

Démarrer avec les outils de développement de Microchip

Sommaire

Getting Started with Development Tools	2
ÉTAPE 1 - UN APERÇU DES SYSTEMES INTEGRES	3
Découvrons ensemble l'outil « Microchip Advanced Part Selector » (MAPS).....	5
ÉTAPE 2 - COMMENCER DÈS MAINTENANT AVEC LES OUTILS DE DÉVELOPPEMENT MICROCHIP	6
ÉTAPE 3 - Mettre en œuvre un system embarqué avec MPLAB IDE	10
ÉTAPE 4 - LE CYCLE DE DÉVELOPPEMENT.....	12
ÉTAPE 5 - PROJECT MANAGER	13
ÉTAPE 6 - LANGUAGE TOOLS	14
ÉTAPE 7 – DEBOGAGE DE LA CIBLE	15
ÉTAPE 8 - PROGRAMMATION	17
ÉTAPE 9 - PLUS D'INFORMATION sur MPLAB IDE	20



Getting Started with Development Tools



MPLAB Starter Kit pour PIC24H



PICKIT 3 Debug Express

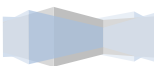
MPLAB[®] REAL ICE™

Les outils de développement Microchip sont une suite de logiciels et matériels pour vous aider à concevoir des applications utilisant composants Microchip. Pour vous guider à travers les divers outils fournis par Microchip, nous présentons cette brève introduction aux outils logiciels et matériels. Sur le panneau de gauche du site www.microchip.com sont affichées différentes sections ainsi que des ressources qui font partie de cette synthèse.

Notez la section intitulée "Development Tools Made Easy", vous pouvez parcourir ces rubriques séquentiellement ou voir l'un d'eux pour obtenir une compréhension des outils de développement Microchip.

Dans le volet de gauche vous trouverez aussi la rubrique "Tools: A Closer Look" vous permettant de découvrir des tutoriaux complémentaires "Start Now" couvrant des projets MPLAB, MPLAB Editor's, Simulator et autres composants de MPLAB IDE.

Enfin dans la partie inférieure de ce volet, des liens sont disponibles vers la documentation indispensable, les manuels des outils de développement, des livres, des groupes de discussion en ligne, et des outils pour les produits Microchip proposés par d'autres sociétés. Mais également d'autres outils, tels que le [sélecteur d'outil de développement](#), un outil facile pour faire correspondre chaque PIC[®] MCU et dsPIC[®] DSC aux outils de développement appropriés, des connecteurs et adaptateurs.



ÉTAPE 1 - UN APERÇU DES SYSTEMES INTEGRES

MPLAB[®] IDE est un logiciel qui s'exécute sur un PC pour développer des applications pour les microcontrôleurs de Microchip. Il est appelé environnement de développement intégré, ou IDE, car il offre un "environnement" intégré unique pour développer du code embarqué pour microcontrôleurs.

Description d'un « Embedded System »

Un système embarqué est généralement un dessin ou modèle faisant usage de la puissance d'un microcontrôleur, comme le MCU Microchip[®] PIC ou dsPIC[®] Digital Signal Controller (DSC). Ces microcontrôleurs combinent une unité à microprocesseur (comme le processeur dans un PC de bureau) avec quelques circuits supplémentaires appelés "périphériques" sur la même puce. Cet élément unique peut ensuite être incorporé dans d'autres appareils électroniques et mécaniques à faible coût pour la commande numérique.



PIC MCU

Différences entre un contrôleur embarqué et un PC:

Embedded Controller	PC
Dédié à une tâche spécifique	Peut faire beaucoup de tâches différentes
Formé d'un seul programme, à la suite, peut être fait à peu de frais pour inclure juste assez de puissance informatique et le matériel pour réaliser cette tâche dédiée	Capable d'exécuter de nombreux types de programmes différents et qui peuvent être reconfigurés pour accomplir chaque tâche de manière optimale
Le programme est généralement inscrit de façon permanente dans la mémoire, lors de la fabrication du MCU	Le programme est chargé en mémoire lors de son appel, généralement à partir du disque dur.
Dispose d'une unité microcontrôleur (MCU) à faible coût, avec de nombreux périphériques sur la même puce, et relativement peu de périphériques externes	Formé d'une unité centrale de traitement (CPU) relativement coûteuse, avec de nombreux autres composants externes (mémoire, disques durs, contrôleurs vidéo, circuits d'interface réseau, etc)
Souvent, un système embarqué est une partie invisible, ou sous-module d'un autre produit, comme une perceuse sans fil, un réfrigérateur ou un ouvre-porte de garage. Le contrôleur de ces produits permet le fonctionnement d'une infime partie de la fonction de l'ensemble du dispositif. Le contrôleur ajoute de l'intelligence à faible coût pour certains des sous-systèmes critiques dans ces dispositifs.	Habituellement, un dispositif important, faisant souvent partie d'un réseau, utilisé pour le traitement de texte, feuilles de calcul, présentations, interface internet, interface graphique pour PC à base de différents outils.

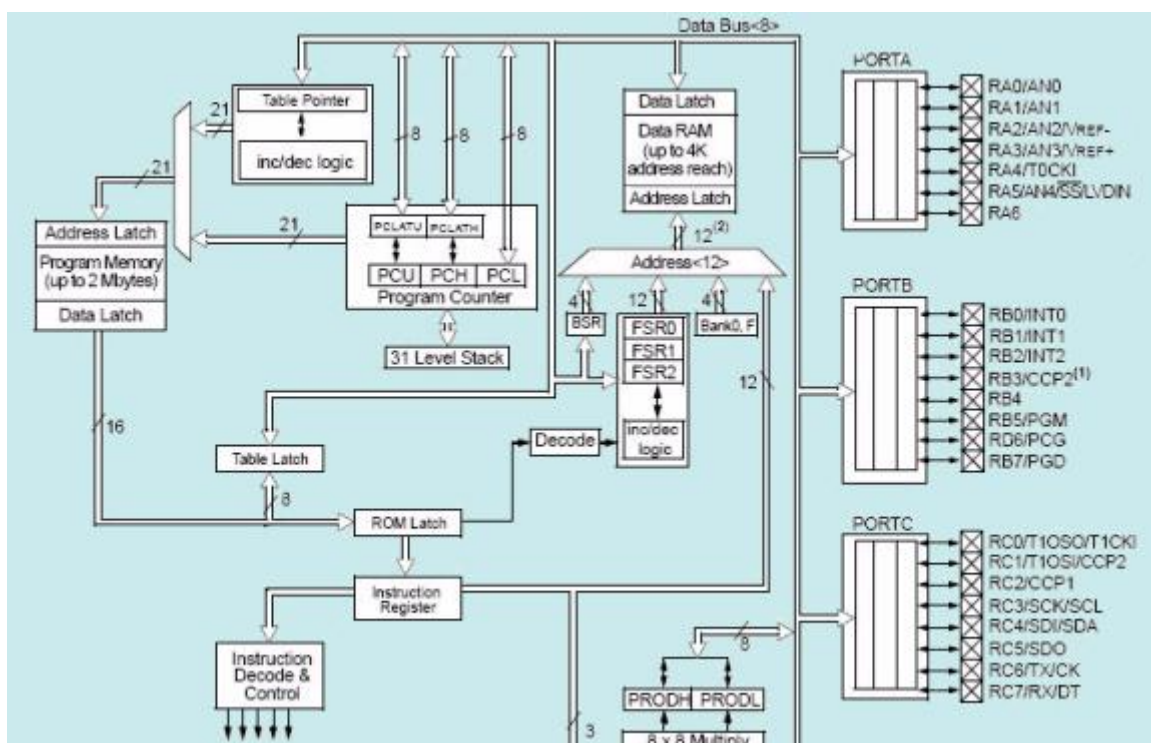


Un exemple d'un système embarqué est un détecteur de fumée. Sa fonction est d'évaluer les signaux d'un capteur et déclenche une alarme sonore si les signaux indiquent la présence de fumée. Un petit programme dans le détecteur de fumée fonctionne soit dans une boucle infinie (échantillonnage du signal du capteur de fumée), ou dans un état latent " mode veille", qui pourra être réveillé par un signal du capteur. Le programme aurait peut-être quelques autres fonctions, comme une fonction de test utilisateur, et une alerte de batterie faible. Même si un PC équipé d'un capteur et sortie audio peut être programmé pour faire la même fonction, il ne serait pas une solution rentable (ni ne pourrait fonctionner sur une pile de neuf volts sans surveillance pendant des années!) Embedded concepts utilisent des microcontrôleurs peu coûteux, pour mettre de l'intelligence dans des choses de tous les jours de notre environnement, tels que les détecteurs de fumée, caméras, téléphones cellulaires, appareils ménagers, automobiles, cartes à puce et systèmes de sécurité.

Composition d'un microcontrôleur

Le MCU PIC dispose d'une mémoire de programme pour le firmware, où les instructions codées, permettent d'exécuter un programme. Il a aussi des registres mémoires pour stocker les variables que le programme aura besoin pour le calcul ou le stockage temporaire. Il possède également un certain nombre de circuits périphérique sur la même puce.

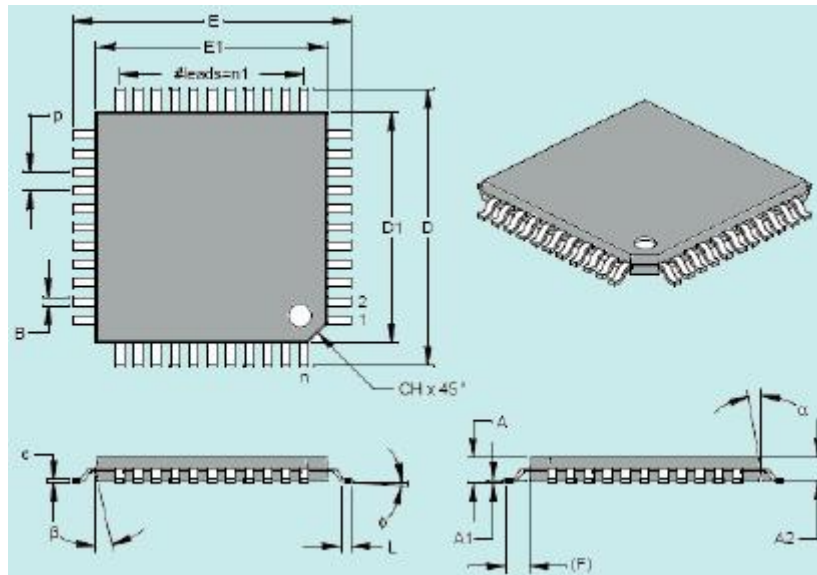
Certains composants sont appelés périphériques d'Entrée/Sortie. Les Ports I/O sont des broches du microcontrôleur qui peuvent être à l'état haut ou bas, permettant d'envoyer des signaux, faire clignoter des leds, produire du son sur un haut parleur, à peu près tout type de signaux qui peut être envoyé par un fil. Souvent, ces broches sont bidirectionnelles et peut également être configuré comme entrée permettant au programme de répondre à une commutation externe, un capteur ou de communiquer avec certains périphériques externes.



Synoptique d'un microcontrôleur PIC

Afin de concevoir un tel système, il faut décider des périphériques qui sont nécessaires pour une application. Les convertisseurs analogique-numérique permettent aux microcontrôleurs d'être connectés à des capteurs et de percevoir l'évolution des niveaux de tension. Les périphériques de communication série vous permettent de mettre en œuvre les communications sur seulement quelques fils vers un autre microcontrôleur, un réseau local ou via Internet. Les périphériques du microcontrôleur PIC appelés "Timers" mesurer avec précision les événements temporels de signaux, permettent de produire des formes d'onde précises, et même de réinitialiser automatiquement le microcontrôleur si une défaillance matérielle est détectée ou en raison d'un problème d'alimentation. D'autres périphériques sont capables de détecter si l'alimentation atteint un niveau dangereusement faible de sorte que le microcontrôleur peut stocker des informations critiques et s'éteindre en toute sécurité avant que celles-ci ne soient complètement perdues.





Dimensions du package d'un microcontrôleur PIC

Le nombre de périphérique et la quantité de mémoire qu'une application a besoin pour exécuter un programme sont autant d'éléments qui déterminent en grande partie le microcontrôleur PIC à utiliser. D'autres facteurs pourraient comprendre la puissance consommée par le MCU et son «facteur de forme », autrement dit, la taille et les caractéristiques du package physique qui doit résider sur la carte cible à concevoir.

Découvrons ensemble l'outil « Microchip Advanced Part Selector » (MAPS)

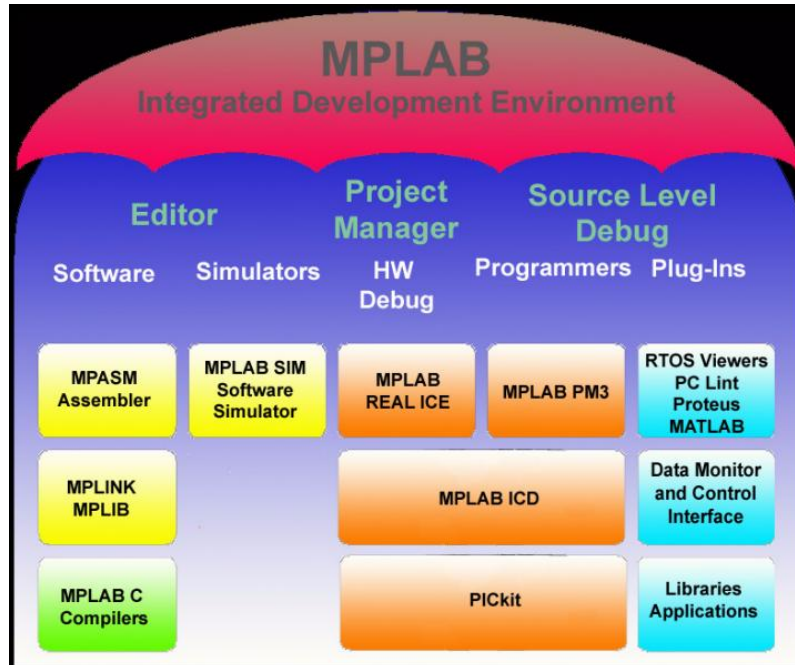
The screenshot shows the Microchip Advanced Part Selector (MAPS) interface. On the left, there are search filters for Family, Prefix, P Memory, RAM, EEPROM, IO Pins, Max CPU Speed, Int. OSC, Comparator, A/D Ch., DIA Ch., UART Ch., SPI, PC™, CAN Ch., USB Ch., Input Capture, and PWM Ch. The 'USB' option is selected in the 'Connectivity' section. The search results on the right show 510 MCHP Parts found, with the PIC32MX895F512L selected. The selected part's specifications are displayed, including P Memory (512 Flash), RAM (128K), Max CPU Speed (80 MHz), and various peripheral features like UART, SPI, and I2C.

L'ensemble des références apparaît dans la fenêtre « Search Results » avec environ 510 microcontrôleurs référencés. Nous allons choisir un MCU USB en boîtier 40 broches, pour cela : Développer la fenêtre « Connectivity » et choisissez l'option l'USB, le nombre de MCU correspondant à ce critère est de 65. Puis choisir dans « Pkge Pins » From 40 - Thru 40 , le choix n'est plus alors que de 5 MCH. Sélectionner le modèle PIC18F4550, puis cliquer sur « Add PIC 18F4550 », faite de même avec la référence 18F4553, puis cliquer sur « Go to side-by-side »

En choisissant le PIC 18F4550, il y a une carte d'évaluation [PICDEM Full Speed USB](#) qui permet de réaliser quelques projets contenu dans « [Microchip Applications Libraries](#) »

ÉTAPE 2 - COMMENCER DÈS MAINTENANT AVEC LES OUTILS DE DÉVELOPPEMENT MICROCHIP

Microchip a un grand ensemble d'outils de développement logiciels et matériels intégrés au sein d'un logiciel appelé **MPLAB Integrated Development Environment**. D'autres sociétés fournissent également des outils qui fonctionnent dans ce cadre. Voici un aperçu "parapluie" de MPLAB IDE et les catégories d'outils logiciels et matériels qui y sont intégrés:



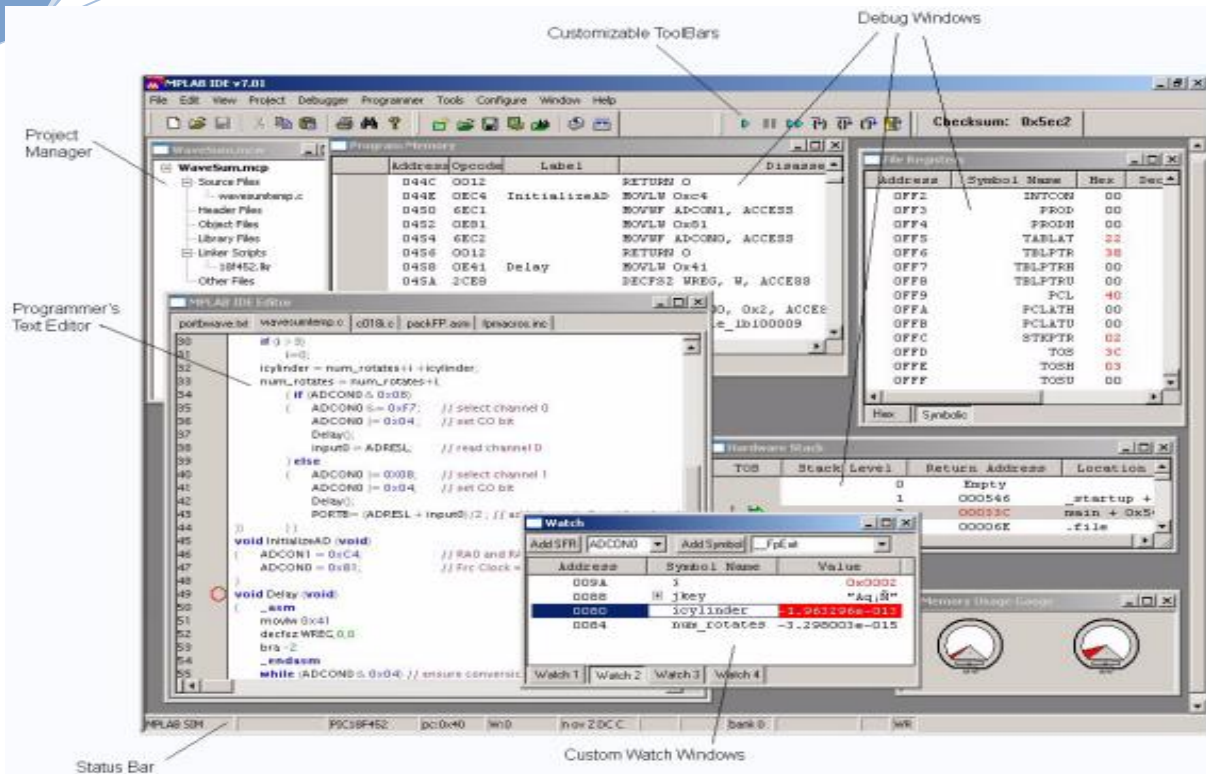
Construit dans MPLAB IDE, un éditeur de programmeur, un gestionnaire de projet et de puissants outils de débogage. D'autres outils libre sont des éléments standard de MPLAB IDE, comme le montrent les blocs jaunes, assembleur MPASM, MPLINK / Mplib éditeur de liens et de bibliothèque, ainsi que MPLAB SIM, le simulateur logiciel. Les compilateurs C MPLAB (bloc vert) sont vendus séparément et fonctionnent de façon transparente au sein de cet environnement. Les outils matériels sont indiqués en rouge et comprennent les programmeurs et les débogueurs ainsi que des émulateurs de circuit complexes. Notez que MPLAB ICD et PICkits permettent d'avoir des attributs de débogage à faible coût, Emulation / analyse et de programmation de périphériques dans un seul outil. Les plug-ins (en bleu) comprennent des tierces partis des compilateurs C, systèmes d'exploitation temps réel, Surveillance de données et contrôle d'interface graphique, ainsi que des bibliothèques qui peuvent être liés à vos propres applications spécifiques.

Microchip Software Tools

MPLAB[®] IDE - L'environnement de développement intégré sert d'application principale à partir de laquelle d'autres outils sont déployés.

MPLAB IDE est une application MS Windows qui vous aide à écrire du code qui s'exécutera dans une application embarquée, et ensuite vous permettre de tester l'application avec différents logiciels et composants matériels.





Télécharger [MPLAB IDE](#) et utiliser [le manuel de démarrage rapide](#) pour voir comment il est facile de créer une application. Ecrire le code assembleur, de construire et d'assembler votre projet avec des assistants MPLAB, puis tester votre code avec les fonctionnalités intégrées dans le simulateur et un débogueur.

Découvrez les capacités de tous les microcontrôleurs Microchip. Lorsque vous êtes prêt à tester votre propre application, l'utilisation d'[MPLAB REAL ICE](#) ou MPLAB ICD 2 ou 3 pour programmer un dispositif et analyser votre matériel, ou choisissez PICSTART Plus ou MPLAB PM3 programmeurs pour programmer votre code.

Logiciel gratuit inclus dans MPLAB IDE	
Programmer's Text Editor	Utilisez l'éditeur pour écrire du code source et le sauvegarde. Le code source peut être destiné à l'assembleur ou aux compilateurs C.
Project Manager	Permet de grouper à la source des fichiers, puis de les envoyés vers les outils logiciels pour l'assemblage et / ou la compilation.
MPASM™/MPLINK™	Le code écrit dans MPLAB est transformée en "code machine" par l'assembleur en une suite de un et de zéro, qui seront ensuite programmé dans un MCU. L'éditeur de liens prend soin de mettre le code en mémoire, d'attribuer des registres pour les variables, et de s'assurer que chaque module source (généralement écrites dans un fichier texte) peuvent accéder à des fonctions et des variables à partir de modules d'autres sources.
MPLAB SIM	Ceci est le simulateur qui est intégré à MPLAB IDE. Il prend en charge le débogage des PIC MCU et dsPIC DSC. Il utilise votre PC pour simuler les instructions MCU et simule également de nombreuses fonctions des périphériques, y compris la conversion A/D, l'USARTs, et les broches I/O. Des signaux de stimulus peuvent être injectés dans les registres ou sur des broches et des journaux de changements de registre peuvent être sauvegardés dans un fichier texte pour une analyse ultérieure.

Logiciels en option pour MPLAB IDE

MPLAB C Compilers	Compilateurs C conforme à la norme ANSI pour les microcontrôleurs PIC18, PIC24 et PIC32 et dsPIC DSC. C'est une application 32 bits, utilisable avec la console Windows, mais est également entièrement intégré dans MPLAB IDE. Les projets, les options du compilateur et les personnalisations de l'éditeur de liens peut entièrement s'effectuer dans MPLAB IDE, fournissant une interface graphique pour ce puissant compilateur. Modification des erreurs et basculer instantanément vers des lignes correspondantes dans le code source, des fenêtres de visualisation permettent de voir la structure des données avec les types de données définis, y compris en virgule flottante.
--------------------------	---

Pour programmer l'application dans un dispositif, un programmeur est nécessaire:

Programmateurs optionnels à MPLAB IDE

PICSTART Plus	Un programmeur de périphérique bas-coût, soutenant principalement PIC dispositifs DIP.
MPLAB PM3	Un programmeur universel supporter la plupart des PIC et des dispositifs dsPIC et forfaits.
MPLAB ICD	Un débogueur à faible coût / programmeur qui travaille avec de nombreux dernière version de Flash microcontrôleurs PIC et dsPIC DSC.
PICkit développement Programmeur / Debugger	Permet le débogage in-circuit et la programmation des microcontrôleurs PIC ® sélectionné.

Bien que MPLAB SIM puisse être utilisé pour l'essai du code programme, souvent d'autres dispositifs entre en interaction avec le code de manière imprévisible, ou les actions en temps réel des périphériques sont difficiles à simuler avec précision. Un débogueur matériel est utilisé pour exercer le code dans un prototype ou la mise en œuvre finale.

Microchip In-Circuit Emulators and Debuggers

MPLAB REAL ICE	MPLAB REAL ICE est la prochaine génération d'émulateur de Microchip à grande vitesse pour les microcontrôleurs. Il débogue des programmes de microcontrôleurs PIC et dsPIC avec l'interface utilisateur graphique puissante de l'environnement de développement intégré MPLAB IDE.
MPLAB ICD	Certains Flash-based microcontrôleurs PIC et dsPIC DSC ont la logique supplémentaire sur chaque puce qui permet à un débogueur en circuit pour télécharger du code dans la puce, puis de l'exécuter. La logique supplémentaire permet à l'application du firmware de s'arrêter à des points d'arrêt, où l'ingénieur peut voir les registres internes et vérifier l'état de la puce au point d'arrêt. Le code peut être exécuté en mode pas à pas, les variables et les registres peuvent être modifiés afin de déterminer si le code fonctionne correctement. La mémoire programme du microcontrôleur Flash cible est utilisée pendant le débogage. Un circuit de debug permet l'interconnexion même après que le produit soit entré en production, il peut être débogué et reprogrammé après coup.
PICkit Development Programmer/Debugger	Enables in-circuit debugging and programming of selected PIC® microcontrollers.

Démonstration et cartes d'évaluation

Pour démarrer rapidement avec des dispositifs Microchip, envisager d'étudier les nombreuses démo et cartes d'évaluation. Depuis le microcontrôleur à usage général, aux modèles très spécifiques, tels que TCP / IP, ZigBee™ et la communication USB, ces planches sont entièrement fonctionnels, avec un logiciel de démonstration qui peuvent être modifiés et reprogrammés pour se conformer à vos propres besoins. La plupart des cartes ont de nombreuses fonctionnalités, LEDs, commutateurs, potentiomètres, capteurs de température, ainsi que d'une zone de prototypage rapide pour ajouter votre circuit personnalisé. Avec l'émulateur et le débogueur intégré à MPLAB, ces conseils vous permettront d'apprendre les bases, et de passer rapidement à votre propre prototypage de votre application avec un minimum d'efforts.



MPLAB Starter Kits

MPLAB Starter Kits offre un système complet avec une démonstration qui fonctionne dès la sortie de la boîte. Il suffit de brancher sur le port USB d'un PC et de regarder l'application s'exécuter. Un débogueur et programmeur intégré permet d'inspecter l'application du projet visés, de modification et reprogrammées le kit. Ceux-ci fournissent une rapide introduction à une application ou une nouvelle technologie, et sont des systèmes peu coûteux permettant de faire des démonstrations d'une application spécifique.



ÉTAPE 3 - Mettre en œuvre un system embarqué avec MPLAB IDE

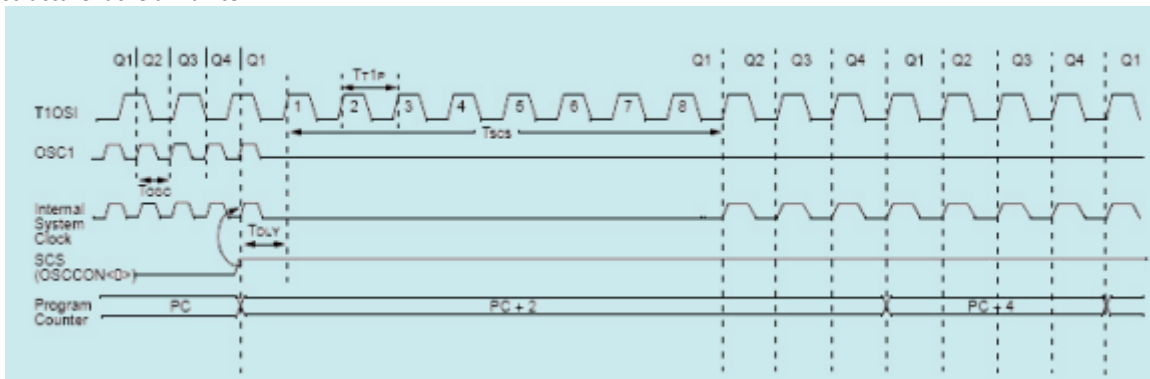
Un système de développement pour les contrôleurs embarqués est un système de programmes exécutés sur un ordinateur de bureau pour aider à écrire, éditer et déboguer le code du programme - l'intelligence des applications des systèmes intégrés - dans un microcontrôleur. MPLAB IDE, fonctionne sur un PC et contient tous les éléments nécessaires pour concevoir et déployer des applications de systèmes embarqués.

Les tâches typiques pour développer une application contrôleur embarqué sont:

1. Créer le design de haut niveau. Depuis les caractéristiques et les performances désirées, décider quel MCU PIC ou dsPIC DSC et le mieux adapté à la demande, puis de concevoir les circuits et matériel annexe.

Après avoir déterminé les périphériques et les broches de commande du matériel, écrire le firmware, c'est le logiciel qui contrôle les aspects matériels de l'application embarquée. L'assembleur est directement traduisible en langage machine, ou un compilateur qui permet à un langage plus naturel de créer des programmes, celui-ci devrait être utilisé pour écrire et éditer du code.

Assembleur et compilateur aidé à rendre le code plus compréhensible, permettant à des fonctions ou routines de code avec des variables qui ont des noms associés à leur utilisation, et avec des constructions qui aident à organiser le code dans une structure facile à maintenir.



Timing Diagram de Data Book

2. Compiler, assembler et lier votre programme en utilisant l'assembleur et/ou un compilateur puis un éditeur de liens pour convertir votre code en «zéros et en uns», code machine pour votre microcontrôleurs. Ce code machine finira par devenir le firmware (le code programmé dans le microcontrôleur).

3. Tester votre code. Habituellement, un programme complexe ne fonctionne pas exactement de la manière imaginée, et des "bugs" doivent être retirés de la conception pour obtenir un résultat convenable. Le débogueur vous permet de voir les "zéros et de uns" exécuté, lié au code source que vous avez écrit, avec les symboles et les noms de fonction dans votre programme.

Le débogage, vous permet d'expérimenter avec votre code pour voir la valeur des variables à différents points dans le programme, et de faire des contrôles, en changeant ces valeurs et en observant en mode pas à pas.

4. "Flashé" le code dans votre microcontrôleur et vérifier qu'il s'exécute correctement dans votre application finale.

Documentation d'une instruction du PIC tiré de la datasheet >>>

RRCF	Rotate Right f through Carry								
Syntax:	[label] RRCF f[d,a]								
Operands:	0 ≤ f ≤ 255 d ∈ {0,1} a ∈ {0,1}								
Operation:	(f←n) → dest←n-1, (f←0) → C, (C) → dest←7								
Status Affected:	C, N, Z								
Encoding:	0011 00da ffff ffff								
Description:	The contents of register T are rotated one bit to the right through the Carry Flag. If 'd' is 0, the result is placed in W. If 'd' is 1, the result is placed back in register T (default). If 'a' is 0, the Access Bank will be selected, overriding the BSR value. If 'a' is 1, then the bank will be selected as per the BSR value (default).								
Words:	1								
Cycles:	1								
Q Cycle Activity:	<table border="1"> <thead> <tr> <th>Q1</th> <th>Q2</th> <th>Q3</th> <th>Q4</th> </tr> </thead> <tbody> <tr> <td>Decode</td> <td>Read register T</td> <td>Process Data</td> <td>Write to destination</td> </tr> </tbody> </table>	Q1	Q2	Q3	Q4	Decode	Read register T	Process Data	Write to destination
Q1	Q2	Q3	Q4						
Decode	Read register T	Process Data	Write to destination						
Example:	RRCF XB0, 0, 0								
Before Instruction									
REG	= 1110 0110								
C	= 0								

Bien sûr, chacune de ces étapes peuvent être assez complexes. L'important est de se concentrer sur les détails de votre propre conception, tout en s'appuyant sur MPLAB IDE et de ses éléments pour obtenir à chaque étape sans se heurter continuellement à de nouvelles courbes d'apprentissage.

L'étape 1 est conduite par le concepteur, même si MPLAB IDE peut aider dans la modélisation des circuits et le code afin que les décisions cruciales de la conception puissent être faites.

MPLAB IDE permet de faciliter les étapes 2 à 5.

- Son éditeur de programme aide à écrire du code correct avec les outils et le langage de votre choix. L'éditeur reconnaît automatiquement le type de langage et colorie le code source pour aider le concepteur à s'assurer de la bonne syntaxique.
- Le gestionnaire de projet vous permet d'organiser les différents fichiers utilisés dans votre application: les fichiers source, les fichiers d'en-tête, de la description du processeur et des fichiers de bibliothèque. Lorsque le code est généré, vous pouvez contrôler de façon rigoureuse que le code sera optimisé pour la taille ou la vitesse par le compilateur et que les variables et les données du programme seront programmés dans le microcontrôleur. Vous pouvez également spécifier un modèle pour la «mémoire» afin de faire le meilleur usage de la mémoire du microcontrôleur pour votre application.
- Si les outils du langage utilisé détecte une erreur lors de la construction du code, la ligne incriminée est affichée et peut être "double-cliquée" pour aller immédiatement sur le fichier source correspondant. Après la modification, appuyez sur le bouton de "construction" pour essayer de nouveau. Souvent, cette boucle écriture-compilation-modification est fait de nombreuses fois pour du code complexe, comme pour les sous fonctions sont écrits et testés. MPLAB IDE passe par cette boucle avec une vitesse maximale, ce qui vous permet d'atteindre l'étape suivante.

Essais en utilisant le debugger et le simulateur

Une fois le code construit sans aucune erreur, celui-ci doit être testé.

Même si le matériel n'est pas encore terminé, vous pouvez commencer à tester le code avec le simulateur, un logiciel qui simule l'exécution du microcontrôleur. Le simulateur peut accepter une entrée simulée (stimulus), afin de modéliser la façon dont le microprogramme répond aux signaux externes. Le simulateur permet de mesurer le temps d'exécution du code, de regarder les variables et les périphériques, et de tracer le code pour générer un compte rendu détaillé de la façon dont le programme a démarré.

Une fois le matériel au stade du prototype, un débogueur matériel, tel que MPLAB REAL ICE ou MPLAB ICD 2 peut être utilisé. Ces débogueurs exécuter le code en temps réel sur votre application réelle.

MPLAB REAL ICE et MPLAB ICD 2 utilisent des circuits intégrés spéciaux dans de nombreux microcontrôleurs Microchip avec une mémoire programme Flash et peut "voir" dans le programme du microcontrôleur cible la mémoire des données. Ils peuvent arrêter et démarrer l'exécution du programme, vous permettant de tester le code avec le micro en place sur la carte.

Après l'exécution correctement de l'application, vous pouvez programmer un microcontrôleur avec l'un des programmeurs Microchip, comme le PICSTART Plus ou MPLAB PM3. Ces programmeurs «Flash» le code final dans les MCU. MPLAB REAL ICE et MPLAB ICD 2, bien que principalement débogueurs, peuvent également être utilisés comme programmeurs.

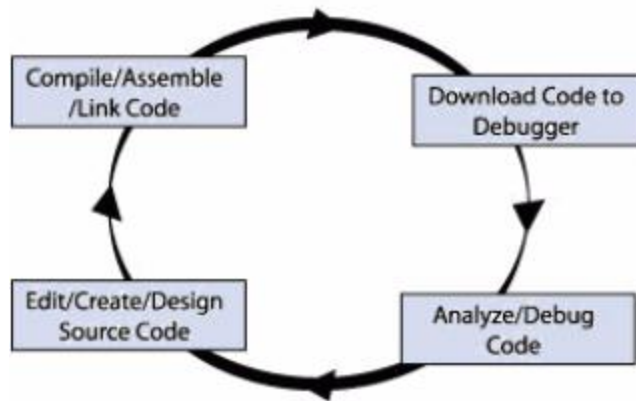
En programmant un appareil, les programmeurs peuvent écrire les bits de configuration de la MCU, des emplacements spéciaux qui déterminent le type d'oscillateur, activer ou désactiver le minuteur de surveillance et de circuits de détection de baisse de tension et le contrôle des paramètres de configuration. Certains microcontrôleurs ont des bits de sécurité qui permettent soit en totalité ou seulement une partie de la mémoire à programmer, une fois "verrouillé", celles-ci ne pourront être lues à nouveau. Cela fournit la sécurité pour le code ainsi que les données qui est programmé dans un dispositif.

REGISTER 19-2: CONFIGURATION REGISTER 2 LOW (CONFIG2L: BYTE ADDRESS 300002h)	
	U-0 U-0 U-0 U-0 R/P-1 R/P-1 R/P-1 R/P-1
	bit 7 bit 6 bit 5 bit 4 bit 3 bit 2 bit 1 bit 0
bit 7-4	Unimplemented: Read as '0'
bit 3-2	BORV1:BORV0: Brown-out Reset Voltage bits 11 = VBOR set to 2.5V 10 = VBOR set to 2.2V 01 = VBOR set to 4.2V 00 = VBOR set to 4.5V
bit 1	BOREN: Brown-out Reset Enable bit 1 = Brown-out Reset enabled 0 = Brown-out Reset disabled
bit 0	PWRTEN: Power-up Timer Enable bit 1 = PWRT enabled 0 = PWRT disabled

ÉTAPE 4 - LE CYCLE DE DÉVELOPPEMENT

Le processus d'écriture d'une application est souvent décrit comme un cycle de développement, comme il est rare que toutes les étapes de la conception à la mise en œuvre peuvent se faire sans faille dès la première fois. Le plus souvent, le code est écrit, testé, puis modifié fréquemment afin de produire une application qui fonctionne correctement.

L'environnement de développement intégré (IDE) permet à l'ingénieur de conception de systèmes embarqués pour progresser dans ce cycle sans la distraction de la commutation entre une palette d'outils. En utilisant MPLAB[®] IDE, toutes les fonctions sont intégrées, permettant à l'ingénieur de se concentrer sur l'achèvement de la demande, sans l'interruption des outils distincts et les différents modes de fonctionnement.



L'IDE est un "wrapper" qui coordonne tous les outils à partir d'une seule interface utilisateur graphique - le plus souvent automatiquement. Par exemple, une fois que le code est écrit, il peut être transformé en instructions exécutables et téléchargées dans un microcontrôleur pour voir comment il fonctionne. Dans ce processus, plusieurs outils sont nécessaires: un éditeur pour écrire le code, un gestionnaire de projet pour organiser les fichiers et les paramètres, un compilateur ou de l'assembleur pour convertir le code source au code machine, et une sorte de matériel ou logiciel qui se connecte soit à une cible microcontrôleur ou simulent le fonctionnement d'un microcontrôleur.

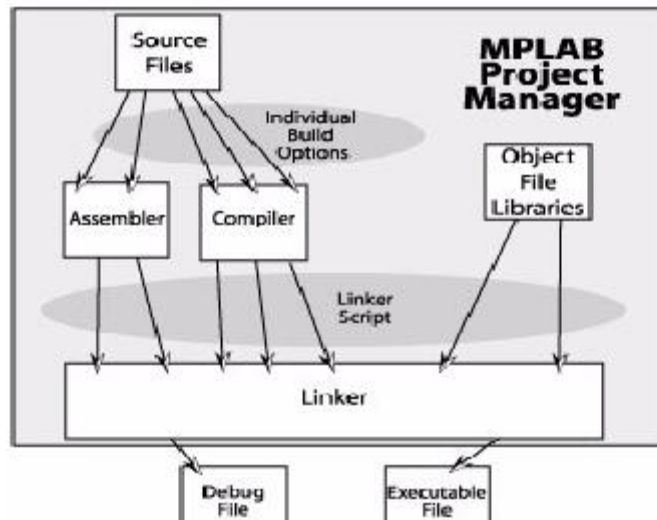
Plus vite le concepteur peut parcourir cette boucle, plus vite la demande sera terminée et prête à être envoyée. Cette boucle sera suivie de nombreuses fois, pour les fautes de frappe dans l'écriture de code, lorsque les bugs se posent dans la mise en œuvre, comme les optimisations sont réalisées, et que les performances de l'application sont adaptées. MPLAB IDE permet au concepteur de se déplacer rapidement entre ces diverses phases de développement, résultant en une courte période de lancement de l'application.



ÉTAPE 5 - PROJECT MANAGER

Le gestionnaire de projet organise les fichiers pour être édité avec les autres fichiers associés afin qu'ils puissent être transmis aux outils du langage comme l'assembleur ou la compilation, et finalement à un éditeur de liens. L'éditeur de liens a pour mission de placer les fragments de code objet à partir de l'assembleur, du compilateur et des bibliothèques dans les domaines de la mémoire proprement dite du contrôleur embarqué, et s'assurer que les modules fonctionnent les uns avec les autres (ou sont «liés»).

Toute cette opération d'assemblage et de compilation à travers le processus de liaison est appelé «construction du projet». A partir du gestionnaire de projet MPLAB, les propriétés des outils du langage peut être invoquée différemment pour chaque fichier, si désiré.



Fichiers d'entrée et de sortie gérés par le gestionnaire de projet

Les fichiers source sont des fichiers texte qui sont écrits conformes aux règles de l'assembleur ou du compilateur. Puis est ensuite converti en modules intermédiaires de code machine incomplet avec des espaces réservés pour les références aux fonctions et données de stockage.

L'éditeur de liens résout ces mots-clés et combine tous les modules dans un fichier de code machine exécutable. L'éditeur de liens produit également un fichier de débogage qui permet MPLAB IDE de relier l'exécution de code machine aux fichiers source, et crée des liens symboliques pour le stockage des variables en mémoire.

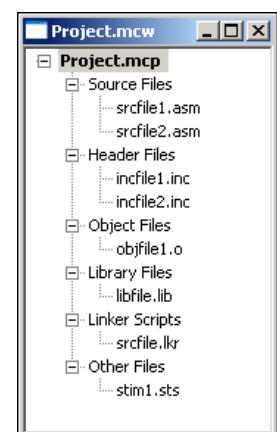
Un éditeur de texte est utilisé pour écrire le code. Ce n'est pas un traitement de texte, mais un éditeur spécifiquement conçu pour l'écriture de code pour microcontrôleurs Microchip. Il reconnaît les constructions dans le texte et utilise des codes de couleur pour identifier les différents éléments, tels que les codes mnémoniques d'instruction, les constructions de langage C et les commentaires. L'éditeur supporte les opérations couramment utilisés dans l'écriture du code source, comme la recherche de correspondants accolades en C, en commentant et dé-commentant les blocs de code, recherche de texte dans plusieurs fichiers et l'ajout de signets spéciaux.

Après l'écriture du code, l'éditeur travaille avec les autres outils pour afficher l'exécution du code dans le débogueur. Des points d'arrêt peuvent être définis dans l'éditeur, et les valeurs des variables peuvent être inspectées par un survol du pointeur de la souris sur le nom des variables. Les noms des variables peuvent être glissés depuis les fenêtres de texte source, puis a chuté dans une fenêtre montre.

Les projets sont facilement créés à l'aide de l'Assistant de projet dans MPLAB IDE. Les projets peuvent être copiés et les fichiers peuvent être archivés en utilisant un logiciel intégré de contrôle de version avec MPLAB IDE.

Le gestionnaire du projet associe les fichiers dans un projet d'une suite «outil», tels que MPASM Microchip / MPLINK, MPLAB C.

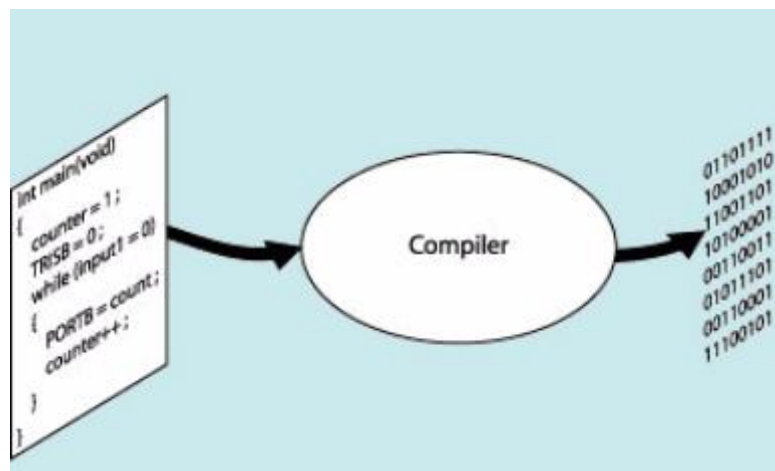
Une fois que les fichiers sont enregistrés dans le gestionnaire de projet, les options de construction peuvent être définies sur le projet dans son ensemble ou sur des dossiers individuels. Build option comprend des éléments tels que les répertoires de fichiers de paramètres de sortie pour l'optimisation des compilateurs, ainsi que la mémoire "modèles".



ÉTAPE 6 - LANGUAGE TOOLS

Les [language tools](#) sont des programmes comme par exemple l'assemblage croisé ou la compilation croisée. La plupart des gens sont familiers avec ceux type de programme qui s'exécutent sur un PC, tel que Visual Basic ou les compilateurs C. Lors de l'utilisation de ces outils pour les systèmes embarqués, un "cross-assembleur" ou «cross-compilateur» est utilisé. Ces outils diffèrent des compilateurs typiques en ce sens qu'elles s'exécutent sur un PC, mais pour produire du code qui fonctionnera sur un autre microprocesseur, par conséquent, la «cross-compilation" génère du code pour un microcontrôleur qui utilise un ensemble d'instructions complètement différent de l'ordinateur. Ces outils permettent également de produire un fichier de débogage que MPLAB IDE utilise pour corrélér les instructions de la machine et les emplacements mémoire avec le code source. Ce point d'intégration permet à l'éditeur de MPLAB de définir des points d'arrêt, permet de visualiser sur une fenêtre le contenu des variables, et vous permet exécuter le code source de l'application en mode pas à pas.

Les outils pour système embarqué diffèrent également des compilateurs fonctionnent sur PC, en ce sens qu'ils doivent permettre de réduire l'espace utilisé dans le microcontrôleur. Plus le code produit est petit, mieux c'est, parce que cela permet la plus petite mémoire possible pour la cible, réduisant ainsi les coûts. Cela signifie que les techniques pour optimiser et améliorer le code du microcontrôleur grâce à une bonne connaissance de ses spécificités est souhaitable. La taille des programmes pour les ordinateurs s'étend généralement plusieurs mégaoctets pour les programmes de complexité moyenne. La taille des programmes d'un simple système embarqué n'est seulement que de mille octets. Un système embarqué moyenne pourrait nécessiter 32 à 64 Ko de code pour des fonctions relativement complexes. Certains systèmes embarqués utilisent plusieurs mégaoctets d'espace de stockage pour les grandes tables, l'utilisateur des messages texte ou l'enregistrement de données.



Traduction par le compilateur d'un code source en langage machine

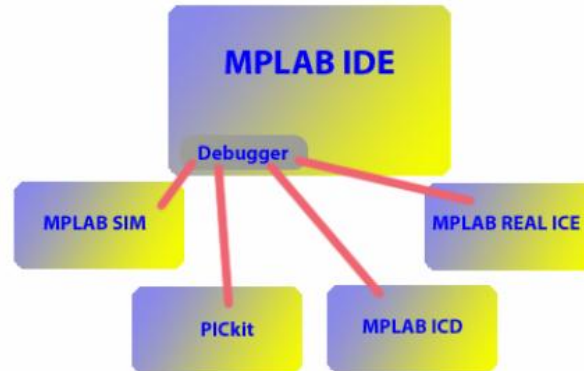
Microchip Language Tools		
Nom	Description	Inclus gratuitement avec MPLAB
MPASM	Macro Assembler	Oui
MPLINK	Edition des liens du code objet des fichiers C et assembleur	Oui
Mplib	Bibliothèque de fichiers réutilisables	Oui
MPLAB C Compilers	Conforme à la norme ANSI, compilateurs optimisés	demos gratuites disponibles séparément



ÉTAPE 7 – DEBOGAGE DE LA CIBLE

Dans un environnement de développement, l'exécution du code est testée sur un débogueur. Le débogueur peut être un **logiciel** qui simule le fonctionnement du microcontrôleur pour des essais ou peut être un **instrument spécial** pour analyser le programme tel qu'il exécute sur la carte cible.

Le simulateur est intégré à MPLAB IDE et permet à un programme d'être testé sans aucun matériel supplémentaire. Un simulateur est un débogueur logiciel, et les fonctions de débogage pour le simulateur sont pratiquement identiques au débogueur matériel, permettant un apprentissage en toute simplicité.



Simulateur et Debugger

Habituellement, un simulateur tourne un peu plus lentement qu'un microcontrôleur en fonctionnement réel, puisque le CPU dans le PC est utilisé pour simuler le fonctionnement du microcontrôleur. La vitesse de la simulation dépend de la vitesse du PC, système d'exploitation du PC et de combien d'autres tâches en arrière-plan sont en cours d'exécution, ainsi que de la complexité de la simulation demandé.

Matériels de débogage et de programmation

Il existe deux types de matériel qui peut être utilisé avec MPLAB IDE:

- Les programmeurs
- Les débogueurs

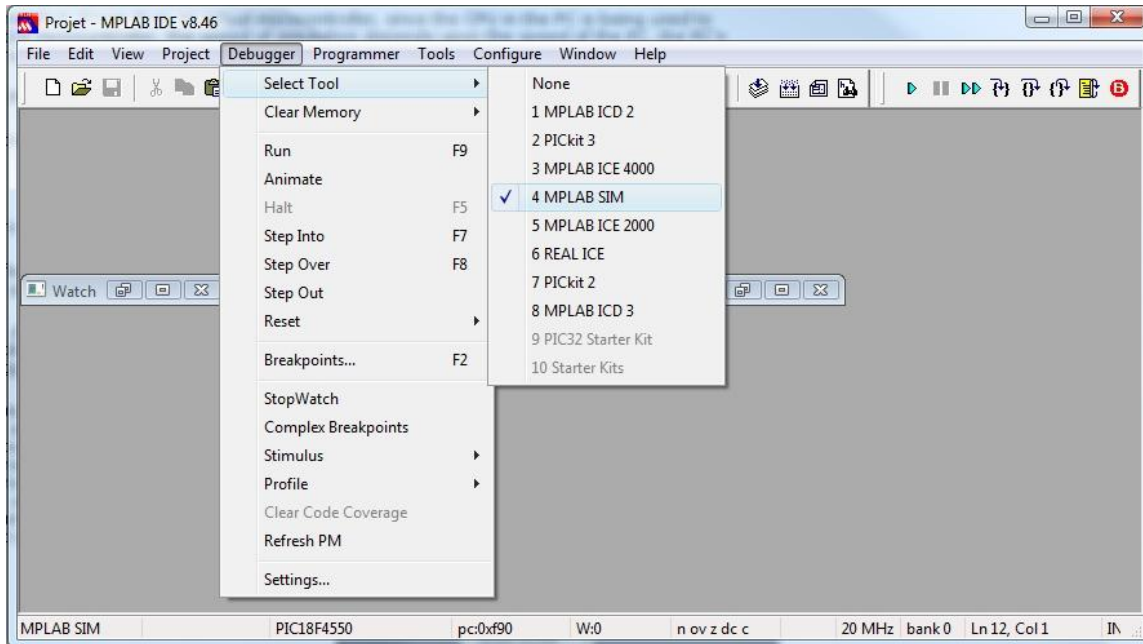
Un programmeur transfère simplement le code machine à partir du PC dans la mémoire interne du microcontrôleur cible. Le microcontrôleur peut ensuite être intégré à la carte cible afin d'exécuter les commandes prévus.

Cependant, il est habituel, que le code ne fonctionne pas exactement comme prévu, et l'ingénieur est chargé d'examiner le code dans son fonctionnement sur l'application afin de déterminer comment modifier le code source original pour réaliser l'exécution désiré. Ce processus est appelé le débogage.

Comme indiqué précédemment, le simulateur peut être utilisé pour tester la façon dont le code va fonctionner, mais une fois le microcontrôleur programmé avec le firmware, beaucoup de choses en dehors du champ d'application du simulateur entrent en jeu. En utilisant simplement un programmeur, le code pourrait être modifié, reprogrammé dans le microcontrôleur et branché sur la cible pour les reprises, mais cela pourrait être un cycle long et laborieuse si le code est complexe. En utilisant cette technique, il est difficile de comprendre exactement ce qui ne va pas dans le matériel.

C'est là où le débogueur matériel est utile. Les débogueurs peuvent être de types différents, en utilisant du matériel spécialisé à la place du microcontrôleur cible réelle.





Un débogueur matériel, comme un simulateur, permet à l'ingénieur d'inspecter les variables à différents points dans le code, une seule étape est nécessaire pour suivre pas à pas les instructions en interaction avec le circuit spécialisé.

Le débogage devient généralement urgente vers la fin du cycle de conception. Pour obtenir le fonctionnement de l'application désiré initialement, la dernière étape avant d'entrer dans le déploiement du produit, présente généralement la plus grande influence sur les retards de production. C'est là où un environnement de développement intégré a un rôle de la plus grande importance.

Faire des réglages a la volet du code, le recompiler, puis le télécharger pour faire les essais, tout cela exige du temps. En utilisant tous les outils dans un environnement unique, cela vous permettra de réduire le temps autour du cycle de développement. Cette dernière étape, où les bugs critiques sont analysés constituent un défi pour le concepteur de systèmes embarqués. Le bon outil permet d'économiser beaucoup de temps.

Ce tableau affiche les compromis entre les différents outils:

Comparaison des fonctions du Microchip Hardware Debuggers			
Émulation du système	PICKit	MPLAB ICD	MPLAB REAL ICE
Coût	environ 80 \$	environ 220 \$	environ 500 \$
Trace Analyzer	Non	Non	Oui
Points d'arrêt complexe	Non	Oui	Oui
Option d'opto-isolation	Non	Non	Oui
Haute vitesse, option faible bruit	Non	Non	Oui



ÉTAPE 8 - PROGRAMMATION

Après que le projet ait été débogué et exécute dans l'environnement de développement, il doit être testé sur dans l'environnement réel. Un microcontrôleur peut être programmé avec le débogueur ou un programmeur.

Lors de la programmation, le MCU PIC ou la mémoire du programme du dsPIC DS, les données mémoire non-volatile, les bits d'identifications et les bits de configurations sont fixés en fonction des données et des paramètres de la boîte de dialogue de configuration MPLAB IDE.

Address	Value	Field	Category	Setting
300000	04	PLLDIV	96MHz PLL Prescaler	Divide by 5 (20MHz input)
		CPUDIV	CPU System Clock Postscaler	Divide by 12 (48MHz input)
		USBPLL	Full-Speed USB Clock Source Selection	Divide by 10 (40MHz input)
300001	0C	OSC	Oscillator	Divide by 6 (24MHz input)
		FCMEN	Fail-Safe Clock Monitor Enable	Divide by 5 (20MHz input)
		IESO	Internal External Switch Over Mode	Divide by 4 (16MHz input)
300002	01	PUT	Power Up Timer	Divide by 3 (12MHz input)
		BODEN	Brown Out Detect	Divide by 2 (8MHz input)
		BODENV	Brown Out Voltage	No Divide (4MHz input)
		VREGEN	USB Voltage Regulator	Disabled-Controlled by SWDTEN bit
300003	00	WDT	Watchdog Timer	1:1
		WDTPS	Watchdog Postscaler	RB3
300005	80	CCP2MUX	CCP2 Mux	PORTB<4:0> configured as digital I/O on RESET
		PBADEN	PortB A/D Enable	Disabled
		LPT1OSC	Low Power Timer1 Osc enable	MCLR Enabled, RE3 Disabled
300006	80	MCLRE	Master Clear Enable	Disabled
		STVR	Stack Overflow Reset	Disabled
		LVP	Low Voltage Program	Disabled
		ENICPORT	Dedicated In-Circuit Port (ICD/ICSP)	Disabled
		XINST	Extended Instruction Set Enable bit	Disabled
300008	0F	CP_0	Code Protect 00800-01FFF	Disabled
		CP_1	Code Protect 02000-03FFF	Disabled
		CP_2	Code Protect 04000-05FFF	Disabled
		CP_3	Code Protect 06000-07FFF	Disabled
300009	C0	CPB	Code Protect Boot	Disabled
		CPD	Data EE Read Protect	Disabled
30000A	0F	WRT_0	Table Write Protect 00800-01FFF	Disabled
		WRT_1	Table Write Protect 02000-03FFF	Disabled
		WRT_2	Table Write Protect 04000-05FFF	Disabled
		WRT_3	Table Write Protect 06000-07FFF	Disabled
30000B	E0	WRTC	Config. Write Protect	Disabled
		WRTB	Table Write Protect Boot	Disabled
		WRTE	Data EE Write Protect	Disabled
30000C	0F	EBTR_0	Table Read Protect 00800-01FFF	Disabled
		EBTR_1	Table Read Protect 02000-03FFF	Disabled
		EBTR_2	Table Read Protect 04000-05FFF	Disabled
		EBTR_3	Table Read Protect 06000-07FFF	Disabled
30000D	40	EBTRB	Table Read Protect Boot	Disabled

Fenêtre de dialogue « Configuration Bits »

L'application cible peut maintenant être observé dans son état presque final. La programmation et l'évaluation avec les techniques de prototypages rapides doivent permettre de faire l'évaluation de l'application finale.

Certaines applications peuvent être programmées après que le microcontrôleur ait été soudé sur la carte cible. L'utilisation In-Circuit Serial Programming (ICSP), le firmware peut alors être programmé dans l'application au moment de la fabrication, ce qui permet des révisions et des mises à jour pour être programmés dans une application embarquée plus tard dans son cycle de vie.

Les composants prenant en charge le débogage In-Circuit Serial Programming, peuvent-être reconnectés au MPLAB REAL ICE, ainsi qu'au PICKit et à MPLAB ICD après la fabrication, permettant de tester la qualité du développement du firmware de la prochaine génération.

Ces programmeurs sont disponibles auprès de Microchip pour s'exécuter avec MPLAB IDE. MPLAB PM3 peut également être exécuté de façon indépendante, simplifier l'interface utilisateur graphique (Visual PROCMD), ou d'un programme en ligne de commande qui permet MPLAB PM3 d'être exécuté à partir d'un fichier batch (PROCMD3).

MPLAB PM3

- RS-232 ou USB
- Intégrée l'interface In-Circuit Serial Programming (ICSP)
 - Mode PC hôte pour un contrôle total
 - Mode sans échec pour sécuriser les données
 - Mode autonome pour la programmation sans PC
- Ligne complète de sockets interchangeables pour soutenir tous la gamme de Microchip et les options de boîtier (vendu séparément)
- Sérialisation SQTP pour la programmation des numéros de série unique.
- Une deuxième ligne de commande DOS interface est disponible pour le contrôle des lots
- Secure Digital (SD) et MultiMedia Card (MMC)



MPLAB® REAL ICE™

Bien qu'il s'agisse principalement d'un débogueur, MPLAB REAL ICE sert en tant que programmeur pour les microcontrôleurs Flash. MPLAB REAL ICE, utilise le dispositif de Microchip In-Circuit Serial Programming et tant que les trois broches peuvent être correctement conduites aux microcontrôleurs PIC et dsPIC DSC, le firmware peut être reprogrammation après la fabrication du produit de l'application.



MPLAB ICD

Aussi et surtout un débogueur, MPLAB ICD permet de programmer plusieurs microcontrôleurs Flash. MPLAB ICD utilise la technologie de Microchip In-Circuit Serial Programming.

MPLAB ICD peut également être utilisé pour tester et déboguer une application finale, utilisant une interface simple. De nombreux concepteurs laisser un connecteur ICD sur l'application finale (ou au moins une place sur le PCB sur lequel un socket peut être soudé si nécessaire) à utiliser pour le débogage sur le terrain, faire des modifications du firmware, ou la fabrication et les essais en ligne.



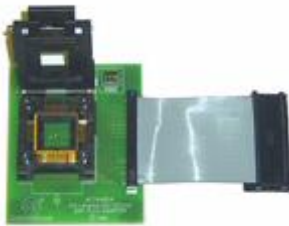
PICkit développement Programmeur / Debugger

PICkit Debug Express permet le débogage in-circuit sur certains microcontrôleurs PIC. Les fonctions de débogage simple, permettent de stopper le programme pendant le fonctionnement sur le microcontrôleur PIC qui intégré l'application. Lorsque le programme s'arrête sur un point d'arrêt, les registres de fichiers peuvent être examinées et modifiées.



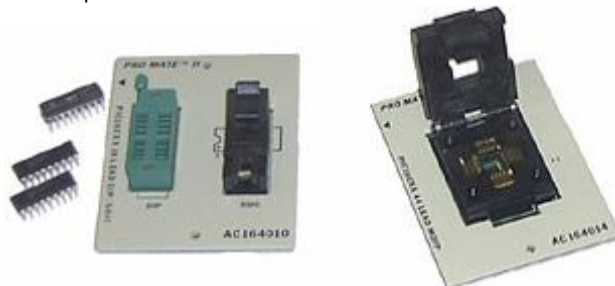
Comparaison des fonctions des programmeurs Microchip				
Programmeur	PICkit Flash Starter Kit	MPLAB ICD	MPLAB REAL ICE	MPLAB PM3
Coût	environ 80 \$	environ 220 \$	environ 500 \$	environ 900 \$
Supports ICSP	Oui	Oui	Oui	Oui
Prend en charge les dsPIC	Oui	Oui	Oui	Oui
Fonctionnement autonome				Oui
Fonctionne également comme débogueur	Oui	Oui	Oui	
Nécessite un module de connexion				Oui
Permet SQTP (numéro de série unique pour chaque MCU programmé)				Oui
Mode sans échec pour l'intégrité des données				Oui

Différents adaptateurs et modules de socket sont disponibles. Même si le PICSTART Plus est conçu avant tout comme un programmeur de sockets DIP, certains adaptateurs sont disponibles afin de lui permettre de programmer d'autres paquets. Le [module de programmation universelle](#) est disponible pour une utilisation avec MPLAB ICD et MPLAB REAL ICE.



PICSTART Plus Adapter

MPLAB PM3 permet la programmation de différents types de modules Socket. D'autres modules peuvent être commandés en tant qu'accessoires.



MPLAB PM3 Socket Modules



ÉTAPE 9 - PLUS D'INFORMATION sur MPLAB IDE

DOCUMENTATION de MPLAB IDE

Le document qui suit porte sur MPLAB IDE:

- MPLAB IDE User's Guide (DS51519)

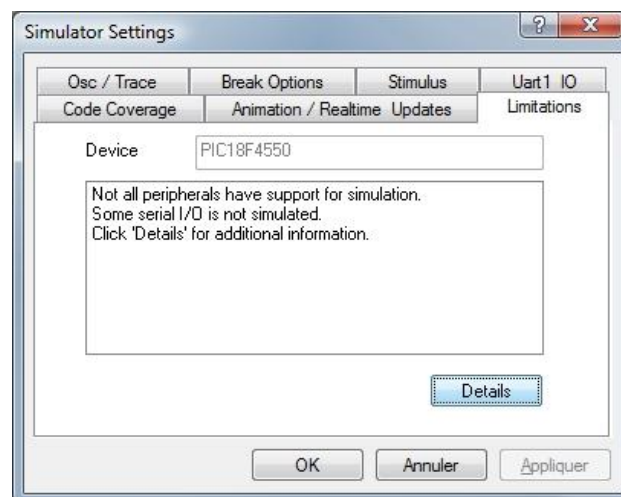
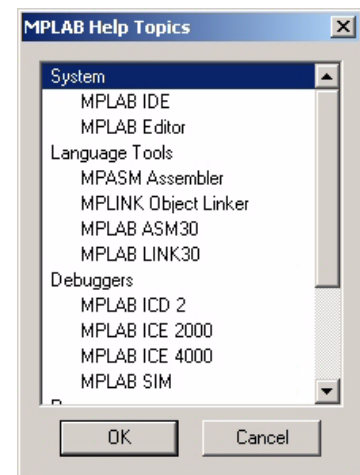
D'autres documents existent pour les différents logiciels et outils de Microchip permettant de travailler avec MPLAB IDE. Consultez le [site web de Microchip](#) pour les versions téléchargeables en PDF de tous ces documents. L'information la plus actuelle est contenue dans l'aide en ligne, téléchargés avec chaque version de MPLAB.

MPLAB IDE ON-LINE HELP

MPLAB IDE est constamment mis à jour avec de nouvelles caractéristiques et fonctionnalités pour l'ensemble de ces outils. Les boîtes de dialogue peuvent ne pas apparaître exactement comme dans ce manuel, les listes des menus peuvent différer. Pour cette raison, l'aide en ligne est la meilleure référence à la version de MPLAB IDE utilisé par vos soins.

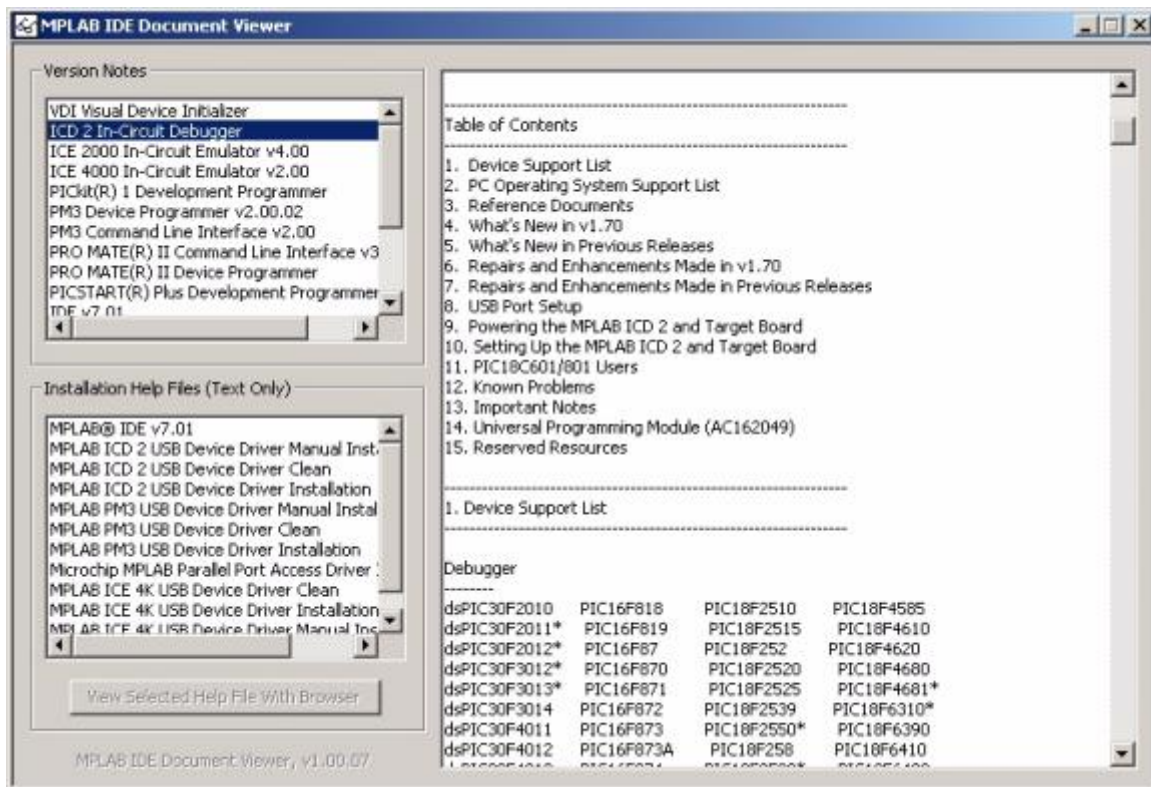
MPLAB IDE est installé avec son aide en ligne, qui est constamment mis à jour. Si des questions surgissent lors de l'utilisation MPLAB IDE, n'oubliez pas de consulter l'aide en ligne pour y trouver des réponses. Plus important encore, l'aide en ligne indique des restrictions qui pourraient exister pour un outil particulier. Essayez de toujours consulter cette section avant de travailler avec une nouvelle combinaison d'outils.

L'onglet limitations du menu Debugger>Settings permet d'afficher une boîte de dialogue où sont indiqués les restrictions dans le simulateur, l'émulateur ou dans le débogage. Les limitations sont généralement indiquées dans la zone texte.



Pour les renseignements courants sur les différentes composantes de MPLAB, ouvrez le fichier README de chaque version. Le détail Notes de version qui sont des dispositifs pris en charge dans cette version pour chaque composant. Les notes des versions des fichiers contiennent aussi des renseignements opportuns sur les problèmes connus et les solutions proposées.





MPLAB IDE Document Viewer permet de consulter des informations sur les outils

MPLAB IDE UPDATES AND VERSION NUMBERING

MPLAB IDE est un programme évolutif avec des milliers d'utilisateurs. Microchip conçoit continuellement de nouveaux microcontrôleurs avec de nouvelles fonctionnalités. Beaucoup de nouvelles caractéristiques de MPLAB IDE proviennent des demandes des clients et de son usage interne. Les mises à jour d'MPLAB IDE sont prévues environ tous les quelques mois pour ajouter le support de nouveaux périphériques et de nouvelles fonctionnalités.

L'aide en ligne est la meilleure source pour toutes questions concernant MPLAB IDE, et les notes d'applications contiennent des informations sur les nouvelles fonctionnalités et le fonctionnement des périphériques.

Pour être averti des mises à jour de MPLAB IDE et de ses composantes, abonnez-vous à la section [service de notification de changement des outils de développement](http://www.microchip.com/) sur <http://www.microchip.com/>.

