Visual Studio au secours d'Arduino

LOÏC JOSSE*

Les cartes Arduino font désormais partie de nos outils pédagogiques, plus particulièrement en STI2D spécialité SIN et en S-SI où, au sein du projet, elles rendent des services très appréciables. Elles méritent d'avoir une interface logicielle de qualité, ce que propose Microsoft avec son « Visual Micro ».

environnement de développement Arduino permet la réalisation facile, rapide et à moindre coût de projets électroniques variés. Le grand nombre d'interfaces ainsi que la richesse des ressources disponibles en font un produit prisé par beaucoup d'enseignants et d'élèves. Le choix de ce type de carte est aussi guidé par le référentiel de notre enseignement en SIN. En effet, le Bulletin officiel nº 3 du 17 avril 2011 indique, dans le chapitre consacré à la démarche de projet : « La réalisation se limite à la conception d'un prototype de qualité industrielle. Les cartes électroniques sont conçues par association de fonctions et/ou de constituants intégrés. » Là encore, la plateforme Arduino se positionne comme un atout non négligeable. Seul bémol à l'engouement qu'elle provoque, son environnement graphique minimaliste et son manque de débogueur. Nous vous proposons donc de découvrir Visual Micro, l'IDE¹ Arduino de Microsoft. Il s'agit en fait d'un complément (plugin) Arduino à l'environnement de développement Visual Studio [voir En ligne].

L'environnement de travail

Une fois l'installation terminée, ouvrir « Visual Studio », puis « Nouveau projet » et, dans le modèle C++, « Visual Micro » **1**. Donner un nom à votre projet (ici potled), puis lancer l'application.

À ce moment, vous pouvez remarquer que le programme source, pourvu de l'extension « .ino », se trouve dans la fenêtre de l'explorateur de projet. Lorsque vous l'ouvrez, vous constatez que le travail de préparation réalisé par l'éditeur de texte est semblable à celui de l'environnement Arduino, à l'exception des couleurs. Remarquez, au-dessus de la fenêtre d'édition, la version d'Arduino installée précédemment, le type de carte à programmer, la possibilité

M 0 T S - C L É S

programmation, logiciel, projet d'insérer une bibliothèque ou un code source. D'ailleurs, et pour information, les bibliothèques et les codes sources nécessaires au bon fonctionnement de vos différentes cartes d'extension (*shields*) sont toujours installés dans le répertoire « Libraries » d'Arduino.

Afin de tester l'application, nous vous proposons de saisir un petit code source qui permet de régler la luminosité d'une LED connectée sur D9 à l'aide d'un potentiomètre connecté sur A0 2.

On constate rapidement que la saisie est facilitée grâce à la fonction *Intellisense* (assistant de saisie). Lorsqu'elle est terminée, compiler votre programme (*Build*), la fenêtre de sortie vous renseigne quant au déroulement des opérations. Si tout est correct, vous constaterez que le programme occupe 12 % de l'espace de stockage contre 4 % si on l'avait compilé avec l'IDE Arduino. En fait, Microsoft ajoute des lignes de code permettant d'utiliser un pseudo-débogueur que nous allons décrire plus loin.

Si vous envisagez de récupérer le code hexadécimal pour effectuer une simulation sous Proteus par exemple, nous vous conseillons d'afficher plus d'informations dans la fenêtre de sortie. Pour cela, aller dans le menu « Déboguer », puis « Options et paramètres », choisir le menu « Visual Micro » et, dans « Compiler », passer l'option « Verbose » sur *True.* Lors de la prochaine compilation, vous verrez apparaître le chemin d'accès vers le code hexadécimal que vous pourrez copier et coller dans le composant Arduino de Proteus **1**.

En appuyant sur CTRL + F5 (exécuter sans débogage), le code est compilé puis transféré dans la carte cible reliée à la sortie USB de votre PC, vous permettant de tester le fonctionnement sur la maquette réelle.

Le pseudo-débogueur

Cliquer sur la gauche de l'éditeur, devant la ligne « PotVar = map(PotVar, 0, 1023, 0, 255); » afin de placer un point d'arrêt. Cliquer droit sur ce point d'arrêt, puis sélectionner « Lorsqu'il est atteint ». Sélectionner « Afficher un message » et indiquer « la valeur de PotVar est : {PotVar} » <a>[4]. En appuyant sur F5 (Démarrer le débogage), vous observerez les valeurs prises par la variable PotVar lorsque vous agissez sur le potentiomètre du prototype réel <a>[5].

* Enseignant SII option information et numérique, lycée Louis-le-Grand, Paris.

FORUM DES TECHNOLOGIES



1 Fenêtre de lancement d'un nouveau projet Visual Micro



FORUM DES TECHNOLOGIES



5 Débogage en mode trace, visualisation de la variable PotVar



Visualisation des entrées/sorties

EN LIGNE

Page de téléchargement de Visual Micro : www.visualmicro.com Vous devez installer sur votre machine trois applications (quatre pour 2013) : I'IDE Arduino à partir de l'adresse : www.arduino.cc/en/Main/Software puis l'IDE Visual Studio sur : www.microsoft.com/fr-fr/download/details.aspx?id=44914 et son module linguistique en français pour la version 2013 (c'est celle que j'ai testée sur Windows 7 et 8.1) sur : www.microsoft.com/fr-fr/download/details.aspx?id=40783 ou pour la version 2015 (mais je ne l'ai pas testée !) sur : www.visualstudio.com/downloads/download-visual-studio-vs et enfin le complément Visual Micro sur :

visualstudiogallery.msdn.microsoft.com/069a905d-387d-4415-bc37-665a5ac9caba

Tous les liens sur http://eduscol.education.fr/sti/revue-technologie



Mise en place du débogage des entrées/sorties

Ici le débogage s'effectue automatiquement en mode trace, mais vous pouvez, en changeant les options disponibles sur le point d'arrêt, contrôler le déroulement en mode pas à pas. Dans ce cas, un pas correspond à un intervalle entre deux points d'arrêt ou, à défaut, une boucle (*loop*) sur le même point d'arrêt. On peut aussi, comme sur l'IDE Arduino, mettre en place une visualisation de variables en utilisant la liaison série.

Un autre aspect très intéressant de ce débogueur est de pouvoir visualiser les tensions sur les entrées analogiques A0 à A5, ainsi que les états logiques sur les entrées/sorties D0 à D15. Pour cela, vous devez positionner les onglets « Report Analogs » et « Report Digitals » du menu « Micro Debug Reporting » sur *True* G et démarrer le débogage avec F5. Un état logique « 0 » apparaît en rouge et un « 1 » en vert. Les tensions analogiques sont représentées sous forme de chronogrammes Z. On peut placer les entrées non utilisées du prototype à la masse pour ne pas encombrer l'affichage.

Visual Micro n'est pas un logiciel portable, et même s'il occupe plus de place dans l'espace mémoire de travail, son fonctionnement plus professionnel et son pseudo-débogueur en font un IDE très appréciable des programmeurs avancés.

1. Integrated Development Environment : environnement de développement intégré ou, ici, environnement de développement logiciel.

TECHNOLOGIE 204 SEPTEMBRE-OCTOBRE 2016