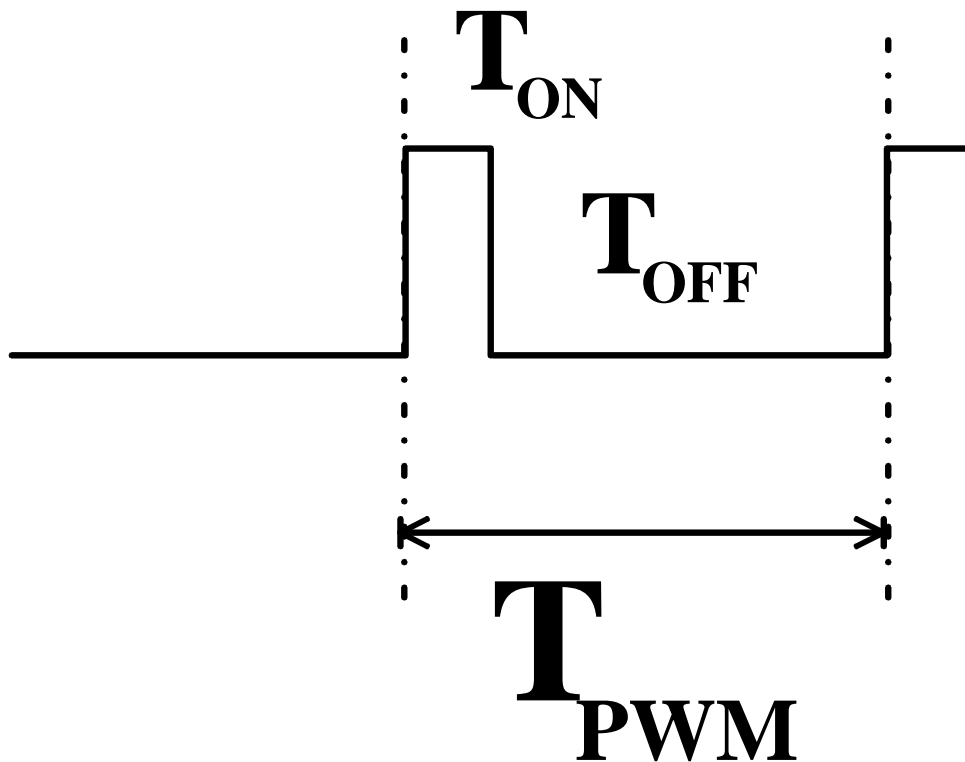


Le signal PWM ou MLI



La période est fixe

$$T_{PWM} = \text{constante}$$

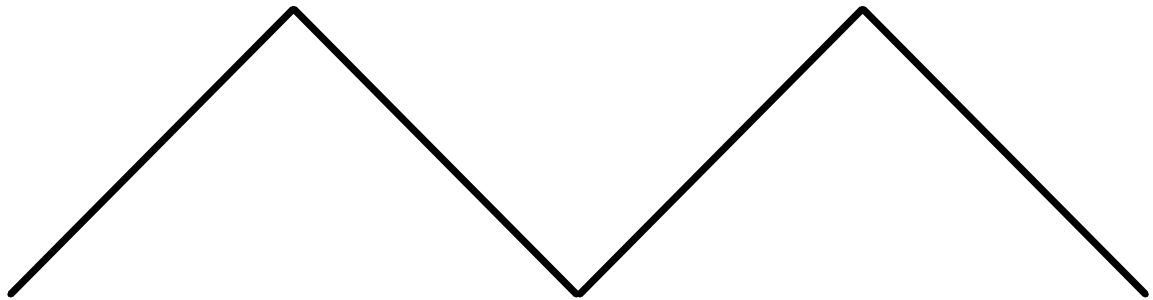
Le rapport cyclique est variable

$$n = \frac{T_{ON}}{T_{PWM}}$$

Microcontrôleur ATMEL

Principe de génération

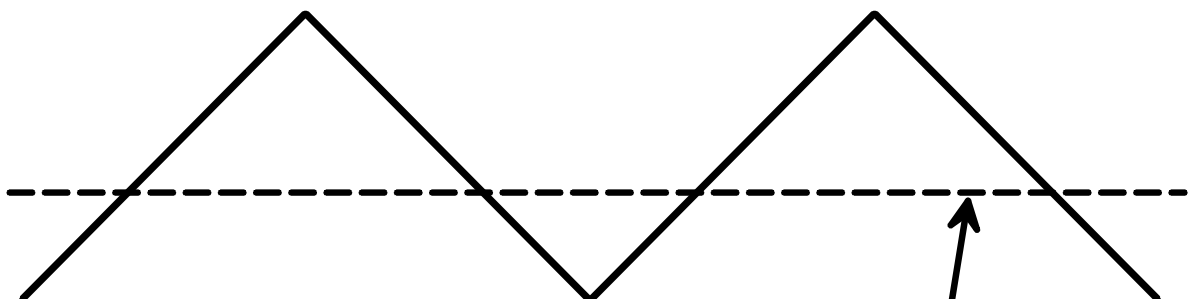
TOP



\$0000

Un compteur interne au μ C balaye les valeurs depuis \$0000 jusqu'à une valeur maximale notée TOP

TOP



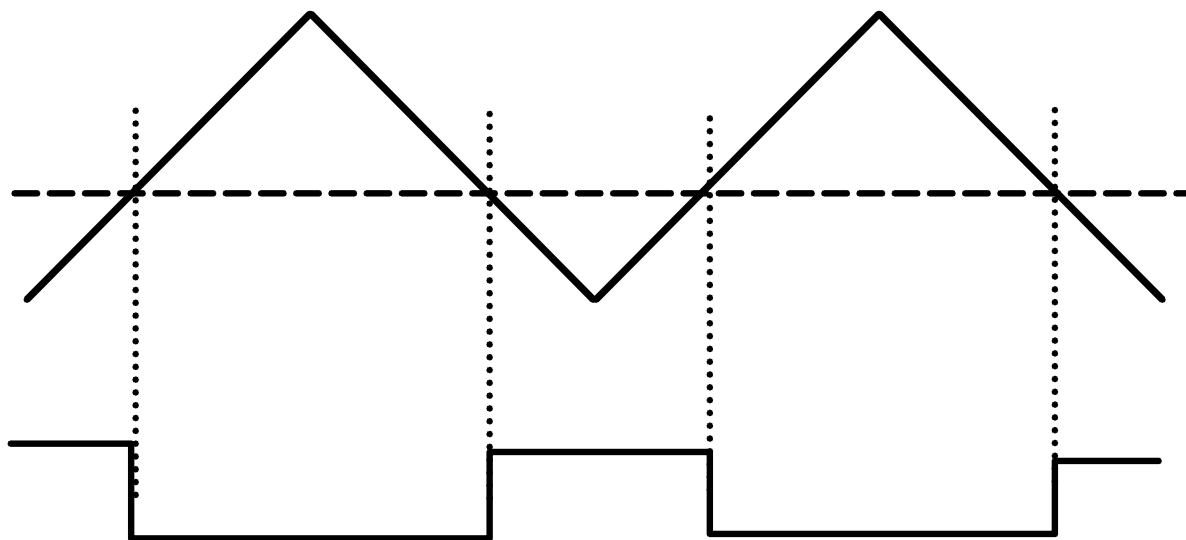
\$0000

Cette valeur sera ensuite comparée à la valeur de comparaison comprise entre \$0000 et TOP

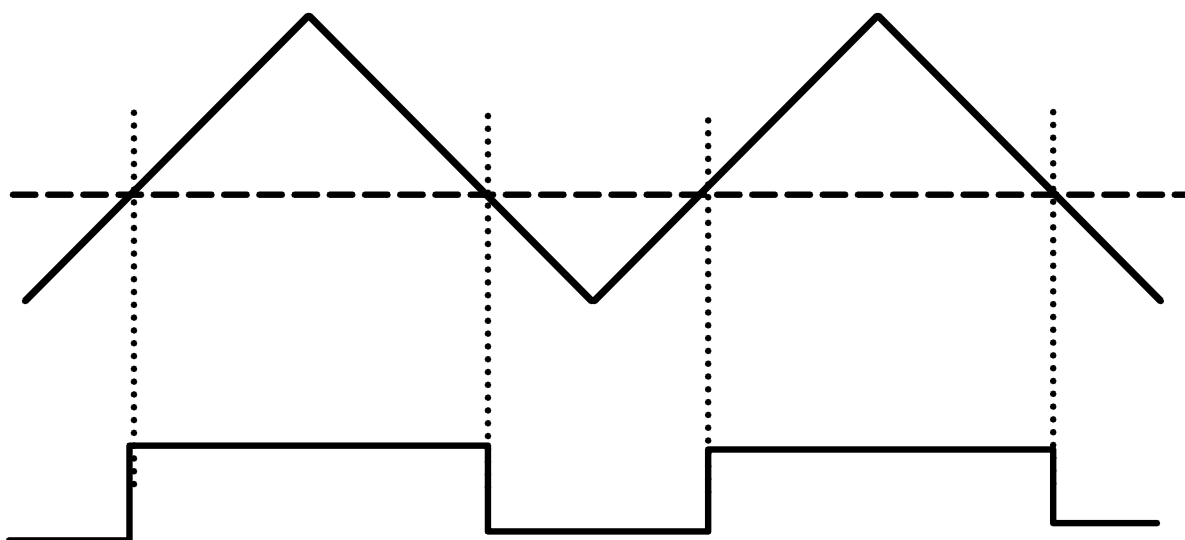
Microcontrôleur ATMEL

Choix de deux modes PWM :

Mode PWM



Mode PWM inversé



Microcontrôleur ATMEL

Comment déterminer F_{PWM}

Résolution : choix de la valeur TOP:

PWM Résolution	Timer TOP Value	Fréquence F_{PWM}
8-bit	\$00FF (255)	$F_{ct1}/510$
9-bit	\$01FF (511)	$F_{ct1}/1022$
10-bit	\$03FF (1023)	$F_{ct1}/2046$

Choix du prédiviseur :

La fréquence du timer1 est obtenue par division de la fréquence du uC notée F_{ck} par une valeur N

$$F_{CT1} = F_{CK} / N$$

$$N = 1, 8, 64, 256, 1024$$

$$F_{CK} = 4 \text{ MHz pour les kits STK200}$$

Microcontrôleur ATMEL

Bilan pour le 2313 :

Registre de comptage : Timer1

Registre de configuration :

TCCR1A Choix du mode PWM
 Choix de la résolution

TCCR1B Choix du facteur de prédivision

Registre de comparaison :

OCR1AL (Résolution 8 bit)

OCR1AH,OCR1AL (Résolution 9 et 10 bit)

Cette valeur est ajustée en fonction du rapport cyclique désiré

Signal PWM sur le PortB,3

Microcontrôleur ATMEL

Configuration des registres

TCCR1A

7

0



1 0 mode PWM

1 1 mode PWM inversé

PWM sur 8 bit 0 1

PWM sur 9 bit 1 0

PWM sur 10 bit 1 1

Microcontrôleur ATMEL

Configuration des registres

TCCR1B

7

0

0	0	-	-	0	⋮	⋮	⋮
---	---	---	---	---	---	---	---

Choix du facteur de
prédivision N

N = 1	0	0	1
N = 8	0	1	0
N = 64	0	1	1
N = 256	1	0	0
N = 1024	1	0	1

Microcontrôleur ATMEL

Détermination de N :

$$N_{8 \text{ bit}} = \frac{F_{CK}}{F_{PWM} \cdot 510}$$

$$N_{9 \text{ bit}} = \frac{F_{CK}}{F_{PWM} \cdot 1022}$$

$$N_{10 \text{ bit}} = \frac{F_{CK}}{F_{PWM} \cdot 2046}$$

**Il faut choisir N= 1,8,64,256,1024
et la résolution correspondante**

Microcontrôleur ATMEL

Exemple de configuration :

$$F_{\text{PWM}} = 20 \text{ KHz}$$

$$F_{\text{CK}} = 10 \text{ MHz}$$

Mode PWM

Valeur de N :

Résolution :

Microcontrôleur ATMEL

Fréquences F_{PWM} possibles :

	Fck	4 000 000 Hz				
	N	1	8	64	256	1 024
Résolution	Fct1	4 000 000	500 000	62 500	15 625	3 906
8 bits	Fpwm	7 843	980	123	31	8
9 bits	Fpwm	3 914	489	61	15	4
10 bits	Fpwm	1 955	244	31	8	2

F_{PWM} en Hz avec $F_{CK} = 4M_{HZ}$

Compléter :

	Fck	10 000 000 Hz				
	N	1	8	64	256	1 024
Résolution	Fct1					
8 bits	Fpwm					
9 bits	Fpwm					
10 bits	Fpwm					

F_{PWM} en Hz avec $F_{CK} = 10M_{HZ}$

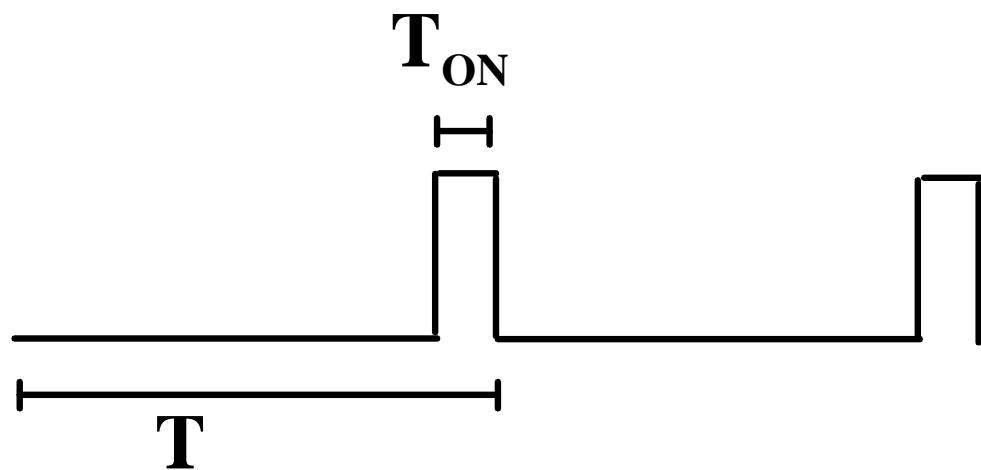
Microcontrôleur ATMEL

Commande d'un servomoteur :



Caractéristiques:

- Excellent rapport qualité/prix
- Couple maxi.: 3 Kg/cm
- Vitesse: 0,22 s / 60°
- Alimentation: 4,8 à 6 Vcc
- Pignons nylon, paliers bronze
- Poids: 49 g
- Dimensions: 40 x 20 x 36 mm



Signal de commande

Période = 20 mS (Radio commande)

$T_{ON} = 1,5$ mS position neutre

Course utile

1 mS $\leq T_{ON} \leq 2$ mS

Microcontrôleur ATMEL

Choix de la fréquence de travail :

Résolution = 10 Bits

N = 64

F_{CK} = 4 MHz

Mode PWM

Donner :

TCCR1A

--	--	--	--	--	--	--	--

TCCR1B

--	--	--	--	--	--	--	--

Compléter :

T = 31 mS

T_{ON} = 1 mS Valeur PWM = 0x1F

T_{ON} = 1,5 mS Valeur PWM = 0x--

T_{ON} = 2 mS Valeur PWM = 0x--

Microcontrôleur ATMEL

Mise en oeuvre du servomoteur

```
#include <90s2313.h>
```

```
#include <delay.h> On inclus la librairie delay
```

```
// Declare your global variables here
```

```
void main(void)
```

```
{
```

```
// Declare your local variables here
```

```
unsigned char pwm = 0x1F;
```

```
unsigned char retard = 1000;
```

*déclarations locales de
pwm et retard*

```
// Input/Output Ports initialization
```

```
// Port B initialization
```

```
PORTB=0x00;
```

```
DDRB=0x08;
```

*Initialisation des ports
(code wizard)*

```
// Port D initialization
```

```
PORTD=0x00;
```

```
DDRD=0x00;
```

```
// Timer/Counter 1 initialization
```

```
// Clock source: System Clock
```

```
// Clock value: 62,500 kHz
```

```
// Mode: Ph. correct PWM top=03FFh
```

```
// OC1 output: Non-Inv.
```

```
// Noise Canceler: Off
```

```
// Input Capture on Falling Edge
```

```
TCCR1A=0x83;
```

```
TCCR1B=0x03;
```

```
TCNT1H=0x00;
```

```
TCNT1L=0x00;
```

```
OCR1H=0x00;
```

```
OCR1L=0x00;
```

*Initialisation du timer 1
PWL (code wizard)*



Microcontrôleur ATMEL

Programme principal

```
while (1) Boucle principale (perpétuelle)
{
    // Place your code here

    while ( pwm < 0x3F )
    {
        pwm = pwm + 1;
        OCR1AL = pwm;
        OCR1AH = 0;
        delay_ms(retard);
    }
    while ( pwm > 0x1F )
    {
        pwm = pwm - 1;
        OCR1AL = pwm;
        OCR1AH = 0;
        delay_ms(retard);
    }
};
```

Que fait le programme précédent ?



Microcontrôleur ATMEL

Exemple de programmation :

INIT:

```
ldi R16,$C2  
out TCCR1A
```

Initialisation
faite une fois
pour toute

```
ldi R16,$01  
out TCCR1B
```

quel est le mode
et la résolution
initialisé ici ?

MAIN:

```
ldi R16,$00  
out OCR1AH
```

Envoi de la valeur
du rapport cyclique
à chaque fois que
nécessaire

```
ldi R16,$42  
out OCR1AL
```