

## Sommaire

- 1 Objectifs du projet
- 2 Domaines d'applications
- 3 Caractéristiques du PIC16F877A
- 4 Fonctions PWM
  - 4.1 PWMx\_Init()
  - 4.2 PWMx\_Set\_Duty()
  - 4.3 PWMx\_Start()
  - 4.4 PWMx\_Stop()
- 5 Analyse de Fonctionnement
  - 5.1 Régimes de fonctionnement
  - 5.2 Exemple de calcul
  - 5.3 Résultat de simulation
  - 5.4 PWM 25%
  - 5.5 PWM 50%



- 5.6 PWM 75%
- 6 Programme MikroC

## Objectifs du projet

- Savoir les caractéristiques du PIC16F877A
- Savoir utiliser les fonctions de la gestion de générateur PWM
- Savoir ajuster le rapport cyclique d'un signal PWM et la fréquence
- etc.

## Domaines d'applications

- Variateurs de vitesse d'un moteur à CC
- Contrôle de puissance d'une charge
- Alimentations de puissance ajustables (DC/DC converter)
- Régulateurs
- etc.

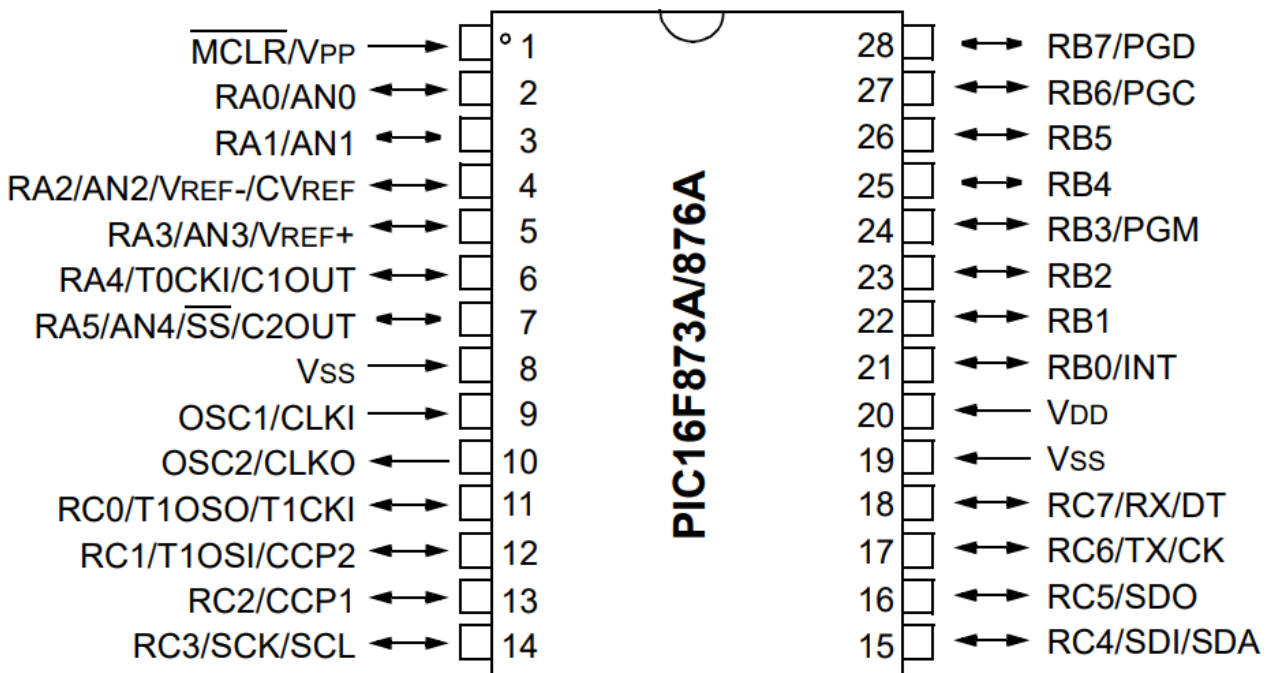
## Caractéristiques du PIC16F877A

- Fréquence : DC-20 MHz
- Resets: POR, BOR, PWRT, OST
- Mémoire flash programme: 8k
- Mémoire de données: 368
- EEPROM: 256
- Interruptions: 15
- Ports E/S: A, B, C, D et E
- Timers: 3
- Capture/Comparateur/PWM: 2
- Communication série: MSSP (SPI, I<sup>2</sup>C), USART



- Communication //: PSP
- Convertisseur A/N: 8 canneaux 10 bits
- Compérateurs analogiques: 2
- Instructions: 35
- [Datasheet PIC16F877A](#)

Device	Program Memory		Data SRAM (Bytes)	EEPROM (Bytes)	I/O	10-bit A/D (ch)	CCP (PWM)	MSSP		USART	Timers 8/16-bit	Comparators
	Bytes	# Single Word Instructions						SPI	Master I <sup>2</sup> C			
PIC16F873A	7.2K	4096	192	128	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F874A	7.2K	4096	192	128	33	8	2	Yes	Yes	Yes	2/1	2
PIC16F876A	14.3K	8192	368	256	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F877A	14.3K	8192	368	256	33	8	2	Yes	Yes	Yes	2/1	2



## Fonctions PWM



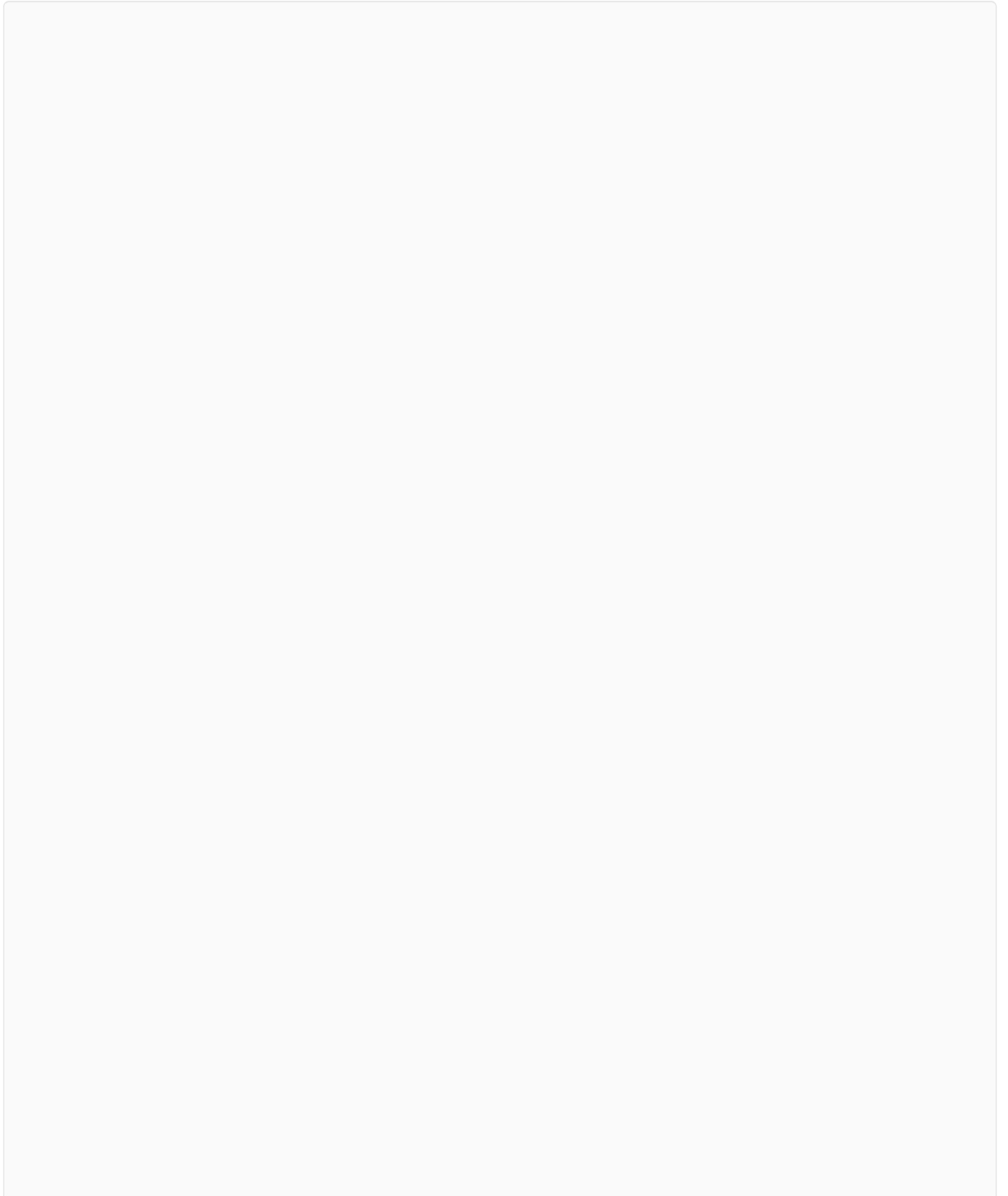
## PWMx\_Init()

La fonction PWMx\_Init(long freq) permet de configurer le contrôleur PWM numéro x (x=1, 2, ...) du [microcontrôleur](#) avec la fréquence égale à « freq » en Hz. Exemple : Initialiser le PWM 1 à 1KHz PWM1\_Init(1000). Par défaut la PWM est connecté avec le pin RC2 et PWM2 avec le pin RC2. Il faut bien s'assurer que le microcontrôleur contient des interfaces PWM, dans le cas du PIC16F877A on dispose de deux sorties PWM (voir les caractéristiques ci-dessus). Réfère au datasheet pour le choix de la fréquence. Lorsque la valeur choisie ne respecte pas celle du microcontrôleur, la fonction retourne une erreur. La fonction retourne aussi une erreur lorsque le µC ne dispose pas du contrôleur PWM.

## PWMx\_Set\_Duty()

```
Syntaxe : void PWMx_Set_Duty(unsigned short duty_ratio);
```

Elle permet de définir le rapport cyclique du signal PWM. La valeur « duty\_ratio » peut varier entre 0(0%) et 255 (100%). Lorsque on fait appel à la fonction d'initialisation PWMx\_Init(long freq), le rapport cyclique par défaut est de 0%. Donc, il faut obligatoirement suivre la fonction PWMx\_Set\_Duty(unsigned short duty\_ratio). Exemple :





```
// Initialisation du PWM à 1KHz  
PWM1_Init(1000);  
// Mise à jour du rapport cyclique à 50%  
PWM1_Set_Duty(127);
```





## PWMx\_Start()

```
Syntaxe : void PWMx_Start(void);
```

Après l'étape d'initialisation du contrôleur, il suit l'étape de mise en route! C'est grâce à la fonction PWMx\_Start(void). Il suffit de préciser le numéro du PWM et la fonction s'occupe du reste. D'une autre façon la sortie RC1 ou RC2 reste à 0 (absence du signal PWM) tant que la fonction de mise en route n'est pas activée. Ci-dessous la fonction d'arrêt du contrôleur.

## PWMx\_Stop()

```
Syntaxe : void PWMx_Stop(void);
```

Désactiver la sortie PWM (mise à zéro du pin RCx)

## Analyse de Fonctionnement

On considère le schéma ci-dessous du contrôleur PWM commandé par une tension. Le fonctionnement consiste à varier le rapport cyclique du signal PWM1 en utilisant une source de tension analogique. On utilise un [potentiomètre](#) monté en pont diviseur afin d'ajuster la tension à la borne du pin PA2 entre 0V (0%) et 5V (100%). Le potentiomètre est branché avec l'entrée analogique AN2 ou RA2.

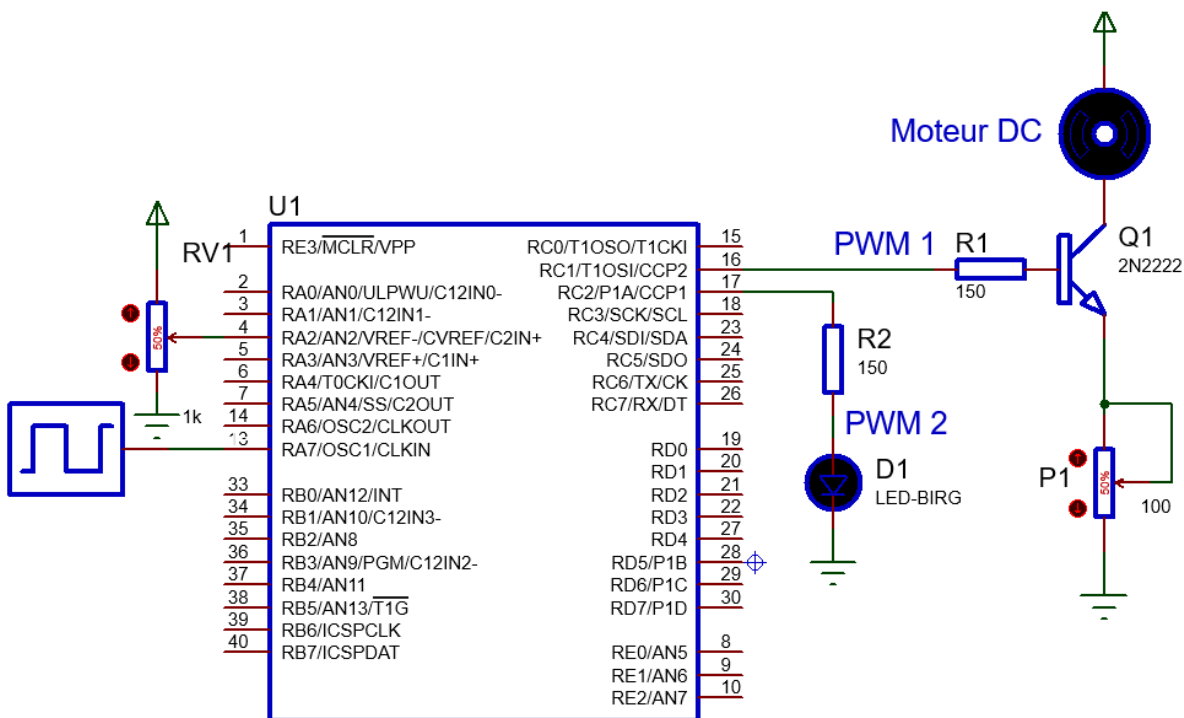
Note : La valeur 100% peut être inférieure à 5V si l'alimentation est inférieure à 5V (Exemple 2V ou 3.3V). De préférence il faut alimenter le potentiomètre RV1 avec la tension de l'alimentation du  $\mu\text{C}$ .

Dans notre exemple on utilise deux sorties PWM (PWM1 liée avec RC1 et PWM2 liée avec RC2) de mêmes fréquences et rapports cycliques. La sortie PWM1 est ensuite sert à contrôler

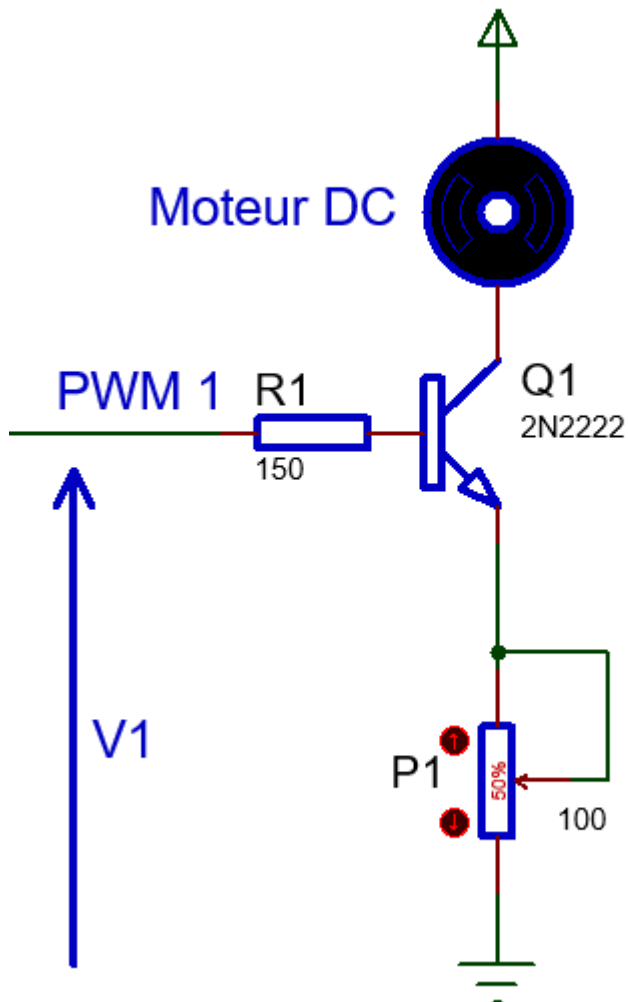




un moteur à courant continu via un **transistor** bipolaire 2N2222. La sortie PWM2 est branchée avec une **LED** à travers une **résistance** de 150 ohms pour rôle la limitation du courant à la borne de la LED. Nous avons utilisé une LED afin de visualiser par l'œil l'influence de la variation du potentiomètre RV1 (ou du rapport cyclique) d'une part sur la luminosité de la LED, d'autre part sur la vitesse de rotation du moteur DC.



## Régimes de fonctionnement



- $PWM1=0V \Rightarrow V1=0$ , le courant de base est nul  $I_B=0 \Rightarrow I_C=0 \Rightarrow$  Le courant à la borne du moteur DC est nul  $\Rightarrow$  Moteur en état d'arrêt
- $PWM1=VCC \Rightarrow V1=VCC=5V \Rightarrow$  courant  $I_B$  non nul, ci-dessous la formule du courant  $I_C$  en fonction de  $V1$ ,  $P1$  et  $R1$ :



$$V1 = V_{be} + P1 * I_C + R1 * I_B = V_{be} + P1 * I_C + R1 * \frac{I_C}{\beta} =$$

$$V1 = V_{be} + I_C \left( P1 + \frac{R1}{\beta} \right) \Rightarrow I_C = \frac{V1 - V_{be}}{P1 + \frac{R1}{\beta}}$$

## Exemple de calcul

A.N: Lorsque  $P1=100$ ,  $V1=5V$ ,  $V_{be}=0.7V$ ,  $R1=150$ ,  $\beta=300 \Rightarrow I_C = (5-0.7)/(100+0.5) = 4.3/100.5 = 42.7mA$

La tension  $V_{be}$  est de l'ordre de  $0.6-0.7V$ , elle est faible par rapport à  $V1$ . De plus, le rapport  $R1/\beta$  est faible ( $\beta$  de l'ordre de  $100-300$ ), alors:

$$I_C \approx \frac{V1}{P1}$$

A.N: Lorsque  $P1=100$ ,  $V1=5V \Rightarrow I_C = 5/100 = 50mA$ .

Note: La formule montre que le rôle de la résistance de la base  $R1$  est négligeable et n'a pas d'effet sur le courant  $I_C$ . Ainsi qu'on peut contrôler le courant à la borne de la charge ( $I_C$ ) avec le potentiomètre  $P1$ . D'où l'utilisation d'un potentiomètre à la place d'une résistance fixe !



## Résultat de simulation

Afin de simuler le courant  $I_C$ , on branche une source de tension DC de 5V à la base, on alimente le transistor avec une tension VCC égale à 12V. On branche deux sondes de courants: Une pour le courant  $I_B$  l'autre pour le courant  $I_C$ . Les résultats sont les suivants:

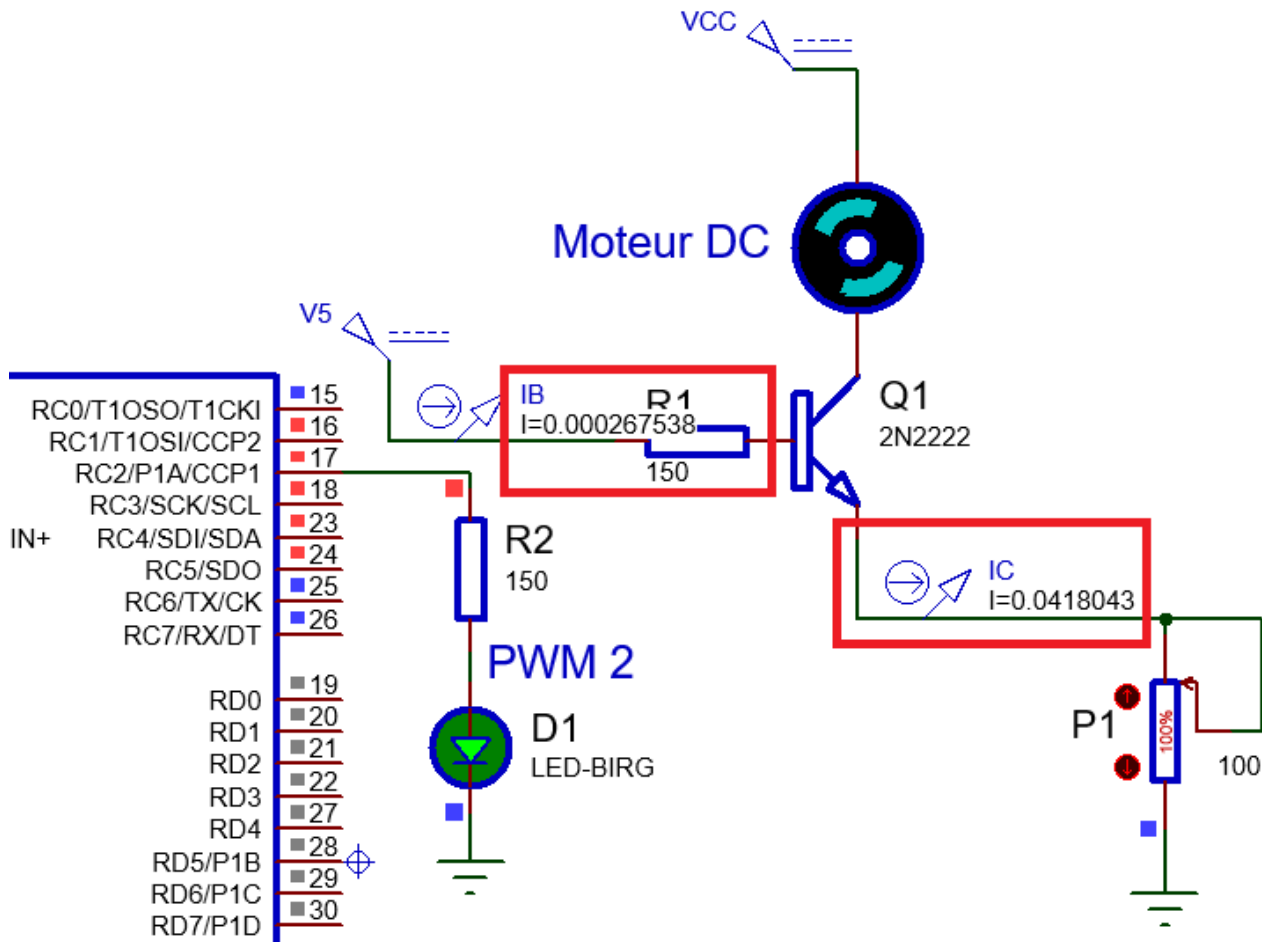
- $I_B=0.26 \mu A$
- $I_C=41.8 \text{ mA}$

En résumé, le résultat théorique 42.7 mA est plus proche de simulation 41.8 mA. On en déduit que notre calcul est bon ainsi que le modèle de simulation utilisé par le [logiciel ISIS](#).

Note: Le  $\mu C$  n'a pas d'importance car la sortie PWM 1 est débranchée.



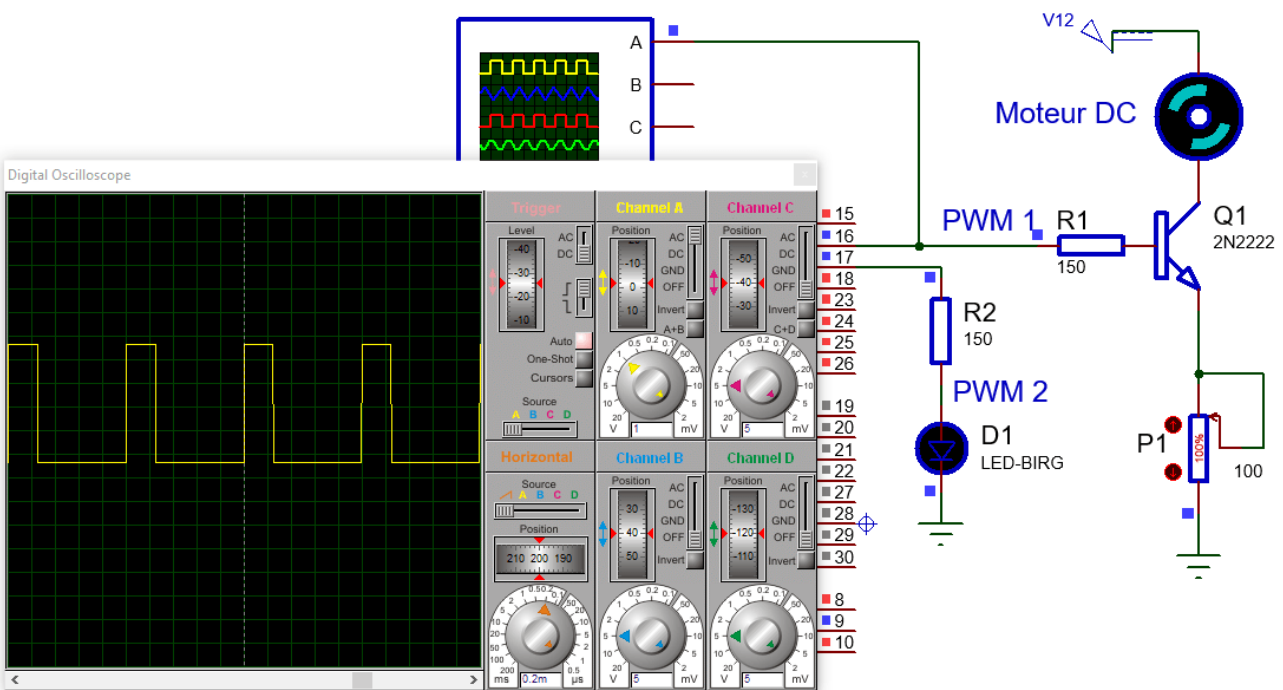
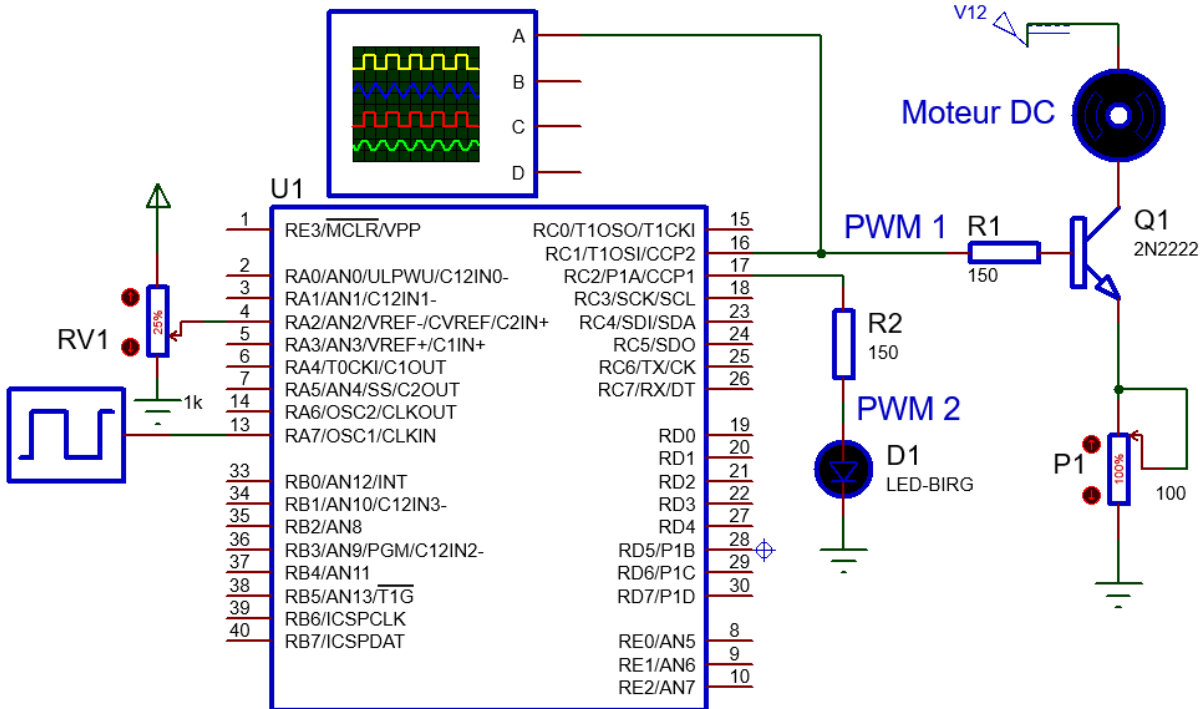
# Contrôleur PWM commandé en tension



PWM 25%



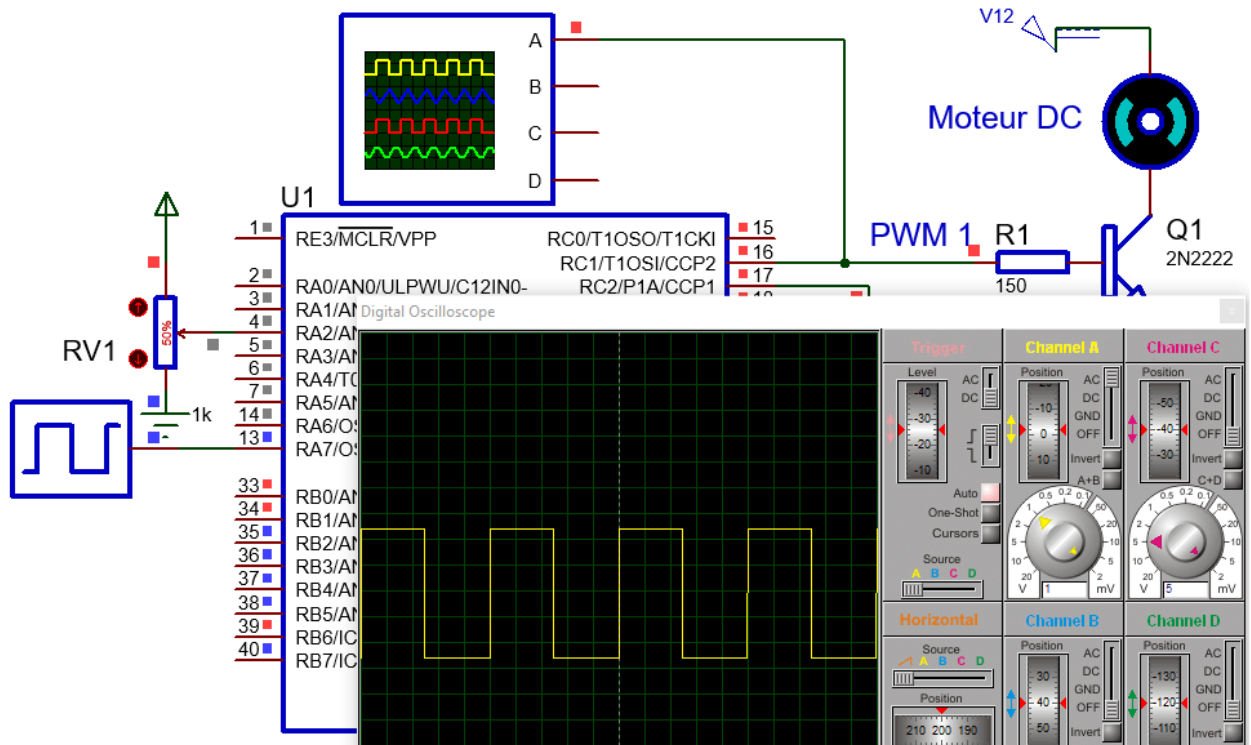
# Contrôleur PWM commandé en tension





# Contrôleur PWM commandé en tension

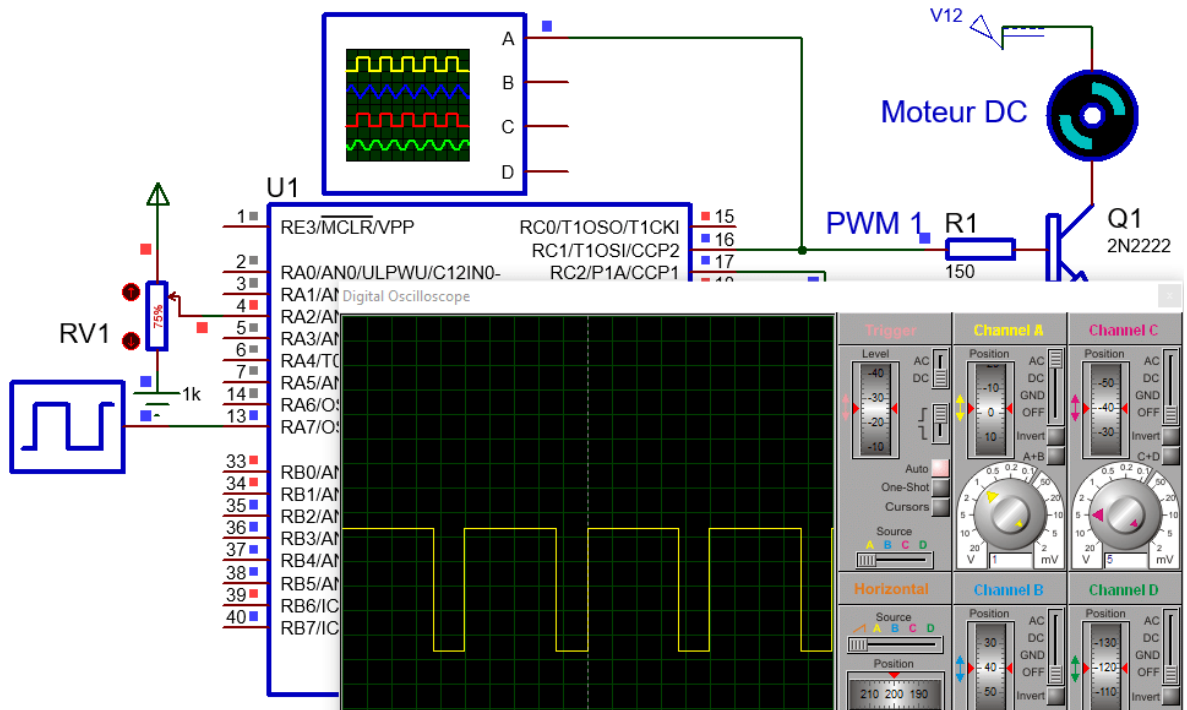
PWM 50%



PWM 75%



## Contrôleur PWM commandé en tension

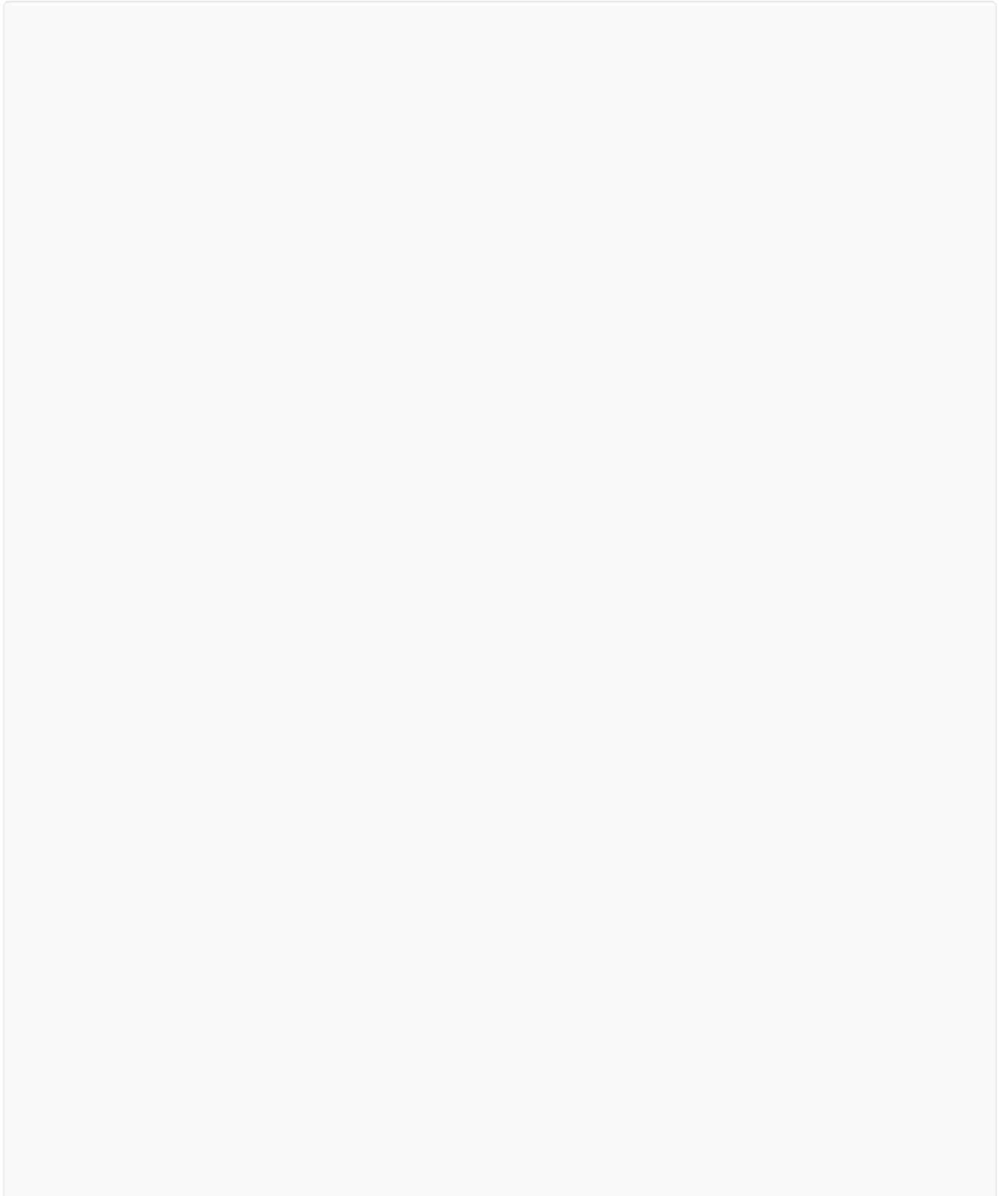


## Programme MikroC





## Contrôleur PWM commandé en tension





```
#include <built_in.h>

unsigned int adc_val;

void main() {

    // Initialisation ADC
    ADC_Init();

    // Initialisation PWM
    PWM1_Init(1000);
    PWM2_Init(1000);

    // Mise à jour des rapports cycliques
    PWM1_Set_Duty(128);
    PWM2_Set_Duty(128);

    // Start PWM
    PWM1_Start();
    PWM2_Start();

    // Port A en entrée
    TRISA=0xFF;

    // Configuration en sortie des ports
    TRISB=0;
    TRISC=0;
    TRISE=0;

    // Lecture et affichage de la donnée ADC
    while(1)
    {
```



```
// Lecture entrée analogique
adc_val=ADC_Read(2);
adc_val=adc_val>>2;

// Mise à jour des rapports cycliques
PWM1_Set_Duty(adc_val);
PWM2_Set_Duty(adc_val);
}
```



```
}
```



## Contrôleur PWM commandé en tension

- [Programme MikroC](#)
- [Simulation ISIS](#)

[Retour à l'accueil MikroC](#)

[Total : 0 Moyenne : 0/5]