Installation des outils de programmation pour le microcontrôleur ATMEL AtMega128
Guide de programmation de base

INRIA Rhône-Alpes - Service Support des Expérimentations et de Développement logiciel (SED)

BRUNEAUX Jérôme

Table des matières

Chapitre 1

Pré-Requis

Les programmes utilisés pour ce manuel requièrent :
- Windows 98/NT/2000/XP
- Pack d'outils WinAVR (http://winavr.sourceforge.net/)
- Pack de librairies AvrLib (http://hubbard.engr.scu.edu/avr/avrlib/)
- Atmel AVR Studio 4 (http://www.atmel.com/)
- SP12 AVR Programmer (http://www.xs4all.nl/~sbolt/e-spider_prog.html)

La programmation du microcontrôleur requiert un programmateur parallèle dont le schéma est donné en annexe A.

Chapitre 2

Installation des outils

2.1 Ordre d'installation

L'installation doit se faire de préférence dans l'ordre donné ci-dessous afin que les librairies et outils soient automatiquement reconnus pas Atmel AVR Studio.

- Pack d'outils WinAVR
- Pack de librairies AvrLib
- SP12 AVR Programmer
- Atmel AVR Studio 4

2.2 Installation

Lors de l'installation, regrouper tous les outils dans un même répertoire (ex : c :\AVR) Ainsi, lors de la configuration, il est plus aisé de retrouver les différents composants.

L'installation des outils se fait de façon automatique et ne requiert que le réglage du répertoire d'installation pour chacun des outils. SP12 requiert le redémarrage impératif de l'ordinateur pour permettre l'installation d'un pilote d'accès au port parallèle sous Windows 2000/NT/XP.

2.3 Création d'un projet

Lorsque l'installation des logiciels set terminée, vous pouvez alors utiliser AVR Studio pour commencer un nouveau projet.

2.3.1 Lancement de AVR Studio

Au lancement de AVR Studio, vous obtenez la fenêtre de la figure 2.1.

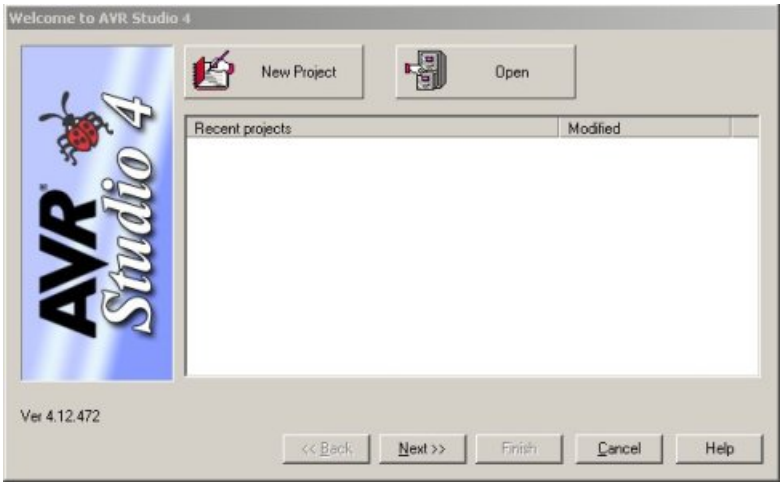

FIG. 2.1 – Fenêtre d'accueil de AVR Studio

Lorsque vous aurez créé un projet, celui-ci apparaitra dans cette fenêtre au lancement et pourra ainsi facilement être rechargé.

2.3.2 Création d'un nouveau projet

Cliquez sur 'New Project' pour démarrer un nouveau projet. Vous obtiendrez alors la fenêtre suivante :

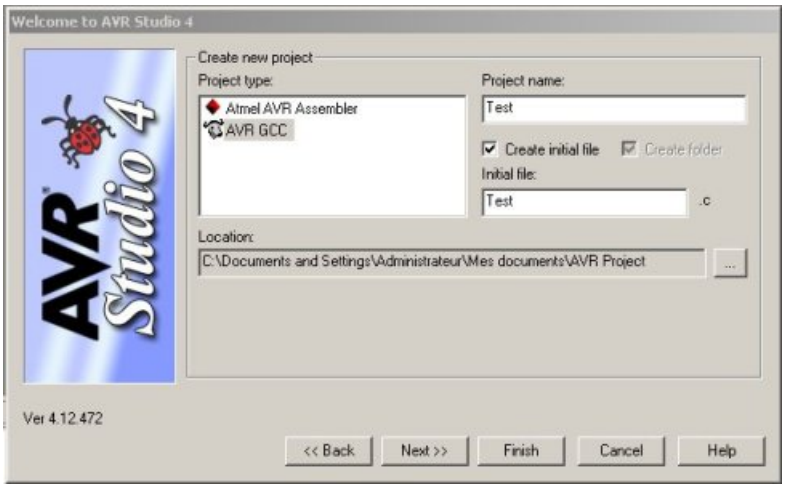

FIG. 2.2 – Sélection du type de projet

Sélectionnez 'AVR GCC' dans la liste 'Project Type'. Remplissez le champ 'Project name' et choisissez le répertoire du projet dans le champ 'Location'. Une fois ces renseignements complétés, cliquez sur 'Next'.

Vous obtiendrez la fenêtre suivante :

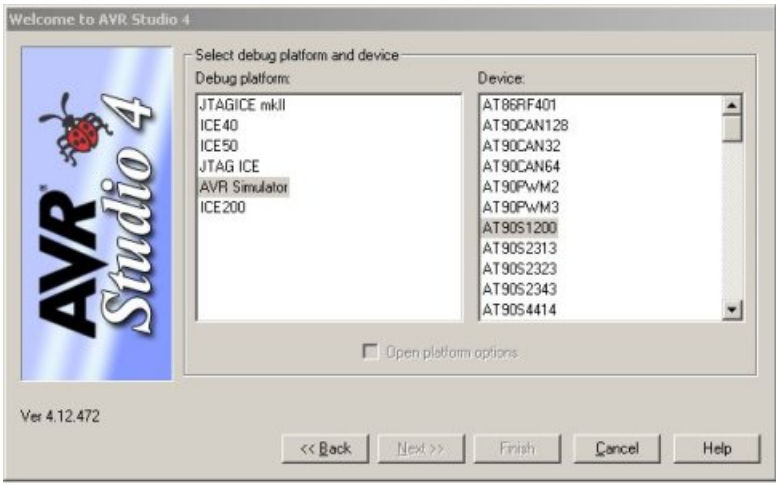

FIG. 2.3 – Sélection de la plateforme de déboguage

Dans cette fenêtre, sélectionnez 'AVR Simulator' dans la liste 'Debug Platform'. Dans la liste de droite, sélectionnez 'ATmega128' et cliquez sur le bouton 'Finish'.

Vous obtiendrez la fenêtre suivante :

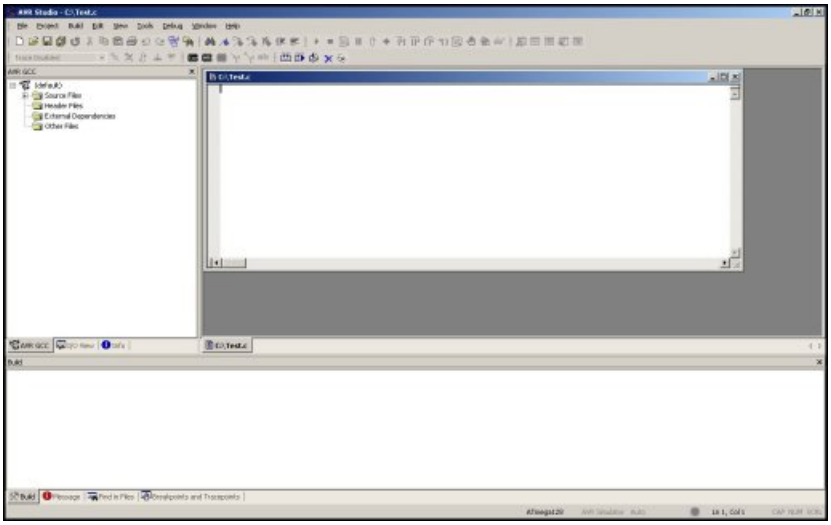

FIG. 2.4 – Fenêtre principale de AVR Studio

La création du projet est maintenant terminée. Pour tester les outils de compilations, entrez les lignes suivantes dans le fichier c :

```
int main(void)
{
    return 0;
}
```

Pour compiler ce programme, appuyez sur la touche F7 ou allez dans le menu 'Build' puis 'Build'. Si les outils sont biens intallés, la fenêtre 'Build' doit afficher ceci :

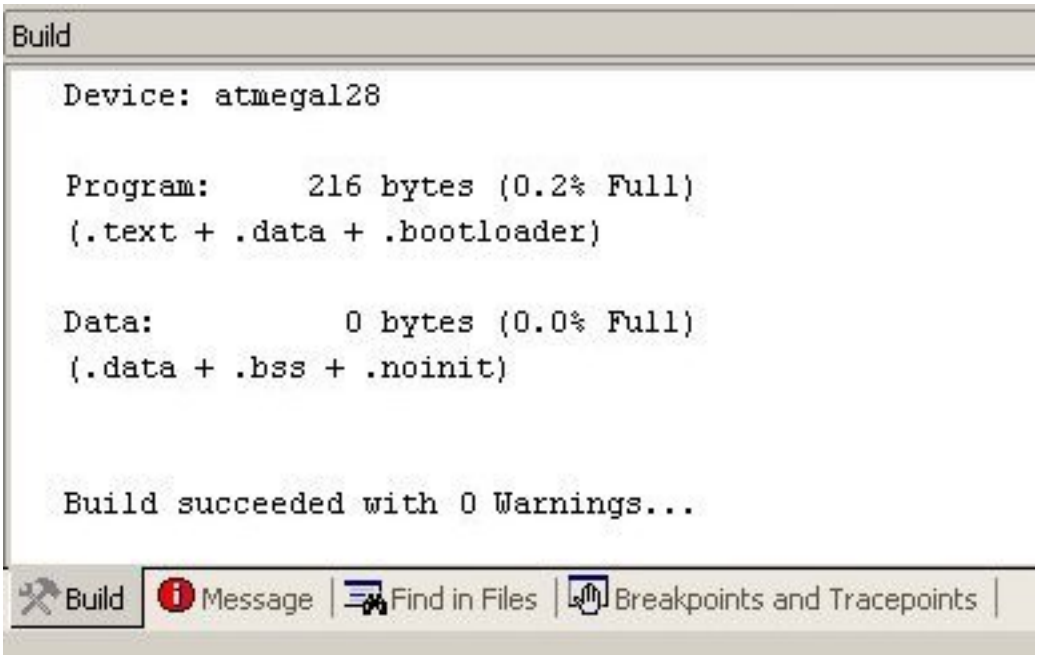

FIG. 2.5 – Log de la compilation

Ce log de compilation résume les principales informations du programme (taille, pourcentage d'utilisation de la mémoire, erreurs, avertissements, ...). Les erreurs de compilation y seront affichées de même que les avertissements.

2.4 Programmation du microcontrôleur

Lorsque le programme est complètement réalisé, il peut être programmé sur le microcontrôleur. Lors de la première programmation du microcontrôleur AtMega128, il faut régler les fusibles internes afin de permettre au microcontrôleur d'exécuter le programme.

2.4.1 Paramétrage des fusibles internes

Les paramètres des fusibles internes sont à déterminer à partir de la documentation technique du microcontrôleur ATmega128. Se référer au tableaux ci dessous reprenant ces réglages :

Lock Fuse	Byte Bit No.	Description	Default Value
BLB12	5	Boot lock bit	1 (unprogrammed)
BLB11	4	Boot lock bit	1 (unprogrammed)
BLB02	3	Boot lock bit	1 (unprogrammed)
BLB01	2	Boot lock bit	1 (unprogrammed)
LB2	1	Lock bit	1 (unprogrammed)
LB1	0	Lock bit	1 (unprogrammed)

Tab. 4.1 - Paramètres Lock Fuses

Low Fuse	Byte Bit No.	Description	Default Value
BODLEVEL	7	Brown out detector trigger level	1 (unprogrammed)
BODEN	6	Brown out detector enable	1 (unprogrammed, BOD disabled)
SUT1	5	Select start-up time	1 (unprogrammed)
SUT0	4	Select start-up time	0 (programmed)
CKSEL3	3	Select Clock source	0 (programmed)
CKSEL2	2	Select Clock source	0 (programmed)
CKSEL1	1	Select Clock source	0 (programmed)
CKSEL0	0	Select Clock source	1 (unprogrammed)

Tab. 4.2 - Paramètres Low Fuses

High Fuse	Byte Bit No.	Description	Default Value
OCDEN(4)	7	Enable OCD	1 (unprogrammed, OCD disabled)
JTAGEN(5)	6	Enable JTAG	0 (programmed, JTAG enabled)
SPIEN(1)	5	Enable Serial Program and Data Downloading	0 (programmed, SPI prog. enabled)
CKOPT(2)	4	Oscillator options	1 (unprogrammed)
EESAVE	3	EEPROM memory is preserved through the Chip Erase	1 (unprogrammed, EEPROM not preserved)
BOOTSZ1	2	Select Boot Size	0 (programmed)
BOOTSZ0	1	Select Boot Size	0 (programmed)
BOOTRST	0	Select Reset Vector	1 (unprogrammed)

Tab. 4.3 - Paramètres High Fuses

Extended Fuse	Byte Bit No.	Description	Default Value
-	7	-	1
-	6	-	1
-	5	-	1
-	4	-	1
-	3	-	1
-	2	-	1
M103C	1	ATmega103 compatibility mode	0 (programmed)
WDTON	0	Watchdog Timer always on	1 (unprogrammed)

Tab. 4.4 - Paramètres Extended Fuses

Les paramètres des fusibles à appliquer dans le cas d'une application générale sont :
- Lock Fuses : L[111111]
- Low Fuses : F[11101111]
- High Fuses : H[11011001]
- Extended fuses : X[11111111]

Pour appliquer ces paramètre, il faut utiliser SP12. Ouvrir une invite de commande puis aller dans le répertoire de SP12.
Executer la commande :

sp12 -wL111111 -wF11101111 -wH11011001 -wX0xFF

Cette commande va paramétrer les fusibles du microcontroleur et permettre l'exécution des programmes.

2.4.2 Programmation des mémoires Flash et EEPROM

Une fois le programme achevé et compilé, nous obtenons deux fichiers qui serviront à programmer le microcontrôleur. Ces fichiers sont situés dans le sous-répertoire 'Default' du répertoire du projet. Ces deux fichiers sont les fichiers %NomduProjet%.hex et %NomduProjet%.eep.

Le fichier %NomduProjet%.hex est la partie Flash du programme du microcontrôleur. Le fichier %NomduProjet%.eep est la partie EEPROM du programme du microcontrôleur.

La programmation se fait par SP12. Ouvrir une invite de commande puis aller dans le répertoire de SP12.
Executer les commandes :

sp12 -wpf %RepertoireDuProjet%\Default\%NomduProjet%.hex
sp12 -wef %RepertoireDuProjet%\Default\%NomduProjet%.eep

Annexe A

Schéma du programmeur SP12