Programmation de µContrôleurs

En environnement Windows

Ce tutoriel est la suite directe de la formation de soudure CMS. Il présente les différentes étapes à franchir afin de programmer le μ Contrôleur présent sur la carte d'initiation CMS à l'aide d'un câble USB.

1. Installation des logiciels requis

Deux suites logicielles sont requises :

- **AVR Studio** : permet d'écrire/compiler un programme <u>http://www.atmel.com/microsite/atmel_studio6/</u>

 Atmel Flip : permet de transférer un programme précédemment compilé dans le µcontrôleur, via un câble USB http://www.atmel.com/tools/FLIP.aspx

2. Création d'un programme à l'aide d'AVR Studio

Nous allons dans un premier temps créer un premier programme pour le µContrôleur embarqué sur la carte d'intro CMS. Après avoir lancé AVRStudio, cliquez sur File->New->Project.

Cliquez sur « C », puis sélectionnez C Executable Project, puis entrez un nom de projet dans la section « Name » (ici bonjour), puis enfin cliquez sur OK

New Project					? ×
Recent Templates		Sort by:	Default		Search Installed Templates
Installed Templates		GCC	C Executable Project	AVR GCC	Type: AVR GCC
Atmel Boards C 1 User Boards		ecc	C Static Library Project	AVR GCC	Creates an AVR 8-bit or 32-bit GCC project
AVR Assembler AVR Studio Solutio	n				
	3				
Name:	bonjour			_	
Location:	C:\Users\Bapou\	Docume	nts\AVRStudio	•	Browse
Solution name:	bonjour				Create directory for solution
					4 OK Cancel

Sélectionnez ensuite le μ Contrôleur qui est présent sur la carte, à savoir le modèle **AT90USB162**.

Device Family:	megaAVR, 8-bit 🔹					Search for device	۶
Name	App./Boot Memory (Kbytes)	Data Memory (bytes)	EEPROM (byte		Device Info:		
AT90CAN128	128	65280	4096	*	Device Name	AT90USB162	
AT90CAN32	32	65280	1024		Speed	0	
AT90CAN64	64	65280	2048	Ξ	Vec	27/55	
AT90PWM1	8	512	512		VCC;	2,7/3,5	
AT90PWM216	16	1024	512		Family:	megaAVR	
AT90PWM2B	8	512	512		Datashee	ets -	
AT90PWM316	16	1024	512				
AT90PWM3B	8	512	512		AVR Studio Su	upported Tools	
AT90PWM81	8	256	512		Note: AVR Drag	on	
AT90USB1286	128	65024	4096		AVRISP n	nkli	
AT90USB1287	128	65024	4096				
AT90USB162	16	512	512		AVR ONE	!	
AT90USB646	64	60928	2048		JTAGICE:	3	
AT90USB647	64	60928	2048		JTAGICE	mkli	
AT90USB82	8	512	512				
ATmega128	128	65280	4096				
ATmega1280	128	65024	4096				
ATmega1281	128	65024	4096	_			
4 4004	100	46004	+0005	-			

Le projet s'auto génère alors, et vous arrivez sur la page contenant le « main » du programme.

♦	main 👻 💽
	□/* * bonjour.c *
	* Created: 19/05/2012 21:24:35 * Author: Bapou */
	<pre>#include <avr io.h=""></avr></pre>
	<pre>Dint main(void) { while(1) { //TODO:: Please write your application code } }</pre>

Vous pouvez tenter de compiler ce programme (celui-ci est actuellement vide) afin de voir si la chaîne de compilation est correctement installée sur l'ordinateur. Pour ce faire, faites un clic droit sur le projet à droite et cliquez sur « Build... ».

	 Solution Explorer 	▼ ₽ ×
	- Coo 🖶 🕼	
	🚔 😡 Solution 'bon	jour' (1 project)
Build	bonjour	dencies
Rebuild	Dutpu	t Files
Clean	📋 bonjo	ur.c
Add	•	

La compilation s'exécute alors et le résultat s'affiche dans la fenêtre « Output » :



ATTENTION : Un bug dans la chaîne Atmel existe actuellement pour le μ Contrôleur **AT90USB162**.

Il se peut qu'à la compilation, l'erreur suivante soit générée : avr/io90usb162.h: No such file or directory

Le problème vient simplement d'une librairie qui est mal renommée. Pour corriger ce problème, rendez-vous à l'adresse suivante dans votre disque dur : C:\Program Files (x86)\Atmel\AVR Studio 5.0\AVR Toolchain\avr\include\avr

Copiez le fichier **iousb162.h** dans le même dossier et renommez-le en **io90usb162.h**.

Retentez la compilation, l'erreur doit alors avoir disparue.

Une fois le programme compilé, vous pouvez retrouver votre fichier de sortie (généralement un fichier héxadécimal .hex) dans le dossier « DEBUG » de votre projet. C'est ce fichier qui va nous servir à programmer le µContrôleur.

3. Programmation du µContrôleur via l'USB

Une fois le fichier .hex généré à partir de l'étape précédente, nous allons utiliser FLIP afin de programmer le μ Contrôleur inclut sur la carte Intro CMS à l'aide d'un câble USB.

Comprendre le bootloader :

Un μ C voit sa mémoire flash partagée en plusieurs sections. La section du bootloader contient un programme qui permet de flasher le reste de la mémoire (avec un programme custom).

Pour programmer le μ C, il va donc falloir le passer en mode bootloader. De cette façon, le μ C sera prêt à être programmé.

Le μ C **AT90USB162** contient de base un bootloader USB permettant de le programmer grâce au logiciel FLIP (et via l'USB). Lorsque le μ C ne contient aucun programme, le bootloader va automatiquement se lancer à l'allumage du μ C. Dans le cas où un programme est déjà présent dans le μ C, ce programme va s'exécuter automatiquement à la mise sous tension du μ C. Il est possible de retourner en mode bootloader en mettant la pin RESET et la pin D7 à la masse, puis de relâcher la pin RESET.

Programmer le μC :

Il faut dans un premier temps lancer le μC en mode Bootloader, puis le brancher au PC via l'USB.



Lancez FLIP et sélectionnez le type de μ C à programmer :

Puis connectez le logiciel au μ C (sélectionnez la connexion USB) :



Un message d'erreur apparaîtra si la connexion s'est mal effectuée. Peut-être est-ce parce que le mode bootloader n'est pas actif. Tentez de le relancer et retentez la connexion.

Sélectionnez ensuite le fichier hexadécimal à envoyer dans le μ C :



Vérifiez ensuite que les case « Erase », « Program » et « Verify » soient cochées. Enfin cliquez sur le bouton RUN pour lancer la procédure de programmation.

Redémarrez le µC afin de lancer le programme.