

Beispiel für die Lied-Definition mit dem Tone-Array

Ein Beispiel-Notensatz:



Konsequenzen daraus:

1. Anzahl Tracks (= Anzahl gleichzeitig auftretender Noten): 2
2. Anzahl Noten in den Tracks:
Die Zuordnung der Noten zu den Tracks ist frei wählbar, sollte aber systematisch erfolgen. Z.B.:
Track 1 (oben): 5 Noten
Track 2 (unten): 4 Noten
Parametrierung des Tone-Array: `NUM_TRACKS=2; MAX_TONES_PER_TRACK=5;
struct tone to[NUM_TRACKS][MAX_TONES_PER_TRACK];`
3. Kürzeste Note: Achtel-Note (dritte Note im unteren Track)
4. Dauer der kürzesten Note:
Geschwindigkeits-Vorgabe: 120 BPM (beats per minute) bezogen auf Viertel-Noten.
Eine Viertel-Note dauert also $1/120$ Minute = $1/2$ Sekunde = 500 ms
Eine Achtel-Note dauert dann 250 ms
5. Anzahl Samples für die kürzeste Note: Audio-Codec: 48000 Samples pro Sekunde
Für eine Achtel-Note werden also 12000 Samples benötigt.
6. Anzahl Tone-Slots für das Lied:
Die Dauer entspricht 8 Viertel-Noten; das sind 16 Achtel-Noten.
Ein Tone-Slot hat die Länge der kürzesten Note. Also gibt es 16 Tone-Slots.
Eine Viertel-Note belegt also 2 Tone-Slots, eine halbe Note belegt 4 Tone-Slots
7. Eigenschaften der Noten im Track 1:
1.Note: Tonhöhe: c' / Frequenz: 261.625565 / Beginn-Slot: 0 / Ende-Slot: 1
2.Note: Tonhöhe: e' / Frequenz: 329.627557 / Beginn-Slot: 4 / Ende-Slot: 5
3.Note: Tonhöhe: f' / Frequenz: 349.228231 / Beginn-Slot: 6 / Ende-Slot: 7
4.Note: Tonhöhe: g' / Frequenz: 391.995436 / Beginn-Slot: 8 / Ende-Slot: 11
5.Note: Tonhöhe: e' / Frequenz: 329.627557 / Beginn-Slot: 12 / Ende-Slot: 15
8. Eigenschaften der Noten im Track 2:
1.Note: Tonhöhe: d' / Frequenz: 293.664768 / Beginn-Slot: 2 / Ende-Slot: 3
2.Note: Tonhöhe: c' / Frequenz: 261.625565 / Beginn-Slot: 4 / Ende-Slot: 5
3.Note: Tonhöhe: g / Frequenz: 195.997718 / Beginn-Slot: 11 / Ende-Slot: 11
4.Note: Tonhöhe: c / Frequenz: 130.812783 / Beginn-Slot: 12 / Ende-Slot: 15
9. Amplitudenwerte vorgeben; z.B. in 2 Variablen A1, A2 (für Track 1 bzw. 2)
10. Daraus folgendes Tone-Array definieren:
`to[0][0].bSlot=0, to[0][0].eSlot=1, to[0][0].freq=261.625565, to[0][0].amp=A1; // c'
to[0][1].bSlot=4, to[0][1].eSlot=5, to[0][1].freq=329.627557, to[0][1].amp=A1; // e'
to[0][2].bSlot=6, to[0][2].eSlot=7, to[0][2].freq=349.228231, to[0][2].amp=A1; // f'
to[0][3].bSlot=8, to[0][3].eSlot=11, to[0][3].freq=391.995436, to[0][3].amp=A1; // g'
to[0][4].bSlot=12, to[0][4].eSlot=15, to[0][4].freq=329.627557, to[0][4].amp=A1; // e'

to[1][0].bSlot=2, to[1][0].eSlot=3, to[1][0].freq=293.664768, to[1][0].amp=A2; // d'
to[1][1].bSlot=4, to[1][1].eSlot=5, to[1][1].freq=261.625565, to[1][1].amp=A2; // c'
to[1][2].bSlot=11, to[1][2].eSlot=11, to[1][2].freq=195.997718, to[1][2].amp=A2; // g
to[1][3].bSlot=12, to[1][3].eSlot=15, to[1][3].freq=130.812783, to[1][3].amp=A2; // c`