

● OSC omzetten naar MIDI ●

Huib Groenewegen

Over MIDI zijn de meningen verdeeld. MIDI zou achterhaald zijn door andere technieken, zoals OSC of Pure Data (PD) en rtpMIDI. Midi over USB bestaat al lang en geeft ook al extra mogelijkheden. MIDI blijft toch vaak het hardware-koppelvlak met de 5-polige DIN-connector of steeds meer met USB-connector. OSC is geen hardware-koppelvlak, maar heeft vele andere mogelijkheden; vooral de vrijheid om over ethernet met veel hogere snelheid meerdere commando's over te sturen. Daar komt dan uiteraard de mogelijkheid bij om over wifi data te verzenden. Dit is het thema van dit artikel: hoe stuur ik OSC-commando's over via wifi en vertaal ik deze vervolgens naar MIDI; in dit geval een 'ouderwets' DIN-plug-koppelvlak.

Voor we ingaan op de techniek: de reden waarom ik hiermee ben begonnen. Een aantal jaren geleden heb ik het idee opgevat om met een sampler een flexibele opzet te maken voor het maken van een elektronisch pijporgel. Een sampler is gemaakt om geluidsamples op te slaan die je vervolgens met MIDI kunt selecteren en zo kunt afspelen. Ik heb een oude AKAI Z4 sampler op de kop getikt voor € 200,- (zie afb. 1).

Deze samplers werden door Akai (Z4 en Z8) gebouwd in 2002 (toen ca. €1500, afhankelijk van de opties) en dat waren de laatste samplers die gebouwd zijn. Daarna kwam de MPC (Music Production Centre) een combinatie van Sampler en Sequencer. De specificaties zijn nog steeds indrukwekkend, de beperking is vooral het interne RAM-geheugen. De originele versie komt met een IDE-harde schijf, die ik heb vervan-

gen door een unit met een SD-kaart, die minder kwetsbaar en ook geruisloos is. Het leuke van deze sampler is dat hij ook via het softwarepakket AKSYS een grafische user-interface heeft om makkelijker te programmeren via USB. Alle functies kunnen ook met het bedieningspaneeltje worden uitgevoerd, maar dat is omslachtiger.

De belangrijkste specificaties van de Z4 zijn:

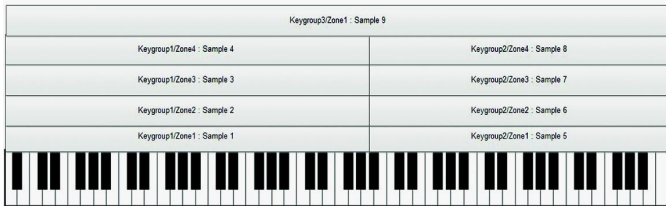
<i>Analog Inputs</i>	(2) balanced 1¼" (-10 dBV)
<i>Analog Outputs</i>	(2) balanced 1¼" (-10 dBV); (1) 1¼" stereo headphone
<i>Digital I/O</i>	coaxial S/PDIF
<i>Other Connectors</i>	(2) MIDI In/Out; (2) USB; (1) 50-pin SCSI; (1) BNC word-clock in
<i>Sampling Resolution</i>	16-bit, 24-bit
<i>Sampling Rates</i>	44.1, 48, 96 kHz
<i>RAM</i>	272 MB (expandable to 512 MB)
<i>Hard Drive</i>	3.5" 60 GB IDE drive
<i>Maximum Sampling Time</i>	96 minutes
<i>Polyphony</i>	64 notes (44.1 or 48 kHz), 32 notes (96 kHz)

Om een instrument te maken ofwel een bepaald orgel-register, maak je eerst een Program (afb. 2); daar kun je heel flexibel mee omgaan: je kunt meerdere samples combineren en eventueel een aanslag-afhankelijke gevoelige sam-

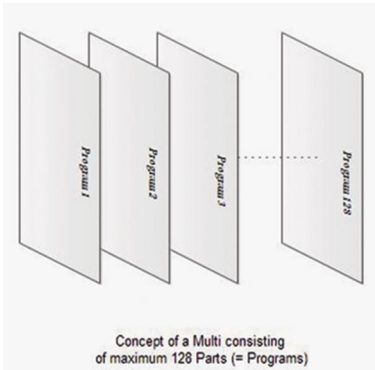
ple bank opbouwen; niet echt interessant voor een orgel, maar je kunt wel meerdere samples samenvoegen om nog mooiere klanken te maken. Vervolgens kun je dan een Program opnemen als een Part in een Multi (afb. 3) en het aantal Multi's is dan je verzameling orgelregisters; je kunt er in totaal tot 128 maken. Nogmaals: heel veel is mogelijk, en de 512 MB aanwezige RAM blijft de beperkende factor, maar er is zeker voldoende ruimte om iets moois te maken.



Afbeelding 1



Afbeelding 2



Afbeelding 3

Om een orgelregister aan of uit te zetten (of wel Program) moet je een partnummer Mute of Unmute aansturen binnen een Multi. Dat kan met het paneel van de Z4 door een keuzemenu door te bladeren, maar het is niet echt handig als je even een register wilt opentrekken tijdens het orgelspelen ... De Z4 heeft een compleet handboek van 52 pagina's gewijd aan het be-

sturen van de meest uiteenlopende functies via een SYSEX-commando. Dat is in de MIDI-wereld een System Exclusive Message die voor elk MIDI-apparaat verschillend is. Werkelijk indrukwekkend wat je er allemaal mee kunt doen.

Het SYSEX commando voor mute/unmute is F0475F001A11nn0XF7 waarbij nn het part (Program = Orgelregister) nummer is en daarna 01 voor mute en 00 voor unmute. In eerste instantie had ik een midi-converterkastje gemaakt met een ARDUINO-boardje waar ik standaard Program change (Cn xx) messages gebruikte afkomstig van de control pads die op een keyboard aanwezig zijn. Deze filterde ik er dan uit en zette ze vervolgens om in een SYSEX-boodschap om een bepaald orgelregister te selecteren. Dat werkte goed, maar de beperking was dat ik geen inzicht had in welke registers er nu open of dicht stonden. Met andere woorden: er ontbrak een grafische interface.

Er is een bijzonder interessante en zeer goed gedocumenteerde app (voor Android en IOS) die het mogelijk maakt om samen met een pc (Windows of Mac) OSC-commando's te sturen en deze vervolgens als MIDI-messages te gebruiken. De app heet TouchOSC en kost € 5,50. (<https://hexler.net>). Het interessante is dat je er een grafisch ontwerp pakket bij krijgt (TouchOSC Editor) dat je een eigen ontwerp laat maken van een display-layout. Er zijn verschillende elementen ter beschikking, zoals faders, potmeters, encoder controls, pushbuttons, toggle buttons, XY pads, multifaders, multi push, multitoggles, multi XY pads, LED's, labels en tijd- en batterydisplay. Ook is het mogelijk om de bewegingssensor van je telefoon aan een control te koppelen. Je kunt ook meerdere pagina's binnen een ontwerp realiseren. De typische toepassing is om je eigen control-layout te maken voor

een programma als Cubase, Renoise of Ableton Live, waar je daarna met MIDI-learning de controls kunt koppelen.

Bingo, alles opgelost, maar ik wil geen pc aan mijn orgel laten hangen alleen voor een grafische interface. Een bijkomstige omstandigheid is dat TouchOSC om te functioneren een extra programma nodig heeft dat TouchOSC Bridge heet. Dat werkt als een op de achtergrond draaiend programma dat via een niet-open communicatieprotocol de OSC-parameters in de Android/iOS omgeving uitleest.

De volgende stap was misschien een Raspberry Pi als mini PC, maar ja dan hebben we het over Linux en begin je eigenlijk opnieuw, omdat er geen TouchOSC Bridge is voor Linux en dan is Pure Data een optie, maar met een totaal andere invalshoek. Daar is ene Albert Gräf diep ingedoken: (<http://lac.linuxaudio.org/2014/papers/5.pdf>). Een zeer interessant artikel en hij heeft ook een oplossing gemaakt voor Linux om de TouchOSC Bridge voor Linux te maken met Pure Data. (<https://bitbucket.org/agraef/osc2midi-utils/src>).

Na veel speuren op internet vond ik een goede opzet. Iemand die met een ESP8266 een applicatie had geschreven om controlcommando's met TOUCHOSC te realiseren en ze via MIDI naar een synthesizer te sturen. De ESP8266 is makkelijk binnen de Arduino-omgeving te programmeren, dus dat was de oplossing!

Referentie:

<https://tadas-s.github.io/toys/2016/08/14/esp8266-osc-to-midi-wifi-bridge>

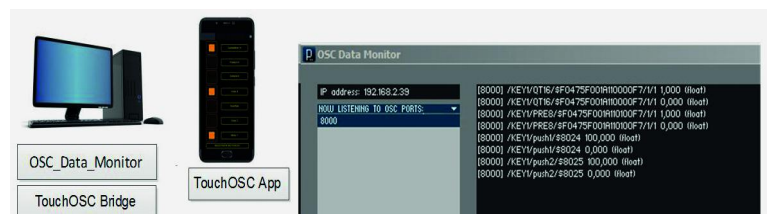
Het programma dat ik gemaakt heb is aangepast om de SYSEX-boodschappen over te sturen en ook MIDI-note on/off, de control messages zijn origineel in het programma gelaten. Het sturen van de SYSEX-boodschappen was niet een-twee-drie te doen. Maar uiteindelijk heb ik via een omweg mijn doel bereikt.

Daarom is het belangrijk is om te begrijpen hoe de OSC-boodschappenstructuur werkt vanuit de TouchOSC App naar de ESP8266.

Zie ook de algemene documentatie over OSC op <http://opensoundcontrol.org>

Een makkelijke manier om de OSC-data te analyseren is het opzetten van een test met behulp van de pc; zie afb. 4. Configureer in de TouchOSC App de MIDI-Bridge naar je pc en vul het Host IP-adres in van je wifi-netwerk. Met het gratis Java-programma van Kasper Kamperman OSC Data Monitor op hetzelfde IP-adres ingesteld als je pc kun je de OSC data zien die de app verstuurt.

<http://www.kasperkamperman.com/blog/osc-datamonitor>



Afbeelding 4

Een niet-standaardmanier die ik heb toegepast is het gebruiken van het OSC-naamveld om daar de MIDI-codes in over te sturen. Vervolgens kun je dan met het float-variabele veld de waarden in de MIDI-code m.b.v. een routine in de ESP8266 wijzigen om ze naar de MIDI-out te sturen. De uitvoering is als volgt gedaan: zet in het naamveld en voor het begin van de MIDI-code een \$-teken als scheidings-karakter dat aangeeft dat daarna de MIDI-code begint; dat is dan makkelijk in de ESP8266-software te testen.

Voor het oversturen van notes:

/KEY1/pushx/\$nnote number(HEX!),velocity waarde; waar /KEY1/pushx/ het naamveld is met x als positie van de toets. Na het \$-teken: '8' is note off-code en 'n' is channel number,

gevolgd door het midi note number in hexadecimaal. Via de OSC-data wordt de velocity-waarde gelezen; is deze float-waarde >0 dan wordt in het ESP8266 programma de MIDI-code veranderd naar: 9n/note number/velocity, zodat de note aan gaat. Met de vingerpositie op het touchkeyboard is ook de velocitywaarde nog eventueel over te sturen. Dat moet je dan wel aanklikken in de TouchOSC-editor. Met OSC aangevinkt in auto wordt het adresveld automatisch gestuurd; als Y position velocity niet wordt aangevinkt moet er wel een Value Range worden ingevuld van 0 !(anders gaat de note niet uit) tot de maximale velocity-waarde die je wilt gebruiken in decimaal! Wordt de note losgelaten dan is de velocity waarde =0 en wordt de 8n/notenumber/velocity=0 naar de midi out-poort gestuurd.

Voor het oversturen van SYSEX:

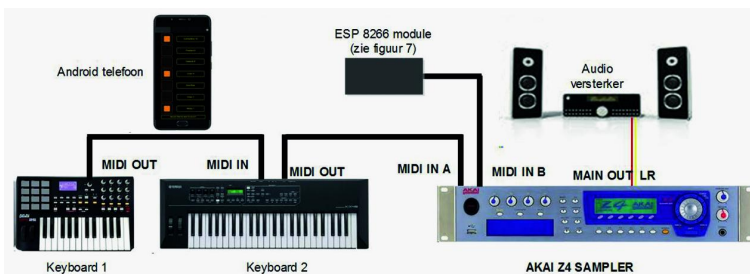
Met een toggle switch in TouchOSC kun je een SYSEX-code /KEY1/PRE8/\$F0475F001A11nn00F7/1/1 toggle-waarde sturen. Met het Mute SYSEX-command voor de Akai is het partnummer 'nn' van de multi vast geprogrammeerd in de positie van de toggle switch en de aan/uit info staat in het float-variabele veld om het muten van het Program aan en uit te zetten. Deze waarde 0 of 1 wordt door het ESP8266 programma in de juiste positie van de SYSEX MIDI-boodschap geplaatst.

Omzetten van float naar byte-informatie:

Omdat MIDI niet overweg kan met float-getallen, moet de float naar byte-info worden omgezet. Na veel zoeken is een eenvoudige methode gevonden via een UNION in de Arduino C-taal en het werkt als volgt.

Een UNION kan verschillende variabelen herbergen en met een andere naam benaderd worden. Het is één plek in het geheugen die ruimte heeft voor de grootste variabele in de definitie. Een bijkomstigheid is dat je, omdat het één geheugenplek is, er ook conversie mee kunt maken. In dit geval 4 HEX-waarden van een float array MSB b[3] ==> LSB b[0]. Als je de vier bytes definieert kun je de float-waarden lezen, maar ook als je de float-waarde definieert kun je de byte array lezen! Het complete ESP8266-programma en de TouchOSC-layout kun je downloaden via

In afbeelding 5 zie je het totale plaatje van het flexibele orgel. Uiteraard kun je meer keyboards aansluiten of een MIDI-pedaal. De Akai Z4 heeft ook geluidseffecten aan boord die je kunt gebruiken, zoals bv. echo.

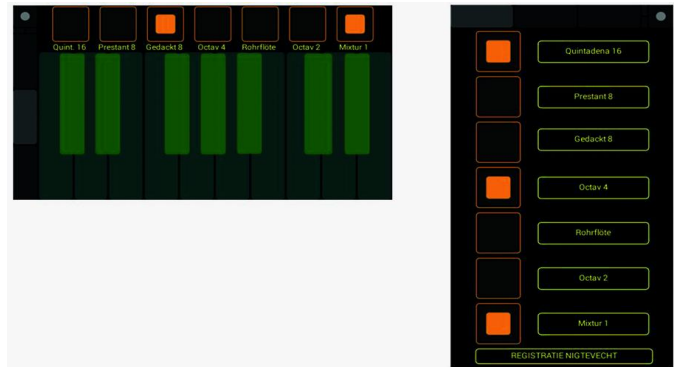


Afbeelding 5: Totale systeemopbouw elektronisch pijporgel

In afbeelding 6 zie je twee gemaakte layouts op een Android-telefoon.

Er zijn weinig elementen gebruikt: De labels, pushbuttons (toetsen) en de toggles voor de registers en de led voor de indicatie aan/uit. De toetsen zijn handig om even iets te testen zonder dat je een keyboard hoeft aan te sluiten.

Afbeelding 7 geeft de hardware weer, waarbij ik ervoor gekozen heb om zo weinig mogelijk zelf te solderen, je kunt ook het Midi-boardje (€ 8,50) vervangen door een zelfgemaakt ontwerp, wat niet moeilijk is. Ik heb de goedkoopste ESP8266 (ESP01: € 1,50) gebruikt, maar wel op een 8-polig adapterboard (€ 0,75), zodat ik hem er uit kan halen en met een externe programmeeradapter (€ 1,25) via de usb-poort kan herprogrammeren. Het ingebouwde adapterboardje functioneert tevens als 3,3V- naar 5V-converter. Het is mij



Afbeelding 6: Layoutvoorbeelden op Android-telefoon



Afbeelding 7: ESP8266 OSC naar MIDI module

nog steeds niet duidelijk of de IO van de ESP8266 wel of niet 5V tolerant is. Zoek maar op internet en de meningen zijn verdeeld. Dus beter voorkomen dan genezen en level-conversie toepassen! De totale kosten van de hardware zijn minder dan 20 euro.

Dit concept is niet alleen beperkt tot het aansturen van MIDI, je kunt hiermee ook ander systemen aansturen waarbij je de plaatsen van je objecten in het scherm kunt definiëren. Het leuke is dat je én een grafische layout kunt bouwen én deze kunt vastknopen aan bepaalde commando's die via wifi worden uitgevoerd. MIDI kun je ook vergeten en gewoon RS232 serieel of via USB te gebruiken (wat MIDI eigenlijk ook is). Meestal heb je al een tablet of een telefoon die je als interface hiervoor kunt gebruiken. Uiteraard is dit geen volledige beschrijving van de software, mochten er vragen of opmerkingen zijn dan kan ik daar altijd met plezier op reageren.

Mijn uiteindelijke doel heb ik nog niet geheel bereikt. Mijn vader is ruim 33 jaar organist geweest in een mooi dorp aan de Vecht, en wat mij nog rest is de pijpen van dat orgel te gaan sampelen om deze vervolgens te gaan inrichten in de Akai Z4, zodat ik het orgel dat hij tijdens zijn leven bespeelde dan thuis in mijn hobbykamer heb.

