

## Introduction (1/4)

Un **microcontrôleur** est un processeur dédié au contrôle, il contient généralement les fonctions suivantes :

- **Control Process Unit (CPU)** est l'Unité de calcul Arithmétique et Logique
- **Timer**
- **Traitement de valeurs analogiques numériques**
- **PORTS** d'E/S
- **Interfaces communication série.**

Présentation générale

Introduction

Info Indus 8 bits

Delfieu

Présentation générale

**Introduction**

Architecture

Mémoire

Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique

Convertisseur Analogique-Numérique

: **ADC**

Les Registres de l'ADC

Les **timers**

Le **timer** 0

Le pré-diviseur de  $T_0$

et  $T_1$

Le **timer** 1

Les ITs du **timer** 1

Les modes de la **MLI**

Programmation

Langage C

Fonctions

Fonction de

mise en oeuvre d'une

interruption

## Introduction (2/4)

La mémoire de l'**ATMEGA<sub>8</sub>** est constituée des blocs suivant :

- 1 k octets de Mémoire vive (**SRAM**)  $\Rightarrow$  : registres et pile système.
- 8 k octet **FLASH** (10.000 cycles)  $\Rightarrow$  programme et données statiques
- 512 octet d'**EEPROM** : 100.000 cycles  $\Rightarrow$  données permanentes

Présentation générale

Introduction

Info Indus 8 bits

Delfieu

Présentation générale

**Introduction**

Architecture

Mémoire

Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique

Convertisseur Analogique-Numérique

: **ADC**

Les Registres de l'ADC

Les **timers**

Le **timer** 0

Le pré-diviseur de  $T_0$

et  $T_1$

Le **timer** 1

Les ITs du **timer** 1

Les modes de la **MLI**

Programmation

Langage C

Fonctions

Fonction de

mise en oeuvre d'une

interruption

# Introduction (3/4)

- **Timer**
  - Base de temps ;
  - comptage d'événements ;
  - **MLI**.
- Valeurs analogiques
  - **ADC** : 0-5v  $\Rightarrow$  mots de 10 bits
  - Comparateur Analogique [0, 5v]
- **Interfaces**
  - **RS232** : Loader un programme ;
  - Communiquer avec d'autres composants :
    - \* asynchrone : **USART** : Communication synchrone ou asynchrone
    - \* synchrone : **SPI** : Protocole série
    - \* synchrone : **TWI (I<sup>2</sup>C)** : Protocole série

Présentation générale

Introduction

Info Indus 8 bits

Delfieu

Présentation générale

**Introduction**

Architecture

Mémoire

Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique

Convertisseur Analogique-Numérique : **ADC**

Les Registres de l'ADC

Les **timers**

Le **timer** 0

Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer** 1

Les ITs du **timer** 1

Les modes de la **MLI**

Programmation

Langage C

Fonctions

Fonction de **manipulation** de bits

Mise en oeuvre d'une interruption

# Introduction (3/4)

- **Timer**
  - Base de temps ;
  - comptage d'événements ;
  - **MLI**.
- Valeurs analogiques
  - **ADC** : 0-5v  $\Rightarrow$  mots de 10 bits
  - Comparateur Analogique [0, 5v]
- **Interfaces**
  - **RS232** : Loader un programme ;
  - Communiquer avec d'autres composants :
    - \* asynchrone : **USART** : Communication synchrone ou asynchrone
    - \* synchrone : **SPI** : Protocole série
    - \* synchrone : **TWI (I<sup>2</sup>C)** : Protocole série

Présentation générale

Introduction

Info Indus 8 bits

Delfieu

Présentation générale

**Introduction**

Architecture

Mémoire

Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique

Convertisseur Analogique-Numérique : **ADC**

Les Registres de l'ADC

Les **timers**

Le **timer** 0

Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer** 1

Les ITs du **timer** 1

Les modes de la **MLI**

Programmation

Langage C

Fonctions

Fonction de **manipulation** de bits

Mise en oeuvre d'une interruption

## Introduction (3/4)

- **Timer**
  - Base de temps ;
  - comptage d'événements ;
  - **MLI**.
- Valeurs analogiques
  - **ADC** : 0-5v ⇒ mots de 10 bits
  - Comparateur Analogique [0, 5v]
- **Interfaces**
  - **RS232** : Loader un programme ;
  - Communiquer avec d'autres composants :
    - \* asynchrone : **USART** : Communication synchrone ou asynchrone
    - \* synchrone : **SPI** : Protocole série
    - \* synchrone : **TWI (I<sup>2</sup>C)** : Protocole série

Présentation générale

Introduction

Info Indus 8 bits

Delfieu

Présentation générale

**Introduction**

Architecture

Mémoire

Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique

Convertisseur Analogique-Numérique : **ADC**

Les Registres de l'ADC

Les **timers**

Le **timer** 0

Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer** 1

Les ITs du **timer** 1

Les modes de la **MLI**

Programmation

Langage C

Fonctions

Fonction de mise en oeuvre d'une interruption

3/122 de bits

Mise en oeuvre d'une interruption

## Introduction (4/4)

Puce	Flash	E <sup>2</sup>	RAM	I0	MLI	Interfaces	CAN
ATM <sub>8</sub>	8K	512	1k	23	3	SPI-USART	10
ATM <sub>16</sub>	16K	512	1k	32	4	SPI-USART	10
ATM <sub>32</sub>	32K	1k	2k	32	4	SPI - USART	10
ATM <sub>64</sub>	64K	2k	4k	53	8	SPI - USART(2)	10
ATM <sub>128</sub>	128K	4k	4k	53	8	SPI - USART(2)	10
ATM <sub>256</sub>	256K	4k	8k	53	16	SPI - USART(2)	10

Présentation générale

Introduction

Info Indus 8 bits

Delfieu

Présentation générale

**Introduction**

Architecture

Mémoire

Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique

Convertisseur Analogique-Numérique : **ADC**

Les Registres de l'ADC

Les **timers**

Le **timer** 0

Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer** 1

Les ITs du **timer** 1

Les modes de la **MLI**

Programmation

Langage C

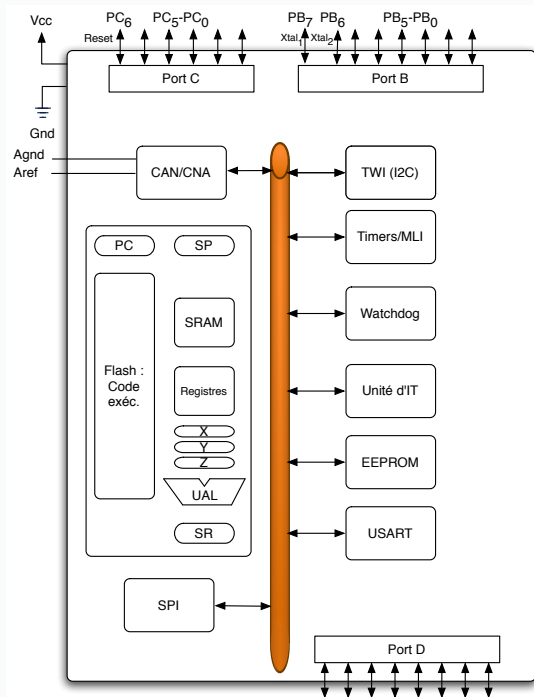
Fonctions

Fonction de mise en oeuvre d'une interruption

4/122 de bits

Mise en oeuvre d'une interruption

# Synoptique



Présentation générale

Architecture

Info Indus 8 bits

Delfieu

## Présentation générale

Introduction

**Architecture**

Mémoire

Registres Systèmes

## Les interruptions externes

Les **PORTS**

## Le traitement des valeurs Analogiques

Le Comparateur Analogique

Convertisseur Analogique-Numérique

: **ADC**

Les Registres de l'ADC

## Les timers

Le **timer 0**

Le pré-diviseur de  $T_0$

et  $T_1$

Le **timer 1**

Les ITs du **timer 1**

Les modes de la **MLI**

## Programmation

Langage C

Fonctions

Fonction de **mode** de bits

5/122

Mise en oeuvre d'une interruption

# Caractéristiques générales

- Architecture RISC : 1 instruction par cycle
- Quartz carte TP à 12MHz, 1 instruction prends 83 *nanosec*
- Registres : 8 et 16 bits;
- Registre d'état : **SREG**
- Pile par adresse croissante : **SP**.

Présentation générale

Architecture

Info Indus 8 bits

Delfieu

## Présentation générale

Introduction

**Architecture**

Mémoire

Registres Systèmes

## Les interruptions externes

Les **PORTS**

## Le traitement des valeurs Analogiques

Le Comparateur Analogique

Convertisseur Analogique-Numérique

: **ADC**

Les Registres de l'ADC

## Les timers

Le **timer 0**

Le pré-diviseur de  $T_0$

et  $T_1$

Le **timer 1**

Les ITs du **timer 1**

Les modes de la **MLI**

## Programmation

Langage C

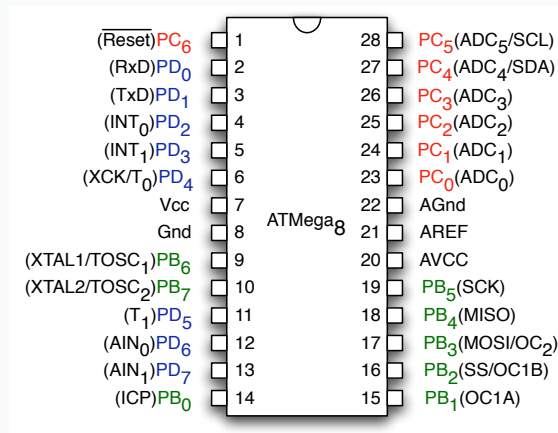
Fonctions

Fonction de **mode** de bits

6/122

Mise en oeuvre d'une interruption

# DIP 28 broches



Présentation générale

Architecture

Info Indus 8 bits

Delfieu

Présentation générale

Introduction

Architecture

Mémoire

Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique

Convertisseur Analogique-Numérique

: ADC

Les Registres de l'ADC

Les timers

Le timer 0

Le pré-diviseur de T<sub>0</sub>

et T<sub>1</sub>

Le timer 1

Les ITs du timer 1

Les modes de la MLI

Programmation

Langage C

Fonctions

Fonction de

minim

122 de bits

Mise en oeuvre d'une

interruption

# Les Ports d'Entrées-sorties

Regroupements de 8 pattes désignés par une lettre.

- 3 ports **PORTB**, **PORTC** et **PORTD** :
  - \* Entrée : lecture signal [0, 5v]
  - \* Sortie : 20 ma par patte.
  - \* 200 ma pour un port.
- Fonctions secondaires :
  - \* **PORTB** : Quartz externe, timer 2, interface série,...
  - \* **PORTC** : 6 Entrées analogiques, reset.
  - \* **PORTD** : RS232, interruptions, comparateur.

Présentation générale

Architecture

Info Indus 8 bits

Delfieu

Présentation générale

Introduction

Architecture

Mémoire

Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique

Convertisseur Analogique-Numérique

: ADC

Les Registres de l'ADC

Les timers

Le timer 0

Le pré-diviseur de T<sub>0</sub>

et T<sub>1</sub>

Le timer 1

Les ITs du timer 1

Les modes de la MLI

Programmation

Langage C

Fonctions

Fonction de

minim

122 de bits

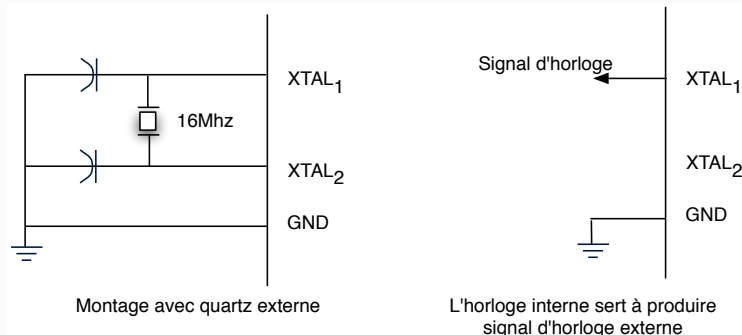
Mise en oeuvre d'une

interruption

# Horloges

On peut utiliser les résonateurs suivant :

- Résonateur externe type céramique ou crystal ou *RC*
- Oscillateur calibré interne de type *RC*
- Horloge externe



Présentation générale

Architecture

Info Indus 8 bits

Delfieu

Présentation générale

Introduction

Architecture

Mémoire

Registres Systèmes

Les interruptions externes

Les *PORTS*

Le traitement des valeurs Analogiques

Le Comparateur Analogique

Convertisseur Analogique-Numérique

: **ADC**

Les Registres de l'ADC

Les **timers**

Le **timer 0**

Le pré-diviseur de  $T_0$

et  $T_1$

Le **timer 1**

Les ITs du **timer 1**

Les modes de la **MLI**

Programmation

Langage C

Fonctions

Fonction de mise en oeuvre d'une interruption

10 / 122

# Les mémoires

Il y a trois sortes de **mémoires** :

- **FLASH** (8 k): stocke le programme (10.000 cycles)
- **SRAM** (mémoire donnée) (1024 octets):
  - *Accumulateurs* ;
  - Registre d'état : *SREG*
  - Pile : *SP*
  - Etat du micro-contrôleur : *MCUCR, MCUSR*
  - Contrôle de l'horloge : *OSCCAL*
- **EEProm** (512 o): on y place des données stratégiques (100 000 cycles)

Présentation générale

Mémoire

Info Indus 8 bits

Delfieu

Présentation générale

Introduction

Architecture

**Mémoire**

Registres Systèmes

Les interruptions externes

Les *PORTS*

Le traitement des valeurs Analogiques

Le Comparateur Analogique

Convertisseur Analogique-Numérique

: **ADC**

Les Registres de l'ADC

Les **timers**

Le **timer 0**

Le pré-diviseur de  $T_0$

et  $T_1$

Le **timer 1**

Les ITs du **timer 1**

Les modes de la **MLI**

Programmation

Langage C

Fonctions

Fonction de mise en oeuvre d'une interruption

10 / 122

# Registres Systèmes

- Registre d'état *SREG*
- Pile : *SP*
- Etat du micro-contrôleur : *MCUCR, MCUSR*
- controle de l'horloge : *OSCCAL*

Présentation générale

Registres Systèmes

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire

**Registres Systèmes**

Les interruptions externes

Les *PORTS*

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer** 0  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer** 1  
Les ITs du **timer** 1  
Les modes de la **MLI**

Programmation

Langage C  
Fonctions

Fonction de **11** / **122** de bits  
11 / 122  
Mise en oeuvre d'une interruption

# Les bits systèmes les plus importants

*SREG* : *I T H S V N Z C*

- *C*: Carry
- *Z* : Zero
- *N*: Negative
- *V* : oVerflow
- *S*: Sign =  $V \oplus N$
- *H* : Half carry
- *T* : sTorage : bit tampon pour manipuler un bit
- *I* : autorisation générale des Interruptions

Présentation générale

Registres Systèmes

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire

**Registres Systèmes**

Les interruptions externes

Les *PORTS*

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer** 0  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer** 1  
Les ITs du **timer** 1  
Les modes de la **MLI**

Programmation

Langage C  
Fonctions

Fonction de **12** / **122** de bits  
12 / 122  
Mise en oeuvre d'une interruption

# Modes de sommeil

*MCUCR* : *SE SM<sub>2</sub> SM<sub>1</sub> SM<sub>0</sub> ISC<sub>11</sub> ISC<sub>10</sub> ISC<sub>01</sub> ISC<sub>00</sub>*

Les bits de configuration des modes de sommeil *SM<sub>j</sub>*:

SM2	SM1	SM0	Mode de Sommeil
0	0	0	Mode attente
0	0	1	Mode réduction de bruit pour le CAN (ADC)
0	1	0	Mode Power down
0	1	1	Mode Power save
1	0	0	Réservé
1	0	1	Réservé
1	1	0	Non utilisé
1	1	1	En pause

Présentation générale

Registres Systèmes

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire

Registres Systèmes

Les interruptions externes

Les *PORTS*

Le traitement des valeurs Analogiques

Le Comparateur Analogique

Convertisseur Analogique-Numérique : **ADC**

Les Registres de l'ADC

Les **timers**

Le **timer** 0

Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer** 1

Les ITs du **timer** 1

Les modes de la **MLI**

Programmation

Langage C

Fonctions

Fonction de **13** / **122** de bits

Mise en oeuvre d'une interruption

# Interruption Externe

## Interruption Externe

Une interruption externe correspond à la prise en compte d'un évènement impromptu ou peu fréquent, dont l'occurrence est aléatoire.

Ces évènements peuvent être par exemple :

- Un arrêt d'urgence provoqué par l'appui d'un bouton ad hoc.
- Capteur de choc mécanique : Un évènement lié à l'environnement mais non attendu au moment ou il se produit
- Dépassement de seuil sur un capteur de température,...

Les interruptions externes

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire

Registres Systèmes

Les interruptions externes

Les *PORTS*

Le traitement des valeurs Analogiques

Le Comparateur Analogique

Convertisseur Analogique-Numérique : **ADC**

Les Registres de l'ADC

Les **timers**

Le **timer** 0

Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer** 1

Les ITs du **timer** 1

Les modes de la **MLI**

Programmation

Langage C

Fonctions

Fonction de **14** / **122** de bits

Mise en oeuvre d'une interruption



## interruption externe et sous-programme

Associer un événement à un sous-programme : Instruction **ISR**

```
ISR(INT0_vect){
    PORTD ^= 0x01;
}
int main(){
    DDRD=0xFF;
    GICR |= 1<<INT0;
    sei();
    while(1){}
```

**ISR** Associe l'évènement *INT0\_vect* au code entre {...}

Si l'évènement *INT0* se produit quel que soit le comportement du programme principal *main* le travail est interrompu pour exécuter le code associé. A la fin de l'interruption on revient là où l'on a été interrompu.

Les interruptions externes

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les *PORTS*

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer** 0  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer** 1  
Les ITs du **timer** 1  
Les modes de la **MLI**

Programmation

Langage C  
Fonctions

Fonction de mise en oeuvre d'une interruption

## Mise en oeuvre des interruptions externes

**GICR** :  $INT_0$   $INT_1$  - - - - **IVSEL** **IVCE**

- $INT_1$  : Autorise l'interruption externe sur  $PD_3$
- $INT_0$  : Autorise l'interruption externe sur  $PD_2$

**MCUCR** :  $ISC_{11}$ ,  $ISC_{10}$   $ISC_{01}$ ,  $ISC_{00}$

00 : un niveau bas génère une interruption

01 : front montant ou descendant

10 : front descendant

11 : front montant

Les interruptions externes

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les *PORTS*

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer** 0  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer** 1  
Les ITs du **timer** 1  
Les modes de la **MLI**

Programmation

Langage C  
Fonctions

Fonction de mise en oeuvre d'une interruption

# LES PORTS

- Les *PORTS* sont programmables en entrée ou en sortie.
- Les *PORTS* ont la capacité d'être à la fois source ou drain de courant.
- Entrée : Les *pattes* peuvent être tirés à la masse (resp VCC) si les résistances de *pull-down* (resp. *pull-up*) sont activées.
- Sortie : On peut sortir 20 *ma* par *patte*.
- Les *pattes* sont de type 3-états. Elles sont en haute impédance lorsque un RESET survient.

Les interruptions externes

Les *PORTS*

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les *PORTS*

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les *timers*

Le *timer* 0  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le *timer* 1  
Les ITs du *timer* 1  
Les modes de la **MLI**

Programmation

Langage C  
Fonctions

Fonction de **17 / 122** de bits  
17 / 122  
Mise en oeuvre d'une interruption

# Programmer le sens des *PORTS*

*DDRB*, *DDRC*, *DDRD* : Programmer le sens des lignes :

- 1 : sortie,
- 0 : entrée.

`DDRB = 0b1111 0011;`

`DDRC = 0xFE;`

`DDRD = 255;`

Les interruptions externes

Les *PORTS*

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les *PORTS*

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les *timers*

Le *timer* 0  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le *timer* 1  
Les ITs du *timer* 1  
Les modes de la **MLI**

Programmation

Langage C  
Fonctions

Fonction de **18 / 122** de bits  
18 / 122  
Mise en oeuvre d'une interruption

# Lecture de *PORTS*

*PINB*, *PINC*, *PIND* : Permettent de lire la valeur instantanée des 8 *pattes* des *PORTS*

```
int B = PIND;  
int C = PINC;
```

Présentation générale

- Introduction
- Architecture
- Mémoire
- Registres Systèmes

Les interruptions externes

Les *PORTS*

Le traitement des valeurs Analogiques

- Le Comparateur Analogique
- Convertisseur Analogique-Numérique : **ADC**
- Les Registres de l'ADC

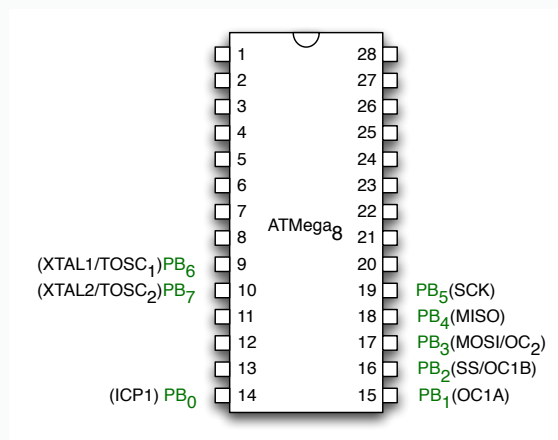
Les **timers**

- Le **timer** 0
- Le pré-diviseur de  $T_0$  et  $T_1$
- Le **timer** 1
- Les ITs du **timer** 1
- Les modes de la **MLI**

Programmation

- Langage C
- Fonctions
- Fonction de mise en oeuvre d'une interruption

# *PORTB*



Présentation générale

- Introduction
- Architecture
- Mémoire
- Registres Systèmes

Les interruptions externes

Les *PORTS*

Le traitement des valeurs Analogiques

- Le Comparateur Analogique
- Convertisseur Analogique-Numérique : **ADC**
- Les Registres de l'ADC

Les **timers**

- Le **timer** 0
- Le pré-diviseur de  $T_0$  et  $T_1$
- Le **timer** 1
- Les ITs du **timer** 1
- Les modes de la **MLI**

Programmation

- Langage C
- Fonctions
- Fonction de mise en oeuvre d'une interruption



# PORTC

- *DDRC*, *PORTC*, *PINC*
- *PC<sub>0</sub>*, *PC<sub>5</sub>* : Entrées analogiques

Les interruptions externes

Les *PORTS*

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les *PORTS*

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

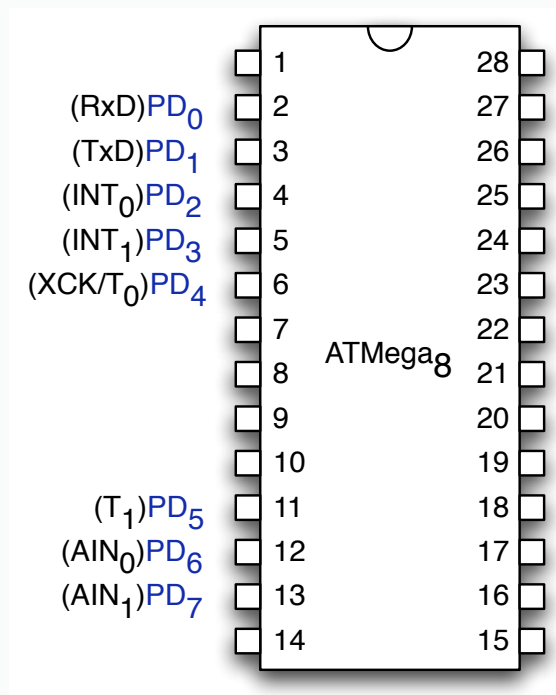
Le **timer** 0  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer** 1  
Les ITs du **timer** 1  
Les modes de la **MLI**

Programmation

Langage C  
Fonctions  
Fonction de mise en oeuvre d'une interruption

23 / 122 de bits

# PORTD



Les interruptions externes

Les *PORTS*

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les *PORTS*

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer** 0  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer** 1  
Les ITs du **timer** 1  
Les modes de la **MLI**

Programmation

Langage C  
Fonctions  
Fonction de mise en oeuvre d'une interruption

24 / 122 de bits

# PORTD

- *DDRD, PORTD, PIND*
- *PD<sub>2</sub>, PD<sub>3</sub>* : interruptions externes
- *PD<sub>0</sub>, PD<sub>1</sub>* : Protocole RS232

Les interruptions externes

Les PORTS

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : ADC  
Les Registres de l'ADC

Les timers

Le timer 0  
Le pré-diviseur de T<sub>0</sub> et T<sub>1</sub>  
Le timer 1  
Les ITs du timer 1  
Les modes de la MLI

Programmation

Langage C  
Fonctions  
Fonction de mise en oeuvre de bits  
Mise en oeuvre d'une interruption

25 / 122

# Exemple d'utilisation d'un port

```
int main(void) {  
  
    DDRD = 0xFF; // port D en sortie  
    PORTD= 0x7A; //Allume leds 2,4,5,6,7  
  
    while(1){ //boucle infinie  
        PORTD^=0x7A; // ^= : ou-exlcusif fait  
        _delay_ms(20); // clignoter les leds  
    }  
}
```

Les interruptions externes

Les PORTS

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : ADC  
Les Registres de l'ADC

Les timers

Le timer 0  
Le pré-diviseur de T<sub>0</sub> et T<sub>1</sub>  
Le timer 1  
Les ITs du timer 1  
Les modes de la MLI

Programmation

Langage C  
Fonctions  
Fonction de mise en oeuvre de bits  
Mise en oeuvre d'une interruption

26 / 122

# Comparateur Analogique (1/7)

- Comparaison analogiques de  $PD_6$  et  $PD_7$ .
- $AIN_0$  ( $PD_6$ ) : broche positive ;
- $AIN_1$  ( $PD_7$ ) : broche négative ;
- Lorsque  $AIN_0 > AIN_1$  alors  $ADC_0$  est mis à un
- *interruption* de comparaison :  $ACIE$  du registre  $ACSR$

Le traitement des valeurs Analogiques

Le Comparateur Analogique

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

**Le Comparateur Analogique**

Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

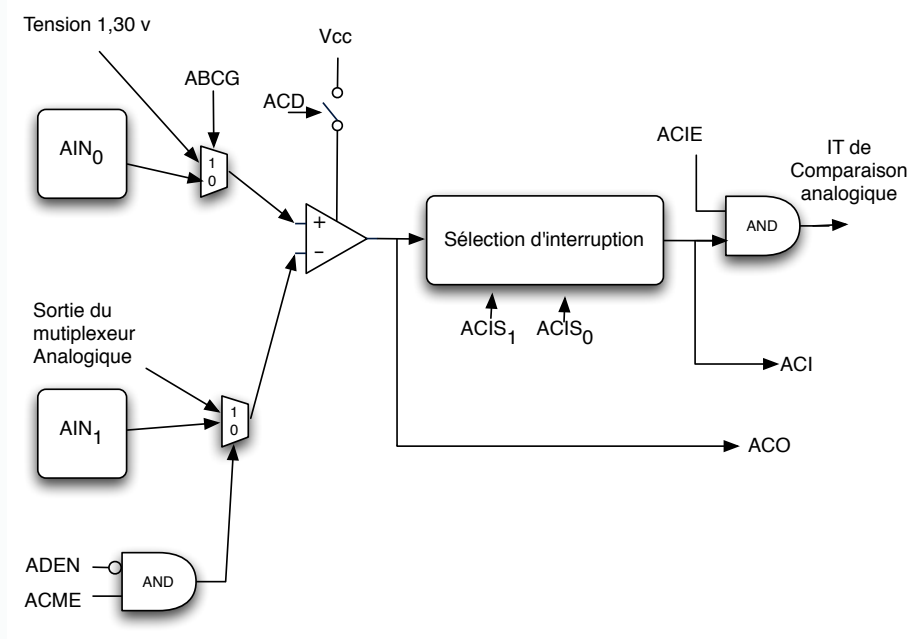
Le **timer 0**  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer 1**  
Les ITs du **timer 1**  
Les modes de la **MLI**

Programmation

Langage C  
Fonctions

Fonction de **multiplicateur** de bits  
27 / 122  
Mise en oeuvre d'une interruption

# Le Comparateur Analogique Numérique (2/7)



Le traitement des valeurs Analogiques

Le Comparateur Analogique

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

**Le Comparateur Analogique**

Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer 0**  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer 1**  
Les ITs du **timer 1**  
Les modes de la **MLI**

Programmation

Langage C  
Fonctions

Fonction de **multiplicateur** de bits  
28 / 122  
Mise en oeuvre d'une interruption

## Le Comparateur Analogique (3/7)

- $ADC_0, ADC_1, \dots, ADC_5$  peuvent remplacer l'entrée négative  $AIN_1$  à l'aide du registre  $ADMUX$
- *interruption* déclenchable sur le front montant ou le front descendant de  $ACO$ .

Le traitement des valeurs Analogiques

Le Comparateur Analogique

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les *PORTS*

Le traitement des valeurs Analogiques

**Le Comparateur Analogique**  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les *timers*

Le *timer* 0  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le *timer* 1  
Les ITs du *timer* 1  
Les modes de la *MLI*

Programmation

Langage C  
Fonctions  
Fonction de  $20/122$  de bits  
29/122  
Mise en oeuvre d'une interruption

## Comparateur Analogique : $ADMUX$ (4/7)

$REFS_1$   $REFS_0$   $ADLAR$   $MUX_4$   $MUX_3$   $MUX_2$   $MUX_1$   $MUX_0$  ( $ADMUX$ )

Permet le choix d'une entrée analogique à la place de  $AIN_1$  :

$ACME(SFIOR)$	$ADEN(ADCSRA)$	$MUX_{2:0}$	$AIN_1$
0	x	xxx	0
1	1	xxx	0
1	0	000	$ADC_0$
1	0	001	$ADC_1$
1	0	010	$ADC_2$
1	0	011	$ADC_3$
1	0	100	$ADC_4$
1	0	101	$ADC_5$
1	0	110	$ADC_6$

Le traitement des valeurs Analogiques

Le Comparateur Analogique

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les *PORTS*

Le traitement des valeurs Analogiques

**Le Comparateur Analogique**  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les *timers*

Le *timer* 0  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le *timer* 1  
Les ITs du *timer* 1  
Les modes de la *MLI*

Programmation

Langage C  
Fonctions  
Fonction de  $30/122$  de bits  
30/122  
Mise en oeuvre d'une interruption



## Comparteur Analogique : ACS (5/7)

ACS : Analog Comparator Control and status

ACD ACO ACI ACIE ACIC ACIS<sub>1</sub> ACS<sub>0</sub> (ACS)

- **ACD** Analog Comparator Disable :  
Contrôle du comparateur analogique, 0 : ON, 1 : OFF !
- **ACO** Analog Comparator Output :  
si  $A_{IN0} > A_{IN1}$  alors  $ACO = 1$ .
- **ACI** Analog Comparator Interrupt Flag :  
Drapeau d'*interruption* (cf  $ACIS_1$ ,  $ACS_0$ ). Ce bit est  
raz automatiquement après le traitement de  
l'interruption.

Le traitement des valeurs Analogiques

Le Comparateur Analogique

Info Indus 8 bits

Delfieu

Présentation  
générale

Introduction

Architecture

Mémoire

Registres Systèmes

Les interruptions  
externes

Les PORTS

Le traitement  
des valeurs  
Analogiques

Le Comparateur  
Analogique

Convertisseur  
Analogique-Numérique  
: ADC

Les Registres de l'ADC

Les timers

Le timer 0

Le pré-diviseur de  $T_0$   
et  $T_1$

Le timer 1

Les ITs du timer 1

Les modes de la MLI

Programmation

Langage C

Fonctions

Fonction de  
mise en oeuvre d'une  
interruption

31 / 122 de bits

31 / 122 de bits

31 / 122 de bits

## Comparteur Analogique : ACSR ( 6/7)

- **ACIE** Analog Comparator Interrupt enable : Bit de  
validation de l'interruption.
- **ACIC** Analog Comparator Input Capture Enable :  
Autorise la datation de comparaison par le **Timer1**.
- **ACIS<sub>1</sub>&ACS<sub>0</sub>** gère la sortie **AC<sub>0</sub>** :

Activation de $AC_0$	$ACIS_1$	$ACS_0$
$1 \rightarrow 0$ ou $0 \rightarrow 1$	0	0
non utilisé	0	1
<i>front montant</i>	0	0
<i>front descendant</i>	1	1

Le traitement des valeurs Analogiques

Le Comparateur Analogique

Info Indus 8 bits

Delfieu

Présentation  
générale

Introduction

Architecture

Mémoire

Registres Systèmes

Les interruptions  
externes

Les PORTS

Le traitement  
des valeurs  
Analogiques

Le Comparateur  
Analogique

Convertisseur  
Analogique-Numérique  
: ADC

Les Registres de l'ADC

Les timers

Le timer 0

Le pré-diviseur de  $T_0$   
et  $T_1$

Le timer 1

Les ITs du timer 1

Les modes de la MLI

Programmation

Langage C

Fonctions

Fonction de  
mise en oeuvre d'une  
interruption

32 / 122 de bits

32 / 122 de bits

32 / 122 de bits

# ACME (7/7)

**SFIOR** : Special Function IO Registers

**ADTS2 ADTS1 ADTS0 ACME PUD PSR2 PSR10 (SFIOR)**

**ACME** Analog Comparator Multiplexer Enable

- Quand **ACME**=1 et **ADEN**=0 alors le multiplexeur de l'**ADC** choisit **AIN<sub>1</sub>** comme entrée négative du comparateur analogique.
- Quand **ACME**=0 alors **AIN<sub>0</sub>** est appliqué à l'entrée négative du comparateur analogique.

Présentation générale

- Introduction
- Architecture
- Mémoire
- Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

**Le Comparateur Analogique**  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

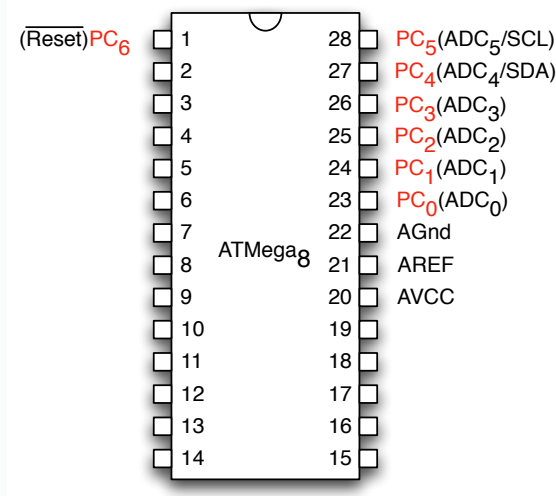
Le **timer 0**  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer 1**  
Les ITs du **timer 1**  
Les modes de la **MLI**

Programmation

Langage C  
Fonctions

Fonction de **33/122** de bits  
33/122  
Mise en oeuvre d'une interruption

# Convertisseur Analogique-Numérique : ADC(1/19)



Présentation générale

- Introduction
- Architecture
- Mémoire
- Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

**Le Comparateur Analogique**  
**Convertisseur Analogique-Numérique : ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer 0**  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer 1**  
Les ITs du **timer 1**  
Les modes de la **MLI**

Programmation

Langage C  
Fonctions

Fonction de **34/122** de bits  
34/122  
Mise en oeuvre d'une interruption

## ADC(2/19)

- L'ADC convertit une entrée analogique sur 10 bits par approximations successives.
- Erreur correspond à 1 "Least Significant Bit" (LSB)
- Temps de conversion (10b) : 13 cycles d'horloge ( $\approx 1 \mu s$ )
- 6 entrées différentielles dont 2 avec gain (10x)
- L'ADC a une alimentation découplée du micro :  $AV_{CC}$  tq  $AV_{CC} \in [V_{CC} - 0,3, V_{CC} + 0,3]$
- Tension de référence externe ou interne :  $AREF$  ou  $AV_{CC}$  ou  $= 2,56 \text{ v}$ .

## ADC (3/19)

- *interruption* de fin de conversion
- Limitation du bruit en mode de sommeil est possible.
- conversion numérique

$$\text{Res} = (\text{P. Ent.}(V_{in}/V_{REF}) * 1024)$$

$$V_{REF} = AREF \text{ ou } AV_{CC} \text{ ou } 2,56 \text{ v.}$$

### Exemple

$$ADC_0 = 2.5\text{v}, AREF = 3.3\text{v}$$

$$ADCH \text{ ADCL} = PE((2.5/3.3) \times 1024) = 774$$

- Le temps de conversion est égale à 13 fois l'horloge système multiplié par le facteur de pré-division.

## ADC (4/19)

### Remarque

Pour réduire au maximum les erreurs de conversions due à la logique interne du contrôleur :

- Mettre en sommeil l'unité centrale avant le lancement d'une conversion
- Découpler soigneusement l'alimentation  $AV_{CC}$  avec des condensateurs
- Règles de routage : Connexions courtes, plan de masse, . . .
- Filtre électronique ou numérique des résultats (amortissement, moyenne, ...).

## ADC (5/19)

- La valeur minimale de conversion est  $GND$
- La valeur maximale est :
  - Soit la tension sur la broche  $AREF$ ;
  - Soit la tension sur la broche  $AV_{CC}$ ;
  - Soit une tension interne de 2,56V (cf bits  $REFS_n$ ).

## ADC (6/7)

- Chacune des 6 broches d'entrée de l'ADC ainsi que **GND** et la référence de tension, peuvent être choisies comme des entrées simples de l'ADC.
- L'ADC contient un mécanisme d'échantillonneur/bloqueur qui assure que l'entrée est tenue constante pendant 1,5 cycle d'horloge.

## ADC (6/19)

- L'ADC est actif avec **ADEN** à 1 dans **ADCSRA**.
- L'ADC est déclenché par **ADSC** à 1 dans **ADCSRA**.
- L'ADC produit un résultat sur 10 bits présenté dans les registres **ADCH** et **ADCL**.
- Par défaut, le résultat est présenté ajusté à droite. On lit **ADCL** puis **ADCH**.
- Ajusté à gauche en mettant le **ADLAR** de **ADMUX**:
  - Dans ce cas, le résultat est sur 8 bits
  - La lecture de **ADCH** est suffisante.

## ADC (7/19)

### conversion 10 bits

- **ADCL** doit être lu en premier puis **ADCH**.
- Une fois que **ADCL** est lu, l'accès aux registres est bloqué afin d'empêcher une nouvelle conversion.
- Quand **ADCH** est lu, l'**ADC** est à nouveau être opérationnel.

#### Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

#### Les interruptions externes

Les **PORTS**

#### Le traitement des valeurs Analogiques

Le Comparateur Analogique

#### Convertisseur Analogique-Numérique : **ADC**

Les Registres de l'ADC

#### Les **timers**

Le **timer** 0

Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer** 1

Les ITs du **timer** 1

Les modes de la **MLI**

#### Programmation

Langage C

Fonctions

Fonction de mise en oeuvre d'une interruption

41 / 122 de bits

## Lancement d'une conversion (7/19)

- Une conversion simple est lancée par **ADSC** (**ADSCRA**)
- Une conversion simple prends 13 cycles d'horloge.
- **ADSC** reste à un tant que la conversion se réalise et sera remis à 0 par le matériel quand la conversion est achevée.
- Si on change le canal tandis qu'une conversion est en cours, l'**ADC** finira la conversion actuelle avant l'exécution du changement de canal.

#### Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

#### Les interruptions externes

Les **PORTS**

#### Le traitement des valeurs Analogiques

Le Comparateur Analogique

#### Convertisseur Analogique-Numérique : **ADC**

Les Registres de l'ADC

#### Les **timers**

Le **timer** 0

Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer** 1

Les ITs du **timer** 1

Les modes de la **MLI**

#### Programmation

Langage C

Fonctions

Fonction de mise en oeuvre d'une interruption

42 / 122 de bits

## Lancement d'une conversion (8/19)

Free Running mode : Mode de fonctionnement libre

- Dans ce mode, l'**ADC** échantillonne en permanence et met à jour les registres de données (**ADCH** et **ADCL**)
- Ce mode est positionné par la valeur 1 dans **ADFR** **ADCSRA**.
- La première conversion doit être lancée par **ADSC**
- Dans ce mode, la première conversion dure 25 cycles puis les suivantes 15, il n'est pas nécessaire de re-déclencher **ADSC** à chaque conversion.

Le traitement des valeurs Analogiques

Convertisseur Analogique-Numérique : **ADC**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique

**Convertisseur Analogique-Numérique : ADC**

Les Registres de l'ADC

Les **timers**

Le **timer** 0

Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer** 1

Les ITs du **timer** 1

Les modes de la **MLI**

Programmation

Langage C

Fonctions

Fonction de

43 / 122 de bits

Mise en oeuvre d'une interruption

## Le pré-diviseur (9/19)

- L'**ADC** contient un pré-diviseur qui produit une fréquence à partir de celle du **CPU**
- Prédiviseur : **ADPS** dans **ADCSRA**
- Prédiviseur activé dès que **ADEN** à 1

Le traitement des valeurs Analogiques

Convertisseur Analogique-Numérique : **ADC**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique

**Convertisseur Analogique-Numérique : ADC**

Les Registres de l'ADC

Les **timers**

Le **timer** 0

Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer** 1

Les ITs du **timer** 1

Les modes de la **MLI**

Programmation

Langage C

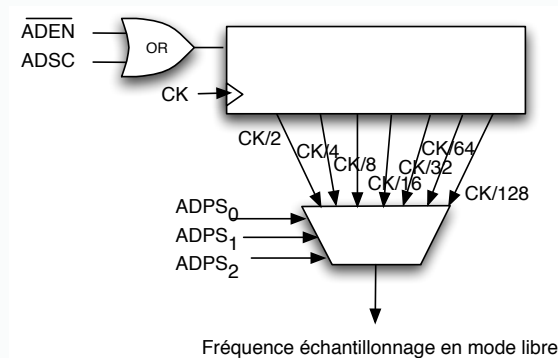
Fonctions

Fonction de

44 / 122 de bits

Mise en oeuvre d'une interruption

## Le pré-diviseur (10/19)



Le traitement des valeurs Analogiques

Convertisseur Analogique-Numérique : **ADC**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique

Convertisseur Analogique-Numérique : **ADC**

Les Registres de l'ADC

Les **timers**

Le **timer** 0

Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer** 1

Les ITs du **timer** 1

Les modes de la **MLI**

Programmation

Langage C

Fonctions

Fonction de mise en œuvre d'une interruption

45 / 122 de bits

## L'ADC(11/19)

- 1 **ADC** : **ADEN** ← 1 du **ADCSRA**
- 2 conversion simple : **ADSC** ← 1 du **ADCSRA**
- 3 Fin conversion : Résultat écrit dans **ADCH**, **ADCL** et **ADIF** ← 1. **ADSC** ← 0
- 4 On peut relancer une conversion par **ADSC** (étape 2)

Le traitement des valeurs Analogiques

Convertisseur Analogique-Numérique : **ADC**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique

Convertisseur Analogique-Numérique : **ADC**

Les Registres de l'ADC

Les **timers**

Le **timer** 0

Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer** 1

Les ITs du **timer** 1

Les modes de la **MLI**

Programmation

Langage C

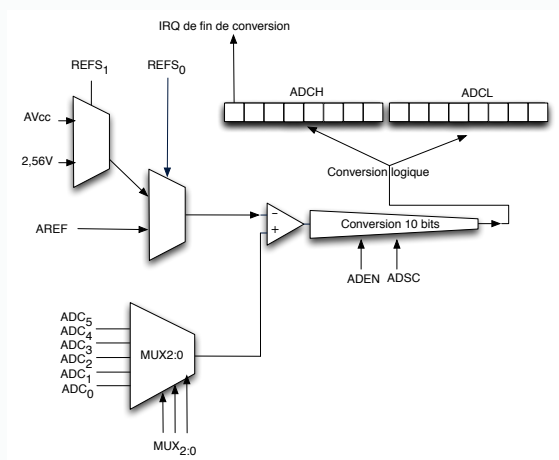
Fonctions

Fonction de mise en œuvre d'une interruption

46 / 122 de bits



# Le Convertisseur Analogique Numérique (12/19)



Le traitement des valeurs Analogiques

Convertisseur Analogique-Numérique : **ADC**

Info Indus 8 bits

Delfieu

Présentation générale

- Introduction
- Architecture
- Mémoire
- Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique

**Convertisseur Analogique-Numérique : ADC**

Les Registres de l'ADC

Les **timers**

Le **timer** 0

Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer** 1

Les ITs du **timer** 1

Les modes de la **MLI**

Programmation

Langage C

Fonctions

Fonction de **multiplication** de bits

Mise en oeuvre d'une interruption

47 / 122

# Convertisseur Analogique Numérique (13/19) : *ADMUX*

$REFS_1$   $REFS_0$   $ADLAR$   $MUX_4$   $MUX_3$   $MUX_2$   $MUX_1$   $MUX_0$  (*ADMUX*)

$REFS_1$	$REFS_0$	Tension de Référence
0	0	<i>AREF</i>
0	1	$AV_{cc}$ avec capacité externe sur <i>AREF</i>
1	0	non utilisé
1	1	2,56 v

Le traitement des valeurs Analogiques

Les Registres de l'ADC

Info Indus 8 bits

Delfieu

Présentation générale

- Introduction
- Architecture
- Mémoire
- Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique

**Convertisseur Analogique-Numérique : ADC**

Les Registres de l'ADC

Les **timers**

Le **timer** 0

Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer** 1

Les ITs du **timer** 1

Les modes de la **MLI**

Programmation

Langage C

Fonctions

Fonction de **multiplication** de bits

Mise en oeuvre d'une interruption

48 / 122

## Convertisseur Analogique Numérique (14/19) : *ADMUX*

- *ADLAR* : Analog Left Adjust Register : Ajustement à gauche à du résultat dans le registre *ADCL* et *ADCH*.
- *MUX<sub>2,1,0</sub>* : Choix du canal de l'*ADC*
- *ADSC* : (AD Start conversion) Lancement de la conversion de la voie sélectionnée.

Le traitement des valeurs Analogiques

Les Registres de l'*ADC*

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les *PORTS*

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : *ADC*

Les Registres de l'*ADC*

Les *timers*

Le *timer* 0  
Le pré-diviseur de  $T_0$  et  $T_1$

Le *timer* 1

Les ITs du *timer* 1

Les modes de la *MLI*

Programmation

Langage C  
Fonctions

Fonction de mise en oeuvre d'une interruption

49 / 122 de bits

Mise en oeuvre d'une interruption

## Convertisseur Analogique Numérique (15/19) : *ADCSRA*

*ADEN ADSC ADATE ADIF ADIE ADPS<sub>2</sub> ADPS<sub>1</sub> ADPS<sub>0</sub>* (*ADCSRA*)

- *ADFR* : (AD Free running Mode) La mise à 1 de ce bit permet de mettre en le convertisseur en mode free running.
- *ADIF* (AD Interrupt Flag) Passe à 1 une fois la conversion terminée et déclenche l'interruption si *ADIE* =1. Ce bit repasse automatiquement à 0 lors du traitement de la routine d'interruption.

Le traitement des valeurs Analogiques

Les Registres de l'*ADC*

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les *PORTS*

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : *ADC*

Les Registres de l'*ADC*

Les *timers*

Le *timer* 0  
Le pré-diviseur de  $T_0$  et  $T_1$

Le *timer* 1

Les ITs du *timer* 1

Les modes de la *MLI*

Programmation

Langage C  
Fonctions

Fonction de mise en oeuvre d'une interruption

50 / 122 de bits

Mise en oeuvre d'une interruption

# Le Convertisseur Analogique Numérique (17/19) : ADSCRA

- **ADIE** Analog Digital Interrupt Enable : 1 valide de l'interruption de l'**ADC**
- **ADSP<sub>2</sub>, ADSP<sub>1</sub>, ADSP<sub>0</sub>** : Prédiviseur de l'horloge interne : 1,2,4,8,16,32,64,128.

Le traitement des valeurs Analogiques

Les Registres de l'ADC

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**

Les Registres de l'ADC

Les **timers**

Le **timer** 0  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer** 1  
Les ITs du **timer** 1  
Les modes de la **MLI**

Programmation

Langage C  
Fonctions  
Fonction de **51/122** de bits  
Mise en oeuvre d'une interruption

# Le Convertisseur Analogique Numérique (18/19) : ADCH – ADCL

- **ADCH ADCL** avec **ADLAR=0**

— — — — — — — — — — **ADC<sub>9</sub> ADC<sub>8</sub>**

**ADC<sub>7</sub> ADC<sub>6</sub> ADC<sub>5</sub> ADC<sub>4</sub> ADC<sub>3</sub> ADC<sub>2</sub> ADC<sub>1</sub> ADC<sub>0</sub>**

- **ADCH ADCL** avec **ADLAR=1**

**ADC<sub>9</sub> ADC<sub>8</sub> ADC<sub>7</sub> ADC<sub>6</sub> ADC<sub>5</sub> ADC<sub>4</sub> ADC<sub>3</sub> ADC<sub>2</sub>**

**ADC<sub>1</sub> ADC<sub>0</sub>** — — — — — — — — — —

Le traitement des valeurs Analogiques

Les Registres de l'ADC

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**

Les Registres de l'ADC

Les **timers**

Le **timer** 0  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer** 1  
Les ITs du **timer** 1  
Les modes de la **MLI**

Programmation

Langage C  
Fonctions  
Fonction de **52/122** de bits  
Mise en oeuvre d'une interruption

## Lecture analogique scrutative sans interruption

```
volatile int lu; // variable globale

void lecture_analogique_scrutative(){
    ADCSRA |= (1<<ADSC);
    while (ADCSRA & (1<<ADSC)) {};
    lu=ADCH; // lecture sur 8 bits
}

int main(void){
    DDRB=0x1F; DDRC=0x00; // C : entree, B :
        Sortie
    ADMUX=(1<<ADLAR); //ajust Gauch, PC0
        entree analog.
    ADCSRA= (1<<ADEN); // Mise On ADC
    do {
        _delay_ms(200);
        lecture_analogique_scrutative();
    } while(1);
    return(0);
}
```

Le traitement des valeurs Analogiques

Les Registres de l'ADC

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les *PORTS*

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**

Les Registres de l'ADC

Les **timers**

Le **timer** 0  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer** 1  
Les ITs du **timer** 1  
Les modes de la **MLI**

Programmation

Langage C  
Fonctions

Fonction de mise en oeuvre d'une interruption  
53 / 122

## Déclenchement échantillonné par le timer 1 de la conversion analogique

```
volatile int F;

ISR(ADC_vect){char L=ADCL;F=ADCH; }

ISR(TIMER1_OVF_vect) {ADCSRA |= (1<<ADSC);
    PORTB ^=1;}

int main(void){
    DDRB=0x01;DDRC=0x00;
    ADMUX=(1<<ADLAR); // adj Gauche, PC0
    ADCSRA= (1<<ADEN)+(1<<ADIE); //can + IT
    TCCR1A=0;TCCR1B = (1 << CS11);
    TIMSK = (1<<TOIE1);
    sei(); // toutes its autorisees
    do {
        PORTB ^=1;_delay_ms(50);
    }while(1);
    return(0);
}
```

Le traitement des valeurs Analogiques

Les Registres de l'ADC

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les *PORTS*

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**

Les Registres de l'ADC

Les **timers**

Le **timer** 0  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer** 1  
Les ITs du **timer** 1  
Les modes de la **MLI**

Programmation

Langage C  
Fonctions

Fonction de mise en oeuvre d'une interruption  
54 / 122

## Free Running

On lance l'**ADC** une fois puis on lit à la volée **ADCH** et **ADCL**.

```
int main(void){
    int H,L;
    DDRB=0x01; DDRC=0x00;
    ADMUX=(1<<ADLAR);
    ADCSRA= (1<<ADEN) + (1<<ADSC)+ (1<<ADFR);
    do {
        PORTB^=1;
        delai(50);
        L=ADCL; H = ADCH;
        Res = (H<<8)+L;
    } while(1);
    return(0);
}
```

Le traitement des valeurs Analogiques

Les Registres de l'ADC

Info Indus 8 bits

Delfieu

Présentation générale

Introduction

Architecture

Mémoire

Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique

Convertisseur Analogique-Numérique : **ADC**

Les Registres de l'ADC

Les **timers**

Le **timer** 0

Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer** 1

Les ITs du **timer** 1

Les modes de la **MLI**

Programmation

Langage C

Fonctions

Fonction de mise en oeuvre d'une interruption

55 / 122

## Mode boggué

Le **timer** relance de l'**ADC** dans l'*interruption* de conversion.

```
volatile int F;

ISR(ADC_vect){F=ADCH; ADCSRA |= (1<<ADSC)}
    // Aie !!!

int main(void){
    DDRB=0x01;DDRC=0x00;
    ADMUX=(1<<ADLAR); //Gauche, PC0
    ADCSRA= (1<<ADEN)+(1<<ADCS)+(1<<ADIE);
        // can + IT
    sei(); // toute les its autorisees
    do {
        PORTB^=1;_delay_ms(50);
    }while(1);
    return(0);
}
```

Le traitement des valeurs Analogiques

Les Registres de l'ADC

Info Indus 8 bits

Delfieu

Présentation générale

Introduction

Architecture

Mémoire

Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique

Convertisseur Analogique-Numérique : **ADC**

Les Registres de l'ADC

Les **timers**

Le **timer** 0

Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer** 1

Les ITs du **timer** 1

Les modes de la **MLI**

Programmation

Langage C

Fonctions

Fonction de mise en oeuvre d'une interruption

56 / 122

# timer

Les fonctions d'un **timer** :

- 1 Les fonctions de temporisations
  - Définir une base de temps ;
  - Définir une forme périodique ;
  - Produire une **MLI**: Pulse Width Modulated (**PWM**)

Commande d'un Moteur à Courant Continu via un Pont H

- 2 Comptage/Capture d'évènements

Les **timers**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction

Architecture

Mémoire

Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique

Convertisseur Analogique-Numérique : **ADC**

Les Registres de l'ADC

Les **timers**

Le **timer** 0

Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer** 1

Les ITs du **timer** 1

Les modes de la **MLI**

Programmation

Langage C

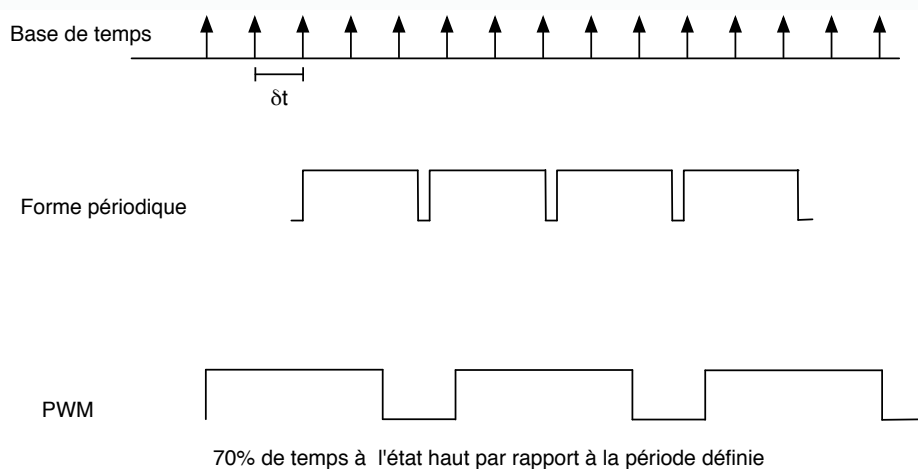
Fonctions

Fonction de **57** / **122** de bits

Mise en oeuvre d'une interruption

# Timer

Les fonctions d'un **timer** : Les fonctions de temporisation



Les **timers**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction

Architecture

Mémoire

Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique

Convertisseur Analogique-Numérique : **ADC**

Les Registres de l'ADC

Les **timers**

Le **timer** 0

Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer** 1

Les ITs du **timer** 1

Les modes de la **MLI**

Programmation

Langage C

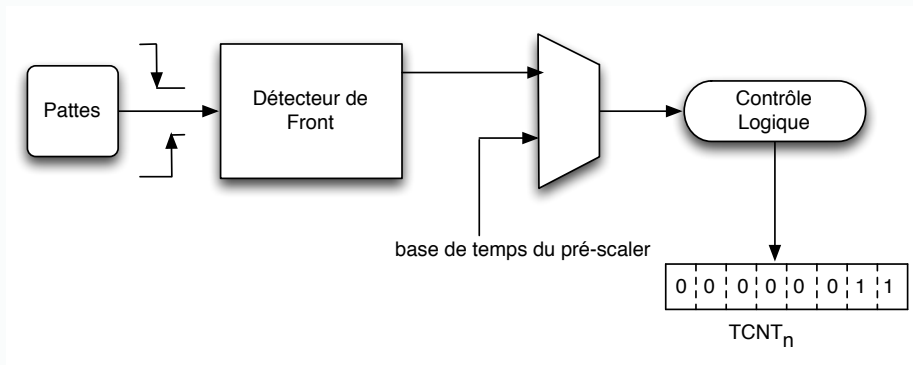
Fonctions

Fonction de **58** / **122** de bits

Mise en oeuvre d'une interruption

## Les timers

Les fonctions d'un **timer** : Le comptage ou la capture d'évènements



Les **timers**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer** 0  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer** 1  
Les ITs du **timer** 1  
Les modes de la **MLI**

Programmation

Langage C  
Fonctions  
Fonction de **59/122** de bits  
Mise en oeuvre d'une interruption

## Les timers

**timer** 0, **timer** 1 et **timer** 2.

- **timer** 0 : 8 bits
- Pré-diviseurs commun aux **timers** 0 et 1
- **timer** 1 : 16 bits, 2 sorties **PWM**
- **timer** 2 : 8 bits, 1 sortie **PWM**

Les **timers**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer** 0  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer** 1  
Les ITs du **timer** 1  
Les modes de la **MLI**

Programmation

Langage C  
Fonctions  
Fonction de **60/122** de bits  
Mise en oeuvre d'une interruption

# Le timer 0

- Simple entrée
- Générateur de fréquence
- Comptage d'événements externes
- Diviseur d'horloge 10 bits

Les timers

Le timer 0

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les timers

Le timer 0

Le pré-diviseur de  $T_0$  et  $T_1$

Le timer 1

Les ITs du timer 1

Les modes de la MLI

Programmation

Langage C

Fonctions

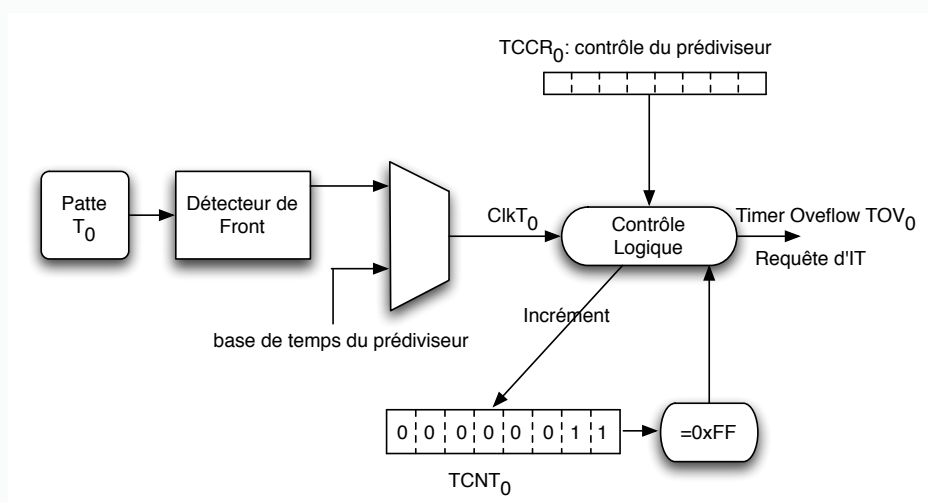
Fonction de

61/122 de bits

Mise en oeuvre d'une interruption

# Le timer 0

Aperçu global du timer 0 :



Les timers

Le timer 0

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les timers

Le timer 0

Le pré-diviseur de  $T_0$  et  $T_1$

Le timer 1

Les ITs du timer 1

Les modes de la MLI

Programmation

Langage C

Fonctions

Fonction de

62/122 de bits

Mise en oeuvre d'une interruption



## Le timer 0

- timer 8 bits.
- Son interruption met à jour le drapeau *TIFR*.
- Le masquage ou l'autorisation de cette *interruption* est réalisé par le bit *TIMSK*.
- Comme le montre le schéma précédent le timer0 peut être excité soit par la patte externe  $T_0$  soit par le pré-diviseur d'horloge. Ce choix est fait par les bits  $CS02 : 0 \in TCCR_0$

Les timers

Le timer 0

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : ADC  
Les Registres de l'ADC

Les timers

Le timer 0

Le pré-diviseur de  $T_0$  et  $T_1$

Le timer 1

Les ITs du timer 1  
Les modes de la MLI

Programmation

Langage C  
Fonctions

Fonction de  $n$  bits

Mise en oeuvre d'une interruption

## Le timer 0

- Le bloc logique ordonne alors l'incrément du registre  $TCNT_0$ . Ce registre s'incrémente de +1.
- Lorsque  $TCNT_0$  atteint la valeur  $0xFF$ , il repart à la valeur 0 et dans le même temps le bit  $TOV_0$  passe à un.
- $TOV_0$  agit comme un 9<sup>ème</sup> bit.
- Si une *interruption* a été mise en place alors ce bit est **raz** lors de l'exécution du sous-programme d'interruption associé.

Les timers

Le timer 0

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : ADC  
Les Registres de l'ADC

Les timers

Le timer 0

Le pré-diviseur de  $T_0$  et  $T_1$

Le timer 1

Les ITs du timer 1  
Les modes de la MLI

Programmation

Langage C  
Fonctions

Fonction de  $n$  bits

Mise en oeuvre d'une interruption

## Les registres associés au timer 0

$TCCR_0$  :

— — — — —  $CS_{02}$   $CS_{01}$   $CS_{00}$  ( $TCCR_0$ )

$CS_{02}$	$CS_{01}$	$CS_{00}$	Description
0	0	0	<i>timer</i> 0 en pause
0	0	1	$clk_{I/O}$ : Horloge Système
0	1	0	$clk_{I/O}/8$
0	1	1	$clk_{I/O}/64$
1	0	0	$clk_{I/O}/256$
1	0	1	$clk_{I/O}/1024$
1	1	0	source externe : front descendant sur la patte $T_0$
1	1	1	source externe : front montant sur la patte $T_0$

REMARQUE : Si un mode source externe (6 ou 7) a été choisi, alors un front sur la patte  $T_0$  agira comme horloge du timer.

Les timers

Le timer 0

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : ADC  
Les Registres de l'ADC

Les timers

Le timer 0  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le timer 1  
Les ITs du timer 1  
Les modes de la MLI

Programmation

Langage C  
Fonctions  
Fonction de mise en oeuvre d'une interruption

65/122 de bits

## Les registres associés au timer 0

- $TCNT_0$  :  
Registre accessible en lecture ou écriture. C'est le registre de comptage courant.
- $TIMSK$  :  
 $OCIE_2$   $TOIE_2$   $TICIE_1$   $OCIE_1A$   $OCIE_1B$   $TOIE_1$  —  $TOIE_0$   
( $TIMSK$ )
  - *interruption* de débordement du timer 0 :  
( $TOIE_0 = 1$  et  $I = 1 \in SREG$ )
  - Sur un débordement de  $TCNT_0$ , on a :  
 $TOV_0 \in TIFR$  qui passe à un, **et** le programme associé à l'IT est alors exécuté.
  - $TOV_0$  est alors automatiquement **raz** dès le début du sous-programme d'*interruption*.

Les timers

Le timer 0

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : ADC  
Les Registres de l'ADC

Les timers

Le timer 0  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le timer 1  
Les ITs du timer 1  
Les modes de la MLI

Programmation

Langage C  
Fonctions  
Fonction de mise en oeuvre d'une interruption

66/122 de bits

# Les pré-diviseurs des timer 0 et 1

- Les timer 0 et 1 partagent le même module de pré-division.
- Le timer 0 :  $CS_{02:0} \in TCCR_0$
- timer 1 :  $CS_{12:0} \in TCCR_1$
- Facteurs de pré-division :  
 $clk_{I/O}/8, clk_{I/O}/64, clk_{I/O}/256, clk_{I/O}/1024.$

Les timers

Le timer 0

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : ADC  
Les Registres de l'ADC

Les timers

Le timer 0

Le pré-diviseur de  $T_0$  et  $T_1$

Le timer 1

Les ITs du timer 1

Les modes de la MLI

Programmation

Langage C

Fonctions

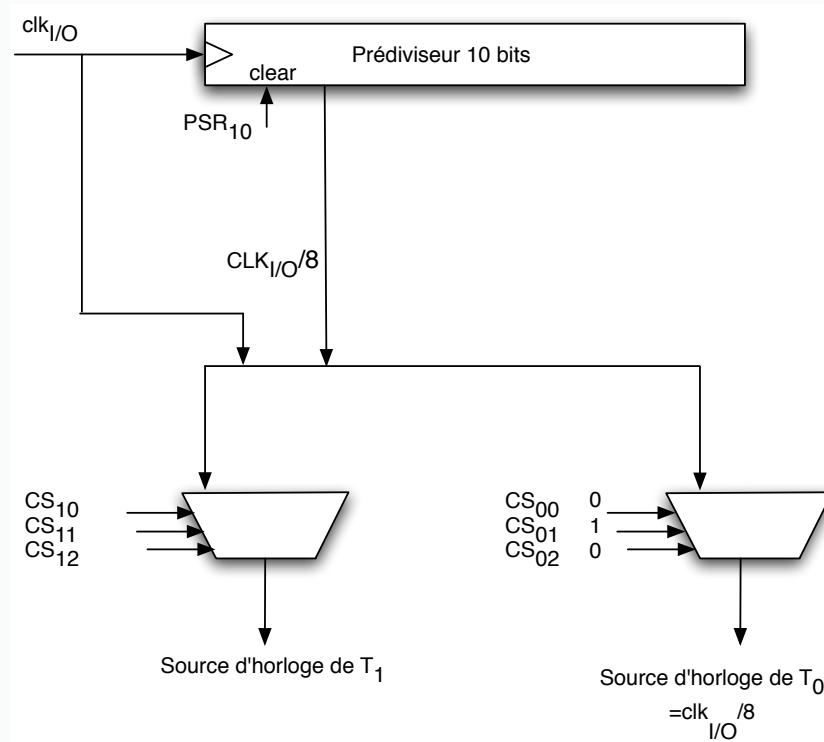
Fonction de

67/122 de bits

Mise en oeuvre d'une

interruption

## Exemple : Pré-division par 8 du timer 0



Les timers

Le pré-diviseur de  $T_0$  et  $T_1$

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : ADC  
Les Registres de l'ADC

Les timers

Le timer 0

Le pré-diviseur de  $T_0$  et  $T_1$

Le timer 1

Les ITs du timer 1

Les modes de la MLI

Programmation

Langage C

Fonctions

Fonction de

68/122 de bits

Mise en oeuvre d'une

interruption

## Registre impliqué : *SFIOR*

### *ADTS2 ADTS1 ADTS0 ACME PUD PSR2 PSR10* (*SFIOR*)

- *PSR10* ← 1 permet un reset du pré-diviseur.
- Attention, un reset du pré-diviseur les affectent tous les deux **timers**.

Les **timers**

Le pré-diviseur de  $T_0$  et  $T_1$

Info Indus 8 bits

Delfieu

#### Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

#### Les interruptions externes

Les *PORTS*

#### Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

#### Les **timers**

Le **timer** 0  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer** 1  
Les ITs du **timer** 1  
Les modes de la **MLI**

#### Programmation

Langage C  
Fonctions  
Fonction de mise en oeuvre d'une interruption

69 / 122

## Timer 1

- 16 bits ;
- 2 sorties indépendantes de comparaison ;
- Une entrée de capture avec annulation de bruit ;
- 2 sorties **MLI**
- Générateur de fréquence
- Compteur externe d'événements
- 4 sources d'*interruption* : *TOV1, OCF1A, OCF1B; ICF1*

Les **timers**

Le **timer** 1

Info Indus 8 bits

Delfieu

#### Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

#### Les interruptions externes

Les *PORTS*

#### Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

#### Les **timers**

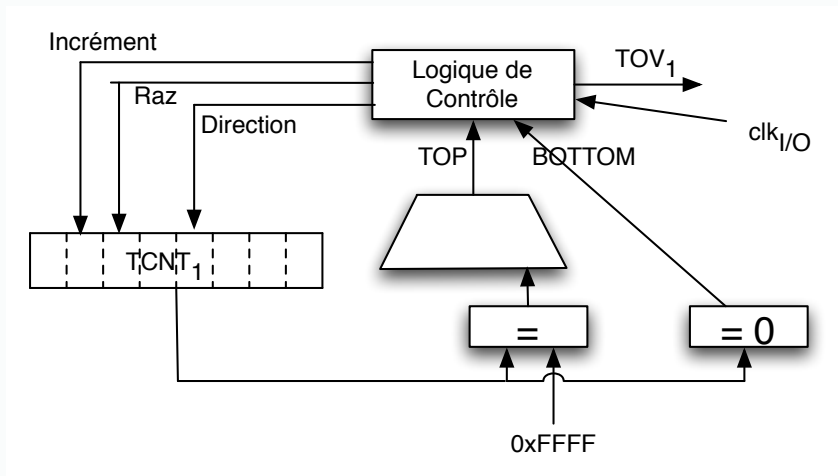
Le **timer** 0  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer** 1  
Les ITs du **timer** 1  
Les modes de la **MLI**

#### Programmation

Langage C  
Fonctions  
Fonction de mise en oeuvre d'une interruption

70 / 122

# Aperçu global du timer 1



Les timers

Le timer 1

Info Indus 8 bits

Delfieu

## Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

## Les interruptions externes

Les PORTS

## Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : ADC  
Les Registres de l'ADC

## Les timers

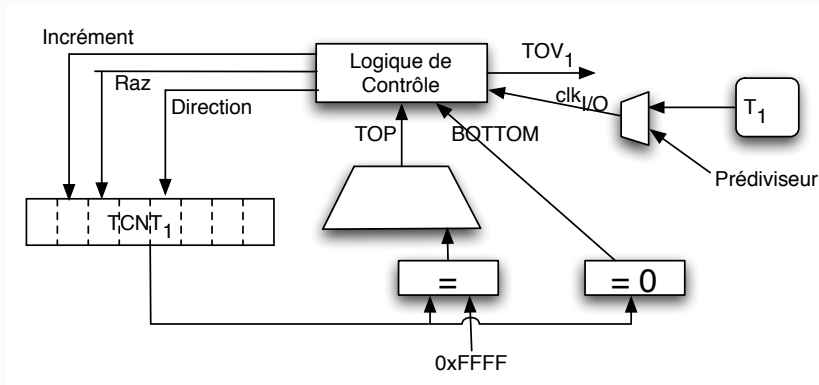
Le timer 0  
Le pré-diviseur de  $T_0$  et  $T_1$

**Le timer 1**  
Les ITs du timer 1  
Les modes de la MLI

## Programmation

Langage C  
Fonctions  
Fonction de  $n$  bits  
71 / 122 de bits  
Mise en oeuvre d'une interruption

# Aperçu global du timer 1



Les timers

Le timer 1

Info Indus 8 bits

Delfieu

## Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

## Les interruptions externes

Les PORTS

## Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : ADC  
Les Registres de l'ADC

## Les timers

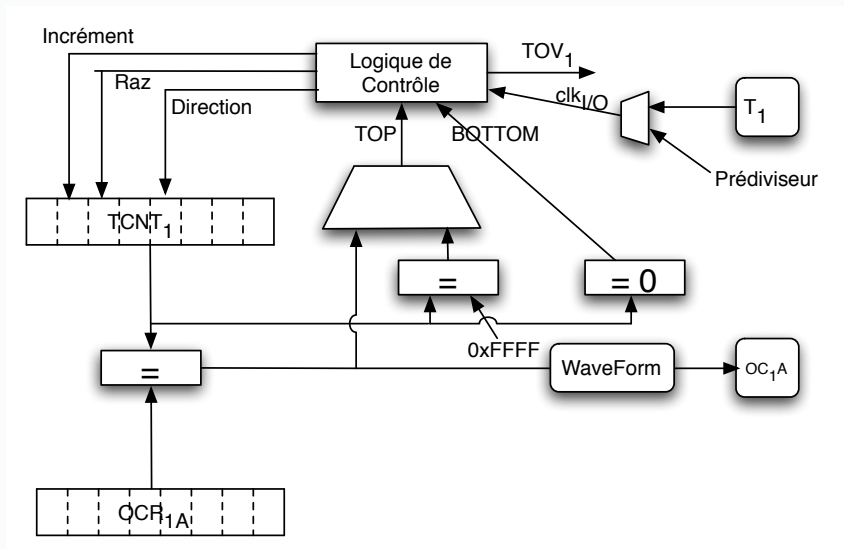
Le timer 0  
Le pré-diviseur de  $T_0$  et  $T_1$

**Le timer 1**  
Les ITs du timer 1  
Les modes de la MLI

## Programmation

Langage C  
Fonctions  
Fonction de  $n$  bits  
72 / 122 de bits  
Mise en oeuvre d'une interruption

# Aperçu global du timer 1



Les timers

Le timer 1

Info Indus 8 bits

Delfieu

## Présentation générale

- Introduction
- Architecture
- Mémoire
- Registres Systèmes

## Les interruptions externes

Les PORTS

## Le traitement des valeurs Analogiques

- Le Comparateur Analogique
- Convertisseur Analogique-Numérique : ADC
- Les Registres de l'ADC

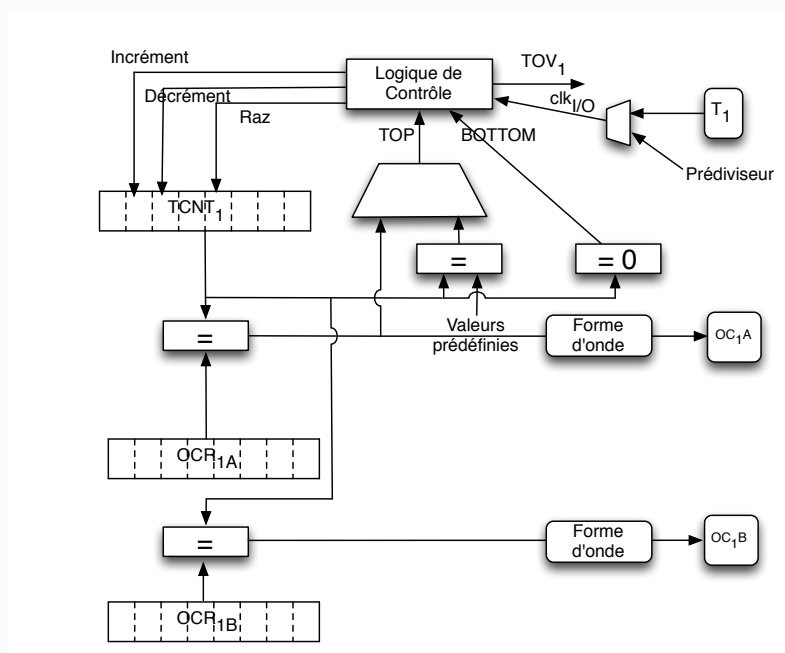
## Les timers

- Le timer 0
- Le pré-diviseur de  $T_0$  et  $T_1$
- Le timer 1
- Les ITs du timer 1
- Les modes de la MLI

## Programmation

- Langage C
- Fonctions
- Fonction de  $73/122$  de bits
- Mise en oeuvre d'une interruption

# Aperçu global du timer 1



Les timers

Le timer 1

Info Indus 8 bits

Delfieu

## Présentation générale

- Introduction
- Architecture
- Mémoire
- Registres Systèmes

## Les interruptions externes

Les PORTS

## Le traitement des valeurs Analogiques

- Le Comparateur Analogique
- Convertisseur Analogique-Numérique : ADC
- Les Registres de l'ADC

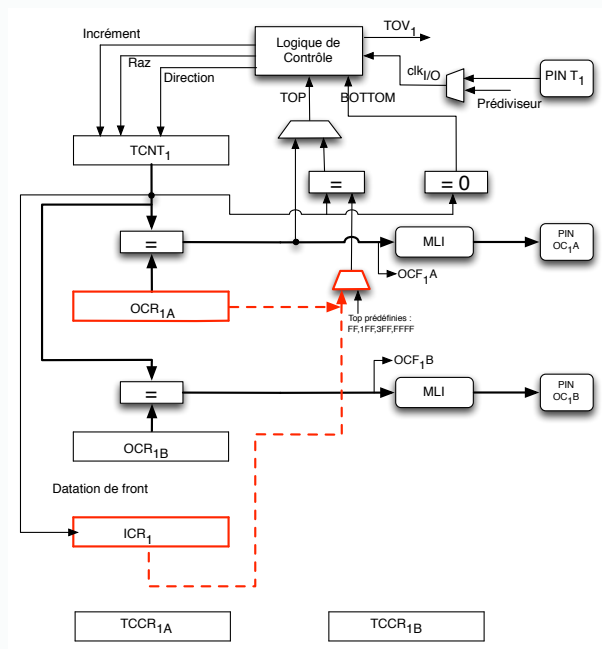
## Les timers

- Le timer 0
- Le pré-diviseur de  $T_0$  et  $T_1$
- Le timer 1
- Les ITs du timer 1
- Les modes de la MLI

## Programmation

- Langage C
- Fonctions
- Fonction de  $74/122$  de bits
- Mise en oeuvre d'une interruption

# Aperçu global du timer 1



Les timers

Le timer 1

Info Indus 8 bits

Delfieu

## Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

## Les interruptions externes

Les PORTS

## Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

## Les timers

Le timer 0  
Le pré-diviseur de  $T_0$  et  $T_1$   
**Le timer 1**  
Les ITs du timer 1  
Les modes de la MLI

## Programmation

Langage C  
Fonctions  
Fonction de  $n$  bits  
75 / 122 de bits  
Mise en oeuvre d'une interruption

# Les registres du timer 1

Les registres 16 bits :

- le registre de comptage  $TCNT_1$
- le registre de sortie de comparaison :  $OCR_{1A/B}$
- le registre de capture d'entrée  $ICR_1$

Les registres de programmation 8 bits :

- Les registres de contrôle du  $TCCR_{1A}$  et  $TCCR_{1B}$
- le registre d'autorisation des ITs :  $TIMSK$
- le registre des drapeaux des ITs  $TIFR$ .

Les timers

Le timer 1

Info Indus 8 bits

Delfieu

## Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

## Les interruptions externes

Les PORTS

## Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

## Les timers

Le timer 0  
Le pré-diviseur de  $T_0$  et  $T_1$   
**Le timer 1**  
Les ITs du timer 1  
Les modes de la MLI

## Programmation

Langage C  
Fonctions  
Fonction de  $n$  bits  
76 / 122 de bits  
Mise en oeuvre d'une interruption

# Les ITs du timer 1

- 4 signaux d'*interruptions*:
  - $TOV_1$  : *Interruption* de débordement du 1 ;
  - $OCF1_A$  : *Interruption* de égalité de seuil :  $TCNT_1$  avec  $OCR1_A$  ;
  - $OCF1_B$  : *Interruption* de égalité de seuil :  $TCNT_1$  avec  $OCR1_B$  ;
  - $ICF_1$  : *Interruption* de capture de front sur  $ICP_1$  ;
- Ces signaux d'*interruption* sont visibles dans avec des drapeaux contenus dans  $TIFR$
- Ces *interruptions* sont masquables individuellement dans  $TIMSK$ .

Les timers

Le timer 1

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les timers

Le timer 0  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le timer 1  
**Les ITs du timer 1**  
Les modes de la MLI

Programmation

Langage C  
Fonctions  
Fonction de  $n$  bits  
77 / 122 de bits  
Mise en oeuvre d'une interruption

# TIMSK

$OCIE_2$   $TOIE_2$   $TICIE_1$   $OCIE_{1A}$   $OCIE_{1B}$   $TOIE_1$  –  $TOIE_0$   
( $TIMSK$ )

- $TICIE_1$ : "Timer/Counter1, Input Capture Interrupt Enable" Dès que  $TICIE_1$  et  $I$  sont à un, l'*interruption* de capture est activée. L'*interruption* ad hoc est alors exécutée lorsque  $ICF_1 \leftarrow 1$
- $OCIE_{1A}$ : "Timer/Counter1, Output Compare A Match Interrupt Enable". Dès que  $OCIE_{1A}$  et  $I$  sont à un, l'*interruption* de comparaison est activée. L'*interruption* ad hoc est alors exécutée lorsque  $OCF1_A \leftarrow 1$

Les timers

Le timer 1

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les timers

Le timer 0  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le timer 1  
**Les ITs du timer 1**  
Les modes de la MLI

Programmation

Langage C  
Fonctions  
Fonction de  $n$  bits  
78 / 122 de bits  
Mise en oeuvre d'une interruption



# TIMSK

$OCIE_2$   $TOIE_2$   $TICIE_1$   $OCIE_{1A}$   $OCIE_{1B}$   $TOIE_1$  –  $TOIE_0$   
(TIMSK)

- $OCIE_{1B}$ : "Timer/Counter1, Output Compare B Match Interrupt Enable". Quand  $OCIE_{1A} \leftarrow 1$ , avec le bit  $I$  du registre d'état mis à un, toutes les *interruptions* sont donc globalement autorisées et en particulier l'interruption de comparaison. L'*interruption* ad hoc est alors exécutée lorsque  $OCF_{1B}$  reçoit un.
- $TOIE_1$ : "Timer/counter 1, Overflow Interrupt Enabled" Dès que  $TOIE_1$  et  $I$  sont à un, l'*interruption* de débordement est activée.

Les **timers**

Le **timer 1**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer 0**  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer 1**  
Les **ITs** du **timer 1**  
Les modes de la **MLI**

Programmation

Langage C  
Fonctions

Fonction de **79/122** de bits  
Mise en oeuvre d'une interruption

# TIFR

$OCF_2$   $TOV_2$   $ICF_1$   $OCF_{1A}$   $OCF_{1B}$   $TOV_1$  –  $TOV_0$  (TIFR)

- $ICF_1$ : "Timer/Counter1, Input Capture Flag"
  - Ce bit est un drapeau (flag) qui reçoit un quand la patte  $ICP_1$  reçoit un signal.
  - Quand  $ICR_1$  est utilisé pour stocker **TOP** avec un mode de  $WGM_{13} : 0$ ,  $ICF_1$  est positionné à un quand le compteur atteint **TOP**.
  - $ICF_1$  est automatiquement **raz** par le sous-programme d'*interruption* est exécuté.
- $OCF_{1A}$ : "Timer/Counter1, Output Compare B Match Flag"
  - Ce bit est un drapeau (flag) qui reçoit un quand l'égalité de  $OCR_{1A}$  avec  $TCNT_1$  se produit.
  - $OCF_{1A}$  est automatiquement **raz** par le sous-programme d'*IT* est exécuté.

Les **timers**

Le **timer 1**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer 0**  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer 1**  
Les **ITs** du **timer 1**  
Les modes de la **MLI**

Programmation

Langage C  
Fonctions

Fonction de **80/122** de bits  
Mise en oeuvre d'une interruption

# TIFR

$OCF2$   $TOV_2$   $ICF_1$   $OCF1A$   $OCF1B$   $TOV_1$  –  $TOV_0$  (*TIFR*)

- $OCF1AB$ : "Timer/Counter1, Output Compare B Match Flag"
  - Ce bit est un drapeau (flag) qui reçoit un quand l'égalité de  $OCR1B$  avec  $TCNT_1$  se produit.
  - $OCF1B$  est automatiquement **raz** par le sous-programme d'*interruption* est exécuté.
- $TOV_1$ : "Timer/Counter1, Overflow Flag"
  - Dans les modes normal et **CTC**,  $TOV_1$  est mis à un sur un dépassement de capacité
  - $TOV_1$  est automatiquement **raz** par le sous-programme d'*interruption* est exécuté.

Les **timers**

Le **timer 1**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer 0**  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer 1**  
**Les ITs du timer 1**  
Les modes de la **MLI**

Programmation

Langage C  
Fonctions  
Fonction de **81/122** de bits  
Mise en oeuvre d'une interruption

# $OCR1A$ et $OCR1B$

- $OCR1A$  peut être utilisé pour définir le **top**
- $OCR1A$ ,  $OCR1B$  sont deux registres de seuil qui peuvent être comparé avec  $TCNT_1$
- Le résultat de la comparaison peut être utilisé pour générer une forme de type **PWM** ou bien une fréquence variable sur les pattes  $OC1A$ ,  $OC1B$
- La comparaison d'égalité va positionner un flag  $OCF1A$ ,  $OCF1B$  qui peut être utilisé pour générer une *interruption*.

Les **timers**

Le **timer 1**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer 0**  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer 1**  
**Les ITs du timer 1**  
Les modes de la **MLI**

Programmation

Langage C  
Fonctions  
Fonction de **82/122** de bits  
Mise en oeuvre d'une interruption

# ICR<sub>1</sub>

- Le registre de capture d'entrée **ICR<sub>1</sub>** peut dater un événement (front) sur la patte **ICP1**.
- Cette partie inclue un filtre réducteur de bruit évitant la capture de fronts parasites.
- **ICR<sub>1</sub>** peut servir à déterminer la valeur **TOP** (Si il n'est pas utilisé pour la capture).

Les **timers**

Le **timer 1**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer 0**  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer 1**  
**Les ITs du timer 1**  
Les modes de la **MLI**

Programmation

Langage C  
Fonctions  
Fonction de **83/122** de bits  
Mise en oeuvre d'une interruption

# TCCR<sub>1A</sub>

**COM1A<sub>1</sub> COM1A<sub>0</sub> COM1B<sub>1</sub> COM1B<sub>0</sub> FOC1<sub>A</sub> FOC1<sub>B</sub> WGM1<sub>1</sub> WGM1<sub>0</sub>**  
(**TCCR<sub>1A</sub>**)

- **COM1A<sub>1</sub>, COM1A<sub>0</sub>** : Mode normal ou inversée de la **MLI A**
- **COM1B<sub>1</sub>, COM1B<sub>0</sub>** : Mode normal ou inversée de la **MLI B**
- **FOC1<sub>A</sub>, FOC1<sub>B</sub>** : Forçage de comparaison sur **OCR1A** ou **OCR1B**
- **WGM1<sub>1</sub>, WGM1<sub>0</sub>** : Modes du **timer**

Les **timers**

Le **timer 1**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer 0**  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer 1**  
**Les ITs du timer 1**  
Les modes de la **MLI**

Programmation

Langage C  
Fonctions  
Fonction de **84/122** de bits  
Mise en oeuvre d'une interruption

# TCCR<sub>1B</sub>

ICN<sub>1</sub> ICES – WGM<sub>13</sub> WGM<sub>12</sub> CS<sub>12</sub> CS<sub>11</sub> CS<sub>10</sub> (TCCR<sub>1B</sub>)

- ICN<sub>1</sub> : Réduction de bruit (Noise Canceler)
- ICES<sub>1</sub> : Sélection du type de front (Edge Select)
- WGM<sub>13</sub>, WGM<sub>12</sub> : Modes du timer
- CS<sub>12:0</sub> : Pré-diviseur de la fréquence de comptage :

Les timers

Le timer 1

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : ADC  
Les Registres de l'ADC

Les timers

Le timer 0  
Le pré-diviseur de T<sub>0</sub> et T<sub>1</sub>  
Le timer 1  
Les ITs du timer 1  
Les modes de la MLI

Programmation

Langage C  
Fonctions  
Fonction de mise en oeuvre d'une interruption

85 / 122

# TCCR<sub>1B</sub>

ICN<sub>1</sub> ICES – WGM<sub>13</sub> WGM<sub>12</sub> CS<sub>12</sub> CS<sub>11</sub> CS<sub>10</sub> (TCCR<sub>1B</sub>)

ICN<sub>1</sub> : Input Capture Noise Canceler

- Si ICN<sub>1</sub> ← 1 cela active le réducteur de bruit de l'entrée de capture.
- Quand le réducteur de bruit est activé l'entrée de capture ICN<sub>1</sub> est filtrée.
- Le filtre produit une sortie si on a 4 sorties successives égales ce qui induira un délai de 4 cycles d'horloges.

Les timers

Le timer 1

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : ADC  
Les Registres de l'ADC

Les timers

Le timer 0  
Le pré-diviseur de T<sub>0</sub> et T<sub>1</sub>  
Le timer 1  
Les ITs du timer 1  
Les modes de la MLI

Programmation

Langage C  
Fonctions  
Fonction de mise en oeuvre d'une interruption

86 / 122

## TCCR<sub>1B</sub>

ICN<sub>1</sub> ICES – WGM<sub>13</sub> WGM<sub>12</sub> CS<sub>12</sub> CS<sub>11</sub> CS<sub>10</sub> (TCCR<sub>1B</sub>)

ICES<sub>1</sub> : Input Capture Edge Select

- Ce bit sélectionne quel type de front permet de déclencher la capture.
- quand ICES<sub>1</sub> = 0 c'est un front descendant.
- quand ICES<sub>1</sub> = 1 c'est un front montant.
- Quand une capture est déclenchée, TCNT<sub>1</sub> est vidé dans ICR<sub>1</sub>.
- ICF<sub>1</sub> est positionné à un et peut générer une *interruption*.
- Quand c'est ICR<sub>1</sub> qui détermine TOP alors ICP<sub>1</sub> est déconnectée et le mécanisme de capture ne fonctionne plus.

Les timers

Le timer 1

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : ADC  
Les Registres de l'ADC

Les timers

Le timer 0  
Le pré-diviseur de T<sub>0</sub> et T<sub>1</sub>  
Le timer 1  
Les ITs du timer 1  
Les modes de la MLI

Programmation

Langage C  
Fonctions  
Fonction de remplissage de bits  
87 / 122  
Mise en oeuvre d'une interruption

## TCCR<sub>1B</sub>

ICN<sub>1</sub> ICES – WGM<sub>13</sub> WGM<sub>12</sub> CS<sub>12</sub> CS<sub>11</sub> CS<sub>10</sub> (TCCR<sub>1B</sub>)

WGM<sub>13</sub> : 2 : Waveform Generation Mode

Abordé à la section "Les 16 modes de la MLI"

Les timers

Le timer 1

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : ADC  
Les Registres de l'ADC

Les timers

Le timer 0  
Le pré-diviseur de T<sub>0</sub> et T<sub>1</sub>  
Le timer 1  
Les ITs du timer 1  
Les modes de la MLI

Programmation

Langage C  
Fonctions  
Fonction de remplissage de bits  
88 / 122  
Mise en oeuvre d'une interruption

# TCCR<sub>1B</sub>

ICN<sub>1</sub> ICES – WGM<sub>13</sub> WGM<sub>12</sub> CS<sub>12</sub> CS<sub>11</sub> CS<sub>10</sub> (TCCR<sub>1B</sub>)  
CS<sub>12</sub> : 0 Clock Select

Ces trois bits sélectionnent la pré-division, ils ont été déjà évoqués à propos du timer0:

Les timers

Le timer 1

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : ADC  
Les Registres de l'ADC

Les timers

Le timer 0  
Le pré-diviseur de T<sub>0</sub> et T<sub>1</sub>

Le timer 1

Les ITs du timer 1  
Les modes de la MLI

Programmation

Langage C  
Fonctions  
Fonction de mise en oeuvre de bits de bits  
Mise en oeuvre d'une interruption

# TCNT<sub>1</sub>

Timer Counter Timer 1 (TCNT<sub>1</sub>)

- TCNT<sub>1</sub> est le registre de comptage du timer1.
- Modifier TCNT<sub>1</sub> pendant le comptage peut produire un effet indésirable si la nouvelle valeur est supérieure aux valeurs de comparaison : on manque alors une comparaison.
- De plus, écrire une nouvelle valeur reporte la comparaison au cycle suivant d'horloge.
- TCNT<sub>1</sub> est co,trôlé par une unité de comptage.

Les timers

Le timer 1

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : ADC  
Les Registres de l'ADC

Les timers

Le timer 0  
Le pré-diviseur de T<sub>0</sub> et T<sub>1</sub>

Le timer 1

Les ITs du timer 1  
Les modes de la MLI

Programmation

Langage C  
Fonctions  
Fonction de mise en oeuvre de bits de bits  
Mise en oeuvre d'une interruption

## TCNT<sub>1</sub> : Valeurs extrêmes

- Le BOTTOM est toujours 0
- TOP est soit :
  - Une valeur prédéfinie : 0x0FF, 0x1FF, 0x3FF, 0xFFFF
  - Ou bien d'autres valeurs libres placées dans ICR<sub>1</sub> ou OCR<sub>1A</sub>

Les timers

Le timer 1

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : ADC  
Les Registres de l'ADC

Les timers

Le timer 0  
Le pré-diviseur de T<sub>0</sub> et T<sub>1</sub>

Le timer 1

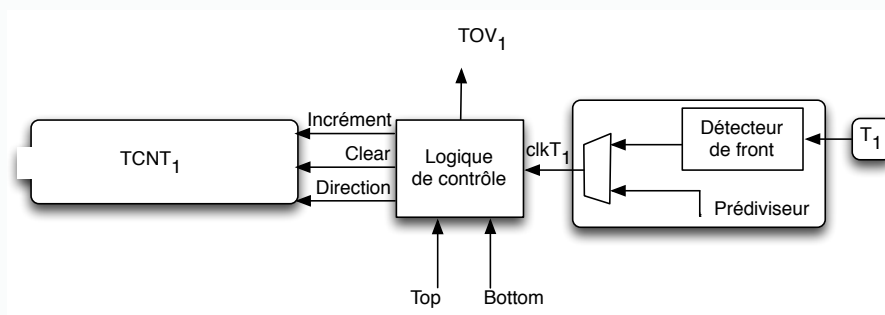
Les ITs du timer 1  
Les modes de la MLI

Programmation

Langage C  
Fonctions

Fonction de mise en œuvre d'une interruption

## TCNT<sub>1</sub> : Unité de Comptage



Les timers

Le timer 1

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : ADC  
Les Registres de l'ADC

Les timers

Le timer 0  
Le pré-diviseur de T<sub>0</sub> et T<sub>1</sub>

Le timer 1

Les ITs du timer 1  
Les modes de la MLI

Programmation

Langage C  
Fonctions

Fonction de mise en œuvre d'une interruption

## TCNT<sub>1</sub> : Unité de Comptage

- TCNT<sub>1</sub> est piloté par le bloc logique par *clear*, *increment* ou *decrement* selon la source *clkT<sub>1</sub>*
- *clkT<sub>1</sub>* peut être une source externe ou interne selon CS12 : 0.
- Quand CS1<sub>2:0</sub> = 0 le timer 1 est arrêté, la valeur de TCNT<sub>1</sub> étant toujours accessible par le CPU
- Une écriture par le CPU a la priorité sur toute autre opérations.

Les timers

Le timer 1

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : ADC  
Les Registres de l'ADC

Les timers

Le timer 0  
Le pré-diviseur de T<sub>0</sub> et T<sub>1</sub>

Le timer 1

Les ITs du timer 1  
Les modes de la MLI

Programmation

Langage C  
Fonctions

Fonction de mise en oeuvre d'une interruption  
93 / 122 de bits

## TCNT<sub>1</sub> : Unité de Comptage

- La séquence de comptage est déterminée par *Waveform Generation* c.a.d. par WGM13 : 0
- Les *waveforms* générées dépendent en fait des modes de comptage sur TCNT<sub>1</sub>
- le bit de débordement TOV<sub>1</sub> est positionné selon le mode choisi sur WGM13 : 0

Les timers

Le timer 1

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : ADC  
Les Registres de l'ADC

Les timers

Le timer 0  
Le pré-diviseur de T<sub>0</sub> et T<sub>1</sub>

Le timer 1

Les ITs du timer 1  
Les modes de la MLI

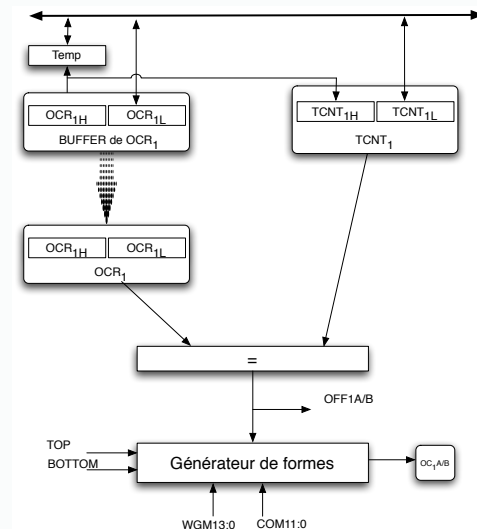
Programmation

Langage C  
Fonctions

Fonction de mise en oeuvre d'une interruption  
94 / 122 de bits



## TCNT<sub>1</sub> : Unité de Comparaison de sorties



Les **timers**

Le **timer 1**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer 0**  
Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer 1**  
Les **ITs** du **timer 1**  
Les modes de la **MLI**

Programmation

Langage C  
Fonctions

Fonction de **95/122** de bits  
Mise en oeuvre d'une interruption

## Les 16 modes de la MLI

- Le générateur de formes utilise **COM1x1 : 0** pour les modes normal, **CTC** et **PWM**.
- En mode **PWM** **COM1x1 : 0** définissent si la **PWM** est inversée ou non.
- Les modes sont définis avec : **WGM13:0** et **COM1x1 : 0**
- Il y a 16 modes possibles.

Les **timers**

Les modes de la **MLI**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer 0**  
Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer 1**  
Les **ITs** du **timer 1**  
Les **modes** de la **MLI**

Programmation

Langage C  
Fonctions

Fonction de **96/122** de bits  
Mise en oeuvre d'une interruption

## Le mode Normal

$WGM_{13:0} = 0$

- Dans ce mode le comptage s'incrmente jusqu'à la valeur  $TOP = 0xFFFF$  puis recommence à partir de  $BOTTOM = 0x0000$ .  $TOV_1$  passe alors à 1 et ce bit peut servir de 17 bits.
- On peut à tout moment écrire une valeur dans  $TCNT_1$
- En mode capture, il faut évidemment que l'intervalle entre les 2 événements extérieurs n'excède pas 65536 cycles d'horloge.
- Sinon, on peut gérer des intervalles plus long avec la prise en compte d'une *interruption*.

Les **timers**

Les modes de la **MLI**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer 0**  
Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer 1**  
Les ITs du **timer 1**

Les modes de la **MLI**

Programmation

Langage C  
Fonctions

Fonction de **97 / 122** de bits  
Mise en oeuvre d'une interruption

## Le mode CTC

$WGM_{13:0} = 4$  ou  $12$

- Dans ce mode  $OCR_{1A}$  ou  $ICR_1$  sont utilisés pour manipuler la résolution du compteur.
- Dans ce mode  $TCNT_1$  est **raz** lorsque il atteint la valeur de  $OCR_{1A}$  (avec  $WGM_{13:0} = 4$ ) ou  $ICR_1$  (avec  $WGM_{13:0} = 12$ ).
- Dans ce mode c'est donc  $OCR_{1A}$  ou  $ICR_1$  qui définissent **TOP**
- Les sorties  $OC_{1X}$  commuteront alors à chaque fois que  $TCNT_1$  atteindra la valeur **TOP** (si  $COM_{1x1} : 0 = 1$ ).

Les **timers**

Les modes de la **MLI**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer 0**  
Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer 1**  
Les ITs du **timer 1**

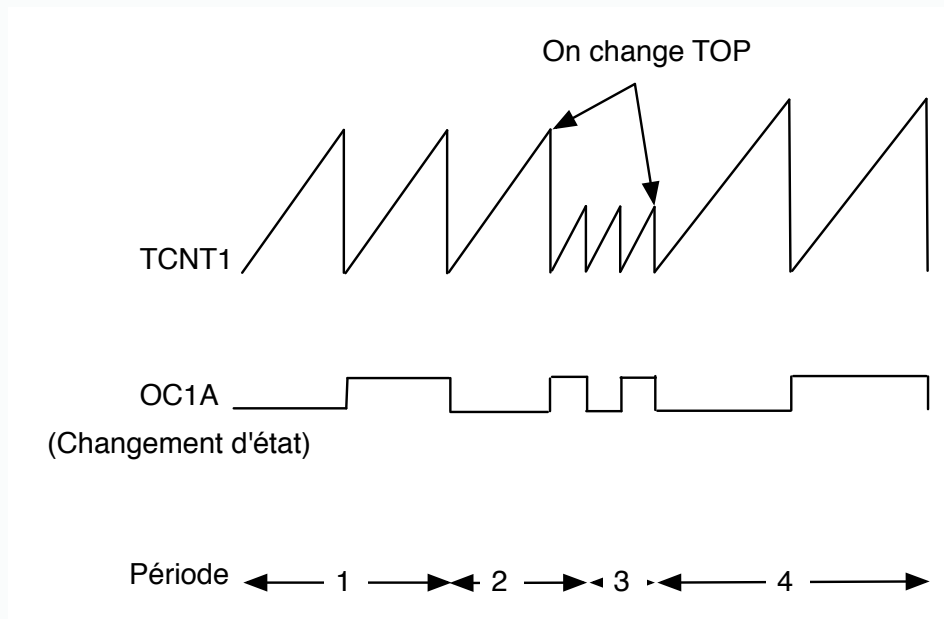
Les modes de la **MLI**

Programmation

Langage C  
Fonctions

Fonction de **98 / 122** de bits  
Mise en oeuvre d'une interruption

## Le mode CTC



Les **timers**

Les modes de la **MLI**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer** 0  
Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer** 1  
Les ITs du **timer** 1

Les modes de la **MLI**

Programmation

Langage C  
Fonctions

Fonction de **99/122** de bits

Mise en oeuvre d'une interruption

## Le mode CTC

- Dans ce mode la fréquence max est :

$$f_{OC1A} = \frac{f_{clk_{I/O}}}{2}$$

- Sinon la fréquence est définie par :

$$f_{OC1A} = \frac{f_{clk_{I/O}}}{2.N.(1 + OCR1A)}$$

( $N$ , facteur de prédivision : 1,8,64,256,1024)

Les **timers**

Les modes de la **MLI**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer** 0  
Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer** 1  
Les ITs du **timer** 1

Les modes de la **MLI**

Programmation

Langage C  
Fonctions

Fonction de **100/122** de bits

Mise en oeuvre d'une interruption

## Le mode Fast PWM

$WGM_{13:0} = 5, 6, 7, 14$  ou  $15$

- PWM à haute fréquence.
- Simple pente : Il y a donc une commutation par triangle. Ce mode est deux fois plus rapide que les autres PWM
- $TCNT_1$  compte de **BOTTOM** à **TOP** et revient à **BOTTOM**.
- Non inversé : La sortie est **raz** sur égalité de  $TCNT_1$  et  $OCR_{1x}$ ,  $OCR_{1x}$  est mise à un sur **BOTTOM**.
- Mode inversé : La sortie est à un sur égalité et **raz** sur **BOTTOM**.
- Dans ce mode deux valeurs permettent d'agir sur  $OC_{1A}$  : la valeur **TOP** et une valeur de comparaison stockée dans  $OCR_{1A}$ .

Les timers

Les modes de la MLI

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les timers

Le timer 0  
Le pré-diviseur de  $T_0$  et  $T_1$

Le timer 1  
Les ITs du timer 1

Les modes de la MLI

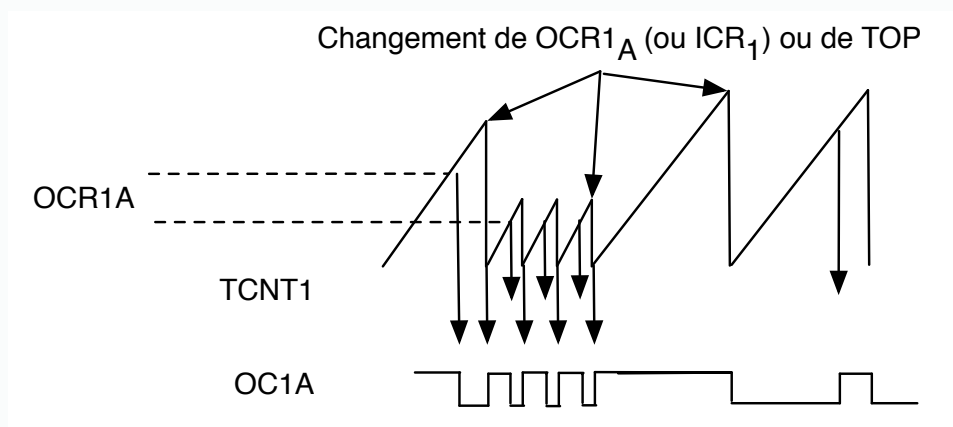
Programmation

Langage C  
Fonctions

Fonction de  $101/122$  de bits

Mise en oeuvre d'une interruption

## Le mode Fast PWM



Les timers

Les modes de la MLI

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les timers

Le timer 0  
Le pré-diviseur de  $T_0$  et  $T_1$

Le timer 1  
Les ITs du timer 1

Les modes de la MLI

Programmation

Langage C  
Fonctions

Fonction de  $102/122$  de bits

Mise en oeuvre d'une interruption

## Le mode Fast PWM

- Dans ce mode la fréquence est définie par :

$$f_{OC_{1A}PWM} = \frac{f_{clk_{I/O}}}{N.(1 + TOP)}$$

(N, facteur de pré-division : 1,8,64,256,1024)

- Attention à ne pas changer la valeur TOP avec une valeur inférieure à la valeur courante du comparateur OCR<sub>1A</sub>.
- Pour une valeur TOP= 0 on aura un quasi front à chaque top d'horloge.

Les timers

Les modes de la MLI

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : ADC  
Les Registres de l'ADC

Les timers

Le timer 0  
Le pré-diviseur de T<sub>0</sub> et T<sub>1</sub>

Le timer 1  
Les ITs du timer 1

Les modes de la MLI

Programmation

Langage C  
Fonctions

Fonction de mise en oeuvre d'une interruption

103 / 122

## Le mode PWM à phase correcte

WGM<sub>13:0</sub>= 1, 2, 3, 10 ou 11

- Ce mode permet d'obtenir une haute résolution de PWM à phase correcte.
- Double pente.
- Le compteur compte de la valeur BOTTOM à TOP puis il dé-crémente jusqu'à BOTTOM et recommence indéfiniment.

Les timers

Les modes de la MLI

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : ADC  
Les Registres de l'ADC

Les timers

Le timer 0  
Le pré-diviseur de T<sub>0</sub> et T<sub>1</sub>

Le timer 1  
Les ITs du timer 1

Les modes de la MLI

Programmation

Langage C  
Fonctions

Fonction de mise en oeuvre d'une interruption

104 / 122

## Le mode *PWM* à phase correcte

$WGM1_{3:0} = 1, 2, 3, 10$  ou  $11$

- En mode non inversé,  $OC_{1X}$  est **raz** sur l'égalité de  $TCNT_1$  en comptant, il est **mis à un** en mode décomptage.
- En mode inversé, l'opération est inversée.
- La résolution varie de la valeur 2 bits à 16 bits, elle dépend de la valeur de  $OCR_{1A}$  ou  $ICR_1$ .
- La résolution de la *PWM* est donnée par l'équation :

$$R_{PCPWM} = \frac{\log(TOP + 1)}{\log(2)}$$

Les **timers**

Les modes de la **MLI**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer 0**  
Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer 1**  
Les ITs du **timer 1**

Les modes de la **MLI**

Programmation

Langage C  
Fonctions

Fonction de **105 / 122** bits

Mise en oeuvre d'une interruption

## Le mode *PWM* à phase correcte

$WGM1_{3:0} = 1, 2, 3, 10$  ou  $11$

- Dans ce mode le compteur est incrémenté jusqu'à ce que la valeur atteigne soit une des valeurs :  $0x00FF$ ,  $0x01FF$ , ou  $0x03FF$  ( $WGM1_{3:0} = 1, 2$ , ou  $3$ ), soit la valeur dans  $ICR_1$  ( $WGM1_{3:0} = 10$ ), soit la valeur dans  $OCR_{1A}$  ( $WGM1_{3:0} = 11$ ).
- Dès que  $TCNT_1$  a alors atteint **TOP** alors  $OC_{1X}$  commute et  $TCNT_1$  décompte dès lors.

Les **timers**

Les modes de la **MLI**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer 0**  
Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer 1**  
Les ITs du **timer 1**

Les modes de la **MLI**

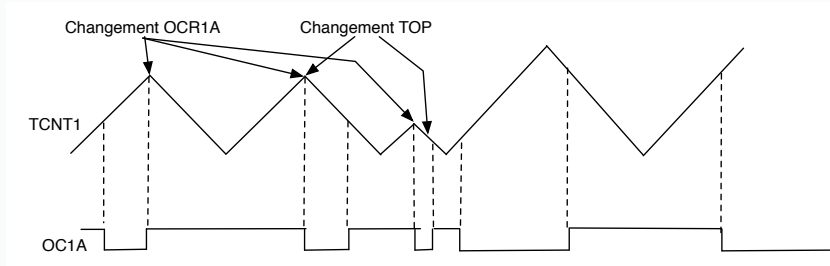
Programmation

Langage C  
Fonctions

Fonction de **106 / 122** bits

Mise en oeuvre d'une interruption

## Le mode *PWM* à phase correcte



Les **timers**

Les modes de la **MLI**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer 0**  
Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer 1**

Les ITs du **timer 1**

Les modes de la **MLI**

Programmation

Langage C

Fonctions

Fonction de **107 / 122**

Mise en oeuvre d'une interruption

## Le mode *PWM* à phase correcte

$WGM_{13:0} = 1, 2, 3, 10$  ou  $11$

- Le comptage se fait de **BOTTOM** à **TOP**, puis de **TOP** à **BOTTOM**.
- En mode non inversé :  $OC_{1x}$  est raz lors de la correspondance entre  $TCNT_1$  et  $OCR_{1x}$  lors du comptage en incrémentant, et mis à un sur égalité en décrémentant.
- En mode inversé, l'opération est inversée.
- Mode non inversé :  $COM_{1x1} : 0$  à 2
- Mode inversé :  $COM_{1x1} : 0$  à 3
- Dans ce mode la fréquence est définie par :

$$f_{OC_{1A}PCPWM} = \frac{f_{clk_{I/O}}}{2.N.TOP}$$

$N$ , facteur de pré-division : 1,8,64,256,1024

Les **timers**

Les modes de la **MLI**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer 0**  
Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer 1**

Les ITs du **timer 1**

Les modes de la **MLI**

Programmation

Langage C

Fonctions

Fonction de **108 / 122**

Mise en oeuvre d'une interruption

## Le mode *PWM* à phase correcte

( $WGM1_{3:0} = 8$  ou  $9$ )

- **MLI** centrée : Mode le plus utilisé en terme de contrôle moteur.
- Double pente : Donne une fréquence max plus basse comparée à celle en simple pente
- Compte indéfiniment de **BOTTOM** à **TOP** et de **TOP** à **BOTTOM**
- Mode non inversé  $OC_{1X}$  est :
  - **raz** sur égalité entre  $TCNT_1$  et  $OCR_{1x}$  en comptant
  - **mis à un** sur égalité en décomptant
- Mode inversé c'est l'inverse.
- Résolution de la *PWM* :

$$R_{PCPWM} = \frac{\log(TOP + 1)}{\log(2)}$$

Les **timers**

Les modes de la **MLI**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer** 0  
Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer** 1  
Les ITs du **timer** 1

Les modes de la **MLI**

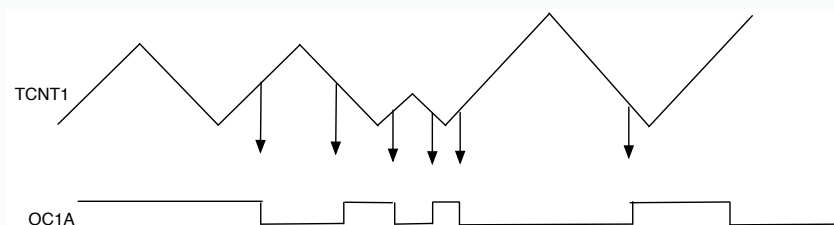
Programmation

Langage C  
Fonctions

Fonction de **109/122** de bits

Mise en oeuvre d'une interruption

## Le mode *PWM* à phase correcte



Les **timers**

Les modes de la **MLI**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer** 0  
Le pré-diviseur de  $T_0$  et  $T_1$

Le **timer** 1  
Les ITs du **timer** 1

Les modes de la **MLI**

Programmation

Langage C  
Fonctions

Fonction de **110/122** de bits

Mise en oeuvre d'une interruption



## Le mode *PWM* à phase et fréquence correcte

( $WGM_{13:0} = 8$  ou  $9$ )

- $OC_{1X}$  sera visible sur le port ssi le port a été programmé en sortie avec *DDRB*.
- Dans ce mode la fréquence est définie par :

$$f_{OC_{1A}PFCPWM} = \frac{f_{clk_{I/O}}}{2.N.TOP}$$

$N$ , facteur de pré-division : 1,8,64,256,1024.

Les **timers**

Les modes de la **MLI**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction

Architecture

Mémoire

Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique

Convertisseur Analogique-Numérique

: **ADC**

Les Registres de l'ADC

Les **timers**

Le **timer** 0

Le pré-diviseur de  $T_0$

et  $T_1$

Le **timer** 1

Les ITs du **timer** 1

Les modes de la **MLI**

Programmation

Langage C

Fonctions

Fonction de

111 / 122

de bits

Mise en oeuvre d'une

interruption

## $COM_{1x1}$ et $COM_{1x0}$ en mode normal et CTC

$COM_{1A_1}$   $COM_{1A_0}$   $COM_{1B_1}$   $COM_{1B_0}$   $FOC_{1_1}$   $FOC_{1_B}$   $WGM_{1_1}$   $WGM_{1_0}$   
( $TCCR_{1A}$ )

$COM_{1x1}$	$COM_{1A_0}$	Description
0	0	Port libre $OC_{1x}$ déconnectés
0	1	$OC_{1A}$ commute sur égalité quand $OC_{1x} = TCNT_1$
1	0	<b>raz</b> sur égalité avec <b>TOP</b>
1	1	<b>mis à un</b> sur égalité avec <b>BOTTOM</b>

Les **timers**

Les modes de la **MLI**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction

Architecture

Mémoire

Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique

Convertisseur Analogique-Numérique

: **ADC**

Les Registres de l'ADC

Les **timers**

Le **timer** 0

Le pré-diviseur de  $T_0$

et  $T_1$

Le **timer** 1

Les ITs du **timer** 1

Les modes de la **MLI**

Programmation

Langage C

Fonctions

Fonction de

112 / 122

de bits

Mise en oeuvre d'une

interruption

## COM1x1 et COM1x0 en mode fast PWM

COM1x1	COM1x0	Description
0	0	OC <sub>1x</sub> déconnectés
0	1	WGM = 15 OC <sub>1A</sub> commute sur égalité OC <sub>1B</sub> déconnecté
1	0	raz OC <sub>1x</sub> sur égalité seuil en comptant mis à un sur <b>BOTTOM</b>
1	1	mis à un sur égalité avec seuil raz sur <b>BOTTOM</b>

Les **timers**

Les modes de la **MLI**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**

Les Registres de l'ADC

Les **timers**

Le **timer** 0  
Le pré-diviseur de T<sub>0</sub> et T<sub>1</sub>

Le **timer** 1  
Les ITs du **timer** 1

Les modes de la **MLI**

Programmation

Langage C  
Fonctions

Fonction de **113 / 122** bits

Mise en oeuvre d'une interruption

## COM1x1 et COM1x0 en mode phase et fréquence correcte

COM1A <sub>1</sub> COM1B <sub>1</sub>	COM1A <sub>0</sub> COM1B <sub>0</sub>	Description
0	0	Port libre OC <sub>1x</sub> déconnectés
0	1	OC <sub>1A</sub> commmte sur égalité, OC <sub>1B</sub> déconnecté
1	0	raz OC <sub>1x</sub> sur égalité du seuil en incrémentant mis à un en décrémentant sur égalité du seuill
1	1	mis à un sur égalité du seuil en incrémentant raz sur égalité du seuil en décrémentant

Les **timers**

Les modes de la **MLI**

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**

Les Registres de l'ADC

Les **timers**

Le **timer** 0  
Le pré-diviseur de T<sub>0</sub> et T<sub>1</sub>

Le **timer** 1  
Les ITs du **timer** 1

Les modes de la **MLI**

Programmation

Langage C  
Fonctions

Fonction de **114 / 122** bits

Mise en oeuvre d'une interruption

# WGM1<sub>3:0</sub> : Waveform Generation Mode

- Ces bits déterminent le sens du comptage et le type de mode.
- Il y a 15 modes : normal, CTC, et les trois modes PWM:
  - Fast PWM
  - Correct Phase PWM
  - Correct Phase and Frequency PWM

Les timers

Les modes de la MLI

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : ADC  
Les Registres de l'ADC

Les timers

Le timer 0  
Le pré-diviseur de T<sub>0</sub> et T<sub>1</sub>

Le timer 1  
Les ITs du timer 1

Les modes de la MLI

Programmation

Langage C  
Fonctions

Fonction de mise en oeuvre d'une interruption

# WGM1<sub>3:0</sub> : Waveform Generation Mode

n	WGM <sub>13</sub>	WGM <sub>12</sub>	WGM <sub>11</sub>	WGM <sub>10</sub>	Mode des Timers	TOP	M.a.j. OCR <sub>1x</sub>	TOV <sub>1</sub>
0	0	0	0	0	Normal	0xFFFF	Immed.	MAX
1	0	0	0	1	Pha. Corr. 8-bit	0x00FF	TOP	BOT.
2	0	0	1	0	Pha. Corr. 9-bit	0x01FF	TOP	BOT.
3	0	0	1	1	Pha. Corr. 10-bit	0x03FF	TOP	BOT.
4	0	1	0	0	CTC	OCR <sub>1A</sub>	Immed.	MAX
5	0	1	0	1	Fast PWM 8-bit	0x0FF	BOT.	TOP
6	0	1	1	0	Fast PWM 9-bit	0x1FF	BOT.	TOP
7	0	1	1	1	Fast PWM 10-bit	0x3FF	BOT.	TOP
8	1	0	0	0	Pha. freq. Corr.	ICR <sub>1</sub>	BOT.	BOT.
9	1	0	0	1	Pha. freq. Corr.	OCR <sub>1A</sub>	BOT.	BOT.
10	1	0	1	0	Pha. Corr.	ICR <sub>1</sub>	TOP	BOT.
11	1	0	1	1	Pha. Corr.	OCR <sub>1A</sub>	TOP	BOT.
12	1	1	0	0	CTC	ICR <sub>1</sub>	Immed.	MAX
13	1	1	0	1	Réservé	—	—	—
14	1	1	1	0	Fast PWM	ICR <sub>1</sub>	BOT.	TOP
15	1	1	1	1	Fast PWM	OCR <sub>1A</sub>	BOT.	TOP

Les timers

Les modes de la MLI

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les PORTS

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : ADC  
Les Registres de l'ADC

Les timers

Le timer 0  
Le pré-diviseur de T<sub>0</sub> et T<sub>1</sub>

Le timer 1  
Les ITs du timer 1

Les modes de la MLI

Programmation

Langage C  
Fonctions

Fonction de mise en oeuvre d'une interruption

## Structure d'un programme

Un programme C est composé de trois parties. Les entêtes, les fonctions nécessaires et le programme principal (main).

```
// ----- ENTETE
-----
#include "mesfonctions.c"
#define F_CPU 12E6
#define Max_del 262
// ----- FONCTIONS
-----
void sleep(int dizaines){
    int i=0;
    for(i=0;i<10*dizaines;i++) _delay_ms(10);
}
// ----- MAIN
-----
int main(void)
{
    DDRD = 0b1111011;
    do {
        PORTD ^=0x0F;
        sleep(10);
    } while(1);
}
```

Programmation

Langage C

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer** 0  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer** 1  
Les ITs du **timer** 1  
Les modes de la **MLI**

Programmation

Langage C

Fonctions  
Fonction de **117 / 122** bits  
 Mise en oeuvre d'une interruption

## Entête

La partie entête définit les bibliothèques nécessaires, des constantes non modifiables ou des fichiers à inclure.

```
#define F_CPU 12E6 // Constantes
#define Max_del 262
#include "mesfonctions.c" //
    Inclusions de fonctions
#include "interface.h" //
    bibliotheque utilisateur
```

Programmation

Langage C

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer** 0  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer** 1  
Les ITs du **timer** 1  
Les modes de la **MLI**

Programmation

Langage C

Fonctions  
Fonction de **118 / 122** bits  
 Mise en oeuvre d'une interruption

# Main

Un main commence par des initialisations et se termine généralement par une boucle infinie :

```
int main(void)
{
    DDRC = 0x00;    // port C en
                  entree
    DDRB = 0x01;    // PBO en sortie
    DDRD = 0b1111011; // PD2 en
                  entree
    do {
        sleep(10);
    } while(1);
    return(1);
}
```

Programmation

Langage C

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les *PORTS*

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer** 0  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer** 1  
Les ITs du **timer** 1  
Les modes de la **MLI**

Programmation

Langage C

Fonctions

Fonction de **119** / **122** de bits  
Mise en oeuvre d'une interruption

# Fonctions

```
int carre(int d){return d*d;}

void sleep(int dzs){
    int i=0;
    int delai=carre(10);
    for(i=0;i<10*dzs;i++) _delay_ms(delai);
}

void entier(int nbtours, int sens){
    int i,j;
    int Tentier[4]={1,2,4,8};
    for(j=0;j<nbtours*12;j++) {
        for(i=3;i>=0;i--){

            sleep(2);
            PORTB=Tentier[i];
        }
    }
}
```

Programmation

Fonctions

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les *PORTS*

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer** 0  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer** 1  
Les ITs du **timer** 1  
Les modes de la **MLI**

Programmation

Langage C

Fonctions

Fonction de **120** / **122** de bits  
Mise en oeuvre d'une interruption

# Manipulation de bits

```
// Mise a zero des 4 bits de poids faible
PORTB=PORTB & 0xF0; // ou
PORTB &= 0xF0;

// Mise a zero du bit numero 3
PORTB=~(1<<PB3);

// Mise a un des 4 bits de poids faible
PORTB = PORTB | 0x0F // ou
PORTB |= PORTB;

//Mise a un de la position 4
PORTD |= (1<<PD4);

// Commutation du bit numero 4
PORTD = PORTD^0x10;
```

Programmation

Fonction de manipulation de bits

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer 0**  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer 1**  
Les ITs du **timer 1**  
Les modes de la **MLI**

Programmation

Langage C  
Fonctions

Fonction de manipulation de bits  
121 / 122  
Mise en oeuvre d'une interruption

# Interruptions

```
ISR(TIMERO_OVF_vect){ // routine d'
    interruption
    PORTD = PORTD^0x10; // Inv. du bit 4
    TCNT0=100;
}
int main(void)
{
    DDRD = 0b1111011;
    TCNT0= 0; // valeur initiale du
    compteur
    TCCR0 = 5; // prediviseur / 1024
    TIMSK |= (1<<TOIE0); // debord. Tim 0
    sei(); // Toutes ITs autorisees
    do {
        ...
    } while(1);
    return(1);
}
```

Programmation

Mise en oeuvre d'une interruption

Info Indus 8 bits

Delfieu

Présentation générale

Introduction  
Architecture  
Mémoire  
Registres Systèmes

Les interruptions externes

Les **PORTS**

Le traitement des valeurs Analogiques

Le Comparateur Analogique  
Convertisseur Analogique-Numérique : **ADC**  
Les Registres de l'ADC

Les **timers**

Le **timer 0**  
Le pré-diviseur de  $T_0$  et  $T_1$   
Le **timer 1**  
Les ITs du **timer 1**  
Les modes de la **MLI**

Programmation

Langage C  
Fonctions

Fonction de manipulation de bits  
121 / 122  
Mise en oeuvre d'une interruption