

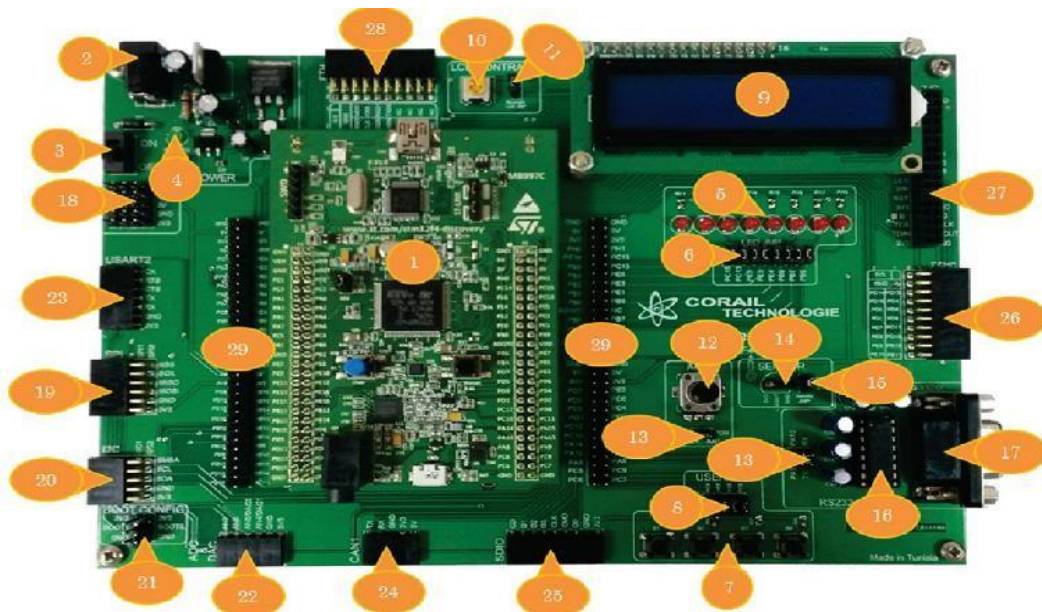
## TP1- Gestion GPIO

### I. Objectifs :

- Initialiser avec le logiciel « Coocox CoIDE »
- Développer des applications simples sur la gestion des GPIOs du microcontrôleur STM32F407.

### II. Environnement matériel

Le STM32F4-Discovery est une carte de développement conçu pour l'outil officiel de la famille STMicroelectronics STM32F4, qui comprend le microcontrôleur STM32F407VGT6. Il est conçu pour permettre aux étudiants de tester facilement et explorer la capacité de ce type de microcontrôleur. Les STM32F4 Discovery sont des cartes de démonstration développée chez STMicroelectronics à prix relativement bas afin de permettre aux développeurs des systèmes embarqués de maîtriser les microcontrôleurs STM32. Le Kit STM32F4 inclut tous les périphériques nécessaires pour s'initier à la programmation des microcontrôleurs de la famille STM32.



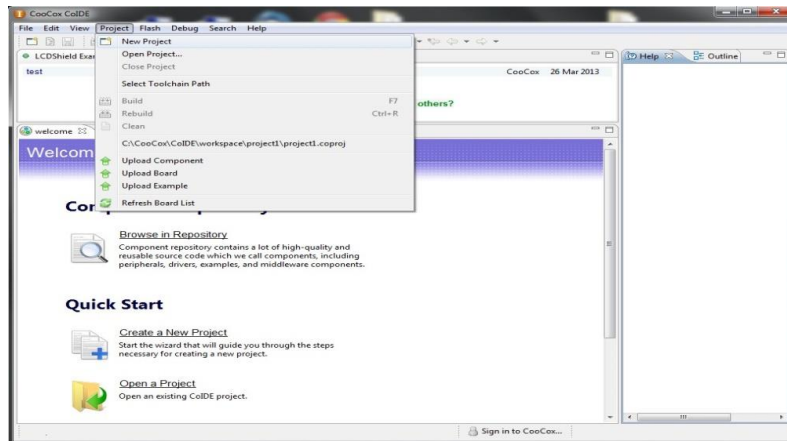
- |   |  |
|---|--|
| 1) STM32F4-Discovery                      | 18) Broches d'alimentation 12V, 5V, 3.3V et GND  |
| 2) Connecteur alimentation 12VDC          | 19) Connecteur de l'interface SPI1 / SPI2 & ADC/DAC  |
| 3) Switcher ON/OFF                        | 20) Connecteur d'interface I2C   |
| 4) Voyant d'alimentation                  | 21) Jumpers de sélection du mode BOOT  |
| 5) 8 LEDs                                 | 22) Connecteur de l'interface ADC/DAC  |
| 6) Jumper LEDs                            | 23) Connecteur de l'interface UART2  |
| 7) 4 Boutons poussoirs                    | 24) Connecteur de l'interface bus CAN1   |
| 8) Jumper Boutons                         | 25) Connecteur de l'interface SDIO   |
| 9) Afficheur LCD 16x2                     | 26) Connecteur de l'interface FSMC   |
| 10) Potentiomètre contraste LCD           | 27) Connecteur LCDG pour un écran LCD tactile  |
| 11) Jumper Back Light LCD                 | 28) Connecteur de l'interface ETHERNET   |
| 12) Potentiomètre d'ajustement ADC        | 29) Connecteur broches MCU: tous les pins de la MCU sont accessibles sur les connecteurs d'extension |
| 13) Jumper d'activation potentiomètre ADC |  |
| 14) TMP36 capteur de température          |  |
| 15) Jumper d'activation capteur TMP36     |  |
| 16) transceiver max3232 pour RS232        |  |
| 17) Connecteur DB9-F pour RS232           |  |

### III. Travail demandé

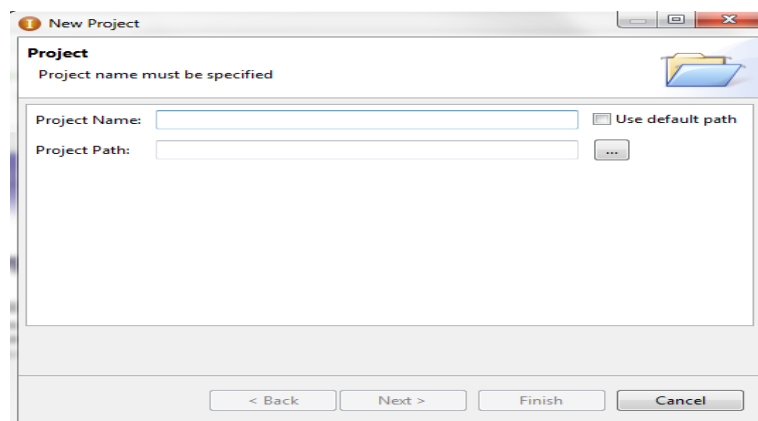
#### ❖ Création d'un nouveau projet :

Suivez les étapes suivantes pour créer un projet sur l'environnement CoIDE.

Step1 : Lancer CoIDE et sélectionner l'option Project >> New project

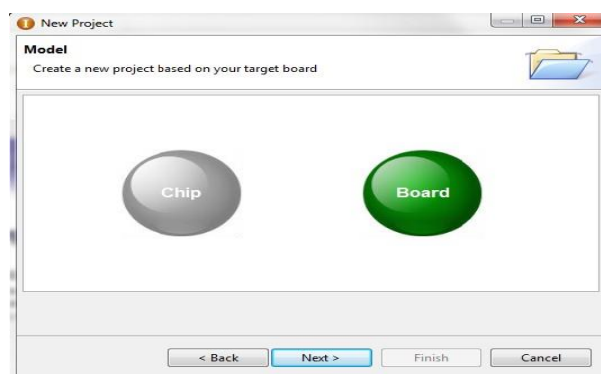


Step 2: Nommer votre projet et créer un répertoire d'enregistrement

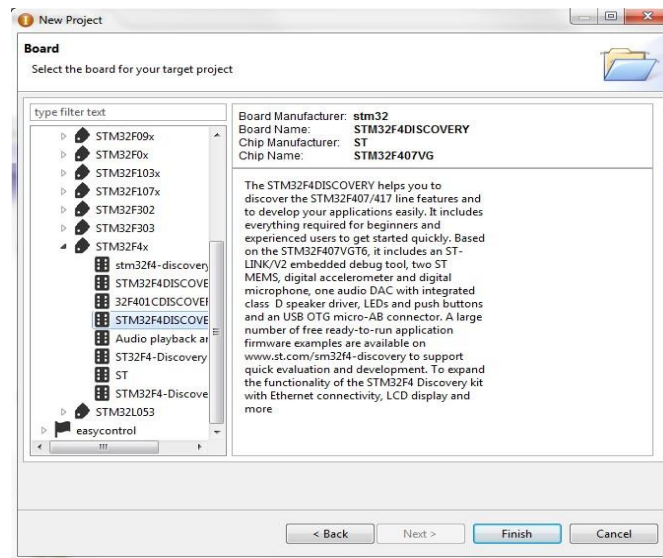


Après avoir donné un nom au projet et précisé leur répertoire d'enregistrement, cliquer sur *Next*

Step 3 : cocher l'option « Board » et cliquer sur « Next »

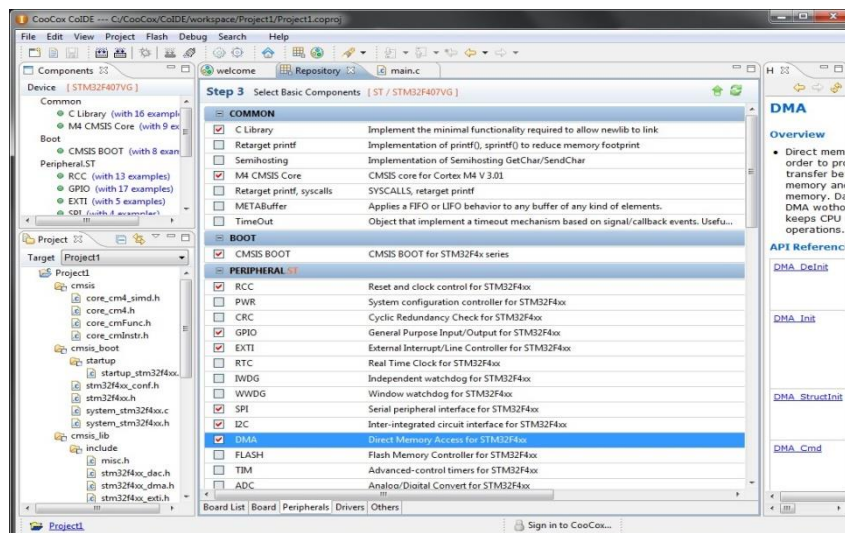


Step4 : Choisir le Board *STM32F4Discovery* et cliquer sur *Finish*



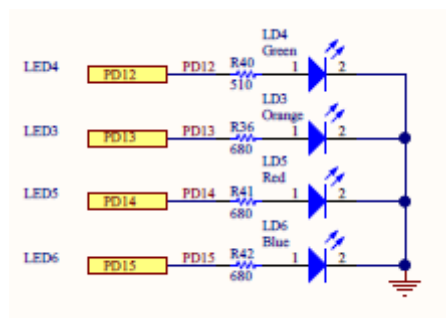
Maintenant votre projet est créé, il vous reste à sélectionner les périphériques à utiliser dans le projet.

**Step5** : Sélectionner les périphériques à utiliser dans le projet.



## Application 1 : Allumer une LED

Le but de cette manipulation est de développer un programme sur l'environnement Coocox CoIDE qui permet d'allumer toutes les LEDs de la carte CT OpenShield.



- 1) Créer un nouveau projet en suivant les étapes décrites précédemment.
- 2) Dans le fichier « stm32f4xx\_GPIO.c », créer une fonction « GPIO\_config » dans laquelle vous configurer le GPIO des LEDs de la carte en sortie (la fonction donnée ci-dessous est à titre indicatif).

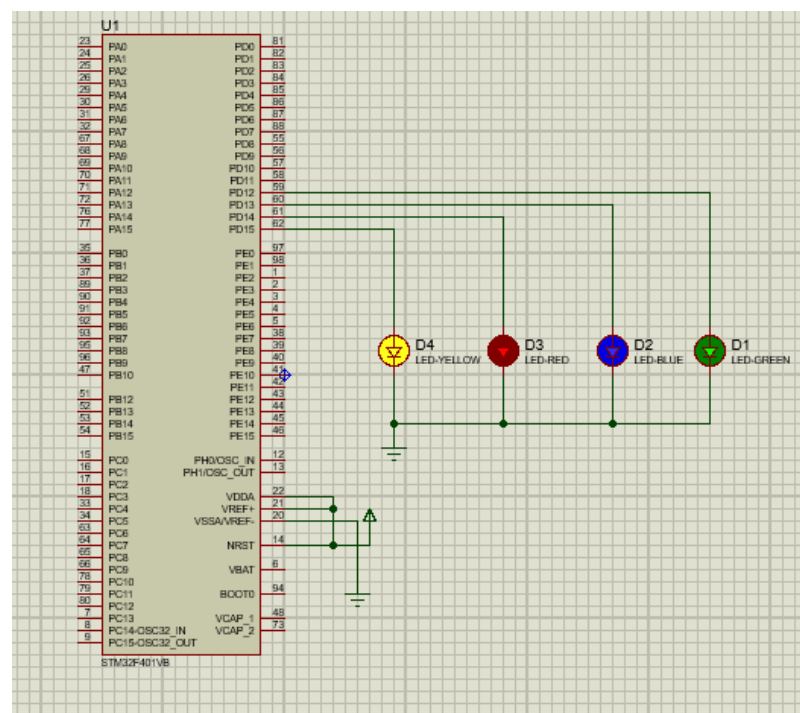
```
void GPIO_config(void)
{
    GPIO_InitTypeDef configuration;
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOD, ENABLE);

    configuration.GPIO_Pin = GPIO_Pin_12;
    configuration.GPIO_Mode = GPIO_Mode_OUT;
    configuration.GPIO_Speed = GPIO_Speed_2MHz;
    configuration.GPIO_OType = GPIO_OType_PP;
    configuration.GPIO_PuPd = GPIO_PuPd_NOPULL;
    GPIO_Init(GPIOD, &configuration);
}
```

Donnée :

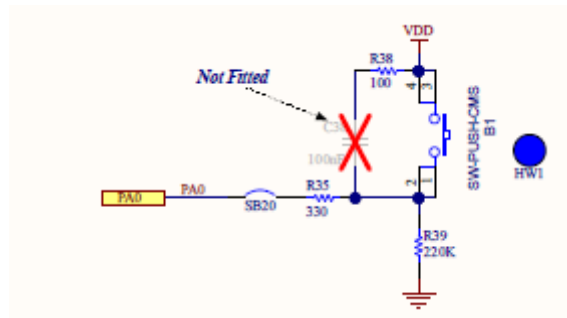
Intitulé des LEDS de la carte STM32					
LED4	GPIOD	PIN12	LED3	GPIOD	PIN13
LED5	GPIOD	PIN14	LED6	GPIOD	PIN15

- 3) Dans le fichier « stm32f4xx\_GPIO.h » ajouter la fonction des configuration des GPIO « GPIO\_config ».
- 4) Dans un programme principal « main.c », créer un code qui permet de clignoter des diodes.
- 5) Compiler votre programme puis tester le sous ISIS suivant le schéma ci-dessous. Sélectionner STM32F401VB à partir de la bibliothèque des composants.

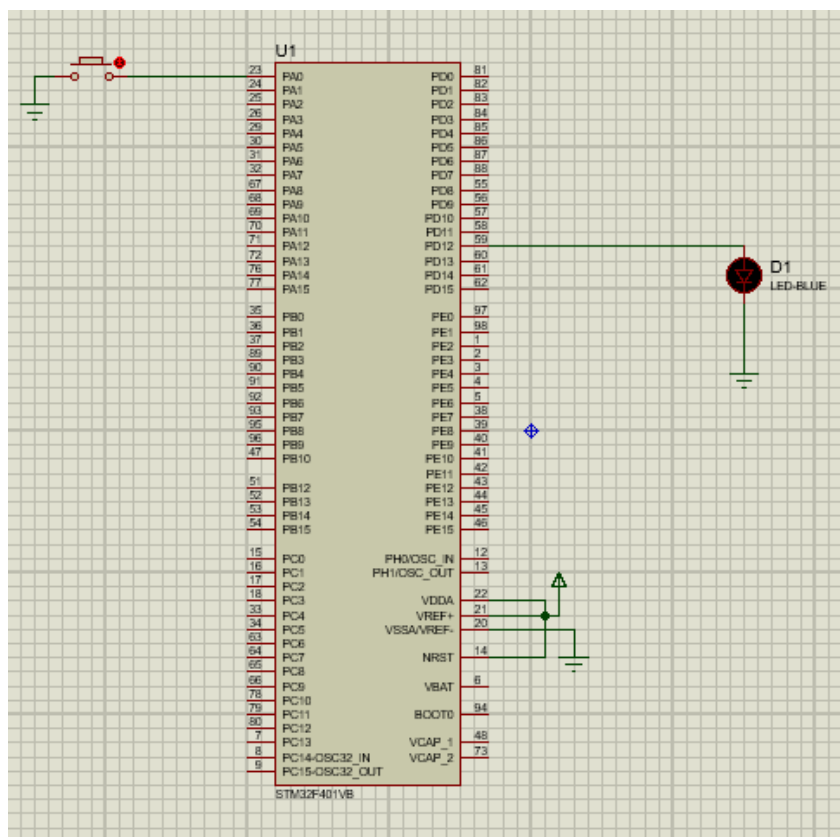


## Application 2 : Boutons poussoir

Le but de cette application est de faire allumer la LED4 (PIN 12) à chaque appui sur un bouton B1 lié à la broche PA0 de la carte STM32F4-Discovery.



- 1) Créer une fonction «PB\_config()» pour configurer et initialiser le bouton B1 (RCC\_AHB1Periph\_GPIOA, PIN\_0)
- 2) Dans un programme principal, appeler ces fonctions pour décrire l'application souhaitée.
- 3) Compiler votre programme puis tester le sous ISIS



### Annexe

```
uint8_t GPIO_ReadInputDataBit(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin)
void GPIO_SetBits(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin)
void GPIO_ResetBits(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin)
void GPIO_ToggleBits(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin)
```