



Département EEA – Faculté Sciences et Ingénierie

Master 1 SME

Projet BE : TP INITIALE

Année 2022-2023

Rapport de projet

Rédigé par HAMDAN Feras et ZAHER Aymen

Réalisation de systèmes : BE-TP-INITIALE

Mini projet : LED

Le but de ce mini projet est de se familiariser avec le microcontrôleur STM32 et de son IDE : STM32CubeIDE, en particulier la création d'un projet, comment le débogger et son exécution. Après avoir créé le projet LED l'IDE génère les fichiers .c et .h. Pour faire clignoter la LED insère le morceau de programme fourni dans le cahier de TP dans la boucle while(1) dans main.c. Ce programme utilise la fonction HAL_GPIO_WritePin pour mettre le pin LD2 à 1 puis à 0, il utilise également la fonction HAL_Delay pour insérer des delay, ceci nous permet d'observer les changements d'état de la LED. Le delay utilisé après la mise à 1 est plus grand que le delay utilisé après la mise à 0, on peut donc déduire, en testant le projet sur le STM32, que la LED est allumée lorsque le pin est mis à 1.

Mini projet : BP-LED

Dans ce mini projet on cherche à faire clignoter la LED que lorsque le bouton est appuyé. Le morceau de programme qu'on a rajouté dans main.c utilise la fonction HAL_GPIO_ReadPin pour tester si le bouton est appuyé.

Dans la boucle while(1):

- Si le bouton n'est pas appuyé, on utilise la fonction HAL_GPIO_WritePin pour mettre le pin LD2 à 0 est donc éteindre la LED
- Dans le cas contraire, donc si le bouton est appuyé, on utilise la fonction HAL_GPIO_TogglePin et des delays pour faire clignoter la LED

Mini projet : BP-LED-interruption

Dans ce mini projet on cherche à ce que la LED change d'état lorsque le bouton poussoir utilisateur B1 est relâché ou si un front descendant apparaît sur PA8 ou PA9. Pour cela on crée un projet de la même façon que dans les mini projets précédents, à l'exception de la configuration des pins, en effet les pins PA8 et PA9 doit être configuré en Pull-Up et mode d'interruption externe avec détection de déclenchement par front descendant, comme on peut le voir sur le figure 1. On doit également autoriser les interruptions sur ces deux pins.

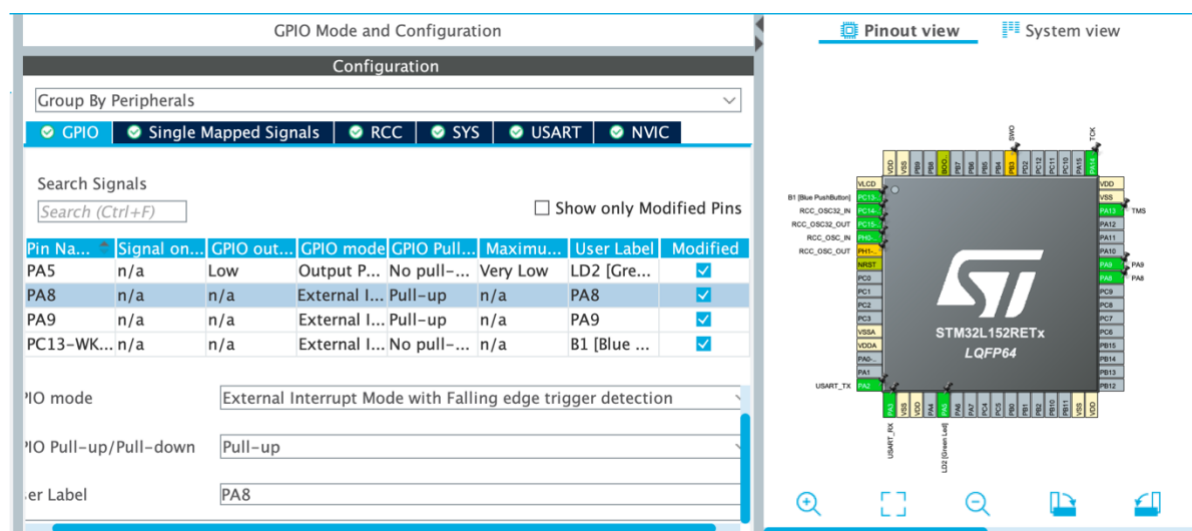


Figure 1

Pour allumer la LED, on utilise la fonction d'interruption HAL_GPIO_EXTI_Callback. Lorsqu'une des conditions est respecté, le programme arrête ce qu'il fait et effectue la fonction d'interruption.

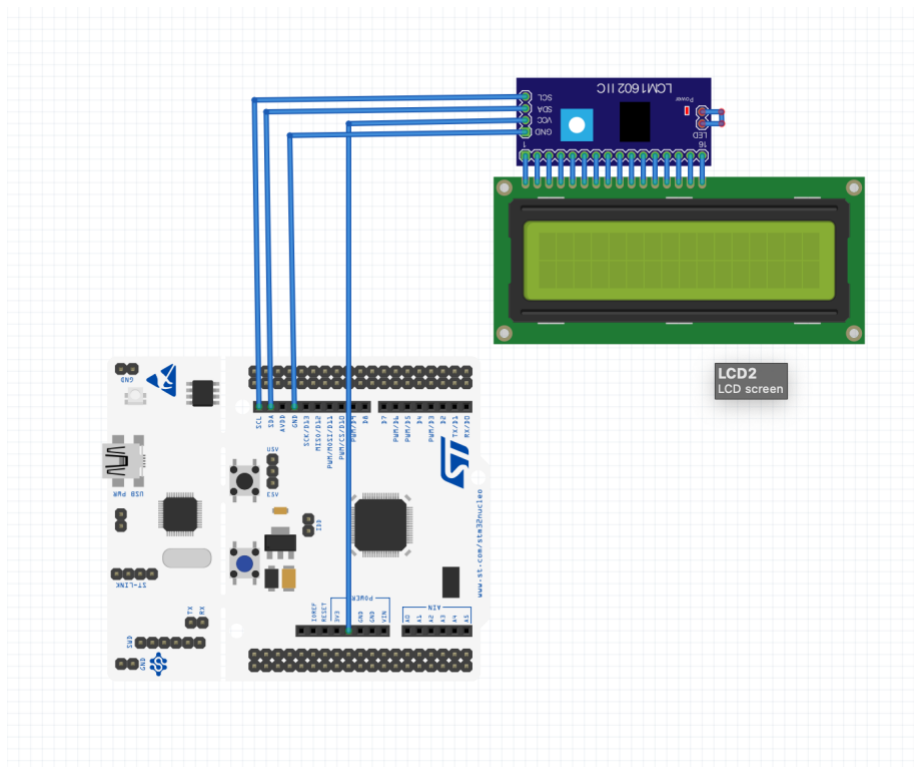
```

/* USER CODE BEGIN 4 */
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    switch(GPIO_Pin)
    {
        case B1_Pin: // GPIO_PIN_13 is the Blue Button
            HAL_GPIO_TogglePin(LD2_GPIO_Port, LD2_Pin); // Green LED - GPIOA,
            GPIO_PIN_5
            break;
        case GPIO_PIN_8:
            HAL_GPIO_TogglePin(LD2_GPIO_Port, LD2_Pin); // Green LED - GPIOA,
            GPIO_PIN_5
            break;
        case GPIO_PIN_9:
            HAL_GPIO_TogglePin(LD2_GPIO_Port, LD2_Pin); // Green LED - GPIOA,
            GPIO_PIN_5
            break;
    }
}

```

Écran LCD

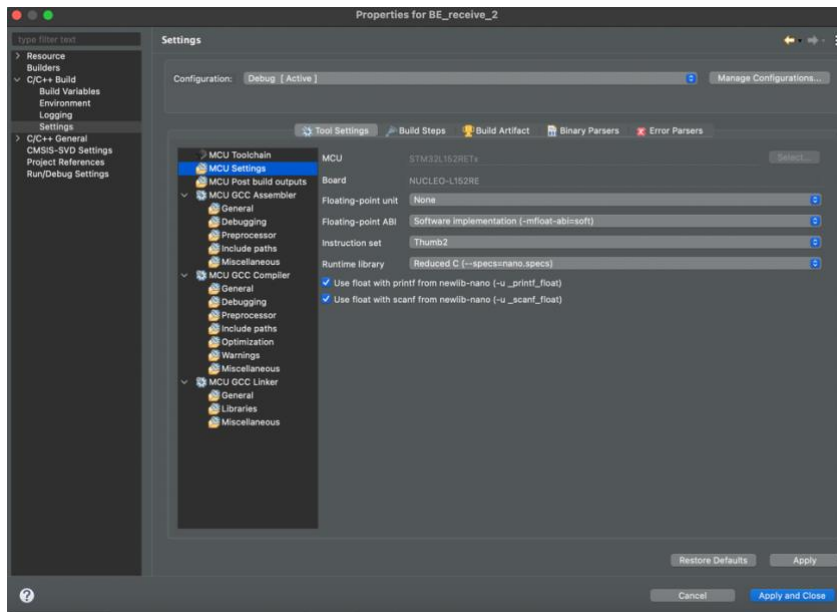
Pour afficher sur un écran LCD, on configure les pin PB8 et PB9 en SDA et SCL respectivement. On configure I2C1 en mode I2C, puisque le LCD nécessite une communication en I2C.



On connecte VCC à VCC, GND à GND, SDA à SDA et SCL à SCL.

A partir du programme la fonction `lcd_print` permet d'afficher des strings (et des chars), pour afficher un entier, on le convertie en string avec la fonction `sprintf`. Pour utiliser cette

fonction il faut d'abord autoriser use float with printf et use float with scanf dans les paramètres du projet.



Pour pouvoir afficher plusieurs strings en même temps, il est possible de les placer à différentes positions sur l'écran avec la fonction `Lcd_position`.