

# TP n°2 : Programmation d'un $\mu\text{C}$ 16F88

## Généralités

Le PIC16F88 est un microcontrôleur de chez Microchip. Un  $\mu\text{C}$  se différencie d'un  $\mu\text{P}$  par ses périphériques qu'un  $\mu\text{P}$  n'a pas. Parmi ceux là, on trouve une mémoire flash, une RAM, des Compteurs, un CAN, des fonctions de dialogue RS232, un générateur MLI, des interruptions...

Dans ce 2<sup>ème</sup> TP du projet, vous allez découvrir la chaîne de développement depuis l'écriture du programme jusqu'au chargement du programme dans le  $\mu\text{C}$ .

## Documents

Data Sheet 16F88, Carte à  $\mu\text{C}$  16F88, Code des couleurs, résistances série E12

## Caractéristiques principales du $\mu\text{C}$ 16F88

Tension d'alimentation	2 à 5,5V
Courant maximal en entrée	+/- 1 $\mu\text{A}$
Courant maximal en sortie	20mA/-25mA
Horloge maximale	20MHz

## Préparation

Donner les schémas de branchement d'une Del et d'un bouton poussoir associés à un  $\mu\text{C}$ . Calculer les valeurs des composants associés.

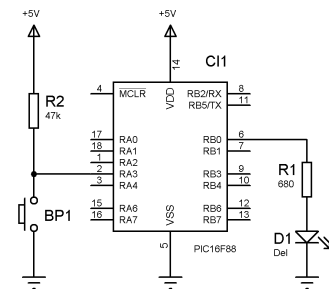
### Calcul de la résistance de polarisation de la Del rouge

**Correction**

Une Del rouge s'éclaire à partir d'un courant de 5mA, sous une tension de 1,6V. Nous en déduisons la valeur de la résistance associée par le calcul suivant :

$$R_1 = \frac{V_{CC} - V_D}{I_D} = \frac{5 - 1,6}{0,005} = 680\Omega$$

Cette valeur est normalisée.



### Calcul de la résistance associée au bouton poussoir

**Correction**

Cette résistance est appelée résistance de tirage bas (pull down). Si le courant maximal en entrée est de 1 $\mu\text{A}$  et la tension pour un niveau logique bas est de 0,15Vcc :

$$R_2 = \frac{V_{CC} - V_{Ilow}}{I_E} = \frac{5 - 0,75}{0,000001} = 4,250M\Omega$$

Ce résultat correspond à la valeur maximale. Mais il est plus sage de diminuer la valeur de cette résistance d'un facteur 10 afin que la tension à ses bornes soit 10 fois plus petite et être sûr du niveau logique bas appliqué. D'autre part, augmenter le courant lorsque le bouton poussoir est appuyé, permet de réduire les parasites et d'améliorer son fonctionnement. La résistance normalisée utilisée en général pour cette fonction est comprise entre 10k $\Omega$  et 100k $\Omega$ . Nous choisirons 47k $\Omega$ .

## Manipulation

### Matériels :

- PC avec logiciels MPLAB + Compilateur C + TinyBootloader + Fichier Exo\_1.c
- Câble Série / TTL ou USB / TTL
- Alimentation régulée 5Volts
- Multimètre
- Plaquette d'essai
- Carte à  $\mu\text{C}$  16F88
- Résistances 680 $\Omega$  et 47k $\Omega$
- Del rouge
- Bouton Poussoir pour CI
- Divers fils de liaison

### Câblage :

Brancher la carte à  $\mu\text{C}$  16F88 sur la plaque d'essai. Ajouter les autres éléments et compléter avec des petits fils. (Câblage d'après Fritzing). Le bouton poussoir sera relié à la ligne RB0 et la Del à la ligne RB7.

### Programmation du $\mu\text{C}$

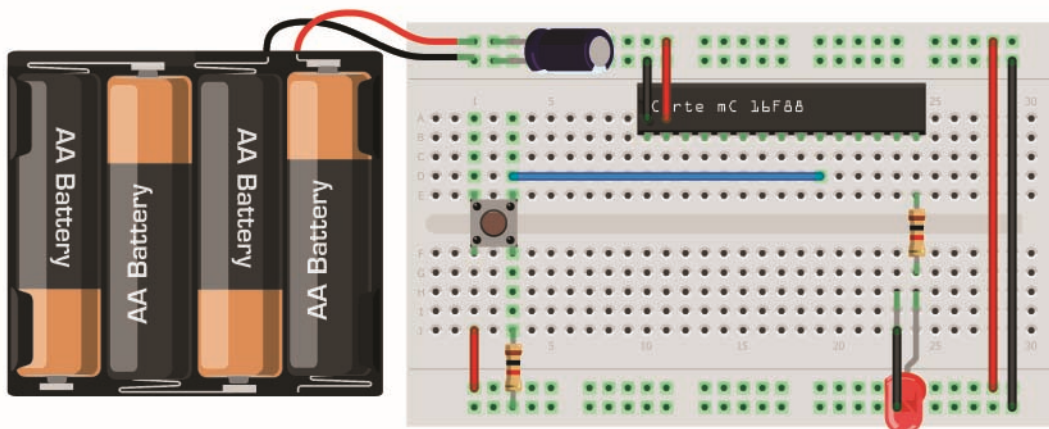
Récupérer un fichier en C contenant la trame de base à compléter ou le fichier corrigé. Placer ce fichier dans un répertoire.

### Compilation du fichier

Lancer MPLAB. Créer un nouveau projet avec Project / Project Wizard... Choisir le composant : PIC16F88. Sélectionner ou valider le compilateur C : CCS C Compiler for PIC... Chercher le répertoire qui contient le fichier précédent et donner un nom au projet. Sélectionner le fichier dans la fenêtre de gauche et cliquer sur ADD pour l'ajouter dans la fenêtre de droite. Valider l'ensemble et Compiler le projet avec l'icône Build All.

### Transfert du programme dans la mémoire du $\mu\text{C}$

Brancher le câble de programmation entre le PC et la carte  $\mu\text{C}$ . Attention au sens de branchement (fil noir du connecteur à la masse de la carte). Lancer TinyBootloader. Sélectionner le fichier avec l'extension .hex. Régler la vitesse de transfert à 19200. Sélectionner un port COM. Et lancer la programmation avec Write Flash.



Vérifier le bon fonctionnement et modifier le programme pour obtenir un flash toute les secondes.