



## Conseil et expertise sur les microcontrôleurs

### **Démarrage de nouveaux projets sur MCU LPC1788 avec l'IDE LPCXpresso et la pile LPCOpen**

Notre client commençait de nouveaux projets basés sur le MCU LPC1788 (Cortex- M3). L'ensemble de modules open-source LPCOpen du site LPCWare devait être utilisé; Il inclut les pilotes MCU, le portage du RTOS FreeRTOS, la pile TCP/IP lwIP, etc. Au-delà d'une simple formation sur la partie matérielle, Ac6 a aidé le client à débiter avec ces logiciels, en se concentrant sur ce dont il avait besoin: la création d'un projet intégrant les différents pilotes et modules à l'aide de bibliothèques configurables. L'IDE utilisé était LPCXpresso. Ac6 a également conseillé le client pour mettre en oeuvre le débogage de projets FreeRTOS; Ac6 a ensuite continué à soutenir le client pendant le développement d'applications utilisant lwIP.

### **Mise en route d'un projet STM32 MCU à un niveau très bas**

Notre client commençait un projet basé sur un MCU STM32. Comme le projet visait le le domaine médical, il était donc très critique, le client devait développer à un niveau très bas et très proche du matériel. Le langage C++ a été utilisé, mais les drivers des périphériques STM32 ont dû être re développés à partir de zéro à cause des contraintes de certification. Ac6 a fourni une description très précise du matériel (au niveau des registres) et a aidé le client à commencer le développement des pilotes. Les pièges les plus courants inhérents à ce MCU (basé sur un Cortex-M3) ont également été détaillés .

### **Conception et programmation d'une plate-forme Microblaze**

Notre client commençait un nouveau projet basé sur un FPGA Spartan-6 de Xilinx. On a utilisé la boîte à outils EDK (avec les outils XPS et SDK) faisant partie de l'IDE ISE. Le client avait besoin de créer sa plate-forme matérielle spécifique basé sur un noyau Microblaze avec ses propres IP périphériques puis développer le logiciel pour les gérer. Ac6 a aidé le client à développer ces périphériques (en utilisant le langage VHDL) et à les connecter au processeur. Nous l'avons également aidé à démarrer le développement des drivers de ces périphériques.