

Classe : 2^{ème} LAII	Devoir Surveillé	Nombre de pages : 2
Enseignants : Neila BADIOUI	Date : 31-10-2019	Durée : 1h30
Barème approximatif :	Documents : autorisé ☐ : non autorisé ☒	Calculatrice : autorisée ☐ : non autorisée ☒

Architecture des calculateurs

Exercice 1

Nous souhaitons implémenter un système de présence d'une personne muni d'alarme sur une carte STM32F4.

Le système est composé d'un capteur numérique qui indique vrai en cas de présence d'individu.

En cas d'intrusion, une alarme se met à sonner, et un message est affiché sur LCD "intrusion".

1) Nous proposons dans ce cadre de déclarer les fonctions suivantes :

- « CTOpenShield_alarmeInit() » dans laquelle vous configurer le GPIO de l'alarme (GPIOC, PIN6) ;
- CTOpenShield_capteurInit() » dans laquelle vous configurer le GPIO du capteur (GPIOD, PIN7) ;
- « CTOpenShield_LCDInit() » dans laquelle vous configurer le module LCD

2) Dans un programme principal, proposer un code permettant de concevoir l'application souhaitée.

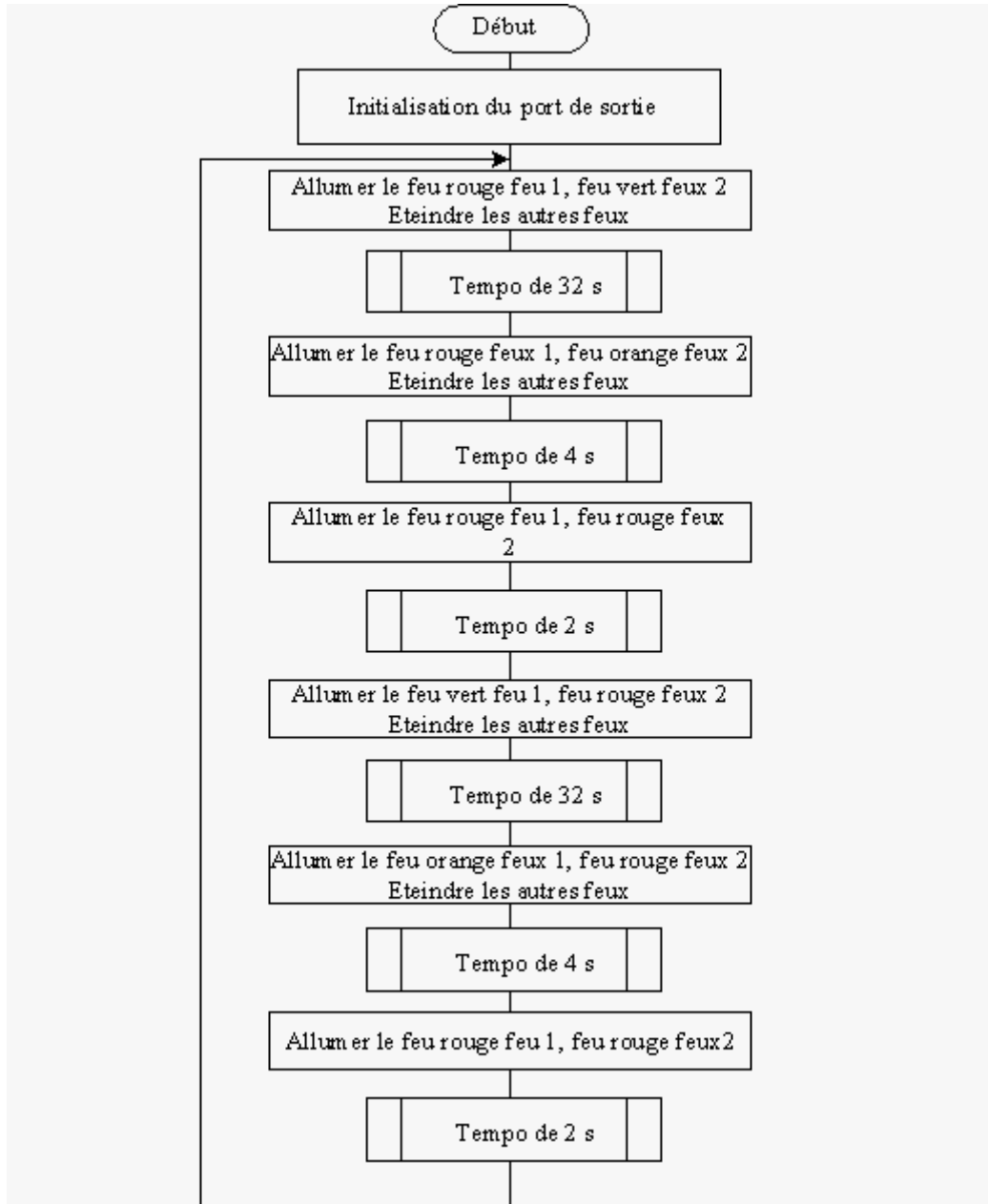
Exercice 2

Le système de gestion de 2 feux de carrefour feux 1 et feux 2 est décrit par l'organigramme ci-dessous. Chaque feu est composé de trois couleurs Rouge, Orange et Vert.

1) Proposer une fonction « CTOpenShield_LEDInit() » dans laquelle vous configurer le GPIO des led L1R, L1V, L1O et L2R, L2V et L2O de la carte en sortie.

L1R	GPIOC	PIN0
L1V	GPIOC	PIN1
L1O	GPIOC	PIN2
L2R	GPIOC	PIN3
L2V	GPIOC	PIN4

- 1) Dans un programme principal, proposer une application de gestion de feux de carrefour répondant au cahier de charge.



Annexe

```
GPIO_InitTypeDef var;
```

```
RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOC, ENABLE);  
var.GPIO_Pin= GPIO_Pin_15;  
var.GPIO_Mode= GPIO_Mode_OUT;  
var.GPIO_Speed= GPIO_Speed_50MHz;  
var.GPIO_OType= GPIO_OType_PP;  
var.GPIO_PuPd= GPIO_PuPd_DOWN;
```

```
GPIO_Init(GPIOC, &var);
```

```
uint8_t GPIO_ReadOutputDataBit(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin)
```

```
uint8_t GPIO_ReadInputDataBit(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin)
```

```
void GPIO_ResetBits(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin)
```

```
void GPIO_SetBits(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin)
```

```
void LCD_Puts(const char *s)
```

```
LCD_InitTypeDef LCD_InitStruct;
```

```
    // LCD Init
```

```
    LCD_InitStruct.CharacterFont = LCD_CharacterFont_5x10Dots;
```

```
    LCD_InitStruct.DataLength = LCD_DataLength_4Bit;
```

```
    LCD_InitStruct.LineNumber = LCD_LineNumber_2Lines;
```

```
    LCD_Init(&LCD_InitStruct);
```