

# Langage du microtruck

---

## Idée générale

Permettre de coder sur papier sans l'assistance d'un ordinateur.

Pour cela, on imagine recréer un langage type assembleur qui pourra être réécrit sur papier à l'aide de case grisées d'un tableau .

Ce tableau fera donc 12 cases de large.

## Instructions

### EDX

**Code :** 0 (0000)

**Datas :** 8b

**Datatype :** Brut

**Description :** Charge dans l'EDX l'octet spécifié.

### MOV

**Code :** 1 (0001)

**Datas :** 4b+4b

**Datatype :** addr + addr

**Description :** Copie dans la variable de la deuxième adresse la valeur de la variable à la première adresse.

### JMP

**Code :** 2 (0010)

**Datas :** 8b

**Datatype :** Numéro d'instruction

**Description :** Se déplace au numéro d'instruction donné.

### JMC

**Code :** 3 (0011)

**Datas :** 8b

**Datatype :** Numéro d'instruction

**Description :** Se déplace au numéro d'instruction donné si EDX contient 1 (0000 0001), sinon, continue.

### AND

**Code :** 4 (0100)

**Datas :** 4b+4b

**Datatype :** addr + addr

**Description :** Charge dans EDX le AND des deux variables.

### OR

**Code :** 5 (0101)

**Datas :** 4b+4b

**Datatype :** addr + addr

**Description :** Charge dans EDX le OR des deux variables.

ADD

**Code :** 6 (0110)

**Datas :** 4b+4b

**Datatype :** addr + addr

**Description :** Charge dans EDX l'addition des deux variables. Si il y a une retenue, elle sera chargée dans le registre dédié.

ADC

**Code :** 7 (0111)

**Datas :** 4b+4b

**Datatype :** addr + addr

**Description :** Charge dans EDX l'addition des deux variables et de la retenue. Si il y a une retenue, elle sera chargée dans le registre dédié.

SUB

**Code :** 8 (1000)

**Datas :** 4b+4b

**Datatype :** addr + addr

**Description :** Charge dans EDX la différence des deux variables.

**Erreur :** Si la différence est négative, une erreur est levée, le programme est stoppé et l'affichage indique le numéro d'instruction actuel.

INC

**Code :** 9 (1001)

**Datas :** 4b+4b

**Datatype :** addr + brut

**Description :** Ajoute à la variable le nombre donné s'il est inférieur ou égal à 127 (0111 111), ou soustrait l'inverse binaire du nombre + 1. Les même règles que ADD et SUB s'appliquent.

SHF

**Code :** 10 (1010)

**Datas :** 4b+4b

**Datatype :** brut + addr

**Description :** Décale vers la gauche la variable du nombre donné si ce nombre est inférieur ou égal à 127 (0111 1111) (la retenue contient alors le bit immédiatement plus fort que le bit de poids fort après l'opération), ou décale vers la droite de l'inverse binaire du nombre + 1.

DEQ

**Code :** 11 (1011)

**Datas :** 2+2+4

**Datatype :** code + addr + addr

- **ISE**

- **Code :** 0 (00)
- **Datas :** 2+4
- **Datatype :** addr + None
- **Description :** Vérifie si la deque est vide et stocke 1 dans l'EDX si oui, 0 sinon.

- **POP**

- **Code :** 1 (01)
- **Datas :** 3+1
- **Datatype :** None+code
- **Description :** Si le code est 1, retire la valeur en haut de la pile, sinon, à gauche de la file. La valeur est chargée dans EDX.

- **PSR**

- **Code :** 2 (10)
- **Datas :** 2 + 4
- **Datatype :** addr + addr
- **Description :** Ajoute la valeur de la variable en haut de la pile.

- **PSL**

- **Code :** 3 (11)
- **Datas :** 2+4
- **Datatype :** addr+addr
- **Description :** Ajoute la valeur de la variable à gauche de la file.

## CMP

**Code :** 12 (1100)

**Datas :** 4b+4b

**Datatype :** addr + addr

**Description :** Compare les deux variables. 1 (0000 0001) pour supérieur, 0 (0000 0000) pour égal et 255 (1111 1111) pour inférieur. Le résultat est stocké dans l'EDX.

## IMP

**Code :** 13 (1101)

**Datas :** 4b+4b

**Datatype :** code + addr

**Description :** Récupère la donnée du capteur et la charge dans la variable.

## EXP

**Code :** 14 (1110)

**Datas :** 4b+4b

**Datatype :** code + addr

**Description :** Envoie la valeur de la variable à l'actionneur.

STP

**Code :** 15 (1111)

**Datas :** 8

**Datatype :** Code

**Description :** Indice de début/fin de programme ou de feuille. 255 (1111 1111) pour le début d'une feuille, 60 (0011 1100) pour la fin d'une feuille, 85 (0101 0101) pour le début d'un programme et 170 (1010 1010) pour la fin d'un programme.