

## Cours ATMEGA

## NAVIGATION

[Brochage](#)[Architecture générale](#)[Les plans mémoires](#)[Les Fusibles](#)[Les Registres](#)[Index](#)

## Bienvenue dans la section des cours Atmel ATMEGA



Les microcontrôleurs Atmel de type ATMEGA connaissent un succès croissant due à leur facilité d'utilisation et leurs performances générales très satisfaisantes. Ils sont de plus en plus appréciés de l'industrie et du monde amateur s'intéressant au domaine de l'électronique programmable. Ce nouveaux chapitre leurs est entièrement dédiés. Ces nouveaux cours s'articuleront autour du modèle **ATMEGA32**, représentant idéal de la gamme .

En effet il est commercialisé sous un package DIL 40 broches le rendant accessible à tout électronicien qui ne souhaite pas se lancer dans l'aventure du CMS. Coté performance, il dispose d'une mémoire programme de 32Ko, un espace RAM de 2 Ko et 1024 octets d'EEProm. Les nombreux périphériques internes seront passés en détail dans les chapitres correspondants

N'hésitez pas à poser vos questions sur le Forum.

Copyright © 2004 - 2009 Les Atmel's en France - CRDD - Atmel@microprog.com

[Accueil](#) - [News](#) - [Cours](#) - [Téléchargements](#) - [Projets](#) - [Forum](#) - [Liens](#)

## NAVIGATION

[Brochage](#)[Architecture générale](#)[Les plans mémoires](#)[Les Fusibles](#)[Les Registres](#)

## Brochage

## ► Brochage de l'ATMEGA32

(XCK/T0) PB0	<input type="checkbox"/>	1	40	<input type="checkbox"/>	PA0 (ADC0)
(T1) PB1	<input type="checkbox"/>	2	39	<input type="checkbox"/>	PA1 (ADC1)
(INT2/AIN0) PB2	<input type="checkbox"/>	3	38	<input type="checkbox"/>	PA2 (ADC2)
(OC0/AIN1) PB3	<input type="checkbox"/>	4	37	<input type="checkbox"/>	PA3 (ADC3)
( $\overline{SS}$ ) PB4	<input type="checkbox"/>	5	36	<input type="checkbox"/>	PA4 (ADC4)
(MOSI) PB5	<input type="checkbox"/>	6	35	<input type="checkbox"/>	PA5 (ADC5)
(MISO) PB6	<input type="checkbox"/>	7	34	<input type="checkbox"/>	PA6 (ADC6)
(SCK) PB7	<input type="checkbox"/>	8	33	<input type="checkbox"/>	PA7 (ADC7)
RESET	<input type="checkbox"/>	9	32	<input type="checkbox"/>	AREF
VCC	<input type="checkbox"/>	10	31	<input type="checkbox"/>	GND
GND	<input type="checkbox"/>	11	30	<input type="checkbox"/>	AVCC
XTAL2	<input type="checkbox"/>	12	29	<input type="checkbox"/>	PC7 (TOSC2)
XTAL1	<input type="checkbox"/>	13	28	<input type="checkbox"/>	PC6 (TOSC1)
(RXD) PD0	<input type="checkbox"/>	14	27	<input type="checkbox"/>	PC5 (TDI)
(TXD) PD1	<input type="checkbox"/>	15	26	<input type="checkbox"/>	PC4 (TDO)
(INT0) PD2	<input type="checkbox"/>	16	25	<input type="checkbox"/>	PC3 (TMS)
(INT1) PD3	<input type="checkbox"/>	17	24	<input type="checkbox"/>	PC2 (TCK)
(OC1B) PD4	<input type="checkbox"/>	18	23	<input type="checkbox"/>	PC1 (SDA)
(OC1A) PD5	<input type="checkbox"/>	19	22	<input type="checkbox"/>	PC0 (SCL)
(ICP1) PD6	<input type="checkbox"/>	20	21	<input type="checkbox"/>	PD7 (OC2)

Comme je le souligne dans l'introduction aux cours, l'ATMEGA32 est disponible en format DIL40. Son implantation sur un circuit imprimé ne posera donc aucun problème. Cette version existe aussi en format TQFP qui est déjà un peu plus difficile à souder. La version TQFP possède une petite différence matériel : le convertisseur analogique/numérique se voit enrichi de quelques fonctions supplémentaires comme la possibilité de faire de l'amplification de signal (gain paramétrable) par rapport à une broche donnée...

A première vue on distingue immédiatement les 4 ports parallèles. Ils sont complets et nous laisse donc disposer de 32 broches d'entrées/sorties.

## ► Fonctions principales et auxiliaires des ports et broches

Tous les ports et broches peuvent être configurés en entrées ou sorties numériques de manière individuelle. Certaines broches ont quelquefois une fonction secondaire, voire même un triple emploi. Pour utiliser les fonctions auxiliaires, il faudrait toujours les configurer de manière logicielle.

N°	Nom	Fonctions Auxiliaires
1	PB0	<b>T0</b> : Source d'horloge externe du compteur/timer 0 <b>XCK</b> : Source d'horloge externe pour L'USART quand il est utilisé en mode synchrone
2	PB1	<b>T1</b> : Source d'horloge externe du compteur/timer 1
3	PB2	<b>AIN0</b> : Entrée positive du comparateur analogique <b>INT2</b> : Entrée d'interruption externe N°2
4	PB3	<b>AIN1</b> : Entrée négative du comparateur analogique <b>OC0</b> : Sortie lors d'une comparaison réussie pour le compteur/timer 0, soit pour le mode PWM

5	PB4	<b>SS</b> : Entrée pour selection du mode esclave de la SPI
6	PB5	<b>MOSI</b> : Sortie des données en mode SPI maitre, entrée en mode esclave
7	PB6	<b>MISO</b> : Entrée des données en mode SPI maitre, sortie en mode esclave
8	PB7	<b>SCK</b> : Sortie d'horloge de synchronisation en mode SPI maitre, entrée horloge en mode esclave
9	RESET-	Un niveau bas sur cette broche effectue un reset du microcontrolleur
10	VCC	Alimentation positive du microcontrolleur
11	GND	Alimentation négative ou "Masse"
12	XTAL1	Entrée d'horloge externe ou entrée de l'inverseur de l'amplificateur de l'oscillateur
13	XTAL2	Sortie de l'inverseur de l'amplificateur de l'oscillateur
14	PD0	<b>RXD</b> : Entrée des données de l'interface de communication USART
15	PD1	<b>TXD</b> : Sortie des données de l'interface de communication USART
16	PD2	<b>INT0</b> : Entrée d'interruption externe "0"
17	PD3	<b>INT1</b> : Entrée d'interruption externe "1"
18	PD4	<b>OC1B</b> : Sortie signal lors d'une comparaison réussie avec le registre OCR1B du Timer 1 ; Sert aussi de sortie PWM
19	PD5	<b>OC1A</b> : Sortie signal lors d'une comparaison réussie avec le registre OCR1A du Timer 1 ; Sert aussi de sortie PWM
20	PD6	<b>ICP1</b> : Entrée de capture pour le Timer/compteur 1
21	PD7	<b>OC2</b> : Sortie signal lors d'une comparaison réussie avec le registre OCR12 du Timer 2 ; Sert aussi de sortie PWM
22	PC0	<b>SCL</b> : Signal d'horloge de l'interface de communication 2 fils (I2C)
23	PC1	<b>SDA</b> : Signal de données de l'interface de communication 2 fils (I2C)
24	PC2	<b>TCK</b> : Signal d'horloge de l'interface JTAG
25	PC3	<b>TMS</b> : Signal Mode select de l'interface JTAG
26	PC4	<b>TDO</b> : Sortie des données de l'interface JTAG
27	PC5	<b>TDI</b> : Entrée des données de l'interface JTAG
28	PC6	<b>TOSC1</b> : broche de quartz pour horloge externe du Timer2
29	PC7	<b>TOSC2</b> : broche de quartz pour horloge externe du Timer2
30	AVCC	Alimentation positive du onvertisseur Analogique / Digital
31	GND	Masse du convertisseur Analogique / Digital
32	AREF	Référence de tension externe pour le convertisseur Analogique / Digital
33	PA7	<b>ADC7</b> : Entrée du signal analogique voie 7 du convertisseur Analogique / Digital
34	PA6	<b>ADC6</b> : Entrée du signal analogique voie 6 du convertisseur Analogique / Digital
35	PA5	<b>ADC5</b> : Entrée du signal analogique voie 5 du convertisseur Analogique / Digital
36	PA4	<b>ADC4</b> : Entrée du signal analogique voie 4 du convertisseur Analogique / Digital
37	PA3	<b>ADC3</b> : Entrée du signal analogique voie 3 du convertisseur Analogique / Digital
38	PA2	<b>ADC2</b> : Entrée du signal analogique voie 2 du convertisseur Analogique / Digital
39	PA1	<b>ADC1</b> : Entrée du signal analogique voie 1 du convertisseur Analogique / Digital
40	PA0	<b>ADC0</b> : Entrée du signal analogique voie 0 du convertisseur Analogique / Digital

Copyright © 2004 - 2006 tous droits réservés - GROUP - ATmicroprog.com

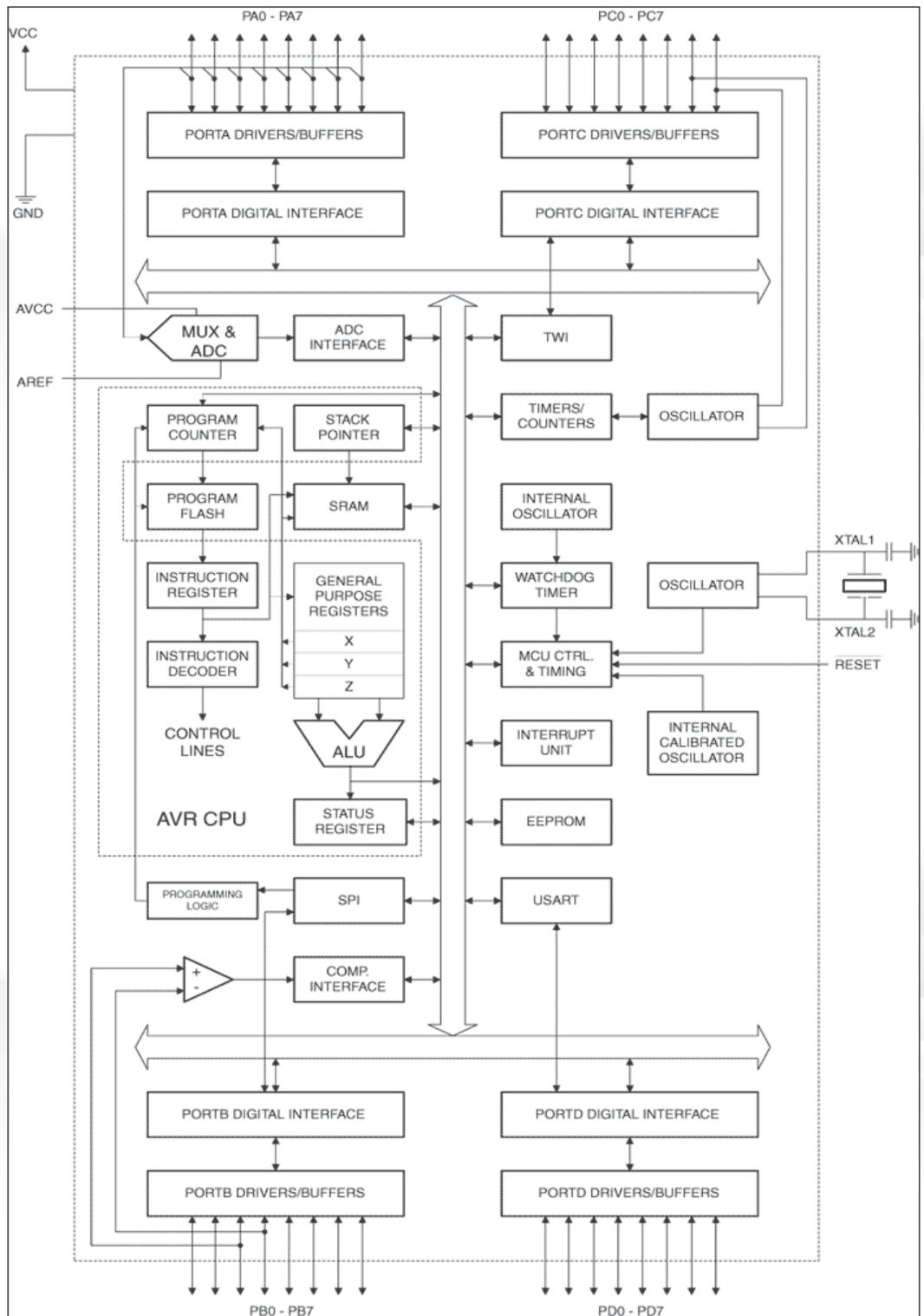
## Cours ATMEGA

## NAVIGATION

[Brochage](#)[Architecture générale](#)[Les plans mémoires](#)[Les Fusibles](#)[Les Registres](#)

## Architecture générale

## Schématique interne



Le cœur du noyau est basé sur une architecture RISC (Circuit à jeu d'instructions réduit), qui compte environ 120 instructions. Ce nombre d'instructions est tout de même assez élevé pour une telle architecture, mais la dénomination de RISC prend en compte d'autres paramètres qui font que ces microcontrôleurs sont ainsi classifiés. L'horloge interne n'est

pas divisé comme sur les classiques 80C51, ce qui donne une vitesse de traitement rapide, par exemple pour un Quartz de 10 Mhz, 10 Millions d'instructions par secondes (MIPS) sont effectués, la majorité des instructions étant réalisés en 1 ou 2 cycles, les microcontrôleurs AVR sont donc très rapide !

Le schéma de l'architecture interne de l'ATMEGA32 nous permet de remarquer les périphériques suivants :

- 2 Timer/Compteur 8 bits avec facteur de pré-division indépendant
- 1 Timer/Compteur 16 bits avec facteur de pré-division indépendant
- 1 Horloge Temps réel avec quartz externe
- 4 canaux PWM
- 1 Convertisseur Analogique/Numérique 8 voies avec une résolution de 10 bits
- 1 Comparateur Analogique à 2 entrées configurables
- 1 Interface de communication synchrone SPI (servant aussi à la programmation In-situ)
- 1 Interface de communication Asynchrone USART
- 1 Compteur Watchdog programmable
- 1 Ports JTAG pour le débogage en temps réel

La liste ci-dessus énumère la liste des périphériques qui ont un rôle majeur dans la communication avec le monde extérieur au microcontrôleur. Il faut compter aussi sur d'autres fonctions typiquement internes, comme le détecteur de Brown-out, les nombreuses sources d'interruption ainsi que les différents type de mémoires...

Nous apprendrons au fil des cours à nous servir de chacun d'entre eux.

Copyright © 2004 – 2008 Tous droits réservés – SRDD – Atmicroprog.com

[Accueil](#) - [News](#) - [Cours](#) - [Téléchargements](#) - [Projets](#) - [Forum](#) - [Liens](#)

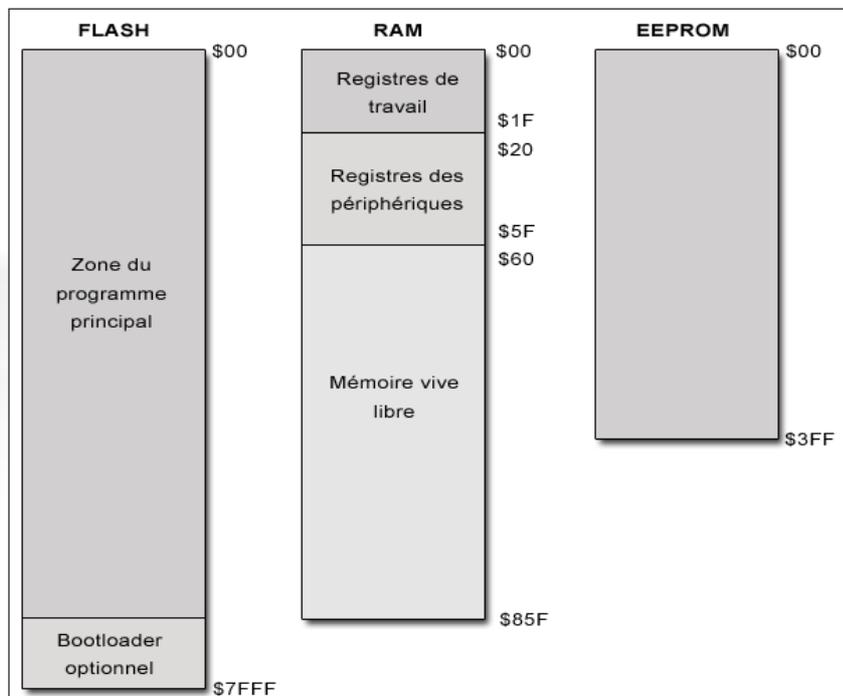
## Cours ATMEGA

## NAVIGATION

[Brochage](#)[Architecture générale](#)[Les plans mémoires](#)[Les Fusibles](#)[Les Registres](#)

## Plans mémoires

## ORGANISATION DES DIFFERENTS PLANS MEMOIRES



L'ATMEGA32 est doté de 3 types de mémoires principales :

- La FLASH** : C'est dans cette mémoire que votre programme est "gravé". C'est une mémoire qui garde son contenu lorsqu'elle n'est plus sous tension. Sa particularité est d'être réinscriptible, ce qui permet de l'effacer et de la reprogrammer environ 10 000 fois. Sa capacité est de 32 Ko. (16K\*16. La représentation est tronquée en 32K\*8)  
 A la fin de cette espace mémoire on remarque une petite zone qui occupe un espace pré-défini qui sert au BOOTLOADER. Le BOOTLOADER est un petit morceau de programme qui s'exécute avant votre propre programme afin d'effectuer diverses opérations. Comme les instructions utilisées à ce niveau peuvent écrire ou effacer les cellules mémoires de la flash, on peut très bien imaginer une petite application permettant la mise à jour du programme via l'interface série du microcontrôleur.
- La RAM** : La ram est une mémoire qui perd son contenu quand elle n'est plus alimentée. Cet emplacement permet de mémoriser différentes informations temporaires lors de l'exécution d'un programme, comme par exemple le résultat d'une opération mathématique qui servira à un moment donné pour une prise de décision du programme. Elle sert aussi à contenir l'adresse de retour lors d'un appel à un sous programme ou d'une routine d'interruption. Attention de ne pas faire chevaucher informations et adresse de retour du pointeur de pile aux memes adresses sous peine de voir son programme "bugger".  
 Le début de l'espace mémoire est occupé par 2 zones qui contiennent les registres; nous nous pencherons sur la fonction de ces Registres un peu plus bas  
 Vous pouvez lire et écrire autant de fois que vous le souhaitez. Sa capacité est de 2 Ko.
- L'EEPROM** : C'est une mémoire qui conserve les données lors de coupure de courant. Elle permet de mémoriser des informations qui doivent être rechargées à la mise sous tension, comme par exemple le point de consigne d'une température pour un régulateur. Sa durée de vie n'est pas infinie, donc n'écrivez pas à outrance dans cet emplacement.  
 Elle est donnée pour un cycle garantie de 100 000 opérations d'écritures ou d'effacements. Sa capacité est de 1 Ko.

## Cours ATMEGA

## NAVIGATION

Brochage

Architecture générale

Les plans mémoires

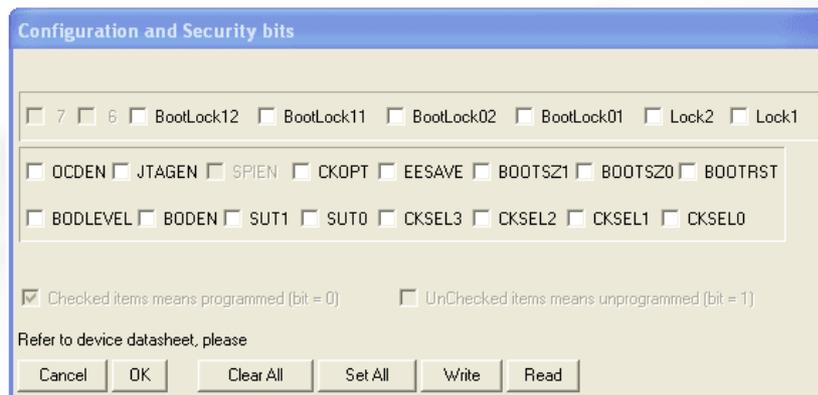
Les Fusibles

Les Registres

## Les fusibles

## Introduction

Les fusibles ont toujours existé dans la famille AVR ne serait ce que pour la protection contre le téléchargement de programme afin d'empêcher la duplication ou le désassemblage; bref de quoi verrouiller le programme de toute curiosité industrielle. La classe ATMEGA c'est vue doté de nouveaux fusibles permettant de multiples réglages : source et vitesse d'horloge, détecteur de sous-tension (brown-out)... d'ou un total de 21 Fusibles :



Copie d'écran des fusibles sur un ATMEGA32 avec le logiciel Pony Prog

Nous étudierons ici l'intégralité des fusibles que possède l'AMEGA32 afin de mettre en service le programme correctement.

Avant de commencer la description de chaque fusible, il est important de revenir sur jargon utilisé par Atmel :

Un fusible programmé à pour valeur "0"

Un fusible non programmé à pour valeur "1"

On peut donc retenir que la logique intellectuelle est **inversée**.

## Les Fusibles

Les valeurs par défauts seront indiquées par une couleur **rouge**

LES FUSIBLES DE PROTECTION

- **Bootlock2, Bootlock1** : Il s'agit d'une paire de fusibles destinés à contrôler les droits d'accès aux mémoires FLASH et EEPROM.

Bootlock2	Bootlock1	Type de protection
1	1	Aucune restriction d'accès aux mémoires.
1	0	Aucune programmation possible de la FLASH ou de l'EEPROM par programmation série et parallèle. La programmation des fusibles est désactivée
0	0	Aucune programmation et vérification possible de la FLASH ou de l'EEPROM par programmation série et parallèle. La programmation des fusibles est désactivée

- **Bootlock02, Bootlock01** : Il s'agit d'une paire de fusibles destinés à contrôler les droits d'accès de la section application (Zone de programme principale) par les instructions SPM (Store Program Memory) et LPM (Load Program Memory)

Bootlock02	Bootlock01	Type de protection
1	1	Aucune restriction pour accéder à la section application à l'aide des instructions SPM ou LPM.

1	0	L'instruction SPM ne peut pas écrire dans la section application.
0	0	l'instruction SPM ne peut pas écrire dans la section application et LPM exécutée depuis le bootloader ne pas lire dans la section application. Si des vecteurs d'interruptions sont placés dans la section bootloader, les interruptions ne pourront pas être exécutées depuis la section application.
0	1	L'instruction LPM exécutée depuis le bootloader de ne pas lire dans la section application et si des vecteurs d'interruptions sont placés dans la section bootloader, les interruptions ne pourront pas être exécutées depuis la section application.

- **Bootlock12, Bootlock11** : Il s'agit d'une paire de fusibles destinés à contrôler les droits d'accès à la section du bootloader par les instructions SPM (Store Program Memory) et LPM (Load Program Memory)

Bootlock12	Bootlock11	Type de protection
1	1	Aucune restriction pour accéder à la section du bootloader à l'aide des instructions SPM ou LPM.
1	0	L'instruction SPM ne peut pas écrire dans la section bootloader.
0	0	l'instruction SPM ne peut pas écrire dans la section bootloader et LPM exécutée dans l'application de ne pas lire dans la section du bootloader. Si des vecteurs d'interruptions sont placés dans la section application, les interruptions ne pourront pas être exécutées depuis le bootloader.
0	1	L'instruction LPM exécutée dans l'application de ne pas lire dans la section du bootloader. Si des vecteurs d'interruptions sont placés dans la section application, les interruptions ne pourront pas être exécutées depuis le bootloader.

### LES FUSIBLES "HAUT"

Fuse	Description	Valeur par défaut
<b>OCDEN</b>	Activation de OCD. Ce fusebit permet d'activer quelques fonctionnalités du système d'horloge pour fonctionner dans les modes sommeil. Ce fusebit fonctionne généralement avec le JTAG. Je vous conseille ne pas changer sa valeur par défaut.	1 - Désactivé
<b>JTAGEN</b>	Activation du JTAG. Le JTAG est une interface de programmation et de communication. Elle permet la programmation du microcontrôleur tout comme la programmation ISP et le débogage des applications en temps réel. Elle nécessite cependant une interface électronique particulière.	0 - JTAG activée
<b>SPIEN</b>	Activation du SPI pour la programmation et la lecture du programme. Ne pas changer la valeur de ce fusebit sous peine de ne plus pouvoir le programmer par la SPI	0 - SPI activée
<b>CKOPT</b>	Options de l'oscillateur (voir mode horloge).	1
<b>EESAVE</b>	Préservation du contenu de l'EEPROM lors de l'effacement du micro.	1 - contenu non préservé
<b>BOOTSZ1</b>	Sélection de la taille du secteur de Boot (voir tableau suivant).	0
<b>BOOTSZ0</b>	Sélection de la taille du secteur de Boot (voir tableau suivant).	0
<b>BOOTRST</b>	Sélection du vecteur de Reset; soit au vecteur \$0000 ou Bootloader. Si un bootloader est écrit, faire démarrer le programme au secteur du Bootloader	1 - vecteur \$0000

### Paramétrage des Fusibles BOOTSZ :

BOOTSZ1	BOOTSZ0	Taille du secteur de BOOT	Section attribuée au programme	Section attribuée au Bootloader
1	1	256 words	\$0000 - \$3EFF	\$3F00 - \$3FFF
1	0	512 words	\$0000 - \$3DFF	\$3E00 - \$3FFF
0	1	1024 words	\$0000 - \$3BFF	\$3C00 - \$3FFF
0	0	2048 words	\$0000 - \$37FF	\$3800 - \$3FFF

### LES FUSIBLES "BAS"

Fuse	Description	Valeur par défaut
<b>BODLEVEL</b>	Choix du niveau trigger du BOD. il s'agit tout simplement de la tension sous laquelle le trigger enclenche le BOD : 1 = 2,7 Volts ; 0 = 4 Volts	1 - 2,7 Volts

<b>BODEN</b>	Activation du BOD ( Brown-Out Detector) Le BOD est l'équivalent du chien de garde externe. Il surveille la tension d'alimentation. En cas de franchissement inférieure du point de consigne donné par BODLEVEL, le BOD met le microcontrôleur hors service (RESET). Cela permet au démarrage d'éviter les corruptions mémoire, puis en fonctionnement, de corrompre le fonctionnement de périphériques reliés au microcontrôleur (LCD, mémoire de masse ect...) C'est une sécurité digne des applications industrielles.	<b>0</b> - BOD désactivé
<b>SUT1</b>	Sélection du temps de démarrage	<b>1</b>
<b>SUT0</b>	Sélection du temps de démarrage	<b>0</b>
<b>CKSEL3</b>	Sélection de la source d'horloge	<b>0</b>
<b>CKSEL2</b>	Sélection de la source d'horloge	<b>0</b>
<b>CKSEL1</b>	Sélection de la source d'horloge	<b>0</b>
<b>CKSEL0</b>	Sélection de la source d'horloge	<b>1</b>

Les microcontrôleurs ATMEGA possèdent une multitude de mode oscillatoire. Ainsi, ils sont livrés par défaut sur une source d'horloge interne calibrée à 1 MHz. Donc la première opération à faire sera de régler les Fusebits suivant les paramètres physique réel. Certain d'entre vous trouverons un peu "lourd" la configuration de ces fusibles, mais ne perdez pas de vue les différents avantages énumérés depuis le début de ce chapitre. Intéressons nous maintenant aux différents modes d'horloges. A noter que seulement les modes les plus utilisés seront décrits.

### ► Le mode oscillateur RC interne calibré

Ce mode permet de se servir de l'oscillateur interne de type RC à des vitesses pré-définies de 1,2,4,8 Mhz. Ces vitesses sont données pour une tension d'alimentation de 5V et 25°C. Les fusibles **CKOPT** doivent toujours être sélectionnés sur "non-programmé".

La calibration de l'oscillateur ne sera pas abordé dans ce chapitre.

#### Sélection de la fréquence d'oscillation :

CKSEL3	CKSEL2	CKSEL1	CKSEL0	Fréquence (MHZ)
<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	1
<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	2
<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	4
<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	8

#### Mode de démarrage :

SUT1	SUT0	Temps de démarrage après mode POWER-DOWN et POWER-SAVE	Temps de démarrage après RESET	Mode
<b>0</b>	<b>0</b>	6 CK	-	BOD Activé
<b>0</b>	<b>1</b>	6 CK	4.1 ms	Fast Rising Power
<b>1</b>	<b>0</b>	6 CK	65 ms	Slowly Rising Power
<b>1</b>	<b>1</b>	<b>RESERVE</b>		

### ► Le mode oscillateur Externe à Quartz

Il s'agit certainement du mode le plus utilisé. Je tenais juste à préciser un point bien précis concernant l'utilisation du fusible **CKOPT** :

ce bit permet de sélectionner 2 mode de fonctionnement différents; il agit sur l'amplificateur d'horloge : quand il est programmé, la sortie de l'oscillateur fonctionne en mode rail to rail, soit la capacité maximum. Ce mode est préférable dans les environnements "bruyants" (entendez par là, perturbation électriques en tout genres) ou quand la sortie XTAL2 pilote une autre source d'horloge. Si il n'est pas programmé, l'amplification est moindre, et ne permettra pas de pilotage externe d'une autre source d'horloge, en contre partie, le circuit d'oscillation consommera moins d'énergie et à une fréquence de fonctionnement limitée. Atmel préconise de programmer ce fusible pour 16 Mhz de fonctionnement avec un résonateur.

#### Sélection de la fréquence d'oscillation :

CKOPT	CKSEL3	CKSEL2	CKSEL1	Plage de Fréquence (MHZ)	Capacité d'oscillation recommandée
<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	0.4 - 0.9 *	-
<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	0.9 - 3.0	12 - 22 pf
<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	3.0 - 8.0	12 - 22 pf
<b>0</b>	<b>101,110,111</b>			1.0 <=	12 - 22 pf

\* Seulement avec résonateur céramique.

#### Mode de démarrage :

CKSEL0	SUT1	SUT0	Temps de démarrage après mode POWER-DOWN et POWER-SAVE	Temps de démarrage après RESET	Mode
1	0	1	16K CK	-	BOD activé
1	1	0	16K CK	4.1 ms	fast rising power
1	1	1	16K CK	65 ms	slowly rising power

Pour tout les autres modes, je vous renvoie au datasheet correspondant. Avant de clore le chapitre, je souhaiterai vous sensibiliser sur la qualité du "parcours" que doivent emprunter les signaux d'horloge sur le circuit imprimé : les pistes devront êtres les plus courtes possibles, évitez les cassures d'angle de piste trop nombreuses, et enfin, arrangez vous pour que les condensateurs soient toujours au plus près du quartz.

Copyright © 2004 - 2008 Tous droits réservés - SRDD - ATmicroprog.com

[Accueil](#) - [News](#) - [Cours](#) - [Téléchargements](#) - [Projets](#) - [Forum](#) - [Liens](#)

## NAVIGATION

[Brochage](#)[Architecture générale](#)[Les plans mémoires](#)[Les Fusibles](#)[Les Registres](#)

## Les Registres

## ▶ La zone des registres

Comme nous l'avons vu précédemment, ce microcontrôleur dispose de 3 plans mémoires. Cependant comme vous l'avez constaté, une partie de la RAM est utilisée pour servir d'emplacement aux registres. Un registre est un espace mémoire réservé à une utilisation spécifique; Ils sont l'interface entre vous et le noyau du microcontrôleur.

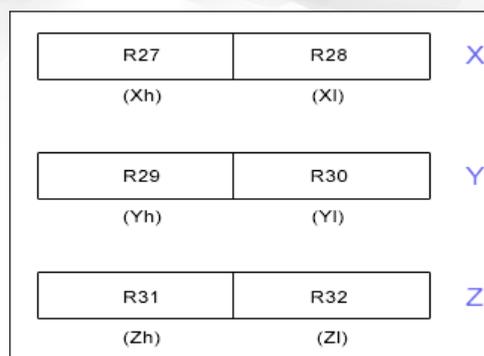
Comme ces registres sont accessibles très rapidement à l'aide du jeu d'instructions assembleur, leurs fonctions premières est principalement de réaliser des opérations mathématiques, des manipulations logiques et des déplacements de données. Certains sont organisés en paires (16 bits) afin de pouvoir pointer la totalité de l'espace adressable.

On compte 32 registres de travail s'appelant R0, R1... R32. Les registres formés en paires s'appellent les registres d'indirection X,Y,Z. Bien entendu il sont individuellement adressable mais lors de leurs utilisations en pointeur de mémoires, ils fonctionneront comme un compteur 16 Bits :

Plan des registres R0 à R31

R0	\$00
R1	\$01
R2	\$02
R3	\$03
R4	\$04
....	
R16	\$10
R17	\$11
R18	\$12
....	
R26	\$1A
R27	\$1B
R28	\$1C
R29	\$1D
R30	\$1E
R31	\$1F

Topographie des registres d'indirection X,Y,Z



- Les registres des périphériques :

Ces registres permettent d'accéder à la configuration des périphériques intégrés (USART,CAN,SPI...) et d'obtenir des états sur leurs fonctionnements. Nous détaillerons la fonction de chacun dans les chapitres correspondants. Le tableau ci dessous énumère le nom de ses registres de l'ATMEGA32 ainsi que les drapeaux qu'ils contiennent :

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
\$3F (\$5F)	SREG	I	T	H	S	V	N	Z	C
\$3E (\$5E)	SPH	–	–	–	–	SP11	SP10	SP9	SP8
\$3D (\$5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0
\$3C (\$5C)	OCR0	Timer/Counter0 Output Compare Register							
\$3B (\$5B)	GICR	INT1	INT0	INT2	–	–	–	IVSEL	IVCE
\$3A (\$5A)	GIFR	INTF1	INTF0	INTF2	–	–	–	–	–
\$39 (\$59)	TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0
\$38 (\$58)	TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0
\$37 (\$57)	SPMCR	SPMIE	RWWSB	–	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN
\$36 (\$56)	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
\$35 (\$55)	MCUCR	SE	SM2	SM1	SM0	ISC11	ISC10	ISC01	ISC00
\$34 (\$54)	MCUCSR	JTD	ISC2	–	JTRF	WDRF	BORF	EXTRF	PORF
\$33 (\$53)	TCCR0	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00
\$32 (\$52)	TCNT0	Timer/Counter0 (8 Bits)							
\$31 <sup>(1)</sup> (\$51) <sup>(1)</sup>	OSCCAL	Oscillator Calibration Register							
	OCDR	On-Chip Debug Register							
\$30 (\$50)	SFIOR	ADTS2	ADTS1	ADTS0	–	ACME	PUD	PSR2	PSR10
\$2F (\$4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10
\$2E (\$4E)	TCCR1B	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10
\$2D (\$4D)	TCNT1H	Timer/Counter1 – Counter Register High Byte							
\$2C (\$4C)	TCNT1L	Timer/Counter1 – Counter Register Low Byte							
\$2B (\$4B)	OCR1AH	Timer/Counter1 – Output Compare Register A High Byte							
\$2A (\$4A)	OCR1AL	Timer/Counter1 – Output Compare Register A Low Byte							
\$29 (\$49)	OCR1BH	Timer/Counter1 – Output Compare Register B High Byte							
\$28 (\$48)	OCR1BL	Timer/Counter1 – Output Compare Register B Low Byte							
\$27 (\$47)	ICR1H	Timer/Counter1 – Input Capture Register High Byte							
\$26 (\$46)	ICR1L	Timer/Counter1 – Input Capture Register Low Byte							
\$25 (\$45)	TCCR2	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20
\$24 (\$44)	TCNT2	Timer/Counter2 (8 Bits)							
\$23 (\$43)	OCR2	Timer/Counter2 Output Compare Register							
\$22 (\$42)	ASSR	–	–	–	–	AS2	TCN2UB	OCR2UB	TCR2UB
\$21 (\$41)	WDTCR	–	–	–	WDTOE	WDE	WDP2	WDP1	WDP0
\$20 <sup>(2)</sup> (\$40) <sup>(2)</sup>	UBRRH	URSEL	–	–	–	–	UBRR[11:8]		
	UCSRC	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL
\$1F (\$3F)	EEARH	–	–	–	–	–	–	EEAR9	EEAR8
\$1E (\$3E)	EEARL	EEPROM Address Register Low Byte							
\$1D (\$3D)	EEDR	EEPROM Data Register							
\$1C (\$3C)	EEDR	–	–	–	–	EERIE	EEMWE	EWE	EERE
\$1B (\$3B)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0
\$1A (\$3A)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0
\$19 (\$39)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0
\$18 (\$38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0
\$17 (\$37)	DRB	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
\$16 (\$36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0
\$15 (\$35)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0
\$14 (\$34)	DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0
\$13 (\$33)	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0
\$12 (\$32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0
\$11 (\$31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0
\$10 (\$30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0
\$0F (\$2F)	SPDR	SPI Data Register							
\$0E (\$2E)	SPSR	SPIF	WCOL	–	–	–	–	–	SPI2X
\$0D (\$2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
\$0C (\$2C)	UDR	USART I/O Data Register							
\$0B (\$2B)	UCSRA	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM
\$0A (\$2A)	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8
\$09 (\$29)	UBRRL	USART Baud Rate Register Low Byte							
\$08 (\$28)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0
\$07 (\$27)	ADMUX	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0
\$06 (\$26)	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0
\$05 (\$25)	ADCH	ADC Data Register High Byte							
\$04 (\$24)	ADCL	ADC Data Register Low Byte							
\$03 (\$23)	TWDR	Two-wire Serial Interface Data Register							
\$02 (\$22)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE
\$01 (\$21)	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0
\$00 (\$20)	TWBR	Two-wire Serial Interface Bit Rate Register							