1 Comptons sur nos doigts...

Utilisons notre main. Nous allons supposer pour cette expérience que notre main comporte 5 doigts et que nous puissions distinguer seulement, pour chaque doigt, le fait d'être refermé ou déployé (on pourrait faire mieux en ne pliant le doigt qu'au niveau des phalanges, mais on ne va pas le faire). On souhaite, à l'aide de cette main, compter à partir de 0. Jusqu'à combien peut-on compter ?

Sans avoir à beaucoup réfléchir, nous savons que nous pouvons compter jusqu'à 5. Attention d'ailleurs, les codes sont différents d'un pays à l'autre. Pour ceux d'entre vous qui ont regardé *Inglourious Basterds*, vous vous rappelez peut-être de cette scène dans la taverne où l'espion britannique se faisant passer pour un capitaine allemand se trahit par sa commande de Whisky : il lève, pour trois verres, l'index, le majeur et l'annulaire. Grossière erreur ! Les Allemands montrent trois à la main avec pouce, index et majeur.

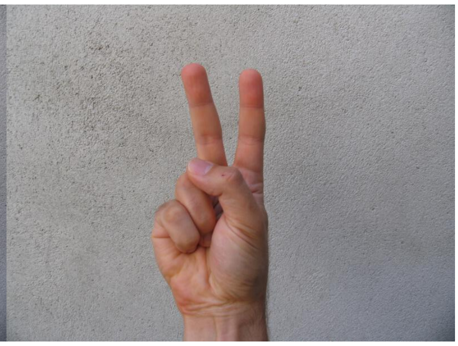
On pourrait également parler de la Chine, où si vous arrivez au bar pour commander deux bières en levant, comme en France, pouce et index, vous vous retrouvez avec 8 bières.

Ces différentes techniques permettent de compter jusqu'à 10 avec une main, selon un code que l'on apprend depuis tout petit (nous, français, il nous faut deux mains pour compter jusqu'à 10). Mais on peut faire mieux : en fait, on peut compter jusqu'à 31 avec une seule main, juste en montrant ou en cachant certains de nos doigts (à supposer qu'on se soit suffisamment entrainés, car lever l'annulaire sans lever aucun autre doigt est un peu difficile).

Avant d'étayer cette prouesse digitale avec une théorie mathématique, donnons un aperçu :

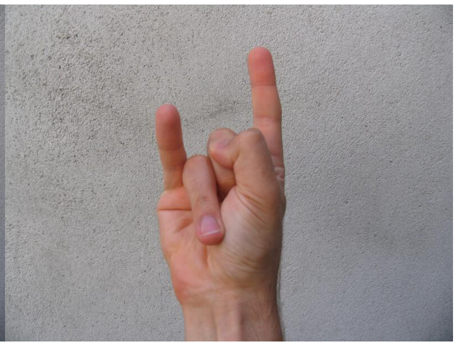
0

1



6

4



18

Comment obtient-on ces nombres ? Avec une théorie mathématique...

2 Le binaire

Il était une fois en Pangée, une manière bien ancestrale de compter : l'unaire. On comptait avec des bâtons (on continue de le faire, généralement en formant des petits carrés qu'on clôt d'une diagonale, ou en formant quatre bâtons verticaux barrés par un cinquième de manière oblique). Sauf que si cette méthode est assez pratique pour additionner (il suffit de mettre tous les bâtons ensemble), elle l'est bien moins pour multiplier.

C'est pourquoi on compte maintenant dans une base. La base 10, donc avec les chiffres de 0 à 9. Mais quand on est une machine, on est un peu plus limité : pour chaque fil, on peut savoir si le courant passe ou ne passe pas, mais pas bien plus. Donc on doit compter avec deux chiffres : le 0 et le 1. Le principe est donc le même :

- le symbole de victoire correspond, quand on regarde l'image, à l'écriture 00110 (un 0 pour un doigt fermé, un 1 pour un doigt ouvert). En base 10, on interprète le nombre de droite comme le chiffre des unités, puis des dizaines, des centaines... il en est de même en base 2 : le chiffre des unités, des « deuzaines », des « quatraines »... : ainsi le nombre 00110 correspond au nombre $1\times4+1\times2+0\times1 = 6$ tout comme 1 789 correspond à $1 \times 1\ 000 + 7 \times 100 + 8 \times 10 + 9 \times 1$.
  Pourquoi le 4 dans le calcul ? Parce qu'en base 10, on a compté par unités (10^0), dizaines (10^1), centaines (10^2), milliers (10^3) et que donc en base deux, cela correspond aux unités (2^0), deuxaines (2^1), quatraines ($2^2 = 4$), huitaines ($2^3 = 8$) etc.
- le symbole du cornu correspond à l'écriture 10010 donc à $1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 16 + 2 = 18$.
- etc. : le chiffre en position i en partant de la droite doit être multiplié par 2^i pour interpréter le nombre.

Ainsi les nombres en binaire s'égrènent de la sorte, de 0 (00000) à 31 (11111) :

0	00000	8	01000	16	10000	24	11000
1	00001	9	01001	17	10001	25	11001
2	00010	10	01010	18	10010	26	11010
3	00011	11	01011	19	10011	27	11011
4	00100	12	01100	20	10100	28	11100
5	00101	13	01101	21	10101	29	11101
6	00110	14	01110	22	10110	30	11110
7	00111	15	01111	23	10111	31	11111

3 Une donnée physique, un codage numérique

Le capteur d'un satellite PLEIADES mesure des luminances en W/cm²/stéradian. Comment faire pour coder en machine cette luminance et l'envoyer ensuite sur terre ?

On l'a compris, un ordinateur compte avec des 0 et des 1. Oui, mais combien en a-t-il à sa disposition ? La taille mémoire de votre ordinateur se mesure en octets. Et un octet, comme son nom l'indique, c'est huit quelque chose. Huit chiffres binaires, en anglais *binary digit* ce qui a donné le mot bit. Un octet donc, c'est huit bits. On en a beaucoup à disposition des bits, sur notre disque dur, puisqu'il y a fort à parier qu'il contient un sacré nombre de gigaoctets c'est-à-dire (à un petit abus près) : un milliard (10^9) d'octets.

Il faut cela dit choisir quelle place va prendre une mesure, c'est-à-dire combien de bits on s'autorise à utiliser pour coder une luminance. Les concepteurs des satellites PLEIADES ont choisi d'utiliser 12 bits pour coder une mesure physique : ils ont donc accès à toute une gamme de nombres, de 0 (0000 0000 0000) à 4 095 (1111 1111 1111).

Cela permet donc de discriminer 4 096 mesures différentes : de 0 (la luminance la plus faible : le noir) à 4 095 (le blanc le plus étincelant mesurable). Reste à échelonner les valeurs, et à dire précisément quelle fonction nous fait passer des mesures en W/cm²/stéradian vers des nombres entre 0 et 4 095. Mais ça, c'est une affaire de physiciens !

4 Un second codage numérique

Vous avez peut-être déjà eu - il y a longtemps dans vos téléchargements via modem 56k ou plus récemment avec votre clef USB passée à la machine à laver - des petits soucis de fichiers. L'ordinateur vous informe que le fichier que vous voulez consulter est endommagé ou corrompu. Cela signifie que, lorsque l'ordinateur lit le fichier, il se rend compte qu'il y a un souci dans le codage : il lit une certaine suite de 0 et de 1, mais s'aperçoit que cette suite n'a pas de sens (de la même manière que certaines combinaisons de lettres n'ont pas de sens, même si « kawax » ferait beaucoup de points au Scrabble).

Il faut parfois assez longtemps pour acquérir une image satellite : de l'ordre de 75 jours pour couvrir le département des Hautes Alpes complètement à mieux que 15° d'inclinaison, 25 jours pour obtenir 2 acquisitions peu nuageuses sur une zone de Guyane. . . Il serait donc préjudiciable, lors de la transmission de données depuis le satellite vers la terre, qu'il y ait des bits qui se transforment ou se perdent, donnant à l'arrivée un morceau d'image corrompu.

C'est pour cela que dans une première étape de codage numérique, on rajoute quatre bits, appelés bits de contrôle. Ils sont là pour à la fois contrôler qu'il n'y a pas eu d'erreur, et en corriger certaines. Il existe plusieurs types de codes correcteurs d'erreurs. On peut présenter un code très simple qui ne permet pas de corriger les erreurs, mais qui permet d'en détecter : le bit de parité.

Prenons notre information de luminance sur 12 bits. Comptons le nombre de 1 dans ce codage. Si ce nombre est pair, on rajoute un 0 à la fin, si ce nombre est impair, on rajoute un 1 à la fin.

<u>Exemples de codage :</u>

- 1110 0101 0000 contient cinq 1 donc on rajoute un 1 à la fin, pour donner 1110 0101 0000 1.
- 1010 0101 0000 contient quatre 1 donc on rajoute un 0 à la fin, pour donner 1010 0101 0000 0.

<u>Exemples à la réception :</u>

- Supposons que nous recevons la donnée 1000 0101 0001 1. Dans les 12 premiers bits, nous comptons quatre 1. Ainsi le 13^{e} bit, qui vaut 1, n'est pas cohérent avec la réception de cette donnée. Nous savons donc qu'il y a au moins une erreur. Malheureusement on ne sait pas où : tout ce qu'on peut faire, c'est redemander à recevoir cette donnée.
- Supposons que nous recevons la donnée 1010 0101 0000 0. Nous constatons que le bit de parité correspond bien à 0 puisque dans les 12 premiers bits il y a quatre 1. Cela veut-il dire qu'il n'y a pas d'erreur ? Presque. Il pourrait y avoir un nombre pair d'erreurs. Sauf qu'avoir plus d'une erreur dans un si petit nombre de bits n'arrive quasiment jamais, donc on peut considérer que la réception de cette donnée s'est faite sans erreur.

Une fois l'image reçue sur terre, et les erreurs éventuellement corrigées, ces quatre bits par mesure de luminance deviennent inutiles, et on s'en sépare alors pour gagner de la place (puisqu'on gagne 4 bits tous les 16 bits, on gagne donc 25%, ce qui est loin d'être négligeable).