Résumé des points intéressants des 17 premières pages.

Table des matières

AVR Studio4 ou Atmel Studio6 ?

Désirant passer de l'Arduino au langage C standard, nous optons pour **Studio**, l'environnement de développement gratuit proposé par Atmel.
L'énorme avantage de Studio sur l'IDE Arduino est la possibilité de debugging, soit par simulation du fonctionnement du programme, soit par debugging en temps réel dans le µC lui-même.
Deux versions sont valables :

- Studio 4.18 (build 684) + Studio 4.18 SP3 (build 716). Lien: http://www.atmel.com/tools/studioarchive.aspx
- Studio 6.2 Lien: http://www.atmel.com/tools/atmelstudio.aspx

Pour choisir

	Studio 4	Studio 6
PROs	Nécessite peu de ressources du PC Rapide	Nécessite un PC assez performant Version suivie pour les nouveaux µC Intègre ATMEL-ICE Mise en œuvre de l'ATMEGA328P-XMINI Contient la dernière version de AVR-GCC Tool Chain
CONs	Studio 4 n'est plus mis à jour ATMEL-ICE non reconnu ATMEGA328P-XMINI non reconnu Dernière version de AVR-GCC Tool Chain à installer manuellement	Lent sur PC ancien Utilisation plus complexe que Studio 4

Voir également ce lien pour le choix: http://www.kanda.com/blog/microcontrollers/avr-microcontrollers/avrstudio-explored/

ATMEL-ICE est un debugger JTAG (in situ) récent bon marché produit par Atmel. Lien: http://atmelcorporation.wordpress.com/2014/03/25/the-new-atmel-ice-debugger-is-here/
Vendu par Farnell.

ATMEGA328P-XMINI est une platine de développement récente d'Atmel avec debugging in situ. Très bon marché. Vendu par Farnell. Lien: http://www.atmel.com/tools/mega328p-xmini.aspx

AVR-GCC Tool Chain : ensemble des outils pour la compilation des programmes C. Lien pour l'installation manuelle: http://forums.futura-sciences.com/electronique/651270-tuto-configurer-avrstudio-4-18-sp3-derniere-toolchaine-avr-gcc.html

Adam_D a fourni un lien intéressant en français (mais surtout Linuxien) décrivant la Tool Chain : http://www.chicoree.fr/w/USnooBie#GNU_AVR_Toolchain

Pour ceux qui éprouvent beaucoup de peine à passer de l'IDE Arduino au langage C de Studio, une solution intermédiaire a été pointée par Fred_du_92 : http://forums.futura-sciences.com/electronique/650737-de-larduino-langage-c-standard-avr-studio-5.html#post4884070. Dans Studio6.2, menu Help → Atmel Gallery → search "Arduino" → possibilité de télécharger *Arduino IDE for Atmel Studio 6.1 & 6.2*. Ce plug-in permet de programmer dans le langage Arduino avec les avantages de Studio, en particulier avec des possibilités de simulation.

Utilisation de Studio6 : http://maxembedded.com/2012/06/25/using-atmel-studio-6/

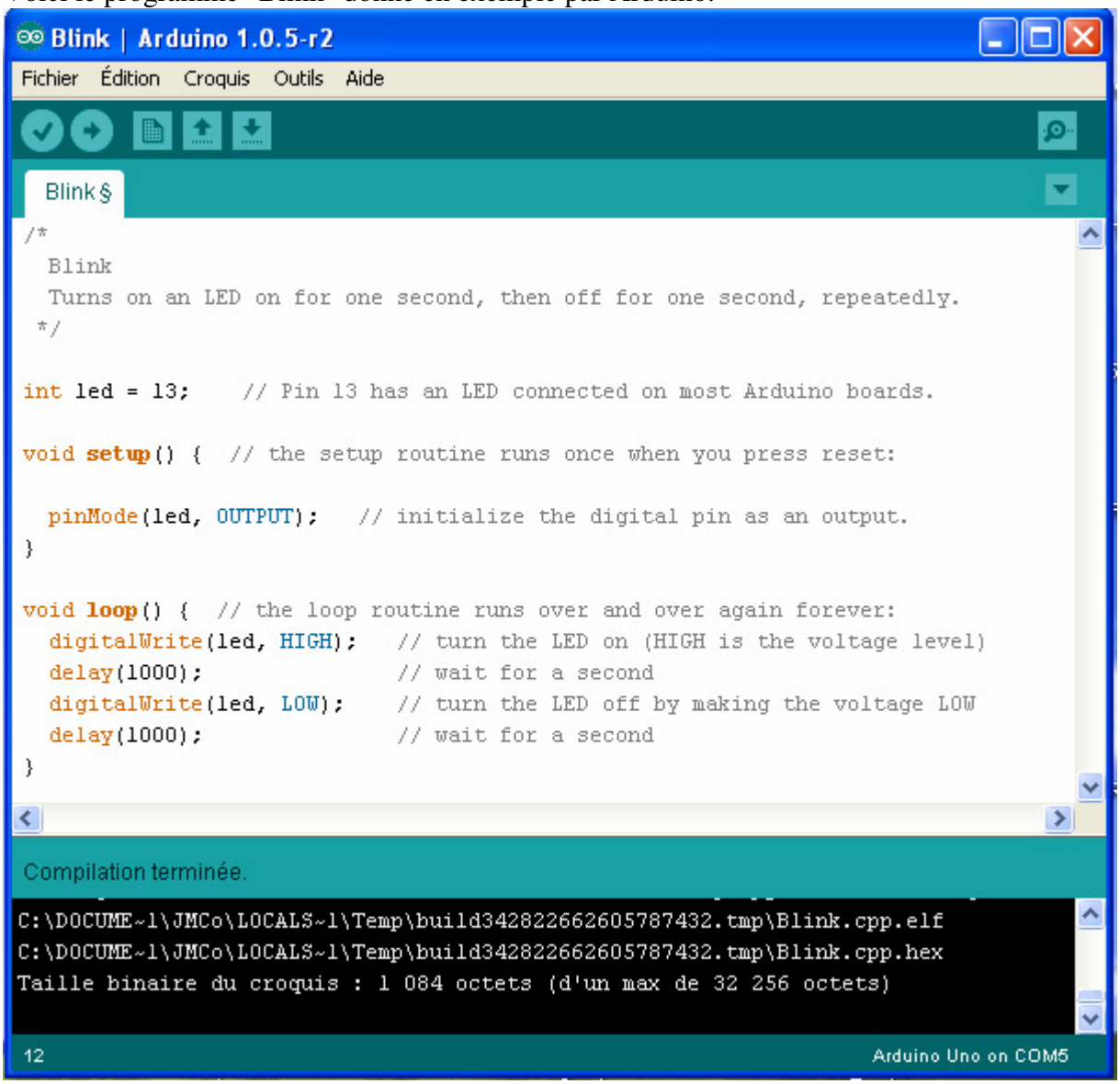
Ecriture du programme

Pour Arduino, un programme se nommait "sketch" jusqu'il y a peu et se nomme actuellement "croquis". C'est un fichier texte avec une extension propre à Arduino (anciennement .pde et actuellement .ino).

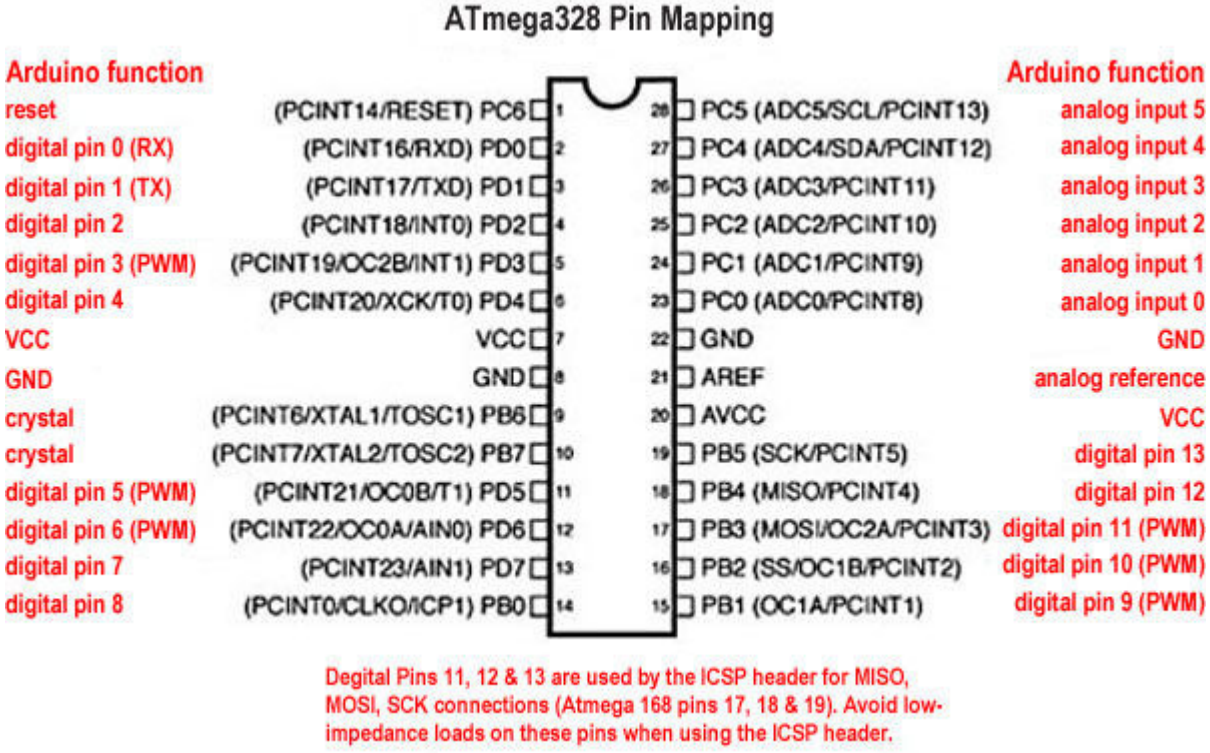
Ce programme comporte au minimum deux fonctions :

- la fonction setup( ) qui ne s'exécute qu'une seule fois après de reset et qui sert à initialiser les pins d'entrée/sortie et les périphériques.
- la fonction loop( ) qui se répète indéfiniment

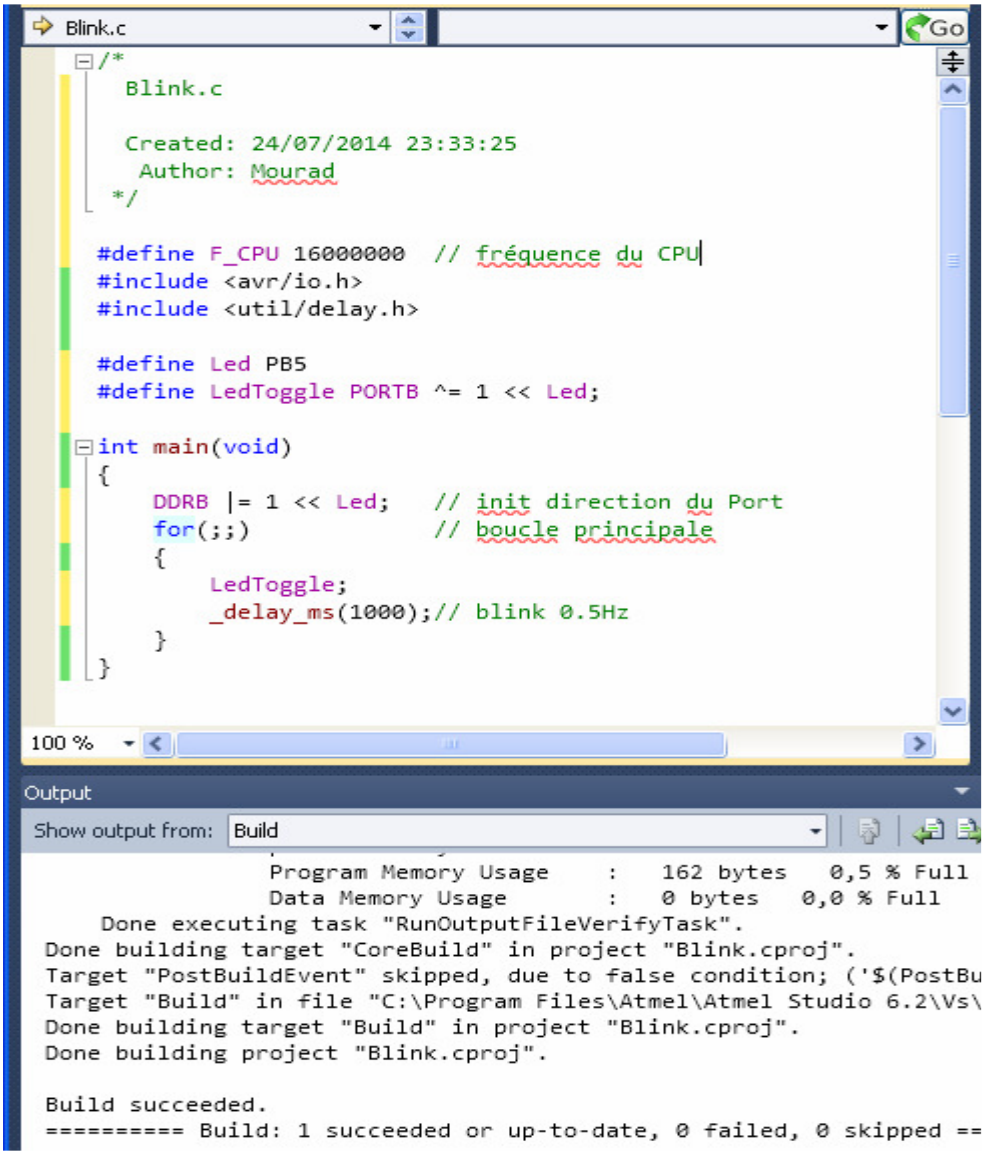
Voici le programme "Blink" donné en exemple par Arduino.



Arduino a donné une dénomination différente d'Atmel pour les pins de l'Atmega.
Pour la programmation en C, on utilise la dénomination officielle d'Atmel.
La photo suivante montre la correspondance entre les pins d'Arduino en rouge et les pins d'Atmel en noir.



La pin 13 d'Arduino correspond donc à la pin 5 du Port B (PB5).
Voici le programme écrit en C dans Studio6.



Remarquez que le programme compte 1084 bytes dans sa version Arduino et 162 bytes dans sa version C.

Le programme C comporte au minimum une seule fonction :

- main( )

A l'intérieur de cette fonction, on crée une boucle sans fin "for(;;)" ou "while(1)" qui est l'équivalent de "loop( )" dans Arduino.

La fonction "_delay_ms" du C ne fait pas appel au Timer0, contrairement à l'Arduino. Ce Timer reste donc libre pour d'autres fonctions.

Question de *variable volatile :* voir ce message et les suivants : http://forums.futura-sciences.com/electronique/650737-de-larduino-langage-c-standard-avr-studio-8.html#post4887384

Voir également cet article : http://www.avrfreaks.net/index.php?name=PNphpBB2&file=viewtopic&t=97382&highlight=optimization

Collection de programmes C à télécharger : http://www.elektor.fr/programmation-en-c-des-microcontroleurs-risc-avr-french

Flashage du programme dans le µC

Dans l'IDE Arduino, le bouton "Téléverser" compile le programme et flashe le fichier hex compilé dans le µC en une seule opération.

Arduino se sert du logiciel **avrdude** pour flasher le fichier hex.

Lorsqu'on travaille en C avec Studio, il existe plusieurs possibilités de flasher le fichier hex selon le matériel dont on dispose.

1. **Le µC est sur une platine Arduino USB et est pourvu de son bootloader**

   Si le µC est pourvu de son bootloader Arduino et que la platine est raccordée au PC par un câble USB, le flashage peut se faire sans autre matériel. On utilisera le logiciel avrdude, soit en ligne de commande avec ses nombreux paramètres, soit plus confortablement sous Windows à l'aide d'un logiciel d'interface (voir plus loin "avrdude et son interface Windows AVRDUDESS").

2. **Le µC a un bootloader et est sur une breadboard**

   On peut utiliser un convertisseur USB-Série TTL (très bon marché sur eBay) avec avrdude comme logiciel.
   Voir ce lien : http://forum.arduino.cc/index.php?topic=66178.0

3. **Le µC n'a pas de bootloader**

   Un programmateur externe est nécessaire pour envoyer un programme ou un bootloader dans le µC.
   (Localisation des bootloaders Arduino : http://forums.futura-sciences.com/electronique/650737-de-larduino-langage-c-standard-avr-studio-4.html#post4882156)
   De nombreux programmateurs sont possibles :
   - Une platine Arduino peut être utilisée comme programmateur ISP. http://hardwarefun.com/tutorials/use-arduino-as-an-isp-programmer-to-program-non-arduino-avr-microcontrollers
   - Un USBasp acheté sur eBay + avrdude + AVRDUDESS est une solution très peu chère.
   - Un AVRISP mkII d'Atmel : http://fr.farnell.com/atmel/atavrisp2/programmateur-avr-mcu-isp/dp/1135517 L'AVRISP mkII s'utilise à partir de Studio (mais peut aussi être utilisé avec avrdude). http://forums.futura-sciences.com/electronique/650737-de-larduino-langage-c-standard-avr-studio-3.html#post4880980
   - Un Atmel-ICE ou une autre platine JTAG : http://fr.farnell.com/atmel/atatmel-ice-basic/debogeur-pour-arm-avr-cable-comm/dp/2407172
     En plus du flashage, ces platines permettent de faire du debugging in situ.

Un petit résumé se trouve ici : http://forums.futura-sciences.com/electronique/650737-de-larduino-langage-c-standard-avr-studio-5.html#post4884212

Laveplusblanc signale des sondes JTAG pas chères sur eBay, mais réservées à Studio4 et pour certains Atmegas seulement. Demander "AVR JTAG" sur eBay.

Quelques précisions ici : http://forums.futura-sciences.com/electronique/650737-de-larduino-langage-c-standard-avr-studio-8.html#post4887343 (Attention, lire AVRISP mkII au lieu de USBisp mkII).

Si vous possédez une platine Arduino avec une prise ICSP, le µC de la platine peut être programmé par cette prise, à l'aide d'un programmateur externe. http://forums.futura-sciences.com/electronique/650737-de-larduino-langage-c-standard-avr-studio-13.html#post4895331 . Dans ce cas le bootloader Arduino sera effacé.

Platine d'expérimentation et µC

Voici quelques réflexions sur le sujet :

Platine ZIF + ATmega32 + USBasp ou AVRISP mkII ou JTAG: http://forums.futura-sciences.com/electronique/650737-de-larduino-langage-c-standard-avr-studio-6.html#post4885083

Platine montée avec ATmega128 + USBasp ou AVRISP mkII ou JTAG: http://forums.futura-sciences.com/electronique/650737-de-larduino-langage-c-standard-avr-studio-6.html#post4885186

Comparaison des platines et des ATmegas : http://forums.futura-sciences.com/electronique/650737-de-larduino-langage-c-standard-avr-studio-12.html#post4891491

Platine ZIF + ATmega1284 + USBasp ou AVRISP mkII ou JTAG: http://forums.futura-sciences.com/electronique/650737-de-larduino-langage-c-standard-avr-studio-14.html#post4897164

Platine à construire soi-même + ATmega1284 + USBasp ou AVRISP mkII ou JTAG: http://forums.futura-sciences.com/electronique/650737-de-larduino-langage-c-standard-avr-studio-14.html#post4897208

Platine ArduinoMEGA2560 + USB ou USBasp ou AVRISP mkII ou JTAG: http://forums.futura-sciences.com/electronique/650737-de-larduino-langage-c-standard-avr-studio-14.html#post4898089

Pas de "Break" pour l'ATmega128 http://forums.futura-sciences.com/electronique/650737-de-larduino-langage-c-standard-avr-studio-15.html#post4898571

Simulation

Essai de simulation avec Studio4 : http://forums.futura-sciences.com/electronique/650737-de-larduino-langage-c-standard-avr-studio-6.html#post4885635

Simulation et débogueur : http://maxembedded.com/2011/06/13/using-the-avr-studio-5-simulator-and-debugger/

Envoi de stimuli pendant la simulation : http://forums.futura-sciences.com/electronique/650737-de-larduino-langage-c-standard-avr-studio-10.html#post4888155

Debugging in situ

Arduino MEGA2560 et AVR ONE : http://forum.arduino.cc/index.php/topic,96401.0.html

Pas de "Break" pour l'ATmega128 http://forums.futura-sciences.com/electronique/650737-de-larduino-langage-c-standard-avr-studio-15.html#post4898571

Terminal (moniteur) par liaison série

Terminal de Bray++ et convertisseur USB-Série TTL : http://forums.futura-sciences.com/electronique/650737-de-larduino-langage-c-standard-avr-studio-15.html#post4898839

Terminal d'Atmel et "Termite" : http://forums.futura-sciences.com/electronique/650737-de-larduino-langage-c-standard-avr-studio-16.html#post4901048

Terminal par la prise USB de l'Arduino: http://forums.futura-sciences.com/electronique/650737-de-larduino-langage-c-standard-avr-studio-17.html#post4902552

Programme USART de base: http://forums.futura-sciences.com/electronique/650737-de-larduino-langage-c-standard-avr-studio-15.html#post4900125

Programme USART pour affichage complexe (Mourad): http://forums.futura-sciences.com/electronique/650737-de-larduino-langage-c-standard-avr-studio-17.html#post4904551

Programme de redirection stdin et out vers USART : http://forums.futura-sciences.com/electronique/650737-de-larduino-langage-c-standard-avr-studio-17.html#post4906804

Transformation d'un USBasp en AVRISP mkII

http://forums.futura-sciences.com/electronique/650737-de-larduino-langage-c-standard-avr-studio-12.html#post4891491

http://forums.futura-sciences.com/electronique/650737-de-larduino-langage-c-standard-avr-studio-12.html#post4891638

Installation des drivers pour USBasp et AVRISP mkII : http://forums.futura-sciences.com/electronique/650737-de-larduino-langage-c-standard-avr-studio-13.html#post4894797

avrdude et son interface Windows AVRDUDESS

Avrdude est un excellent logiciel de type Linux mais dont la version compilée pour Windows s'utilise en ligne de commande avec de très nombreux paramètres. Retenir ces paramètres n'est pas pratique et les retaper dans une fenêtre de commande à chaque fois qu'on veut programmer un µC est encore moins pratique. Heureusement, il existe des logiciels permettant d'enrober Avrdude dans une fenêtre de style Windows. J'ai découvert récemment le logiciel AVRDUDESS http://blog.zakkemble.co.uk/avrdudess-a-gui-for-avrdude/ qui permet d'utiliser un nombre impressionnant de programmateurs. En particulier, il s'accommode de AVRISP mkII, de USBasp et accepte même de programmer directement une carte Arduino par sa prise USB : http://forums.futura-sciences.com/electronique/650737-de-larduino-langage-c-standard-avr-studio-13.html#post4896257 et message suivant. Voir également message #197.

Bibliothèque AVR et C

Envoyez-moi un message privé à cette adresse et je vous enverrai le lien vers ma bibliothèque AVR et C en pdf: http://forums.futura-sciences.com/private.php?do=newpm&u=621561