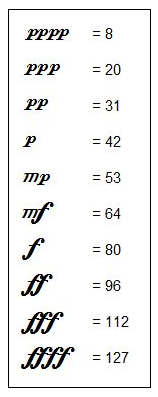
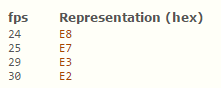
Midi :

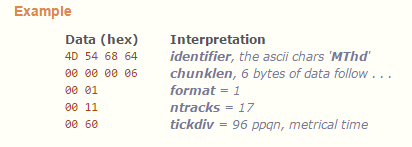
* Se focalise sur lecture et modification de musique (représentée comme une ou plusieurs séquence de notes)
* Différence avec le format audio : étude plus subtile d’un extrait : info à propos des notes, temps de début et de fin, dynamique, tempo, variations du volume, de hauteur + autres info comme type d’instrument qui doit être utiliser pour jouer la note
* Utilisé comme langage universel pour donner info en temps réel dont on a besoin pour jouer
* Usage commun : l’ordi peut contrôler un synthé externe pour lui faire jouer de la musique
* Chaque message est envoyé au moment où il doit être interprété. C’est instantané
* Langage midi définit pas le son lui-même, mais la séquence d’instruction pour que le synthé fasse le bon son
* S’articule en au moins deux plages : entête + données (MThd / MTrk)
* Dans **MThd :** on retrouve trois sortes de trucs : **format**/ **ntracks**/ **tickdiv.** Contient max 14 bytes
* **format :** Organise les différents « morceaux » (MTrk) (ou piste) de morceaux ensembles 3 **type**
  + Type = 0 : une seule piste
  + Type=1 : deux pistes ou plus à jouer simultanément. Première piste = tempo
  + Type=2 : un pistes ou plus joués indépendamment
* **ntracks :** Le nombre de pistes (dans le format type 0, ntracks=1 forcément) pour les autres format, pas de limitation
* **tickdiv :** « intervalle de temps ». Deux schemas de temps , metrical timing et time code. Le premier bit des deux bytes codant le tickdiv indique si on est en metrical timing ou en time code.

**1111 1111 1111 1111**

* + **Bit 15 = 0 : metrical timing** : temps métrique. Les bits de 0 à 14 donnent un nombre qui indique le tempo (*ex 01100000->x0060->96 tempo seconde)*
  + **Bit 15= 1 : time code :** Nombre de mesures par seconde( ?). Le s bites de 8 à 15 donnent ce nombre parmis les 4 standards : 

Les bits de 0 à 7 donnent le nombre de pulsations par mesures( ?)

Valeurs typiques : 4 (corresponding to MIDI Time Code), 8, 10, 80 (corresponding to SMPTE bit resolution), or 100.



* **MTrk :** Contient un tas d’évènements se déroulant dans l’ordre où on les a écrits. Il est possible de mettre des pistes de nature différentes dans un même Mtrk. Les données ont la forme suivante :
  + Le temps où se trouve la note, sur 4 bytes max donnant le temps depuis le dernier évènement (delta time). La première note aura son deltatime=0 . Seulement les 7 derniers bits de chaque bit indiquent temps entre new event et dernier event. Le bit 0 de chaque byte indique si un autre byte suit.
  + Deux bytes ou plus qui décrivent la nature de l’évènement
* Les **EVENTS**: trois types, MIDI/SysEx/Meta
  + **MIDI :** Message standard à un une chaine midi.
* Sequence de 1 bytes ou plus :
  + STATUS byte : type de message. Nb de data byte dépendent du type de message. Contient le numero de la prise Midi (16 possibilités de 0 à 15 en hexa) sous la forme 1xxx xxxx
  + DATA bytes (des fois) : paramètres additionnel
* On peut envoyer des NOTE ON simultanément pour faire des accords
* La note est un nombre entier codé en hexadécimal compris entre 0 et 127. La note C (do) est 1 et on avance de demi ton en demi ton (ex : C# = 2). On ajoute 12 pour monter d’une octave. Le do du milieu (le plus bas de la clé de sol et le plus haut de la clé de fa) est encodé par le nombre 60 (soit 3C)