



## MATIERE : Systèmes Embarqués et Systèmes Temps Réels

### TP N°4 : Introduction au Langage Assembleur

**Objectifs** : les objectifs principaux de ce TP sont :

Utilisation de quelques instructions assembleur pour :

- Configuration des GPIO
- Réalisation de quelques opérations arithmétiques et logiques

Voir DOC\_TP4

**Remplacer le code du dernier TP par le code suivant**

```
int var1 =5 ;
int var2=8;
int var3=0;
unsigned char Temp = 10;

void main() {

    GPIO_Digital_Input(&GPIOA_IDR, _GPIO_PINMASK_0);           // Set PA0 as
digital input
    GPIO_Digital_Input(&GPIOA_IDR, _GPIO_PINMASK_3);           // Set PA3 as
digital OUT PUT
    GPIO_Digital_Output(&GPIOB_ODR, _GPIO_PINMASK_ALL);        //PORT B OUTPUT
    GPIO_Digital_Input(&GPIOC_IDR, _GPIO_PINMASK_ALL);         //PORT C INPUT

    asm{

        LDR R0,=_var1
        LDR R1,=_var2
        add R3, R0,R1
        NOP
        MUL R1,R2,R3
        SDIV R1,R2,R3
        LDR R6, = GPIOD_CRL // @ Load GPIOD control register low (Pin1
is in CRL register)
        LDR R5, = 0x00000020 // @ Set MODE bits to 10 and CNF bits to 00,
thus 0x00000020
        STR R5, [R6]
        LDR R6, = RCC_APB2ENR // @ Load peripheral clock enable register
        LDR R5, [R6] // @ Read its content
        ORR R5, 0x00000020 // @ Set Bit 5 to enable GPIOD clock
        STR R5, [R6]
        MVNS R2, #0xF
        MOVT R0, #0x5

        MOVW R1, #52
        MOVW R6, #10
    }
```

```
}  
}
```

Avec

**Build > Build**

**Run>Start Debugger visualiser la fichier assembleur View > Assembly**

**Ajouter les registres dans Watch values ,avec le bouton step over executer les lignes de codes pas a pas et suivre l'évolution de chaque registre**

**Analyser le programme et chaque instruction assembleur**

### **1. En utilisant les instructions assembleur**

#### **Configurer**

- PA1 **input** en **pull down**
- PA2 **input** en **pull up**
- PA3 **output** en **push pull** avec **max speed 10 MHz.**
- Donner le programme assembleur qui cacul  $a+b=c$  utiliser les register spéciaux
-