

Dans ce tp, nous allons faire clignoter les LED de la carte d'une certaine façon selon le nombre de pression effectuer sur le bouton :

Pour ce faire, nous allons utiliser un compteur qui s'incrémentera à chaque pression du bouton, selon la valeur du compteur, on entre dans un if qui définit le comportement des LED.

```
int count = 1;
while (1)
{
    if(HAL_GPIO_ReadPin(B1_GPIO_Port, B1_Pin)==GPIO_PIN_SET && count == 1){
        HAL_GPIO_WritePin(LD6_BLUE_GPIO_Port, LD6_BLUE_Pin, GPIO_PIN_SET);
        HAL_Delay(500);
        HAL_GPIO_WritePin(LD6_BLUE_GPIO_Port, LD6_BLUE_Pin, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(LD4_VERT_GPIO_Port, LD4_VERT_Pin, GPIO_PIN_SET);
        HAL_Delay(500);
        HAL_GPIO_WritePin(LD4_VERT_GPIO_Port, LD4_VERT_Pin, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(LD3_ORANGE_GPIO_Port, LD3_ORANGE_Pin, GPIO_PIN_SET);
        HAL_Delay(500);
        HAL_GPIO_WritePin(LD3_ORANGE_GPIO_Port, LD3_ORANGE_Pin, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(LD5_ROUGE_GPIO_Port, LD5_ROUGE_Pin, GPIO_PIN_SET);
        HAL_Delay(500);
        HAL_GPIO_WritePin(LD5_ROUGE_GPIO_Port, LD5_ROUGE_Pin, GPIO_PIN_RESET);
        count ++;
    }
```

Pour chaque condition, on regarde si le bouton à été presser (HAL_GPIO_ReadPin(B1_GPIO_Port, B1_Pin)==GPIO_PIN_SET), ainsi que la valeur du compteur (définie à 1 par défaut)

Pour manipuler les LED, on utilise la fonction HAL_GPIO_WritePin à laquelle on passe en paramètre le port et la LED associer, ainsi que l'attribut GPIO_PIN_SET pour l'allumer ou GPIO_PIN_RESET pour l'éteindre, on place des delay entre chaque action.

Enfin, on incrémente le compteur à la fin de chaque condition, si le compteur est égal à 4, on le remet à 1.

```
if(HAL_GPIO_ReadPin(B1_GPIO_Port, B1_Pin)==GPIO_PIN_SET && count == 2){
    HAL_GPIO_WritePin(LD6_BLUE_GPIO_Port, LD6_BLUE_Pin, GPIO_PIN_SET);
    HAL_Delay(500);
    HAL_GPIO_WritePin(LD6_BLUE_GPIO_Port, LD6_BLUE_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(LD5_ROUGE_GPIO_Port, LD5_ROUGE_Pin, GPIO_PIN_SET);
    HAL_Delay(500);
    HAL_GPIO_WritePin(LD5_ROUGE_GPIO_Port, LD5_ROUGE_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(LD3_ORANGE_GPIO_Port, LD3_ORANGE_Pin, GPIO_PIN_SET);
    HAL_Delay(500);
    HAL_GPIO_WritePin(LD3_ORANGE_GPIO_Port, LD3_ORANGE_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(LD4_VERT_GPIO_Port, LD4_VERT_Pin, GPIO_PIN_SET);
    HAL_Delay(500);
    HAL_GPIO_WritePin(LD4_VERT_GPIO_Port, LD4_VERT_Pin, GPIO_PIN_RESET);
    count ++;
}
```

```

if(HAL_GPIO_ReadPin(B1_GPIO_Port, B1_Pin)==GPIO_PIN_SET && count == 3){
    HAL_GPIO_WritePin(LD6_BLUE_GPIO_Port, LD6_BLUE_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(LD5_ROUGE_GPIO_Port, LD5_ROUGE_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(LD4_VERT_GPIO_Port, LD4_VERT_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(LD3_ORANGE_GPIO_Port, LD3_ORANGE_Pin, GPIO_PIN_SET);
    HAL_Delay(500);
    HAL_GPIO_WritePin(LD6_BLUE_GPIO_Port, LD6_BLUE_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(LD5_ROUGE_GPIO_Port, LD5_ROUGE_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(LD4_VERT_GPIO_Port, LD4_VERT_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(LD3_ORANGE_GPIO_Port, LD3_ORANGE_Pin, GPIO_PIN_RESET);
    count ++;
}

```

```

if(HAL_GPIO_ReadPin(B1_GPIO_Port, B1_Pin)==GPIO_PIN_SET && count == 4){
    HAL_GPIO_WritePin(LD3_ORANGE_GPIO_Port, LD3_ORANGE_Pin, GPIO_PIN_SET);
    HAL_Delay(500);
    HAL_GPIO_WritePin(LD4_VERT_GPIO_Port, LD4_VERT_Pin, GPIO_PIN_SET);
    HAL_Delay(500);
    HAL_GPIO_WritePin(LD6_BLUE_GPIO_Port, LD6_BLUE_Pin, GPIO_PIN_SET);
    HAL_Delay(500);
    HAL_GPIO_WritePin(LD5_ROUGE_GPIO_Port, LD5_ROUGE_Pin, GPIO_PIN_SET);
    HAL_Delay(1000);
    HAL_GPIO_WritePin(LD6_BLUE_GPIO_Port, LD6_BLUE_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(LD5_ROUGE_GPIO_Port, LD5_ROUGE_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(LD4_VERT_GPIO_Port, LD4_VERT_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(LD3_ORANGE_GPIO_Port, LD3_ORANGE_Pin, GPIO_PIN_RESET);
    count = 1;
}

```