuľku. Písmeno K v stĺpci status je skratka od slova kanál (pamätajte, že ich číslujeme od 0 po 15):

Status	Hodnota 1	Hodnota 2	Názov	Vysvetlenie
\$8K	Číslo noty 0-127	Sila noty 0-127	Note off	Vypnutie noty
\$9K	Číslo noty 0-127	Sila noty 0-127	Note on	Zapnutie noty
\$AK	Číslo noty 0-127	Sila noty 0-127	Polyphonic aftertouch	Individuálna tlaková citlivosť
\$BK	Číslo kontroléru 0-127	Hodnota 0-127	Controll change mode	Kontrolér
\$CK	Číslo programu 0-127		Program change	Zmena programu
\$DK	Sila úderu 0-127		Chanel aftertouch	Tlaková citlivosť spoločná pre celý kanál
\$EK	LSB 0-127	MSB 0-127	Pitchband	Ohýbanie tónu

Tabuľka tónov a ich midi čísel

Záverom ešte sľúbená tabuľka kompletnej klavírnej klaviatúry s midi číslami tónov:

Tabuľka 1:

Subkontra	Kontra	Veľké	Malé
	C - 24	C - 36	C - 48
	Cis - 25	Cis - 37	Cis - 49
	D - 26	D - 38	D - 50
	Dis - 27	Dis - 39	Dis - 51
	E - 28	E - 40	E - 52
	F - 29	F - 41	F - 53
	Fis - 30	Fis - 42	Fis - 54
	G - 31	G - 43	G - 55
	Gis - 32	Gis - 44	Gis - 56
A - 21	A - 33	A - 45	A - 57
Ais - 22	Ais - 34	Ais - 46	Ais - 58
H - 23	H - 35	H - 47	H - 59

Tabuľka 2:

Jednočiarkové	Dvojiarkové	Trojčiarkové	Štvorčiarkové
C - 60	C - 72	C - 84	C - 96
Cis - 61	Cis - 73	Cis - 85	Cis - 97
D - 62	D - 74	D - 86	D - 98
Dis - 63	Dis - 75	Dis - 87	Dis - 99
E - 64	E - 76	E - 88	E - 100
F - 65	F - 77	F - 89	F - 101
Fis - 66	Fis - 78	Fis - 90	Fis - 102
G - 67	G - 79	G - 91	G - 103
Gis - 68	Gis - 80	Gis - 92	Gis - 104
A - 69	A - 81	A - 93	A - 105
Ais - 70	Ais - 82	Ais - 94	Ais - 106
H - 71	H - 83	H - 95	H - 107

9. Kontroléry

Čo sú kontroléry

Kontroléry – anglické controllers - sú riadiace príkazy, ktoré za nás nastavujú rôzne vlastnosti nástroja, ktoré by sme ručne nastavovali otáčaním koliesok a zapínaním či vypínaním spínačov na ovládacom paneli. Kontrolérmi meníme hlasitosť či stereo polohu skladby, kontrolérmi zapíname a vypíname sustain alebo echo efekty. Skrátka, kontroléry sú tie riadiace nástroje, ktoré nám pomáhajú dať našej skladbe, ktorú sme si už zostavili pomocou nôt, ten správny zvuk.

Kontroléry sú do zariadenia posielané kanálovou správou typu \$BK, pričom B určuje, že ide o kontrolér a K je číslo kanála, na ktorom má byť kontrolér použitý. (O kanálových správach pozri predošlú kapitolu 8.) Prvý dátový bajt môže byť v rozsahu 0 až 127 a hovorí, o aký typ kontroléru ide. Podľa tejto hodnoty teda nástroj vie, ktorú jeho vlastnosť chceme meniť. Správne ste usúdili, že máme k dispozícii 128 kontrolérov, pričom každému zodpovedá jedno číslo. Druhý dátový bajt v rozsahu 0 až 127 určuje hodnotu, na ktorú sa má daná vlastnosť, vyjadrená číslom kontroléra, nastaviť. 0 je pritom minimálna a 127 maximálna hodnota. Takže ak chceme nastaviť hlasitosť prvého kanála zhruba na polovicu, použijeme na to kontrolér 7, ktorý nastavuje hlasitosť a pošleme správu nasledovného typu: \$B0 7 63 (kontrolér na kanáli 1, hlasitosť, hodnota zhruba na stred).

Teraz si už preberme konkrétne kontroléry. Tie môžeme z hľadiska významu a spôsobu používania rozdeliť na šesť skupín a to kontinuálne kontroléry - potenciometre, spínače, modulačné kontroléry - modulátory, špeciálne kontroléry, špeciálne riadiace povely a režimové povely.

Skôr ako si postupne všetky tieto skupiny predstavíme ešte upozorňujem, že toto členenie, ako aj čísla a význam jednotlivých kontrolérov je iba rámcový súhrn, ako vyplýva zo súčasných používaných noriem, ktorých však máme viac (Pozri kapitolu 14). Kým teda niektoré kontroléry sú všeobecne používané tak, ako ich uvádza základná GM norma (hlavne kontroléry 1, 7, 10, 11, 64, 100 a 101), iné sa môžu svojim významom líšiť podľa toho, ako ich výrobca v danom modeli použil.

1. Kontinuálne kontroléry – potenciometre (0-63)

Prvou skupinou kontrolérov sú potenciometre, alebo tiež kontinuálne kontroléry. Je ich až 64 a ovplyvňujú tie funkcie nástroja, ktorých nastavenie si vyžaduje veľmi podrobné krokovanie. Z toho dôvodu boli kontinuálne kontroléry navrhnuté ako dvojbajtové kontroléry. Ako už vieme, správa typu kontrolér má v stavovom bajte vždy šestnástkové číslo v rozsahu \$B0 až \$BF (tým určuje, že ide o kontrolér) a v prvom dátovom bajte číslo kontroléra. Na uloženie hodnoty nám teda ostal už iba druhý dátový bajt, ktorý môže mať iba hodnotu v rozsahu 0 až 127. Tvorcovia midi normy usúdili, že na nastavenie potenciometra je to málo a bolo by potrebné použiť dva kontroléry, aby spolu doručili nástroju dve časti hodnoty, ktorá sa v nástroji zlúči a vytvorí tak výslednú hodnotu. Takýmto spôsobom by muzikant mohol do kontroléru vložiť hodnotu až v rozsahu 0 – 16383, čo je teda riadny rozdiel. Kontinuálne kontroléry sú teda navrhnuté tak, že prvých 32 kontrolérov tvorí MSB a druhých 32 kontrolérov zasa LSB časť toho istého ovládača. Ak teraz neviete čo je MSB a LSB, naštudujte si to v kapitolke 8, kde sme sa tomu podrobnejšie venovali. Tu už len krátko

zopakujme, že MSB a LSB sú dve časti jedného čísla, ktoré sa v cieľovom zariadení zlúči do jednej výslednej hodnoty.

Kontinuálne kontroléry spolu tvoria dvojice 0/32, 1/33, 2/34 a tak ďalej. Pre odoslanie komplexnej hodnoty musíme teda poslať dve správy, ktoré budú vyzerat' takto: \$B0 0 100; \$B0 32 30. Prvá správa mala stavový bajt \$B0, v prvom dátovom bajte nástroju povedala, že prichádza prvá časť čísla a v druhom dátovom bajte odovzdala prvú polovicu informácie. Vzápätí prišla druhá správa, ktorá statusom a prvým dátovým bajtom oznámila, že prichádza dokončenie predošlej správy a v druhom dátovom bajte odovzdala zvyšok informácie. Nástroj si obe čísla zlúči a keďže kontroléry 0 a 32 slúžia na zvolenie banky, aktivuje sa príslušná banka. Vďaka tomuto systému môže byť v nástroji až 16 384 zvukových bánk a my vždy budeme mať v kontroléroch dost' rezerv,

aby sme aj tak vysoké číslo mohli poslať klasickou kanálovou správou. (Viac o bankách a zvukoch sa dočítate v kapitole 10.)

Kompletný zoznam kontinuálnych kontrolérov nájdete v tabuľke na konci tejto kapitoly. Nateraz tu spomeňme iba tie najznámejšie – kontroléry 0/32 slúžia na výber zvukovej banky. Kontrolér 7/39 slúži na nastavenie hlasitosti, 8/40 je vyváženie, 10/42 je stereo poloha (dolné hodnoty posúvajú zvuk doľava, horné doprava), 11/43 je hlasitosť, ktorú nastavujeme v rozpätí 0 až hodnota, nastavená kontrolérom 7/39 (teda ak 7/39 nastavíme na polovicu, 11/43 môžeme síce nastavovať v plnom rozsahu 0 až 127, no reálne hlasitosť skladby nedostaneme vyššie ako nad polovicu, pretože tak nastavil maximum hlasitosti kontrolér 7/39. Je to to isté ako keď na nástroji nastavíme koliesko master volume do určitej polohy a potom nožným ovládačom pridávame a stišujeme hlasitosť, no nikdy sa nožným ovládačom nedostaneme vyššie ako určuje kolečko master), 20/52 až 31/63 nie sú obsadené.

Záverom podotýkam, že systém kontinuálnych kontrolérov ako dvojbjtových bol pôvodný zámer tvorcov normy. Prax však ukázala, že v mnohých prípadoch to nie je potrebné a Kým zvukovú banku niekedy naozaj musíme voliť dvojicou kontrolérov, stereo polohu alebo vyváženie bez problémov zadáme iba jedným kontrolérom 10 alebo 8. Voľné ekvivalentné LSB kontroléry tak neraz výrobcovia obsadzujú inými funkciami.

2. Spínače (64-79)

Ide o riadenie tých ovládačov, ktoré poznajú de facto iba dve polohy a to on a off (zapnutý a vypnutý). Patrí sem sustain (64), portamento (65), sostenuto (66), soft pedal (67), hold (69) a iné, pričom mnohé kontroléry z tejto skupiny ani nie sú obsadené.

Kontrolér tohto typu má v druhom dátovom bajte buď hodnotu 0 pre vypnutie, alebo hodnotu 127 pre zapnutie spínača. To je ideálny stav. Mnohé nástroje však reagujú aj na iné hodnoty, avšak hodnotu do 63 považujú za vypnutie a hodnotu nad 63 za zapnutie daného spínača. Vzhľadom na staršie nástroje je však lepšie používať hraničné hodnoty 0 a 127.

3. Modulačné kontroléry (80-95)

Ide o kontroléry, ktoré v nástroji nastavujú rôzne efekty napríklad chord (80), legato (84), reverb (91), tremolo (92), chorus (93), celeste (94). Aj v tejto skupine sú niektoré kontroléry neobsadené napríklad 85 až 90.

Tieto kontroléry majú v druhom dátovom bajte hodnotu 0 až 127, ktorá určuje, aký výrazný má daný efekt byť.

4. Špeciálne kontroléry (96-119)

V tejto skupine si všimneme najmä kontroléry 98 až 101, ktoré spolu tvoria dve dvojice dvojbajtových kontrolérov podobne, ako tomu bolo v prvej skupine.

Kontroléry 98/99 nám otvárajú svet neregistrovaných parametrov – NRPN (Non registered parameter number) a kontroléry 100/101 svet registrovaných parametrov – RPN (Registered parameter number). O čo ide? Možno už aj vám zišla na um otázka, či 128 kontrolérov vôbec stačí pri tej rozmanitosti funkcií, ktoré majú dnešné nástroje. Veď ak zrátame všetky parametre, ktoré na našom nástroji môžeme meniť, pridáme neraz k číslu, ktoré veru ďaleko prekračuje rozsah, ktorý nám poskytujú kanálové správy typu kontrolér. Preto si tvorcovia midi nechali dve zadné brány presne pre tento prípad. Kontrolér pre registrované a neregistrované parametre nám umožňuje vstúpiť tajnou bránou do ďalšieho sveta. Ak totiž nástroju pošleme kanálovú správu typu registrovaný alebo neregistrovaný parameter a ako hodnotu zadáme konkrétne číslo, ktoré si prečítame v dokumentácii nášho nástroja, nástroj aktivuje vlastnosť, ktorá je priradená danému číslu a čaká na kontrolér 6, ktorý slúži na zadávanie hodnoty registrovaným a neregistrovaným parametrom. Vysvetlíme si to na príklade, ktorý si teraz čisto pre študijné účely vymyslím. Náš nástroj umožňuje pri zvolenom zvuku meniť rýchlosť jeho nástupu. Na to ale v norme nemáme žiadny kontrolér. Preto výrobca určí, že táto vlastnosť sa ovláda neregistrovaným parametrom číslo 80/10, pričom 80 je MSB a 10 LSB časť čísla parametru. Nástroju preto pošleme správu s kontrolérom 98, ktorému priradíme hodnotu 10 a hneď za ňou správu s kontrolérom 99, ktorému priradíme hodnotu 80 (pozor na to, aby sme nezamenili MSB a LSB kontroléry a čísla. Kontrolér 98 musí niesť LSB a kontrolér 99 MSB časť čísla). Správy budú teda vyzerat takto: \$B0 98 10; \$B0 99 80. Týmito dvoma správami sme nástroju povedali, že chceme zmeniť rýchlosť nástupu zvuku. Nástroj teda upriami svoju pozornosť na danú vlastnosť, ale ešte sa nič nezmení, pretože sme ešte nevložili žiadnu hodnotu, na akú sa má daná vlastnosť nastaviť. Samotné kontroléry 98 ani 99 na to proste už nemajú kapacitu. Preto teraz pošleme nástroju správu s kontrolérom 6, ktorý vždy slúži na zadávanie hodnoty, ak predtým boli aktivované registrované alebo neregistrované parametre. Otázkou teraz je, či rýchlosť nástupu zvuku budeme nastavovať jednobajtovým alebo dvojbajtovým kontrolérom. Inak povedané, ak nám stačí zadať hodnotu v rozsahu 0 až 127, stačí poslať iba kontrolér 6. Ak ale potrebujeme zadať hodnotu v rozsahu 0 až 16383, musíme poslať dvojicu kontrolérov a to 6 a 38, pretože tieto dva kontroléry predstavujú MSB a LSB časť toho istého kontroléru. Predpokladajme teda prvý prípad a pošleme do nástroja správu \$B0 6 80. Teraz sme už skutočne nastavili rýchlosť nástupu zvuku.

Vidíte, že registrované a neregistrované parametre otvárajú obrovský svet možností nastavení, pretože tým, že ide o dvojbajtové kontroléry, nám umožňujú nastavovať až 16384 vlastností. Je len na výrobcach, ktoré parametre v nástroji obsadia.

Na záver vám možno ešte vráta hlavou otázka, aký je vlastne rozdiel medzi RPN a NRPN teda registrovanými a neregistrovanými parametrami. Nuž rozdiel je ten, že kým registrované parametre sú spoločné pre viacero výrobcov – ide o výsledok vzájomných dohôd, takže sa viac výrobcov zhodne na používaní jedného parametru pre tú istú vlastnosť, čo by teoreticky mala byť záväzná norma pre všetkých výrobcov, neregistrované parametre si určuje každý výrobca sám, takže sa pri nich môžeme spoliehať jedine na dokumentáciu.

5. Špeciálne riadiace povely (120-123)

Ide o štyri kontroléry, ktoré slúžia spravidla na zabezpečenie proti nežiadúcim efektom:

120 – ASO (All sounds off) – vypnúť všetky zvuky. Tento kontrolér vypne všetky zvuky, ktoré práve neprodukujú tón.

121 – RAC (Reset all controllers) – reset kontrolérov. Tento povel vynuluje všetky kontroléry a otočné potenciometre vráti do stredovej hodnoty.

122 – local controll – miestne riadenie. Ide o zapínanie a vypínanie klaviatúry nástroja. Ak má kontrolér hodnotu 127, je miestna klaviatúra zapnutá. Ak vtedy pustíme midi skladbu zo sekvencera a budeme hrať, budú znieť všetky hlasy súčasne. Ak má tento kontrolér hodnotu 0, miestne riadenie sa vypne, teda klaviatúra nástroja bude nemá. V mnohých prípadoch však aj vtedy vysielala cez midi výstup správy o tom, čo na klaviatúre robíme. Ide však o tiché vysielanie, takže zvukovo znie iba zvuk skladby zo súboru.

123 – ANO (All notes off) vypnúť všetky noty. Tento povel vypne všetky znejúce noty. Tým zaistíme, že žiadny tón neostane visieť.

Okrem kontroléru 122 majú všetky ostatné v druhom dátovom bajte nulu.

6. Režimové povely (124-127)

Napokon sú tu posledné štyri povely, ktoré však dnes už asi málo použijete. Vypínajú a zapínajú omni režim a mono režim v nástroji. Omni režim je stav, kedy nástroj prijíma midi správy na všetkých kanáloch. Ak tento režim vypneme, bude prijímať iba na jednom kanáli, ktorý musíme zadať. Omni režim zapínajú a vypínajú dva kontroléry a to 124 a 125, tie sa však vzájomne vylučujú, takže ak prvým omni režim vypneme, poslaním druhého ho zasa zapneme.

Rovnako je to aj v prípade mono režimu. Ten znamená, že nástroj bude ku každému zvuku priradovať iba jeden tón. Nie je to veľmi bežné, ale môže sa to niekedy hodiť. Mono režim aktivujeme kontrolérom 126. Kontrolérom 127 ho zasa vypneme a nástroj sa prepne do režimu poly.

Tabuľka kontrolérov

Na záver vám ponúkam prehľad najbežnejších kontrolérov. Pripomínam, že mnohé môžu mať vo vašom nástroji iný význam, ako uvádza tabuľka. Rozdiel je závislý od použitej normy (Pozri kapitolu 14.), ale aj od výrobcu a modelu nástroja. Pre plnohodnotné využívanie kontrolérov preto nahliadnite do dokumentácie k vášmu nástroju.

Číslo	Názov	Význam
0/32	Bank Select MSB/LSB	Voľba banky
1/33	Modulation MSB/LSB	Modulácia
2/34	Breath Controller MSB/LSB	Dych
4/36	Foot Controller MSB/LSB	Nožný spínač
5/37	Portamento Time MSB/LSB	Portamento
6/38	Data Entry MSB/LSB	Vloženie dát
7/39	Channel Volume MSB/LSB	Hlasitosť kanálu
8/40	Balance MSB/LSB	Vyváženie
10/42	Pan MSB/LSB	Stereo poloha
11/43	Expression Controller MSB/LSB	Čiastková hlasitosť
12/44	Effect Control 1 MSB/LSB	Efekt 1
13/45	Effect Control 2 MSB/LSB	Efekt 2
16/48	General Purpose Controller 1 MSB/LSB	Všeobecné účely 1
17/49	General Purpose Controller 2 MSB/LSB	Všeobecné účely 2
18/50	General Purpose Controller 3 MSB/LSB	Všeobecné účely 3
19/51	General Purpose Controller 4 MSB/LSB	Všeobecné účely 4
64	Damper Pedal on/off (Sustain)	Sustain
65	Portamento On/Off	Portamento
66	Sostenuto On/Off	Sostenuto
67	Soft Pedal On/Off	Zmäkčenie
68	Legato Footswitch	Legato
69	Hold 2	Hold
70	Sound Controller 1 (default: Sound Variation)	Variácie zvuku
71	Sound Controller 2 (default: Timbre/Harmonic Intens.)	Harmonická zložka
72	Sound Controller 3 (default: Release Time)	Čas doznenia
73	Sound Controller 4 (default: Attack Time)	Čas nástupu
74	Sound Controller 5 (default: Brightness)	Jas zvuku
75	Sound Controller 6 (default: Decay Time - see MMA RP-021)	Čas doznenia
76	Sound Controller 7 (default: Vibrato Rate - see MMA RP-021)	Frekvencia vibráta
77	Sound Controller 8 (default: Vibrato Depth - see MMA RP-021)	Hĺbka vibráta
78	Sound Controller 9 (default: Vibrato Delay - see MMA RP-021)	Odstup vibráta
79	Sound Controller 10 (default undefined - see MMA RP-021)	Nedefinované
80	General Purpose Controller 5	Všeobecné účely 5
81	General Purpose Controller 6	Všeobecné účely 6
82	General Purpose Controller 7	Všeobecné účely 7

83	General Purpose Controller 8	Všeobecné účely 8
84	Portamento Control	Portamento
88	High Resolution Velocity Prefix	Prefix vysokého rozlíšenia citlivosti
91	Effects 1 Depth	Efekt 1 – hĺbka
92	Effects 2 Depth (formerly Tremolo Depth)	Efekt 2
93	Effects 3 Depth	Efekt 3
94	Effects 4 Depth (formerly Celeste [Detune] Depth)	Efekt 4
95	Effects 5 Depth (formerly Phaser Depth)	Efekt 5
96	Data Increment (Data Entry +1) (see MMA RP-018)	Zvýšenie hodnoty o 1
97	Data Decrement (Data Entry -1) (see MMA RP-018)	Zníženie hodnoty o 1
98	Non-Registered Parameter Number (NRPN) - LSB	Neregistrované číslo parametra LSB
99	Non-Registered Parameter Number (NRPN) – MSB	Neregistrované číslo parametra MSB
100	Registered Parameter Number (RPN) - LSB*	Registrované číslo parametra LSB
101	Registered Parameter Number (RPN) - MSB*	Registrované číslo parametra MSB
120	All Sound Off	Všetky zvuky vypnúť
121	Reset All Controllers	Reset všetkých ovládačov
122	Local Control	Lokálne riadenie
123	All Notes Off	Všetky noty vypnúť
124	Omni Mode Off	Omni režim vypnutý
125	Omni Mode On	Omni režim zapnutý
126	Mono Mode On	Mono režim zapnutý
127	Poly Mode On	Poly režim zapnutý

10. Voľba zvukov

Zmena zvuku je asi jedna z najčastejších operácií, ktoré na digitálnom klávesovom nástroji vykonávame. Ak komponujeme skladbu, alebo iba interpretujeme známy hit, je voľba správneho zvuku veľmi dôležitá. Povedzme si preto, ako funguje voľba zvukov v midi.

Norma GM

Kedysi na začiatku mali nástroje iba málo zvukov – tie dokonalejšie až 127, daktore ešte menej. Boli technicky označené číslami, ale ich usporiadanie nebolo nijako štandardizované, takže kým jeden nástroj mal pod číslom 10 organ, iný tam mohol mať trúbku. Do midi súborov sa nikdy nezaznamenáva názov zvuku, iba jeho číslo, preto v začiatkoch midi vládol v systéme voľbe zvukov dokonalý chaos. No len si to predstavte. Sedíte doma a tvoríte midi skladbu – aranžujete známu ľudovku. Použijete bicie, basu, harmoniku, nejakú trúbku a sem-tam sa ozve klarinet. Do midi skladby sa zvolené zvuky zapíšu ako čísla v rozsahu 0 až 127, pričom vaša basa je zapísaná ako číslo 8, harmonika ako číslo 10, trúbka ako 35. (Čísla som si samozrejme vymyslel a len teoreticky predpokladajme, že by použité zvuky vo vašom nástroji mali priradené práve takéto čísla). Skladbu dokončíte, uložíte na disketu a bežíte sa pochváliť kolegovi – hudobníkovi. Vložíte disketu, načítate skladbu a prehráte. No namiesto ľubozvučnej polky sa ozve soundtrack z hororu, pretože kolegov nástroj začne búchať, písať, cinkať a nič z toho nenaznačuje počúvateľnú muziku. Čo sa stalo? Iba to, že nástroj poslúchol príkazy, ktoré dostal a zapol zvuky s číslom 8, 10 a 35. Problém je v tom, že kým u vás sa pod číslom 8 skrýva basa, u kolegu je tam zvonkohra. Kým pod číslom 35 je u vás trúbka, u kolegu je to organ. Skrátka viete si určite predstaviť, že bez jednotného číslovania zvukov sa seriózna muzika vytvoriť nedá a ak sa aj vytvorí, je použiteľná iba na tom nástroji, na ktorom bola vytvorená.

Preto norma General midi priniesla do tejto problematiky systém. Bol vytvorený zoznam 128 zvukov, ktoré boli rozdelené na 16 podskupín, pričom v každej podskupine bolo 8 zvukov. Tak sa do usporiadania zvukov v nástrojoch vniesla aká-taká jednota. Ak potom autor na jednom nástroji použil organ, mal tento číslo v rozsahu 17 až 24 – teda čísla 17 až 24 sú vyhradené organom a ak má nástroj organov viac (maximálne 8), čísľuje ich práve týmito číslami. Teda dajme tomu, že autor použil rock organ a do súboru sa zapísalo číslo 20. Ak potom skladbu prehral na inom nástroji, bola veľká pravdepodobnosť, že sa ozve organ, pretože norma určovala, že čísla od 17 po 24 musia byť organy. Možno to nebol presne ten rock, no už to nebolo také diametrálne vzdialenie sa, ako keď predtým namiesto trúbky hučali zvony. V závere kapitoly vám ponúkam tabuľku zvukových rodín aj tabuľku GM zvukov.

Situácia sa teda čiastočne vyriešila, zvuky a ich číslovania sa štandardizovali a na zvolenie príslušného zvuku sa použila jednoduchá správa \$CK, ktorá obsahovala jediný dátový bajt a to číslo požadovaného zvuku. (Bližšie pozri kapitolu 8.) Tu ešte upozorním na jednu chybu, ktorá sa často v zoznamoch objavuje. Norma GM určuje poradie 128 zvukov. Toto číslo nie je náhodné. Je dané tým, že do dátového midi bajtu vieme zapísať práve 128 čísel, pozor však na to, že prvým číslom v bajte nie je číslo 1 ale 0. Do midi bajtu teda nepíšeme čísla 1 až 128, ale 0 až 127. Takýto je skutočný technický stav. Mnohé tlačené zoznamy však tento fakt nerešpektujú a zvuky čísľujú od 1 po 128. Ak potom zo zoznamu vyberiete zvuk a jeho číslo zadáte do dátového bajtu správy \$CK, dôjde k posunu, pretože vaše číslo (napríklad 10) nástroj síce pochopí ako číslo 10, no zapne jedenásty zvuk, pretože ak rátate od nuly, je desiaty zvuk naozaj v poradí už jedenásty. Bolo by teda korektné, keby aj

rôzne tlačené zoznamy počítali zvuky od 0, čím by sa zaistilo, že autor hudby zadá do midi súboru číslo toho zvuku, ktorý naozaj potrebuje.

Tu by sme mohli našu kapitolu ukončiť. Máme stabilný zoznam zvukov a vieme ich aj vyvolať správou \$CK. Skončiť však nemôžeme, pretože situáciu nám skomplikoval technický pokrok, ktorý spôsobil, že dnešné nástroje nemajú iba 128 zvukov, ale ich majú oveľa viac. Keďže do dátového midi bajtu správy \$CK môžeme zapísať iba číslo v rozsahu 0 až 127, bola vylúčená možnosť, že by zoznam proste pokračoval až do nebeských čísel. Tvorcovia preto vymysleli systém, ktorý je používaný dodnes a to sú zvukové banky.

Zvukové banky

Zvuková banka je sada 128 zvukov, ktorá ich zoradením spĺňa normu GM. Teda ak starý nástroj mal 127 zvukov v jednom zozname, číslovaných od 0 po 127, novší nástroj už mohol mať takéto zoznamy dva alebo tri, pričom v každom bolo číslovanie zvukov rovnaké, teda od 0 po 127, každý zoznam mal na začiatku klavíri a na konci efekty, ale celkový počet zvukov sa už zvýšil zo 128 na 384. Zaiste ste si už domysleli, že konkrétny zvuk sa v takomto prípade nevyvolá správou \$CK a zadaním čísla 300, lebo to technicky nie je možné, ale vyvolá sa tak, že sa najprv jednou správou určí, z ktorej banky chceme zvuk čerpať a následne sa druhou správou už typu \$CK určí, ktorý zvuk to má byť. Teda ak máme v nástroji napríklad 3 sady, hovorme im správne 3 banky, jednou správou povieme: použi banku 2 a druhou správou povieme: zapni zvuk 10.

Nech vás nepomýli, že takéto rozdelenie zvukov nevidíte na ovládacích paneloch nástrojov. V tejto kapitole je reč o vnútornom členení zvukov. Výrobcovia však logicky zobrazujú na paneloch nástrojov zvuky tak, aby to bolo pre hráča čo najpohodlnejšie. Na vašom nástroji teda máte napríklad tlačidlo Piano, ktoré vám zobrazí 18 rôznych klavírov, ktoré v nástroji máte. Javí sa vám to ako jedna ucelená skupina zvukov, ale v skutočnosti vám nástroj iba vyselektoval klavírne zvuky z bánk, ktoré má. Keďže norma stále určuje, že klavíri môžu mať v zozname zvukov čísla od 0 po 7, je pravdepodobne vašich 18 klavírov rozložených vo viacerých bankách. Kým prvý má číslo 0 v prvej banke, druhý môže mať tiež číslo 0, ale v inej banke a tak ďalej. Pre upresnenie ešte dodajme, že všetky banky vo vašom nástroji nemusia byť úplne plné. Môžete mať v nástroji 18 klavírov, ale 83 gitár. Nie je pravidlom, že výrobcovia všetky banky doplna využijú, preto sa ani nedá povedať, že počet zvukov v nástroji sa rovná počtu bánk krát 128. Niektoré banky môžu byť plné, iné môžu byť obsadené iba v rodinách pre gitary, čím vznikne výsledný efekt veľkého počtu gitár a malého počtu klavírov.

Ako v midi zvoliť zvuk

Dnes teda volíme zvuk nástroja tak, že najprv určíme banku, z ktorej chceme zvuk načítať a až potom zvolíme konkrétny zvuk. Výrobcovia však už rátajú s tým, že časom môže pribudnúť veľké množstvo bánk, preto banku volíme dvojbajtovým kontrolérom. Ak teraz neviete o čom hovorím, vráťte sa ku kapitole 8 a 9 a naštudujte si problematiku dvojbajtových kontrolérov.

Na výber banky sa používa dvojbajtový kontrolér 0/32, pričom 0 je MSB select bank a 32 je LSB select bank. Ako už viete, hodnoty MSB a LSB patria k sebe a v nástroji zlúčením

vytvoria jednu konečnú hodnotu, ktorá môže byť v rozsahu 0 až 16383. Ak si predstavíte nástroj, ktorý má 16384 bánk a v každej má 128 zvukov, vynorí sa vám pred očami supermoderný nástroj, ktorého sa asi nedožijeme. Systém voľby banky pomocou MSB a LSB kontroléru má teda dostatočné rezervy na stovky rokov dopredu.

V praxi to vyzerá asi takto:

\$BK 0 MSB – na kanáli K sa zvolí banka, ktorej prvá časť hodnoty je zapísaná v MSB

\$BK 32 LSB – na kanáli K sa zvolí banka, ktorej druhá časť hodnoty je zapísaná v LSB

\$CK 10 – na kanáli K sa zvolí zvuk s číslom 10

Možno sa vám teraz vynorí v mysli otázka, odkiaľ naberiete čísla zvukov vo vašom nástroji? Zoznam všetkých zvukov spolu s ich číslami je súčasťou dokumentácie každého nástroja. Skúste zalistovať v manuáli k vášmu nástroju. Môže sa však stať, že tabuľku zvukov nenájdete. Vtedy použijete google a hľadajte na stránkach výrobcu vášho nástroja. Niektorí výrobcovia totiž dodávajú k nástrojom v tlačenej podobe iba návod na obsluhu. Brožúrka s technickými údajmi (ako je aj zoznam zvukov) často nie je súčasťou balenia, ale je dostupná na stránkach výrobcu.

Tabuľka zvukov vo vašom nástroji by mala obsahovať názvy zvukov tak, ako sú zobrazené na obrazovke nástroja, napríklad Classic piano, Concert guitar a podobne, potom by mala obsahovať číslo banky a číslo zvuku. Ak nástroj obsahuje malý počet bánk a na ich voľbu stačí použiť iba jeden kontrolér, nájdete v tabuľke iba údaj označený ako bank. Vtedy stačí do midi skladby vložiť iba jeden kontrolér a to 0. Ak má nástroj veľa bánk, bude v tabuľke číslo banky uvedené dvoma číslami a to MSB a LSB. V takom prípade musíte vložiť dva kontroléry, pričom kontroléru 0 priradíte hodnotu MSB a kontroléru 32 hodnotu LSB. Napokon vložíte správu \$CK, ktorej priradíte číslo zvuku, v tabuľke uvedené pod skratkou PRG ako program.

Záverom ešte podotknem, nakoľko tento seriál vychádza z používania midi v programe Notedit, že program Notedit dokáže pracovať s takzvanými instrument listami. Sú to textové súbory, ktoré obsahujú zoznam zvukov daného nástroja a ich čísla. Praktické využitie je potom také, že ak v programe Notedit chcete zmeniť zvuk, zobrazia sa vám klasické ľudské názvy zvukov vášho nástroja, vy si vyberiete ten, ktorý chcete a potvrdíte. Program sám vytvorí potrebné kontroléry a vloží hodnoty, ktoré prečítal z instrument listu. Ak takýto instrument list nemáte, môžete požiadať o jeho vytvorenie autora Noteditu.

Tabuľka zvukových rodín podľa normy GM

Poradové číslo	Rozsah (ľudovo)	Rozsah (technicky)	Rodina
1	1-8	0-7	Klavíry
2	9-16	8-15	Chromatické bicie
3	17-24	16-23	Organy
4	25-32	24-31	Gitary
5	33-40	32-39	Basy
6	41-48	40-47	Sláčikové nástroje
7	49-56	48-55	Súborové zvuky
8	57-64	56-63	Plechové nástroje
9	65-72	64-71	Jazýčkové nástroje
10	73-80	72-79	Píšťalové nástroje
11	81-88	80-87	Sólové syntetické
12	89-96	88-95	Sprievodné syntetické
13	97-104	96-103	Syntetické efekty
14	105-112	104-111	Etnické nástroje
15	113-120	112-119	Perkusné nástroje
16	121-128	120-127	Zvukové efekty

Tabuľka zvukov v banke podľa normy GM

Ľudové číslo	Technické číslo	Názov	Rodina
1	0	AcousticGrandPiano	Klavíry
2	1	BrightAcousticPiano	Klavíry
3	2	ElectricGrandPiano'	Klavíry
4	3	'HonkyTonkPiano	Klavíry
5	4	ElectricPiano1	Klavíry
6	5	ElectricPiano2	Klavíry
7	6	Harpsichord	Klavíry
8	7	Clavinet'	Klavíry
9	8	'Celesta	Chromatické
10	9	Glockenspiel	Chromatické
11	10	MusicBox	Chromatické
12	11	Vibraphone	Chromatické
13	12	Marimba	Chromatické
14	13	Xylophone'	Chromatické
15	14	'TubularBells	Chromatické
16	15	Dulcimer'	Chromatické
17	16	'DrawbarOrgan	Organy
18	17	PercussiveOrgan	Organy
19	18	RockOrgan	Organy
20	19	ChurchOrgan'	Organy
21	20	'ReedOrgan	Organy
22	21	Accordion	Organy
23	22	Harmonica	Organy

24	23	TangoAccordion',	Organy
25	24	'AcousticNylonGuitar	Gitary
26	25	AcousticSteelGuitar	Gitary
27	26	JazzElectricGuitar',	Gitary
28	27	'CleanElectricGuitar	Gitary
29	28	MutedElectricGuitar	Gitary
30	29	OverdrivenGuitar',	Gitary
31	30	'DistortionGuitar	Gitary
32	31	GuitarHarmonics	Gitary
33	32	AcousticBass',	Basy
34	33	'FingeredElectricBass	Basy
35	34	PickedElectricBass	Basy
36	35	FretlessBass',	Basy
37	36	'SlapBass1	Basy
38	37	SlapBass2	Basy
39	38	SynthBass1	Basy
40	39	SynthBass2',	Basy
41	40	'Violin	Sláčikové
42	41	Viola	Sláčikové
43	42	Cello	Sláčikové
44	43	Contrabass',	Sláčikové
45	44	'TremoloStrings	Sláčikové
46	45	PizzicatoStrings	Sláčikové
47	46	OrchestralHarp	Sláčikové
48	47	Timpani',	Sláčikové
49	48	'StringEnsemble1	Súborové
50	49	StringEnsemble2	Súborové
51	50	SynthStrings1',	Súborové
52	51	'SynthStrings2	Súborové
53	52	ChoirAahs	Súborové
54	53	VoiceOohs	Súborové
55	54	SynthVoice	Súborové
56	55	OrchestraHit',	Súborové
57	56	'Trumpet	Plechové
58	57	Trombone	Plechové
59	58	Tuba	Plechové
60	59	MutedTrumpet	Plechové
61	60	FrenchHorn',	Plechové
62	61	'BrassSection	Plechové
63	62	SynthBrass1	Plechové
64	63	SynthBrass2',	Plechové
65	64	'SopranoSax	Jazýčkové
66	65	AltoSax	Jazýčkové
67	66	TenorSax	Jazýčkové
68	67	BaritoneSax',	Jazýčkové
69	68	'Oboe	Jazýčkové
70	69	EnglishHorn	Jazýčkové
71	70	Bassoon	Jazýčkové
72	71	Clarinet',	Jazýčkové
73	72	'Piccolo	Píšťalové
74	73	Flute	Píšťalové

75	74	Recorder	Píšťalové
76	75	PanFlute	Píšťalové
77	76	BlownBottle',	Píšťalové
78	77	'Shakuhachi	Píšťalové
79	78	Whistle	Píšťalové
80	79	Ocarina',	Píšťalové
81	80	'SquareLead	Sólové syntetické
82	81	SawtoothLead	Sólové syntetické
83	82	CalliopeLead	Sólové syntetické
84	83	ChiffLead',	Sólové syntetické
85	84	'CharangLead	Sólové syntetické
86	85	VoiceLead	Sólové syntetické
87	86	FifthsLead	Sólové syntetické
88	87	BassandLead',	Sólové syntetické
89	88	'NewAgePad	Sprievodné syntetické
90	89	WarmPad	Sprievodné syntetické
91	90	PolySynthPad	Sprievodné syntetické
92	91	ChoirPad',	Sprievodné syntetické
93	92	'BowedPad	Sprievodné syntetické
94	93	MetallicPad	Sprievodné syntetické
95	94	HaloPad	Sprievodné syntetické
96	95	SweepPad',	Sprievodné syntetické
97	96	'SynthFXRain	Syntetické efekty
98	97	SynthFXSoundtrack	Syntetické efekty
99	98	SynthFXCrystal	Syntetické efekty
100	99	SynthFXAtmosphere',	Syntetické efekty
101	100	'SynthFXBrightness	Syntetické efekty
102	101	SynthFXGoblins	Syntetické efekty
103	102	SynthFXEchoes	Syntetické efekty
104	103	SynthFXSciFi',	Syntetické efekty
105	104	'Sitar	Etnické
106	105	Banjo	Etnické
107	106	Shamisen	Etnické
108	107	Koto	Etnické
109	108	Kalimba',	Etnické
110	109	'Bagpipe	Etnické
111	110	Fiddle	Etnické
112	111	Shanai',	Etnické
113	112	'TinkleBell	Perkusné
114	113	Agogo	Perkusné
115	114	SteelDrums	Perkusné
116	115	Woodblock',	Perkusné
117	116	'TaikoDrum	Perkusné
118	117	MelodicTom	Perkusné
119	118	SynthDrum	Perkusné
120	119	ReverseCymbal',	Perkusné
121	120	'GuitarFretNoise	Zvukové efekty
122	121	BreathNoise	Zvukové efekty
123	122	Seashore	Zvukové efekty
124	123	BirdTweet',	Zvukové efekty
125	124	'TelephoneRing	Zvukové efekty

126	125	Helicopter	Zvukové efekty
127	126	Applause	Zvukové efekty
128	127	Gunshot',	Zvukové efekty

11. Systémové správy

Čo sú systémové správy

Systémové správy v praxi veľmi nevyužívate, Dokonca ani nie sú určené na použitie v midi súbore, ktorý zaznamenáva naše umelecké dielo. Systémové správy sa využívajú skôr vo väčších midi systémoch, keď je napríklad prepojených viac midi zariadení, ktoré spolu musia komunikovať, ladiť a musia byť časovo zosynchronizované. Túto kapitolku teda vnímajte len ako rozšírenie vášho všeobecného rozhl'adu vo svete midi.

Systémové správy sú tie, ktoré majú stavový bajt v rozsahu \$F0 až \$FF. Nie sú však využité všetky. Stavové bajty \$F4, \$F5, \$F9 a \$FD nie sú definované a nemali by byť za žiadnych okolností využité. Ostatné systémové správy môžeme rozdeliť do troch skupín:

1. Zvláštne systémové správy – SysEx

Zvláštne systémové správy sa nazývajú aj exkluzívnymi systémovými správami (z anglického System exclusive messages) a označujú sa skratkou SysEx. Viac si o nich pohovoríme v kapitole 13. Nateraz si povedzme iba to, že SysEx správy začínajú stavovým bajtom \$F0 a končia statusom \$F7.

2. Systémové správy reálneho času

Ide o správy, ktoré zabezpečujú časovú synchronizáciu midi zariadení. Presnou syntaxou sa nebudeme zaoberať, pretože v domácej muzikantskej praxi ich nepotrebujeme. Pre zaujímavosť si ich však vymenujme a len krátko predstavme:

\$F1 – Reálny čas - na prvom dátovom bajte nesie informáciu o reálnom čase. Informácia o čase je zakódovaná do špeciálneho takzvaného MTC formátu a je vysielaná riadiacim master zariadením podriadenému zariadeniu.

\$F2 – Pozícia v skladbe - dva dátové bajty MSB a LSB spolu tvoria hodnotu, ktorá určuje pozíciu v skladbe. Táto informácia je posielaná vtedy, ak sa napríklad majú dve zariadenia nastaviť na to isté miesto v pamäti.

\$F8 – Úder hodín – tento status nemá žiadne dátové bajty. Je vysielaný pravidelne v odstupoch 24-krát za štvrt'ovú dobu a tým vlastne zaisťuje časové riadenie. Ak napríklad spustíme automatický doprovod v našom klávesovom nástroji, je tento riadený midi hodinami, ktoré musia pravidelne tikať. Ak hodiny v našom nástroji vypneme a odtikávaním času poveríme iné zariadenie, napríklad Notedit, bude tento pravidelne vysielat' stavový bajt \$F8.

\$FA – Štart – stavový bajt bez dátových bajtov spustí hodiny a teda aj prehrávanie skladby z pamäte

\$FB – Pokračuj - pokračuje v prehrávaní

\$FC – Stop - zastaví prehrávanie

3. Spoločné systémové správy

Sú rovnakého typu ako predošlá skupina, no netýkajú sa času:

\$F3 – Voľba skladby – táto správa umožňuje nastaviť dve zariadenia na to isté miesto v pamäti. Ak máme v dvoch zariadeniach rovnaký zoznam skladieb a chceme v oboch prehrávať tú istú skladbu, touto správou dosiahneme, že sa na zariadení zvolí požadovaná skladba. Skladba je označená číslom od 0 do 127 a toto číslo je uložené v jedinom dátovom bajte správy.

\$F6 – Žiadosť o naladenie – používala sa hlavne v analógových nástrojoch, ktoré sa pod vplyvom počasia a vlhkosti dokázali mierne rozladiť. Táto správa prinútila mikroprocesor, aby všetky oscilátory opäť naladil.

\$FE – Aktívne vnímanie – ide o bezpečnostný povel. Ak zariadenie obdrží tento povel, očakáva ho znovu každých 300 milisekúnd. Ak ho nedostane, vypne všetky hlasy a nastaví sa do normálneho východzieho stavu. V minulosti sa týmto spôsobom zabráňovalo tomu, aby napríklad po nečakanom odpojení kábla alebo zastavení sekvencera ostali visieť tóny, alebo nástroj zostal v konfigurácii, ktorá nebola korektne zrušená. Dnes sa tento povel už málo využíva, pretože zariadenia dnes majú schopnosť v podobnej situácii automaticky všetky kontroléry vynulovať a noty vypnúť.

\$FF – Reset systému – ide o softvérový reštart, teda niečo, čo dosiahneme na nástrojoch spravidla nie štandardným tlačidlom, ale kombináciou nejakých kláves. Ide o povel, ktorý sa nepoužíva často a aj cez midi by mal byť vysielaný len v najnutnejších prípadoch.

Tabuľka systémových správ

Na záver si v prehľadnej tabuľke ešte raz zhrňme všetky systémové správy, pričom do tejto tabuľky, aby bola kompletná, zahrniem aj SysEx správy, hoci o nich ešte budeme hovoriť v kapitole 13:

Status	Hodnota 1	Hodnota 2	Význam
\$F0			Začiatok SysEx správy
\$F1	MTC informácia o čase 0-119		Reálny čas
\$F2	LSB 0-127	MSB 0-127	Pozícia v skladbe
\$F3	Číslo skladby 0-127		Číslo skladby
\$F6			Žiadosť o naladenie
\$F7			EOX – end of SysEx – Koniec SysEx správy
\$F8			Úder hodín
\$FA			Štart
\$FB			Pokračuj
\$FC			Stop
\$FE			Aktívne vnímanie
\$FF			Reset zariadenia

12. Meta správy

Čo sú meta správy

Ak si spomínate na kapitolu o štruktúre správy, hovorili sme tam o šiestich častiach správy: čas delta (tento teraz nechajme stranou, pretože síce je uložený v midi súbore, ale reálne sa nikdy neposiela), status, dva dátové bajty, meta bajt a políčko sysex. Všetky doteraz prebrané typy správ využívali iba prvé tri polia – status a dva dátové bajty. Zvyšok sme nepotrebovali. Teraz sa situácia obráti. Hoci status budeme naďalej používať, ten musí vždy určovať o akú správu ide, dátové bajty si už vôbec nebudeme všímať, pretože budú vždy prázdne. Zaujímať nás bude políčko meta, ktoré bude svojou hexadecimálnou číselnou hodnotou určovať, o aký druh meta správy ide a políčko sysex, ktoré bude už obsahovať samotný text.

Čo sú teda vlastne meta správy? Sú to nekanálové správy, ktoré nástroju prinášajú rôzne informácie, číselné a dokonca aj textové. Pomocou meta správ je na začiatku každej skladby určené tempo, takt a tónina, pomocou meta správ môžeme na obrazovke zobrazovať text, pomocou meta správ si do skladby môžeme vložiť rôzne textové poznámky. Meta správy nie sú zvukového typu, teda sami neprodukurujú zvuk, ale prinášajú iba spomenuté sprievodné informácie.

Meta správa má vždy stavový bajt s hodnotou \$FF. Nie, nemýlite sa. Ak by za týmto stavovým bajtom už neprišlo nič iné a všetky časti správy by boli prázdne, jednalo by sa o reset nástroja (pozri kapitolu 11). Na rozdiel od resetu však v tomto prípade bude mať typový meta bajt vyššiu hodnotu ako 0. Z toho teda vyplýva, že v prípade meta správy nikdy nie je v meta bajte použitá hodnota 0. Hodnota meta bajtu nástroju povie, aký typ správy prichádza: teda či ide o text piesne, určenie tempa alebo taktu. Napokon je tu časť sysex, ktorá už obsahuje samotnú informáciu, teda onen text alebo tempo.

Preberme si teda konkrétne meta správy. V jednotlivých odsekoch budem uvádzať iba číselnú hodnotu meta bajtu, pretože status je vždy \$FF a dátové bajty sú nuly. Uvedené číslo bude vždy šesťnástkové a bude uložené v poli meta bajtu. Obsah informácie bude v poli sysex.

Meta správy by sme mohli z hľadiska editovateľnosti rozdeliť na tri skupiny.

1. Textové meta správy

Prvá skupina sú textové správy. Je im vyhradený rozsah meta bajtov \$01 až \$0F, pričom nie všetky sú využité. Pole sysex obsahuje klasický text, ktorý môžeme bez problémov prepisovať.

\$01 – Text – pomocou tejto správy môžeme do skladby vložiť nejakú textovú poznámku, ktorá sa zobrazí na obrazovke nástroja. Môže to byť pokojne ťahák, alebo nejaká technická poznámka, na ktorú nechceme zabudnúť.

\$02 – Copyright – do správy tohto typu môžeme vložiť informáciu o autorovi skladby.

\$03 – Track name – do takejto správy sa uloží názov tracku, ak sme ho napríklad v programe Notedit určili. Je dobré dávať trackom názvy, pretože ak potom chceme editovať

midi súbor cez nejaký event editor, môžeme hneď z názvu vidieť, o aký track ide. Názvy trackov spravidla vyjadrujú charakter nástroja. Napríklad ak si v Event editore Noteditu otvoríte midi súbor, ktorý má 8 trackov a tie sú seriózne pomenované, uvidíte zoznam s položkami ako napríklad 1 – piano, 2 – strings, 3 – bass ... a hneď viete, ktorý track máte otvoriť, aby ste opravili chybu v basovej linke.

\$04 – Instrument name – do tejto správy sa ukladá názov nástroja. Niektoré Event editory – ako napríklad Notedit, sa v prípade, že track neobsahuje žiadny názov, snažia nájsť práve túto správu, aby aspoň tak užívateľovi ukázali, čo v danom tracku môže očakávať.

\$05 – Lyric – ide o text piesne. Ak do skladby vložíme správy tohto typu, pričom do políčka sysex vpíšeme postupne slabiky textu, budú tieto zobrazované priebežne na obrazovke, ak to samozrejme nástroj podporuje. Text piesne sa často ukladá do samostatného tracku, takže všetky správy tohto typu idú za sebou a dôležité je len ich správne načasovanie.

\$06 – Marker – ide o vnútornú textovú značku. Bežne túto správu asi nepoužijete, vkladajú si ju rôzne sekvenceri na miesta, kde potrebujú oddeliť jednu časť skladby od inej. Používa sa napríklad pri patternovom prehrávaní, kde značka marker oddeľuje jeden pattern od druhého. (Pre bližšie vysvetlenie patternového prehrávania pozri kapitolu 5 o typoch midi súborov).

\$07 – Cue point – správa obdobného typu ako predošlá – oddeľuje jednu časť od druhej pomocou značky cue.

2. Číselné správy

Druhú skupinu tvoria číselné správy – to sú správy typu tempo, takt, tónina. Všetky tieto hodnoty sú v midi vyjadrené číslami, pričom spôsob prevodu klasickej hodnoty na číslo je vždy iný. Ak v notovom programe klasickým výberom z daných možností nastavíme tempo, takt a tóninu, postará sa program o to, aby tieto nastavenia boli zapísané do midi súboru práve prostredníctvom meta správ. Ak si potom midi súbor otvoríme v Event editore, objavíme tieto správy s číselnými hodnotami, ktoré budú zapísané v poli sysex.

\$51 – Tempo – bežne sa na nástrojoch či v hudobných programoch zadáva v BPM (beats per minute – počet štvrt'ových nôt za minútu). To je však spôsob, určený pre ľudí. Do midi súboru sa tempo zapisuje v mikrosekundách za štvrt'ovú dobu. Štandardné tempo 120 BPM tak zapíšeme ako 500000, tempo 60 BPM ako 1000000. Viac o tom pozri v kapitole 4.

\$58 – Time signature - Takt – dôležitý údaj, ktorý by sa mal nachádzať na začiatku každého midi súboru. Určuje takt skladby. Hodnota je vyjadrená štyrmi číslami, ktoré notové programy preratávajú na takt. Týmto algoritmom sa nebudeme podrobne zaoberať, pretože je dosť zložitý. Radšej vám v závere kapitoly ponúknem prehľad najčastejšie používaných taktov s ich midi číselnými vyjadreniami.

\$59 – Key signature - Tónina – správa, ktorá hovorí, v ktorej tónine skladba je. Nie je to povinný údaj, pretože ako už dobre vieme, v midi majú všetky noty svoje fixné číselné označenia. Ak do nástroja príde midi správa note on s hodnotou 61, tak zaznie Cis bez ohľadu na stupnicu. Táto správa sa však hodí vtedy, ak midi skladbu chceme otvoriť v programe,

ktorý zobrazuje noty. Aby program správne určil, v akej tónine skladba je, je vhodné, ak na začiatku súboru nájde správnu key signature. Teda inak povedané, táto správa nie je potrebná pre hudobné prehrávanie, ale pre zobrazenie v notovom zápise. Tónina je pritom vyjadrená dvoma číslami, pričom prvé označuje počet krížikov a béčok, druhé či je tónina Dur alebo mol. Prvé číslo môže byť v rozsahu -7 až 7. Záporné čísla označujú počet béčok, kladné počet krížikov. 0 označuje tóninu C, ktorá nemá žiadne krížiky ani béčka. Pokiaľ ide o druhé číslo, ak je toto 0, ide o durovú stupnicu, ak 1, ide o molovú stupnicu. Napríklad 0,0 – C Dur (0 krížikov a béčok, Dur); -1,1 – d mol (jedno béčko, mol); 4,0 – E Dur (4 krížiky, Dur).

3. Ostatné meta správy

Tretou skupinou sú ostatné meta správy, ktoré sú do súboru vložené sekvencerom. Neodporúčam ich nikdy mazať ani meniť, pretože ich zmena by mohla výrazne ovplyvniť zvuk skladby (napríklad pri dátach sekvenceru), alebo dokonca čitateľnosť súboru (napríklad ak by sme vymazali správu EOX – end of track). Tieto správy nebudete nikdy vkladať cieľavedome cez hudobný program, no zrejme ich nájdete v súboroch, ktoré boli vytvorené napríklad priamo v hudobných nástrojoch. Ich odstránenie by mohlo spôsobiť, že by pri prehrávaní nástroj neaktivoval nejaké funkcie či efekty.

\$2F – End of track – správa, ktorá musí byť ako posledná v každom tracku a označuje jeho koniec. Ide o samotný stavový bajt, teda správa nemá žiadnu hodnotu v poli sysex.

\$54 - SMPTE format specification – zvláštny časový údaj, ktorý niektoré sekvencery vkladajú napríklad na časové určenie začiatku tracku.

\$7F - Data sequencer – správy podobné SysEx správam – ide o špeciálne správy, ktoré si do súborov ukladajú sekvencery.

Vymenovali sme si najčastejšie meta správy. Sami ste videli, že zďaleka nie je obsadená celá škála možností teda od \$00 po \$7F. Je však na výrobcovi, či široké spektrum možností meta správ využije na aktivovanie nejakých funkcií v nástroji. Informácie o tom vám poskytne dokumentácia k vášmu nástroju.

Tabuľka meta správ

Na záver vám ešte ponúkam tabuľku so všetkými uvedenými meta správami, (aj keď ich váš nástroj môže podporovať omnoho viac).

Vysvetlenie:

- slovíčko text v stĺpci sysex označuje, že hodnotou bude text
- značka HEX v poli sysex znamená, že hodnotou budú hexadecimálne čísla
- značka X v poli sysex znamená, že hodnotou bude desiatkové číslo, pričom syntax je taká, ako je uvedené v tabuľke, teda napríklad X,X znamená dve čísla oddelené čiarkou

Status	Dátový bajt 1	Dátový bajt 2	Meta bajt	Sysex	Názov	Význam
\$FF	0	0	\$01	text	Text	Textová poznámka
\$FF	0	0	\$02	text	Copyright	Autor
\$FF	0	0	\$03	text	Track name	Názov tracku
\$FF	0	0	\$04	text	Instrument name	Názov nástroja
\$FF	0	0	\$05	text	Lyric	Text piesne
\$FF	0	0	\$06	text	Marker	Značka
\$FF	0	0	\$07	text	Cue point	Značka
\$FF	0	0	\$2F		End of track	Koniec tracku
\$FF	0	0	\$51	X	Tempo	Tempo
\$FF	0	0	\$54	HEX	SMPTE	Časová správa
\$FF	0	0	\$58	X,X,X,X	Time signature	Takt
\$FF	0	0	\$59	X,X	Key signature	Tónina
\$FF	0	0	\$7F	HEX	Data sequencer	Správy sekvencera

Tabuľka taktov a ich midi označení

Takt	Číselné midi vyjadrenie
2/2	2,1,48,8
3/2	3,1,48,8
4/2	4,1,48,8
2/4	2,2,24,8
3/4	3,2,24,8
4/4	4,2,24,8
2/8	2,3,12,8
3/8	3,3,12,8
4/8	4,3,12,8
6/8	6,3,12,8

13. Systémové exkluzívne správy – SysEx

Už sme ich okrajovo spomenuli v kapitole 11, kde bola reč o systémových správach. SysEx sú vlastne tiež systémové správy, ale veľmi konkrétne, zostavené pre konkrétne nástroje, vyvolávajúce v konkrétnom nástroji konkrétny efekt. Práve preto sa nazývajú špeciálne, zvláštne, alebo exkluzívne (System exclusive message) a z tohto názvu je odvodená aj skratka SysEx.

Prebrali sme si vlastne už všetky typy midi správ. Stretli sme sa s mnohými vlastnosťami nástrojov, vedeli by sme už do midi jazyka preložiť melódiu či nastavenia nástroja. No možno si poviete, že aj tak sme neprebrali toľko príkazov a správ, koľko máte funkcií na vašom skvelom modernom nástroji. Možno ste v celom tomto seriáli nenašli odpoveď na to, ako pomocou midi aktivovať na vašom nástroji jeho zázračnú schopnosť improvizatívne hrať, alebo ako pomocou midi nastavovať panorámy zvukov či iné veci, ktoré by ste stlačením tlačidla na vašom nástroji poľahky aktivovali. Ak kladiete takúto otázku, tak vlastne udierate klinec po hlavičke. Midi norma totiž vznikala na konci minulého storočia a jej tvorcom sa vtedy ešte ani neprisnilo o mnohých funkciách, ktoré majú nástroje dnes. Preto ani v midi norme nenájdeme konkrétne príkazy a správy na všetky funkcie, ktoré dnes možno nájsť na nástrojoch. A predsa si s tým výrobcovia poradili a problém vyriešili exkluzívnymi systémovými správami. V zmysle povedaného teda môžeme SysEx správy definovať ako také, ktoré vykonávajú v konkrétnom nástroji konkrétnu úlohu. Výrobcovia môžu nástroj vyrobiť tak, aby po prijatí konkrétnej SysEx správy začal štekať či vypisovať na obrazovke ruské nadávky a možno, ak technika pokročí ešte ďalej, raz za nás možno vyvešia prádlo. Samozrejme, to už zveličujem. No podstatu ste snáď pochopili. Výrobca môže nástroj vyrobiť tak, aby na konkrétne SysEx správy reagoval konkrétnym efektom. Zoznam týchto SysEx správ musí byť súčasťou dokumentácie a – ako ste si určite už domysleli – nemožno ich brať všeobecne. To, čo urobí konkrétna SysEx správa na mojom nástroji, určite neurobí na vašom.

Ako vyzerá SysEx správa? Je to vlastne iba zmes hexadecimálnych čísel. Ako prvý musí byť vždy stavový bajt s hodnotou \$F0, ktorý nástroju povie, že prichádza SysEx správa. Ďalej nasleduje číselný kód výrobcu (je uložený v dátových bajtoch), takže aj zdanlivo výsledným efektom rovnaké správy sa v tomto bode líšia. Napokon sú to už samotné čísla, ktoré majú tvar šesťnástkového čísla a môže ich byť ľubovoľné množstvo (uložené sú v políčku sysex, ktoré sme doteraz používali iba pri meta správach). Celý reťazec zakončuje stavový bajt \$F7, ktorý sa tiež nazýva EOX (z anglického End of SysEx) – koniec exkluzívnej správy.

Ak potrebujete do vašej skladby vložiť SysEx správu, stačí ak v programe Notedit zvolíte príslušnú voľbu v menu a do zobrazeného políčka vpíšete všetky čísla tak, ako vám to káže dokumentácia. Notedit sa už postará o to, aby vaša správa bola vložená do súboru a v pravý čas odoslaná.

Problémom pri SysEx správach nie je ani tak ich poslanie či vloženie do skladby, ale skôr ich dešifrovanie. Ak totiž aj cez nejaký event editor nazriete do midi skladby a objavíte tam SysEx správu, spleť čísel vám spravidla nič nepovie o tom, čo daná správa robí. Na rozdiel od správ note on, note off a podobne, kde hneď viete, čo daná správa vykoná, u SysEx správ je dešifrovanie cudzích správ de facto nemožné. Ako príklad SysEx správy uvádzam napríklad túto: F0,41,10,42,12,40,00,7F,00,41,F7. S takýmto tvarom správy sa môžete stretnúť bežne. Rovnako však môže byť správa uvedená aj takto: F0 41 10 42 12 40 00 7F 00 41 F7, alebo aj takto: \$F0 \$41 \$10 \$42 \$12 \$40 \$00 \$7F \$00 \$41 \$F7. Či už ale sú čísla oddelené čiarkou, medzerou, alebo dôsledne označené znakom \$, ak ide o SysEx správu, môžete vždy vedieť, že ide o šesťnástkové čísla, teda aj nuly v dvojiciach sú dôležité. Pri

vkladaní do skladby dodržujte syntax, akú si daný software vyžaduje – v prípade Noteditu by čísla mali byť oddelené čiarkou.

14. Midi normy a režimy

Dostali sme sa na záver nášho seriálu o midi. Prenikli sme do útrobov tohto systému a priblížili si základy jeho fungovania. Samozrejme, neprebrali sme všetko, pretože midi je systém, ktorý sa neustále vyvíja a modifikuje. Napriek tomu však nie je tak chaotický ako bol pred desiatkami rokov. Vďačíme za to normám, ktoré sa v priebehu rokov vyvinuli a naznačili tak hranice a smer vývoja.

Slovo "norma" padlo v našom seriáli viackrát. Väčšinou sme pritom mali na mysli základnú GM (General Midi) normu, ktorá vznikla ako prvá ešte v roku 1988. Bola to prvá, no veľmi jednoduchá norma, ktorá by však dnešným požiadavkám nevyhovovala. Preto rokmi vznikli ďalšie dve, ktoré pôvodnú GM normu dopĺňajú.

GM norma – General midi

Je to prvá norma, ktorá sa pokúsila v roku 1988 zjednotiť protokoly, ktoré výrobcovia vo svojich nástrojoch dovtedy používali úplne rozdielne. Táto norma pozná iba 128 zvukov, ktoré člení na 16 podskupín (rodín) po 8 a sú vyvolávané správou Program change. Voľba banky tu vlastne ešte neexistuje, keďže zvukov je iba 128. GM tiež podporuje iba 7 kontrolérov a to kontroléry číslo 1 (modulácia), 7 (hlasitosť tracku), 10 (stereo poloha), 11 (čiastková hlasitosť), 64 (sustain) a dvojicu 100 a 101 (registrované parametre).

Normu GM dodnes používajú jednoduchšie počítače a mobilné telefóny, ale aj hudobné nástroje najlacnejšej cenovej kategórie.

GS norma – General standard

Túto normu vytvorila a zaviedla vo svojich nástrojoch firma Roland. Obsahuje minimálne 226 zvukov, takže sa tu po prvýkrát objavuje systém zvukových bánk a zavádza sa kontrolér 0 pre výber banky, ktorý je nasledovaný klasickým príkazom Program change. Samozrejme, nie všetky banky musia byť úplne plné, preto ak nástroj v režime GS príjme správu, ktorá od neho žiada zvuk, ktorý neexistuje, bude nástroj tvrdošijne mlčať. Je rozšírený aj počet kontrolérov. GS nástroje musia podporovať základných 7 kontrolérov z normy GM a k tomu ešte kontroléry 0/32 pre výber banky, 5 pre Portamento, 6/38 pre zadanie dát registrovaným a neregistrovaným parametrom, 98 a 99 pre voľbu neregistrovaných parametrov a prípadne mnohé iné.

XG norma – Extended GM – Rozšírené GM

Normu XG vytvorila a do svojich nástrojov zaviedla firma Yamaha. Počet zvukov sa zvýšil na minimálne 520, pričom pribudli aj sady špeciálnych efektov SFX a rôzne bicie sady. Zvuk sa vyberá postupným zadaním banky (kontrolér 0 a 32) a voľbou zvuku (Program change), pričom ak požadujeme zapnúť zvuk, ktorý neexistuje, nástroj sa pokúsi nájsť podobný a zapne ho. Pribudli aj ďalšie kontroléry, takže norma XG podporuje všetky GM a GS kontroléry a k tomu kontroléry ako čas nástupu a doznenia (71 72), efekty a iné.

GM2 norma – General midi 2

Je to norma, ktorá je rozšírením pôvodnej GM normy, preto GM a GM2 možno počítať za jednu normu. Je to najmladšia zo všetkých noriem, preto má už v sebe integrované štandardy GS a XG noriem, no dopĺňa ich o ďalšie možnosti riadenia kontrolérmi, ale aj sysex správami.

Vzájomná kompatibilita a resety

Problematika, ktorú sme v tejto poslednej kapitole načrtli, je veľmi široká. Zmienené informácie treba brať ako naozaj veľmi veľmi stručné. Je však potrebné o týchto normách vedieť, pretože okrem svojich výhod prinášajú občas v praxi aj problémy. Jednotlivé normy totiž nemožno pokladať iba za súbor pravidiel, príkazov a syntaxe midi správ, ale ide aj o rozdielne formáty spracovávania zvukov, ba dokonca o celé režimy, v ktorých midi nástroje pracujú. Skrátka, zvuk klavíra GM znie inak ako klavír GS alebo klavír XG. Teda nie je to len o tom, akým príkazom zvuk vyvoláme, ale aj o tom, ako je zvuk technicky vytvorený a ako znie.

Nástroje sú spravidla vyrobené tak, aby po zapnutí pracovali v jednom z uvedených režimov, teda boli pripravené pracovať s jednou z uvedených noriem. Ak vlastnime nástroj od Yamahy, bude náš kláves pravdepodobne pracovať v režime XG. Naopak, ak máme doma moderného Rolanda, bude zrejme po zapnutí pracovať v GS režime. Ak potom na takomto nástroji vytvoríme midi súbor, bude automaticky predpokladať, že ho budeme prehrávať na nástroji s rovnakou normou. V praxi sa môže stať, že súbor, vytvorený vo formáte GS nám bude v nástroji značky Yamaha hrať zvlášťne, alebo nebude hrať vôbec. Rovnako tak súbor, vytvorený v režime XG bude mať v počítači, ktorý podporuje iba normu GM zvláštny zvuk.

Ako z tohto problému von? Jediným riešením je zachovať dve základné zásady:

1. Nástroj, na ktorom ideme prehrávať midi súbor, by mal podporovať režim, v ktorom bol súbor vytvorený. Drahšie nástroje spravidla obsahujú zvuky všetkých štyroch formátov a teda sú pripravené okrem svojho východzieho režimu, pracovať aj v iných režimoch, do ktorých sa ale musia prepnúť špeciálnymi sysex príkazmi – resetmi. Teda ak máme napríklad doma Yamahu vyššej kategórie, pravdepodobne bez problémov prehráme aj súbor vytvorený v GS, ale nástroj musí byť do tohto režimu prepnutý. Ak máme nástroj lacnejšej triedy, ktorý iné režimy nepodporuje, musíme pre svoje potešenie hľadať midi súbory, ktoré boli vytvorené v našej norme.

2. Každý midi súbor by mal na začiatku obsahovať sysex správu, ktorá zariadenie prepne do požadovaného midi formátu. Ide o takzvané resetovacie príkazy a je korektné, ak autor súboru, alebo software, ktorý súbor vytváral, vložil do súboru príslušný resetovací príkaz. Ak to tak nie je, môže sa stať, že prehrávame GS midi súbor, ktorý náš nástroj prepne do režimu GS, no potom ho už nevieme vrátiť späť, pretože ďalší midi súbor je už vytvorený napríklad v norme GM, ale nemá v úvode potrebný resetovací príkaz. Vďaka tomu sa nástroj neprepne do GM režimu a skladba nebude hrať správne.

Resetovací príkaz môžeme z programu Notedit poslať aj v reálnom čase, teda bez toho, aby sme prehrávali midi súbor. Tým nástroj prepne do potrebnej midi normy. To sa môže hodiť vtedy, ak nástroj zostal zapnutý v nežiadúcom režime, ale aj vtedy, ak chceme pracovať so zvukmi danej skupiny a podobne.

