

---

# Compte-Rendu TP3 :

## Développement à base de CMSIS sur STM32F1

---

LOUHICHI Najmeddine

ATTIA Malek

Classe : II2D

---

### B. Manipuler des composants externes à la carte

- On va utiliser le port GPIOB et les pins de pin0 vers le pin6 tels que:le pin0 va être branché avec Cathode A , pin1 avec Cathode B ... Jusqu'au pin6 avec Cathode G.
- Après modifier la partie configuration pour contrôler les pins :

```
67 //
68 RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB,ENABLE);
69
70 /* Configure pin led as output mode */
71 GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0|GPIO_Pin_1|GPIO_Pin_2|GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_6; /*< Specifies the GPIO pins to be initialized as in this example
72 This parameter can be any value of @ref GPIO_pins_define */
73
```

- Voilà la table qui déterminer quels segments doivent être allumés pour afficher le chiffre ou la lettre correspondante :

	<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>	<b>e</b>	<b>f</b>	<b>g</b>	<b>DP</b>
<b>0</b>	1	1	1	1	1	1	0	x
<b>1</b>	0	1	1	0	0	0	0	x
<b>2</b>	1	1	0	1	1	0	1	x
<b>3</b>	1	1	1	1	0	0	1	x
<b>4</b>	0	1	1	0	0	1	1	x
<b>5</b>	1	0	1	1	0	1	1	x
<b>6</b>	1	0	1	1	1	1	1	x
<b>7</b>	1	1	1	0	0	0	0	x
<b>8</b>	1	1	1	1	1	1	1	x
<b>9</b>	1	1	1	1	0	1	1	x
<b>A</b>	1	1	1	1	0	1	1	x
<b>B</b>	0	1	1	1	1	0	1	x
<b>C</b>	1	0	0	1	1	1	0	x
<b>D</b>	0	1	1	1	1	0	1	x
<b>E</b>	1	0	0	1	1	1	1	x
<b>F</b>	1	0	0	0	1	1	1	x

- La fonction Affichage() :

```
void Affichage( int caract ){
switch (caract){
case 0 : GPIO_SetBits(GPIOB, GPIO_Pin_0|GPIO_Pin_1|GPIO_Pin_2|GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_5);
        GPIO_ResetBits(GPIOB, GPIO_Pin_7);
        break;
case 1 : GPIO_SetBits(GPIOB, GPIO_Pin_1|GPIO_Pin_2);
        GPIO_ResetBits(GPIOB, GPIO_Pin_0|GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_6);
        break;
case 2 : GPIO_SetBits(GPIOB, GPIO_Pin_0|GPIO_Pin_1|GPIO_Pin_4|GPIO_Pin_3|GPIO_Pin_6);
        GPIO_ResetBits(GPIOB, GPIO_Pin_2|GPIO_Pin_5);
        break;
case 3 : GPIO_SetBits(GPIOB, GPIO_Pin_0|GPIO_Pin_1|GPIO_Pin_2|GPIO_Pin_3|GPIO_Pin_6);
        GPIO_ResetBits(GPIOB, GPIO_Pin_4|GPIO_Pin_5);
        break;
case 4 : GPIO_SetBits(GPIOB, GPIO_Pin_1|GPIO_Pin_2|GPIO_Pin_5|GPIO_Pin_6);
        GPIO_ResetBits(GPIOB, GPIO_Pin_0|GPIO_Pin_3|GPIO_Pin_4);
        GPIO_ResetBits(GPIOB, GPIO_Pin_4);
        break;
case 10 : GPIO_SetBits(GPIOB, GPIO_Pin_0|GPIO_Pin_1|GPIO_Pin_2|GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_6);
        GPIO_ResetBits(GPIOB, GPIO_Pin_3);
        break;
case 11 : GPIO_SetBits(GPIOB, GPIO_Pin_2|GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_6);
        GPIO_ResetBits(GPIOB, GPIO_Pin_0|GPIO_Pin_1);
        break;
case 12 : GPIO_SetBits(GPIOB, GPIO_Pin_0|GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_5);
        GPIO_ResetBits(GPIOB, GPIO_Pin_1|GPIO_Pin_2|GPIO_Pin_6);
        break;
case 13 : GPIO_SetBits(GPIOB, GPIO_Pin_1|GPIO_Pin_2|GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_6);
        GPIO_ResetBits(GPIOB, GPIO_Pin_0|GPIO_Pin_5);
        break;
case 14 : GPIO_SetBits(GPIOB, GPIO_Pin_0|GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_6);
        GPIO_ResetBits(GPIOB, GPIO_Pin_1|GPIO_Pin_2);
        break;
case 15 : GPIO_SetBits(GPIOB, GPIO_Pin_0|GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_6);
        GPIO_ResetBits(GPIOB, GPIO_Pin_1|GPIO_Pin_2|GPIO_Pin_3);
        break;
}
}
```

- La fonction Delay()

```
volatile uint32_t TimeDelay = 0;

void Delay(__IO uint32_t nCount)

{

    TimeDelay = nCount ;

    while(TimeDelay !=0) ;

}
```

- La partie exécution :

```

166     int compteur=0;
167     while (1)
168     {
169         Affichage(compteur);
170         Delay(1000);
171         if(compteur == 15 )
172             compteur=0;
173         else
174             compteur ++;
175     }
176 }
177 }
178 }

```

## C. Manipuler des composants externes à la carte avec le bouton

Il faut ajouter au programme précédent ( de la partie B ) la partie de configuration du bouton et ajouter un test dans la boucle while

```

GPIO_InitStructure.GPIO_Pin= GPIO_Pin_0;

GPIO_InitStructure.GPIO_Mode= GPIO_Mode_IN_FLOATING;
GPIO_Init(GPIOA,&GPIO_InitStructure);

```

```

    int compteur=0;
    while (1)
    {
        Affichage(compteur);
        Delay(1000);
        if(compteur == 15 )
            compteur=0;
        else
            compteur ++;
        if ( GPIO_ReadInputDataBit(GPIOA,GPIO_Pin_0)) compteur =0;
    }
}

```

## D. Manipuler des composants externes à la carte avec interruption

La Partie configuration :

```
/* Configure Button EXTI line */
EXTI_InitStructure.EXTI_Line = ENABLE;
EXTI_InitStructure.EXTI_Mode = EXTI_Mode_Interrupt;
EXTI_InitStructure.EXTI_Trigger = EXTI_Trigger_Rising;
EXTI_InitStructure.EXTI_LineCmd = ENABLE;
EXTI_Init(&EXTI_InitStructure);

/* Enable and set Button EXTI Interrupt to the lowest priority */
NVIC_InitStructure.NVIC_IRQChannel = EXTI0_IRQn ;
NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0x0F;
NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0x0F;
NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
NVIC_Init(&NVIC_InitStructure);

Affichage(0);
```

Le changement de la fonction affichage ( pour activer une led , on change la fonction GPIO\_SetBits par la fonction GPIO\_WriteBit(,1) alors que pour désactiver une led on change la fonction GPIO\_ResetBits par la fonction GPIO\_WriteBit(,0)

```

*/
void Affichage( int caract ){
switch (caract){
case 0 : GPIO_WriteBit( GPIOB, GPIO_Pin_0|GPIO_Pin_1|GPIO_Pin_2|GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_5, 1);
GPIO_WriteBit(GPIOB, GPIO_Pin_7,0);
break;
case 1 : GPIO_WriteBit(GPIOB, GPIO_Pin_1|GPIO_Pin_2,1);
GPIO_WriteBit(GPIOB, GPIO_Pin_0|GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_6,0);
break;
case 2 : GPIO_WriteBit(GPIOB, GPIO_Pin_0|GPIO_Pin_1|GPIO_Pin_4|GPIO_Pin_3|GPIO_Pin_6,1);
GPIO_WriteBit(GPIOB, GPIO_Pin_2|GPIO_Pin_5,0);
break;
case 3 : GPIO_WriteBit(GPIOB, GPIO_Pin_0|GPIO_Pin_1|GPIO_Pin_2|GPIO_Pin_4|GPIO_Pin_3|GPIO_Pin_6,1);
GPIO_WriteBit(GPIOB, GPIO_Pin_4|GPIO_Pin_5,0);
break;
case 4 : GPIO_WriteBit(GPIOB, GPIO_Pin_1|GPIO_Pin_2|GPIO_Pin_5|GPIO_Pin_6,1);
GPIO_WriteBit(GPIOB, GPIO_Pin_0|GPIO_Pin_3|GPIO_Pin_4,0);
break;
case 5 : GPIO_WriteBit(GPIOB, GPIO_Pin_0|GPIO_Pin_2|GPIO_Pin_3|GPIO_Pin_5|GPIO_Pin_6,1);
GPIO_WriteBit(GPIOB, GPIO_Pin_1|GPIO_Pin_4,0);
break;
case 6 : GPIO_WriteBit(GPIOB, GPIO_Pin_0|GPIO_Pin_2|GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_6,1);
GPIO_WriteBit(GPIOB, GPIO_Pin_1,0);
break;
case 7 : GPIO_WriteBit(GPIOB, GPIO_Pin_0|GPIO_Pin_1|GPIO_Pin_2,1);
GPIO_WriteBit(GPIOB, GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_6,0);
break;
case 8 : GPIO_WriteBit(GPIOB, GPIO_Pin_0|GPIO_Pin_1|GPIO_Pin_2|GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_6,1);
break;
case 9 : GPIO_WriteBit(GPIOB, GPIO_Pin_0|GPIO_Pin_1|GPIO_Pin_2|GPIO_Pin_3|GPIO_Pin_5|GPIO_Pin_6,1);
GPIO_WriteBit(GPIOB, GPIO_Pin_4,0);
break;
case 10 : GPIO_WriteBit(GPIOB, GPIO_Pin_0|GPIO_Pin_1|GPIO_Pin_2|GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_6,1);
GPIO_WriteBit(GPIOB, GPIO_Pin_3,0);
break;
case 11 : GPIO_WriteBit(GPIOB, GPIO_Pin_2|GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_6,1);
GPIO_WriteBit(GPIOB, GPIO_Pin_0|GPIO_Pin_1,0);
break;
case 12 : GPIO_WriteBit(GPIOB, GPIO_Pin_0|GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_5,1);
GPIO_WriteBit(GPIOB, GPIO_Pin_1|GPIO_Pin_2|GPIO_Pin_6,0);
break;
case 13 : GPIO_WriteBit(GPIOB, GPIO_Pin_1|GPIO_Pin_2|GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_6,1);
GPIO_WriteBit(GPIOB, GPIO_Pin_0|GPIO_Pin_5,0);
break;
case 14 : GPIO_WriteBit(GPIOB, GPIO_Pin_0|GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_6,1);
GPIO_WriteBit(GPIOB, GPIO_Pin_1|GPIO_Pin_2,0);
break;
case 15 : GPIO_WriteBit(GPIOB, GPIO_Pin_0|GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_6,1);
GPIO_WriteBit(GPIOB, GPIO_Pin_1|GPIO_Pin_2|GPIO_Pin_3,0);
break;
}
}
}

```

La partie exécution

```

int compteur=0;
while (1)
{
Affichage(compteur);
Delay(1000);
if(compteur == 15 )
compteur=0;

}
}

```