

nouveaux outils de Microchip!

PICKit 5 et MPLAB ICD 5 sont disponibles maintenant !

Par Tam Hanna, pour Microchip

Il est loin le temps où les passionnés d'électronique bricolaient des programmeurs « maison » pour programmer les PIC et autres microcontrôleurs similaires. Microchip a récemment présenté son dernier débogueur/programmeur en circuit PICKit 5, qui est très abordable et qui est compatible avec la plupart des familles de microcontrôleurs Microchip. Pour le dépannage professionnel, le puissant débogueur/programmeur MPLAB ICD 5 est également disponible. Cet article explore leurs nouvelles fonctionnalités et montre comment intégrer les nouvelles versions dans un écosystème de développement existant.

Maintenant avec l'USB-C!

Microchip s'est toujours engagé à réduire l'encombrement de câbles dans les laboratoires des développeurs MPLAB. La variante précédente d'ICD 3 utilisait un connecteur USB-A, tandis que le PICKit 4 était équipé d'un connecteur Micro-USB.

La normalisation dans le monde des connecteurs USB signifie maintenant que le PICKit 5 [1] et l'ICD 5 [2] sont tous deux livrés avec un connecteur USB-C, qui est en train de devenir la norme pour tous les smartphones modernes. Les soucis engendrés par les différents câbles USB feront partie du passé, grâce à la standardisation de ce connecteur.

Pour les propriétaires de ces appareils, cela présente plusieurs avantages. Tout d'abord, pour un développeur, il est désormais facile d'utiliser le même câble pour recharger votre smartphone et pour vous connecter à votre environnement de développement. Cela permet de vous faire gagner de la place, et parfois de l'argent lors des voyages d'affaires, car vous pouvez emprunter des câbles USB-C presque partout en raison de leur utilisation généralisée dans l'industrie des smartphones.

La deuxième innovation concerne spécifiquement les propriétaires de la version « plus grande » du programmeur ICD 5. Grâce à son accès à une plus grande puissance, l'IC 5 peut fournir jusqu'à 1 A au circuit d'application. Dans de nombreux cas, cela élimine la nécessité de prévoir une deuxième source d'alimentation pour l'appareil testé. L'appareil de programmation plus petit est toujours restreint à une

limitation de courant d'environ 150 mA, ce qui est probablement une concession à cause de sa plus petite taille.

En ce qui concerne la vitesse de programmation réelle, il n'y a pas d'avantages significatifs avec l'USB-C, car les deux programmeurs fonctionnent toujours « seulement » au débit de l'USB 2.0 High Speed, mais cela ne devrait pas être un problème majeur dans la pratique, étant donné la petite taille des fichiers de la plupart des images de microcontrôleurs. Les optimisations matérielles, en revanche, améliorent la « vitesse perçue ».

Et PoE !

Lors du développement d'applications comprenant des sections de tensions élevées, il est important d'assurer une bonne isolation galvanique entre l'étage haute tension et le reste du circuit, y compris le processeur, les signaux basse tension et les signaux des capteurs. L'un des avantages de l'utilisation d'une liaison de communication telle qu'un réseau Ethernet est qu'il utilise généralement des transformateurs d'isolation à chaque connecteur de réseau pour assurer l'isolation galvanique des signaux de données. Cela permet d'assurer une bonne isolation entre votre station de travail ou votre PC coûteux et tout appareil externe.

La dernière version de l'ICD 5 comporte un port réseau à l'arrière (figure 1). Ce qui est nouveau, c'est qu'il peut désormais alimenter « l'ensemble de l'application » par le biais de l'alimentation par Ethernet (PoE), s'il est configuré de manière appropriée dans votre réseau PoE. Cela peut fournir un courant conséquent au circuit de l'application, et signifie que vous n'aurez pas besoin d'un adaptateur secteur supplémentaire pour l'alimenter.

Heureusement, MPLAB X est également prêt à communiquer directement avec l'ICD 5 via Internet, de sorte qu'il n'est pas nécessaire de câbler un câble USB-C supplémentaire entre votre PC et le programmeur.



Figure 1. Cette connexion permet également l'alimentation par Ethernet.

Figure 2-1. System Power Supply Control Using the PIC16F15244 Family of Microcontrollers

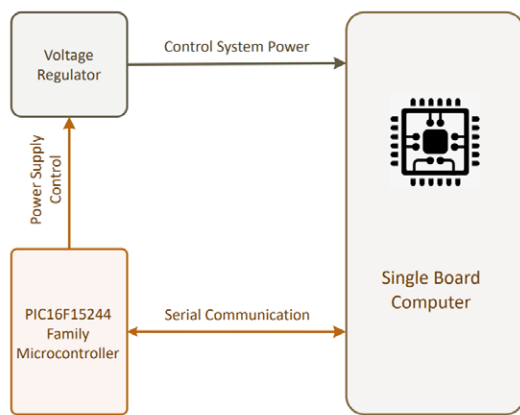


Figure 2. Même un microcontrôleur de gestion nécessite un micrologiciel !
(Source : [3])

Il est important de souligner que cette configuration favorise également l'expérimentation de l'intégration et de la distribution continues, bien que les anciennes versions du programmeur offraient des fonctionnalités similaires lorsqu'elles étaient utilisées avec une connexion Ethernet et un adaptateur secteur 9 V.

Le bloc d'alimentation fourni avec les versions antérieures du kit de développement ne fait plus partie de l'ensemble dans la version 5. Cela indique que l'alimentation peut également être fournie grâce à la capacité d'alimentation plus élevée des ports USB-C ou USB 3.0, par rapport aux versions USB précédentes.

PICKit 5 : fonctionnalités améliorées

Le développement d'un système électronique n'est souvent que la première partie d'une chaîne de valeur complexe. Ceux qui ne prennent pas en compte la « conception en vue de la production », et d'autres aspects similaires, peuvent être confrontés à des coûts élevés inattendus dès le départ ou à des problèmes au cours de la fabrication.

Un bon exemple sont les systèmes qui combinent un « microcontrôleur de gestion » avec un ordinateur de traitement, tels que décrits dans la note d'application AN4121 [3] et résumé par le schéma à la **figure 2**. Une mauvaise planification, par exemple un mauvais positionnement du port de programmation du microcontrôleur, peut avoir un impact significatif sur les coûts d'assemblage et de maintenance du produit. Si les développeurs peuvent gérer le téléchargement de micrologiciels à chaque carte individuelle d'un système prototype à l'aide de MPLAB, cette méthode devient longue et irréalisable lorsque l'appareil entre en production.

Avec le PICKit 5 et sa fonction *Programmer-To-Go*, Microchip offre une solution. En fait, cette fonction est disponible depuis plusieurs versions, et le secret réside dans l'emplacement microSD illustré à la **figure 3**, qui accueille une carte mémoire formatée en FAT32.

En utilisant MPLAB, les développeurs peuvent définir une image de micrologiciel à déployer automatiquement, et l'adaptateur de programmation peut alors la fournir aux appareils cibles qui sont connectés avec un minimum d'effort.

Ce qui est également nouveau avec le PICKit 5, c'est la possibilité de copier plus d'une image de micrologiciel dans la zone de mémoire du *Programmer-To-Go*. Cela permet d'utiliser le programmeur pour le « paramétrage ou l'assemblage final » de plusieurs modules. Le processus de programmation est lancé par le bouton habituel caché sous le logo à l'avant de l'appareil, de sorte que la sélection de l'image doit se faire via une autre interface de communication.



Figure 3. Le logement de la carte mémoire.

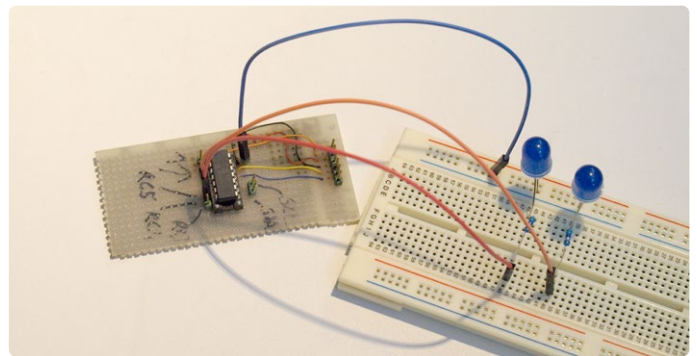


Figure 4. Cette ancienne carte servira de système de test.

...et amis de la norme radio Bluetooth

Pour configurer les nouvelles unités de programmation, vous aurez besoin de l'environnement de développement MPLAB version 6.10 ou supérieure, le processus d'installation en soi est assez standard, comme vous pouvez l'attendre de MPLAB. Outre l'installation classique du compilateur XC8, que Microchip gère séparément, soyez attentifs lors du démarrage initial de MPLAB, à lui accorder l'accès aux réseaux locaux et distants dans le pare-feu Windows. Ceci est particulièrement important si vous prévoyez d'utiliser la version la plus grande du programmeur avec sa connexion Ethernet.

Pour l'étape suivante, vous aurez besoin d'une carte SD. J'ai utilisé une carte de sauvegarde de 16 Go et je l'ai formatée avec une partition FAT32 sous Ubuntu. En recherchant une fonction particulière de la gestion de disques, vous voudrez qu'une des particularités de l'application est qu'elle ne crée pas de partition par défaut, vous devrez donc le faire séparément.

Pour cette configuration, le banc de test matériel que j'utiliserai est une carte à base de contrôleur PIC16F1503 provenant d'un récent projet de consulting pour l'industrie automobile (**figure 4**).

Dans la première étape, comme d'habitude, choisissez l'option *File* → *New Project* pour générer un projet pour le compilateur XC8. Ensuite, démarrez le configurateur de code (MCC) pour générer un squelette de projet de base. Dans cette description, je suppose que vous êtes déjà familiarisé avec la programmation des PIC, je ne vous donnerais donc pas les détails de la manipulation générale.

Non compatible avec Windows 7

Il est important de savoir que Microchip déconseille explicitement l'utilisation de la cinquième version de ses programmeurs avec Windows 7. J'ai utilisé des ordinateurs fonctionnant sous Windows 10 ou 11 pour cet article. Cependant, certaines rumeurs circulent sur l'internet et prétendent que cela « fonctionne », du moins lorsque le programmeur peut être régulièrement connecté à un ordinateur fonctionnant sous Windows 10 ou Windows 11. MPLAB effectuant alors diverses mises à jour du micrologiciel et des tâches de maintenance dans le cadre d'une séquence de test.

Le seul point crucial est que deux broches GPIO doivent être déclarées comme des sorties. Dans les étapes suivantes, je définirai les broches RC4 et RC5 comme des sorties, puis le code sera généré. Pour faciliter la compréhension des différences, j'écrirai un programme suivant un modèle spécifique dans cette structure d'exemple de projet, appelé *RunMeQuick1*. Ce programme allumera les LED ensemble à un rythme relativement lent.

```
void main(void)
{
    // initialize the device
    SYSTEM_Initialize();

    // When using interrupts, you need to
    // set the Global and Peripheral Interrupt Enable bits
    // Use the following macros to:

    // Enable the Global Interrupts
    //INTERRUPT_GlobalInterruptEnable();

    . . .
    IO_RC4_SetHigh() ;
    IO_RC5_SetHigh() ;
    while (1)
    {
        IO_RC4_Toggle() ;
        IO_RC5_Toggle() ;
        __delay_ms(1000);
    }
}
```

Dans l'étape suivante, ouvrez la vue *Project Properties*, où vous trouverez les paramètres permettant de configurer le fonctionnement de *Programmer-To-Go*, comme le montre la **figure 5**. L'aspect le plus important ici est la section *Image Name*, qui est le nom que MPLAB attribue au fichier écrit. L'option *Send image to Tool* (Envoyer l'image à l'outil) indique à l'IDE de charger l'image générée sur la carte microSD du programmeur connecté. Cocher la case *Program Device* est facultative. Car en plus de la fonction *Programmer-To-Go* « actuelle », elle détermine si un microcontrôleur PIC connecté doit être programmé avec l'image fournie. L'utilisation de cette option peut s'avérer utile, par exemple, lorsque vous souhaitez mettre à jour l'image du *Programmer-To-Go*, tout en chargeant

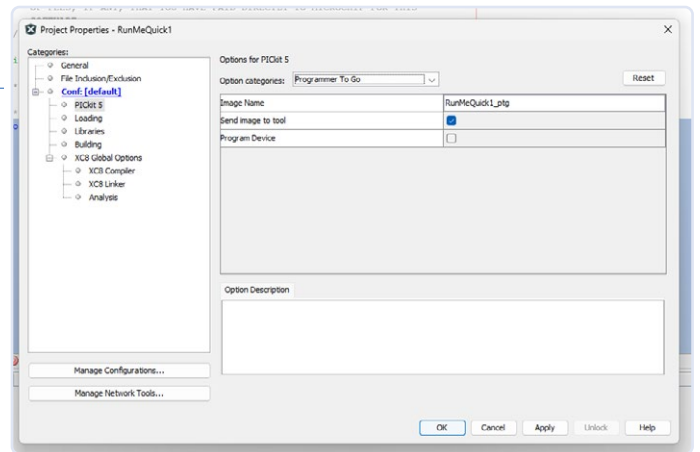


Figure 5. MPLAB assiste les développeurs en configurant le mode *Programmer-To-Go*.

le micrologiciel actuel sur une carte d'évaluation en vue d'un test final. Le déploiement proprement dit doit ensuite *impérativement* être effectué à l'aide du menu illustré à la **figure 6**, qui lance une « mise à jour » de l'image du *Programmer-To-Go*.

Lors du déploiement, il arrive qu'un message d'erreur apparaisse, du type « *Transmission on endpoint 2 failed (err = -109)* ». Dans mon cas, ce problème a presque toujours été résolu en redémarrant le PC. D'ailleurs, si vous venez d'installer MPLAB, il est également recommandé de redémarrer à ce stade. Le déploiement est terminé lorsque la console d'état affiche le message « *The debug tool is in programmer to go mode* ».

Nous allons maintenant retourner à la page d'accueil MPLAB dans cette prochaine étape pour créer un autre projet, quasiment identique, nommé *RunMeQuick2*. Cela permettra d'illustrer les améliorations apportées à la deuxième génération de PICkit 5. Tout d'abord, nous devons redémarrer le MCC à nouveau. Le code destiné à contrôler les LED de signalisation ressemble maintenant à ceci :

```
void main(void)
{
    . . .

    IO_RC4_SetLow() ;
    IO_RC5_SetHigh() ;
    while (1)
    {
        IO_RC4_Toggle() ;
        IO_RC5_Toggle() ;
        __delay_ms(1000);
    }
}
```

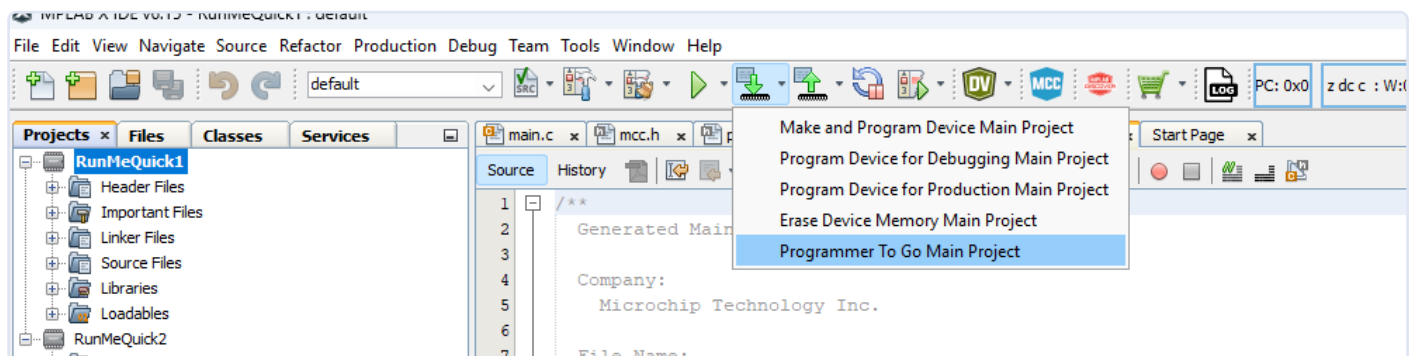


Figure 6. Cette entrée transfère l'image vers la carte SD.

To configure the connected PICkit, logically, you'll need to perform another execution, which should also conclude with the status message *The debug tool is in programmer to go mode*.

Pour configurer le PICkit connecté, logiquement, vous devrez effectuer une autre commande, qui devrait également se terminer par le message d'état « *The debug tool is in programmer to go mode* ». Dans l'étape suivante, vous pouvez déconnecter le programmeur de votre poste de travail. À ce stade, j'ai utilisé un chargeur de téléphone ordinaire pour alimenter le PICkit. La LED verte clignotante indique que l'appareil est en mode « *Programmer-To-Go* » et qu'il attend des instructions.

Nous pouvons maintenant ouvrir le Play Store d'Android ou l'App Store d'Apple pour télécharger la nouvelle application de commande, qui n'est compatible qu'avec le module Bluetooth du PICkit 5. Dans les étapes suivantes, j'utiliserai un smartphone Samsung, et l'application apparaît dans le Play Store comme le montre la **figure 7**.

Lorsque vous utilisez Bluetooth LE, notez que le programme demandera diverses autorisations lors de son lancement initial. Cela est nécessaire en raison des exigences de Google et n'est pas critique ou contournable de quelque manière que ce soit.

À l'étape suivante, la boîte de dialogue de recherche apparaît, indiquant tous les équipements trouvés à proximité. Dans mon cas, en effectuant les tests, j'ai constaté qu'il était parfois nécessaire de cliquer sur le bouton « Annuler » avant de pouvoir accéder à la liste des périphériques de programmation trouvés.

Une fois les tâches terminées, le système affiche les écrans illustrés dans les **figures 8 et 9**, ce qui vous permet de déployer directement le micrologiciel choisi sur les appareils cibles connectés.

De nos jours, le smartphone est l'appareil le plus couramment utilisé qui fonctionne avec une interface graphique familière. L'exécution de l'application sur un tel équipement donne aux débutants une longueur d'avance par rapport à ceux qui partent de zéro en travaillant avec l'explorateur de projet souvent plus complexe de MPLAB sur un ordinateur de bureau. Il convient de noter qu'un smartphone est presque toujours à portée de main, de sorte que la station de travail (relativement plus puissante) n'est plus vraiment nécessaire, et cela peut réduire les coûts de manière significative, en particulier dans les grands déploiements.

Amélioration de mon environnement de développement

L'expérience suivante consiste à revenir sur ma station de travail Windows 10, que j'ai notamment utilisée pour écrire le manuel référencé dans l'encadré ci-dessous. Par défaut, la version 5.45 de MPLAB est installée sur cette machine, que j'utilise habituellement pour mes activités commerciales.

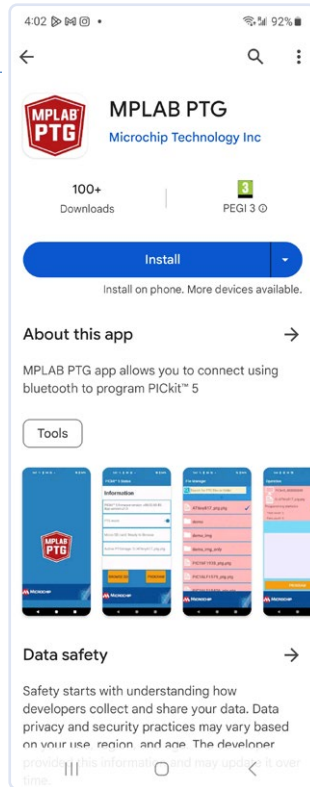


Figure 7. Cette application facilite la programmation à distance.

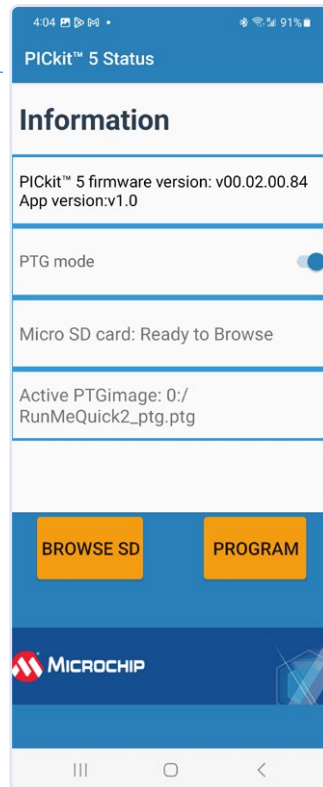


Figure 8. Le menu de démarrage...

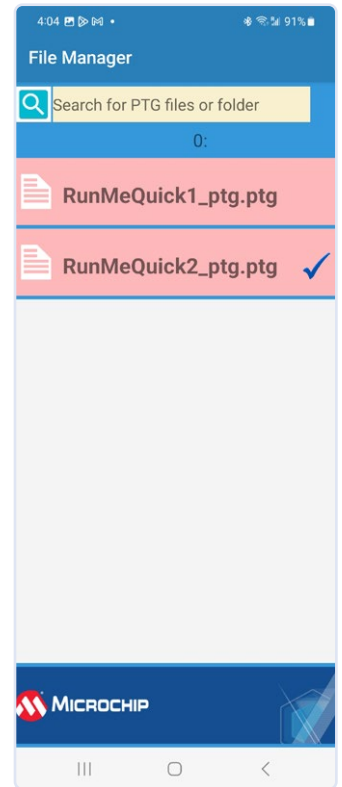


Figure 9. ...et une sélection d'images déployées.

Maintenant, il est temps de visiter à nouveau le site web de MPLAB, où j'ai téléchargé la version 6.15. La dernière version de MPLAB peut être installée directement sur n'importe quelle version déjà existante. Une fois le processus d'installation terminé avec succès, MPLAB 6.15 propose d'utiliser les paramètres de la version précédente comme base de configuration de la nouvelle installation. Comme toujours, vous devrez confirmer certains paramètres dans le pare-feu Windows. L'IDE proprement dit démarre après avoir téléchargé le composant *Microchip Offline Help*.

Lors du démarrage initial de l'IDE, un processus d'analyse approfondie est lancé, qui entre autres tâches, vise à mettre à jour tous les fichiers de projet. En plus, il met à jour divers caches créés dans MPLAB X et les remplit avec des informations provenant des projets.

Pour l'étape suivante, j'ai choisi d'installer l'exemple de démonstration *CH9-Demo1* et j'ai connecté la carte, bien connue de mon manuel, au programmeur et au PC. La compilation s'est déroulée avec succès, mais le déploiement du micrologiciel échoue parfois sous Windows 10 avec l'erreur mentionnée précédemment « *Transmission on endpoint 2 failed (err = -109)* ». En dehors de cela, la nouvelle version de PICkit 5 s'intègre parfaitement dans mon flux de développement existant, et grâce aux optimisations en effet, le déploiement du code est souvent plus rapide dans de nombreux cas.

Mesurer le débit à l'aide de ICD5

Depuis quelque temps, Microchip améliore son IDE MPLAB en y ajoutant diverses fonctions pratiques, afin de permettre aux développeurs de visualiser plus facilement les données de mesure et de suivi générées par l'application intégrée.

L'une des caractéristiques intéressantes de l'ICD 5, est qu'il peut fournir à ce moteur de visualisation de données de MPLAB, des informations sur la consommation d'énergie du circuit d'application connecté. La spécification indique une résolution de mesure de courant de 0,29 µA, plus d'informations de mesure pour le contrôle du courant sont données dans le **tableau 1**.

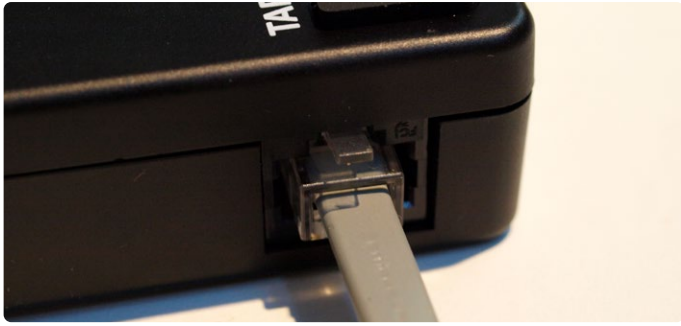


Figure 10. Un petit inconvénient lors de l'utilisation du ICD 5.

Table 1. La précision du contrôle d'énergie, selon Microchip.

Current and Voltage	Resolution	Full Scale
Current	29 μ A / step	1.0 A
Voltage	0.2087 mV / step	6.8 V

Nous pouvons maintenant cliquer sur *Tools Plug-in Downloads* et choisir *Microchip Plug-in Manager* dans la fenêtre qui s'affiche. Vérifiez si *MPLAB Data Visualizer* est déjà listé dans la partie *Installed*. C'est généralement le cas lors d'une nouvelle installation.

Logiquement, vous devrez alimenter le matériel cible à l'aide de la source d'alimentation intégrée de l'ICD 5. Le connecteur utilisé dans l'ICD5 est de type RJ45, alors que les adaptateurs plus anciens tels que l'ICD 3 ont un câble RJ11. Comme le montre la **figure 10**, le connecteur établit en principe un bon contact, mais ce n'est pas le connecteur le plus robuste ou le plus sécurisé.

Pour gagner du temps et de l'effort, nous pouvons utiliser les exemple de LED créés précédemment pour le PICKit pour tester la fonction de mesure de courant de l'ICD 5. Déconnectez le PICKit et connectez l'ICD5. Par souci de commodité, j'éviterai le câble réseau avec ses fonctions TCP/IP et je me connecterai plutôt via un câble USB-C.

Notez que pendant la configuration « initiale » d'un ICD, MPLAB doit effectuer une mise à jour du micrologiciel pour le FPGA, et ce processus unique prend un certain temps..

Il est intéressant de noter que ce processus s'arrête souvent à 93% et continue ensuite « comme d'habitude » (**figure 11**). La raison pour laquelle MPLAB n'indique jamais avoir atteint un état de mise à jour de 100% ici n'est pas tout à fait claire.

A l'étape suivante, cliquez sur *Window Debugging Data Visualizer* pour lancer l'écran de bienvenue de *Data Visualizer*. Ensuite, dans la section *Power*, cliquez sur le bouton *Play* pour commencer l'acquisition des données.

MPLAB répond en affichant la fenêtre de la **figure 12**, indiquant qu'aucun formulaire de visualisation n'a été spécifié.

En réponse, cliquez sur la fonction *Plot Raw*, qui affiche maintenant un graphique. Si vous exécutez le programme qui fait clignoter alternativement les LED, vous obtiendrez des informations sur la consommation d'énergie, comme le montre la **figure 13**.

Des résultats plus intéressants peuvent être obtenus en utilisant *RunMeQuick1*, cet exemple allume et éteint alternativement les LED (**figure 14**).

Pour résumer

Microchip a introduit un certain nombre d'améliorations détaillées avec le PICKit 5 et l'ICD 5, qui aideront les développeurs à la fois dans la phase de débogage et de « production en série » du cycle de vie d'un projet. Les moteurs de recherche de pièces électroniques tels que *oemsecrets.com* sont capables d'identifier les meilleurs prix et niveaux de stock en temps réel pour les composants de divers distributeurs. Le MPLAB PICKit 5 In-Circuit Debugger/Programmer est vendu à 86 €, tandis que le MPLAB ICD 5 In-Circuit Debugger/Programmer est disponible à 360 €, ce qui n'est pas du tout trop cher pour ce qui est proposé. Les utilisateurs du PICKit 4 qui se sentent frustrés par l'*ancienne* interface souhaitent peut-être passer à la nouvelle variante pour réduire le nombre de câbles. La fonction d'analyse de la puissance est également très utile et peut potentiellement permettre de réaliser des économies significatives par rapport à une carte SMU ou GPIB. ◀

VF : Laurent Rauber - 230571-04

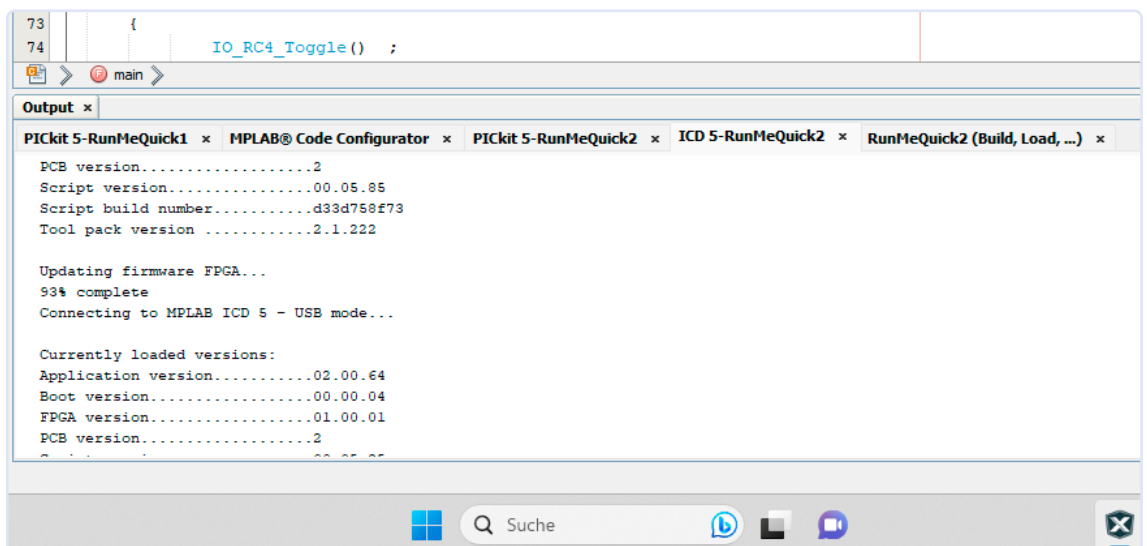


Figure 11. 93% indique ici que la tâche ici est entièrement achevée !

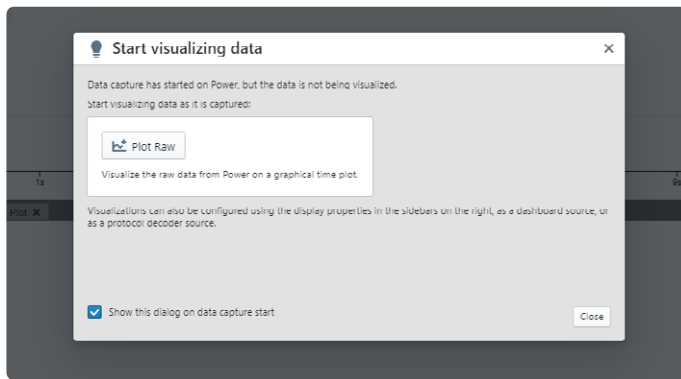


Figure 12. MPLAB demande comment les données capturées doivent être représentées ou visualisées.

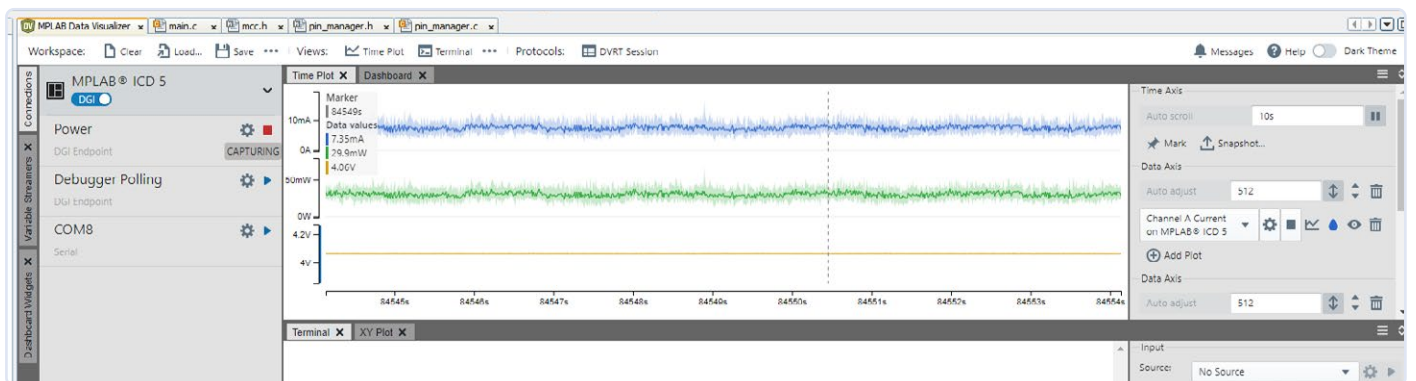


Figure 13. Ici, il y a toujours une LED allumée.

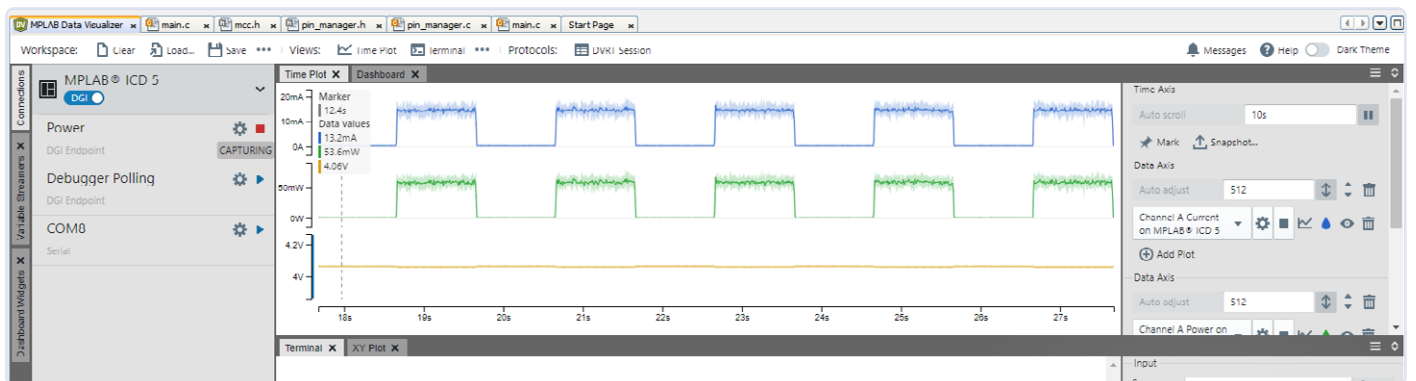


Figure 14. L'exemple montre clairement les changements dans la consommation en courant.

Demandez votre code de réduction de 15 % pour ces outils dès aujourd'hui !

Indiquez vos coordonnées, sélectionnez l'outil pour lequel vous souhaitez obtenir un bon de réduction de 15 % et vous recevrez un code de réduction unique à utiliser chez MicrochipDirect.

https://page.microchip.com/pic5_icd5



LIENS

- [1] MPLAB PICKit 5 de Microchip : <https://microchip.com/en-us/development-tool/pg164150>
- [2] MPLAB ICD5 de Microchip : <https://microchip.com/en-us/development-tool/dv164055>
- [3] Note d'application AN4121 : <https://microchip.com/en-us/application-notes/an4121>