Ce document se retrouve à la fois dans le repo public d'enseignement de dccote et sur le site Web de DCC Lab. Le format Markdown est celui supporté par Typora.io. Une version PDF est disponible.

DAQ: Premiers contacts

Au fil des années, j'ai développé le goût de la programmation des "appareils physiques". J'aime contrôler les choses qui interagissent avec le monde extérieur: une platine de microscope, un laser, une carte d'acquisition, une caméra, une imprimante, etc. Ce goût s'est transformé en expertise au fil des années, et plusieurs étudiants viennent me voir (dans mon groupe, chez Bliq et dans les laboratoires d'enseignement) avec le goût d'en savoir plus. Ce n'est pas toujours facile de transférer ces connaissances, et c'est en partie par la façon dont je les ai acquises.

En effet, mes premiers contacts avec le *hardware* sont venus dès le Commodore-64 en 1983: l'achat de l'ordinateur chez Distribution au Consommateur était bien, mais l'achat par la suite d'une imprimante Star NX-1000 avait subitement décuplé la puissance de notre machine, qui autrement ne faisait que des dessins et des *bips-bips* sans vraiment laisser de traces ou changer quoi que ce soit dans le monde qui m'entourait. Si on envoyait les bonnes commandes à l'imprimante, elle bougeait, elle écrivait, elle dessinait. Un jour, j'ai été obligé de sortir un tournevis parce que le joystick n'arrêtait pas de se briser parce qu'on jouait toujours aux Olympiques (le saut à obstacles demandait de brasser le joystick de gauche à droite sans arrêt le plus rapidement possible). Je voulais jouer, je n'avais plus de joystick parce que ma soeur le brasssait trop fort, je l'ai réparé. Plus tard, plusieurs de mes amis et moi avons appris à utiliser la "Copie-21-secondes", qui était une modification hardware du lecteur de disquette qui permettait d'outrepasser la protection en envoyant l'information lue directement à l'ordinateur au lieu de laisser le lecteur la traiter et la bloquer. J'avais des fils qui dépassaient de mon lecteur de disque. Je ne comprenais absolument rien à ce que ça faisait, mis à part que je pouvais copier tous les jeux que je voulais. C'était tout aussi puissant que mystérieux. J'ai appris le langage assembleur sur le C-64, en partie grâce à un livre obscur qu'un ami informaticien de ma mère à la Ville de québec m'a prêté, j'ai discuté de différentes choses avec d'autres gars de 12 ans sur les *BBS* de Québec (l'ancêtre d'Internet), et j'ai passé mes soirées à programmer un peu n'importe quoi, n'importe comment: c'était tout croche, mais c'était passionnant. À la fin, cependant, ça marchait. Je trippais.

Quand j'ai commencé mon doctorat en physique à l'Université de Toronto en 1995, il y avait un cours qui s'appelait *Microprocessor Interfacing Techniques*, donné par le très expérimenté et passionné Jim Drummond. Le but était de nous montrer comment faire l'interface entre le monde extérieur et nos ordinateurs. Un cours de rêve: nous avions les cours magistraux remplis d'anecdotes de Prof. Drummond, ensuite le laboratoire rempli de pièces, de chips, de breadboards, d'ordinateurs PC 8088 Turbo (rien de moins) et de cartes d'acquisition. De plus, nous avions la *clef* du local pour venir à toute heure du jour et de la nuit. J'ai donc finalement appris la logique TTL, les

flip-flop, les compteurs synchrones et asynchrones, la numérisation, la sortie analogique, comment lire des feuilles de spécification d'un chip, etc... Les multiples bribes d'information glanées à gauche et à droite au fil des années ont fini par converger dans un tout cohérent: j'ai réussi à comprendre les petits détails les plus minuscules d'un circuit logique tout en étant capable de faire une abstraction suffisante en blocs fonctionnels pour finalement construire un système complet.

Avec le recul de 30 ans à taponner des ordinateurs, des circuits, péter des fusibles, briser des cartes d'acquisition, sauter des détecteurs, réparer des *drivers* de *stepper motors* pour graduer, ou passer des soirées entières juste pour faire allumer une DEL dans un circuit de base, je me rends compte que les étudiants qui viennent me voir n'ont pas souvent eu la chance *d'expérimenter* avec les ordinateurs. Ceci vient en partie du fait que les ordinateurs sont de plus en plus compliqués, et plusieurs couches de complexité ont été ajoutées depuis le C-64: les processeurs executent des opérations en parallèles, le fameux port d'imprimante ou de modem RS-232 a été remplacé par le port *Universal Serial Bus*, et les systèmes d'opération, la protection de mémoire, le *preemptive multitasking*, la sécurité font qu'aujourd'hui, il y a rarement "une case de mémoire" qui correspond à des *pins* à l'arrière de l'ordinateur (ce que le C-64 avait), et l'ordre exact de l'exécution des opérations n'est pas toujours connue (ce qu'on pouvait contrôler facilement avec `sys 49152` par exemple sur le C-64, la case de mémoire de prédilection pour les jeux). Ainsi, expérimenter en tant que novice devient de plus en plus difficile parce que la marche est beaucoup plus haute qu'en 1983 quand j'ai commencé, ce qui peut être intimidant.

J'espère donc, avec une série de petits tutoriels, faire une introduction de base au Contrôle d'Appareils. Cela va évidemment commencer par des tutoriels extrêmement simples mais qui devraient vous permettre d'acquérir les connaissances pour faire du Contrôle d'Appareils à travers une démarche basée sur **l'expérimentation méthodique**.

Dans le prochain article, nous commencerons par la construction très simple d'un minuscule circuit à base d'un chip de FTDI UM232R pour obtenir, de façon la plus simple possible, accès à des *pins numériques* de sortie et d'entrée.