Project_5.4_Timer_Encoder-PWM_Mode

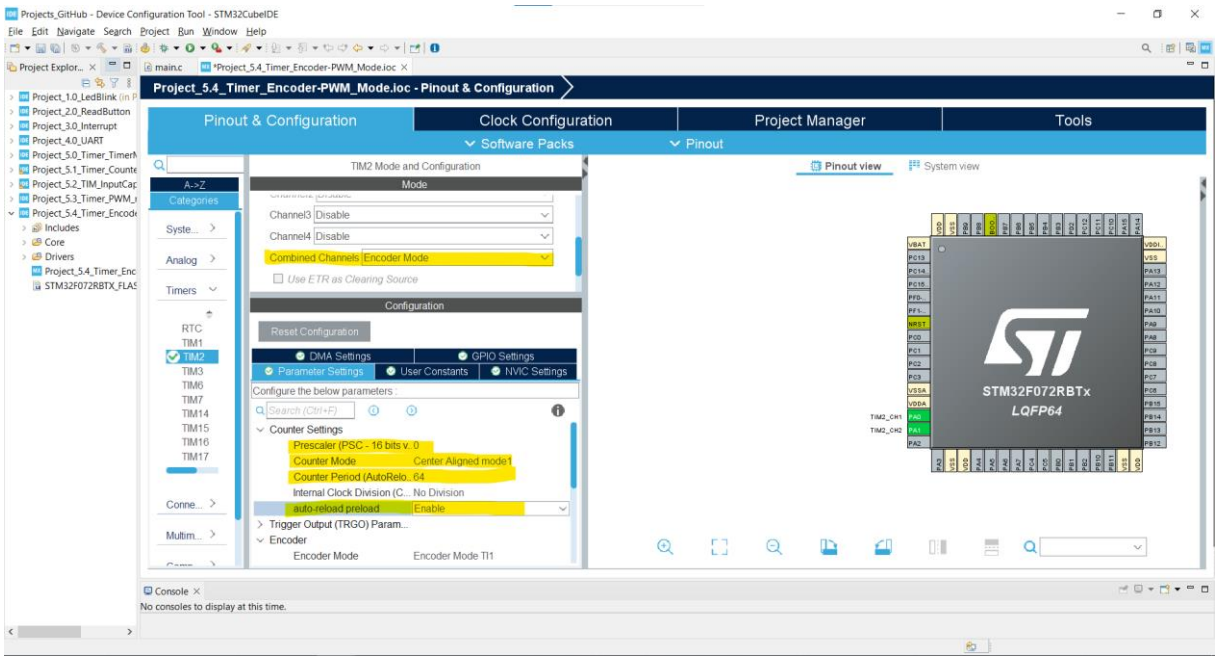
In this project, we will configure TIM2 in encoder mode and TIM3 in PWM mode. We will control the counter of timer 2 with an external rotary encoder and adjust the brightness of the LED.

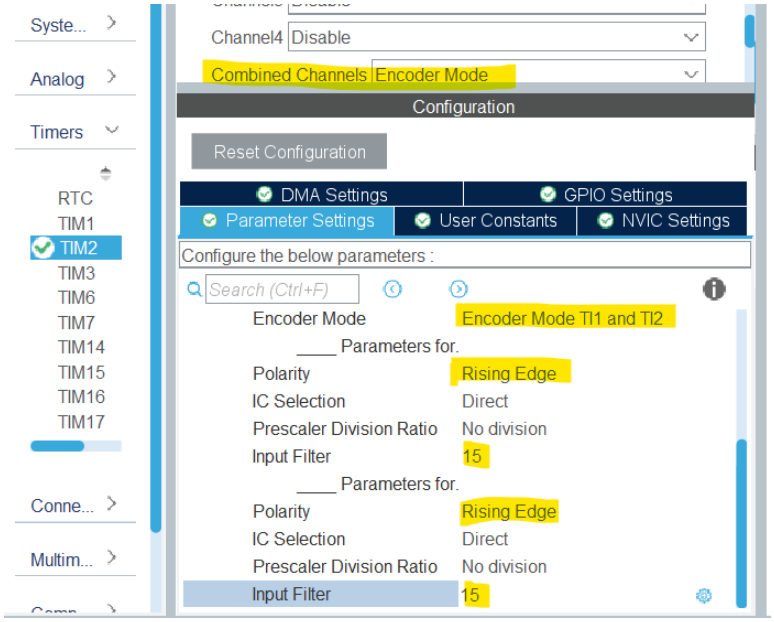
STEP 1 : Open STM32CubeIDE and Create New Project.

STEP 2 : Select Target microcontroller and double Click. My MCU is STM32F072RBT6.

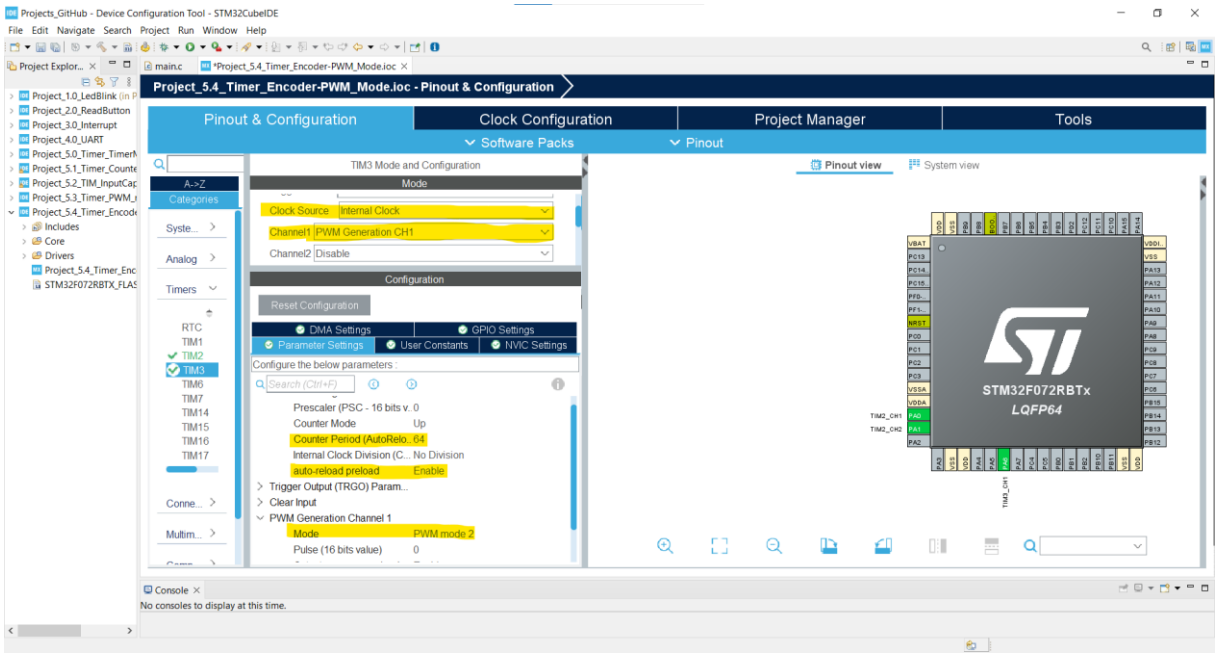
STEP3 : Enter the project name and finish .

Step 4 : Configure Timer 2:

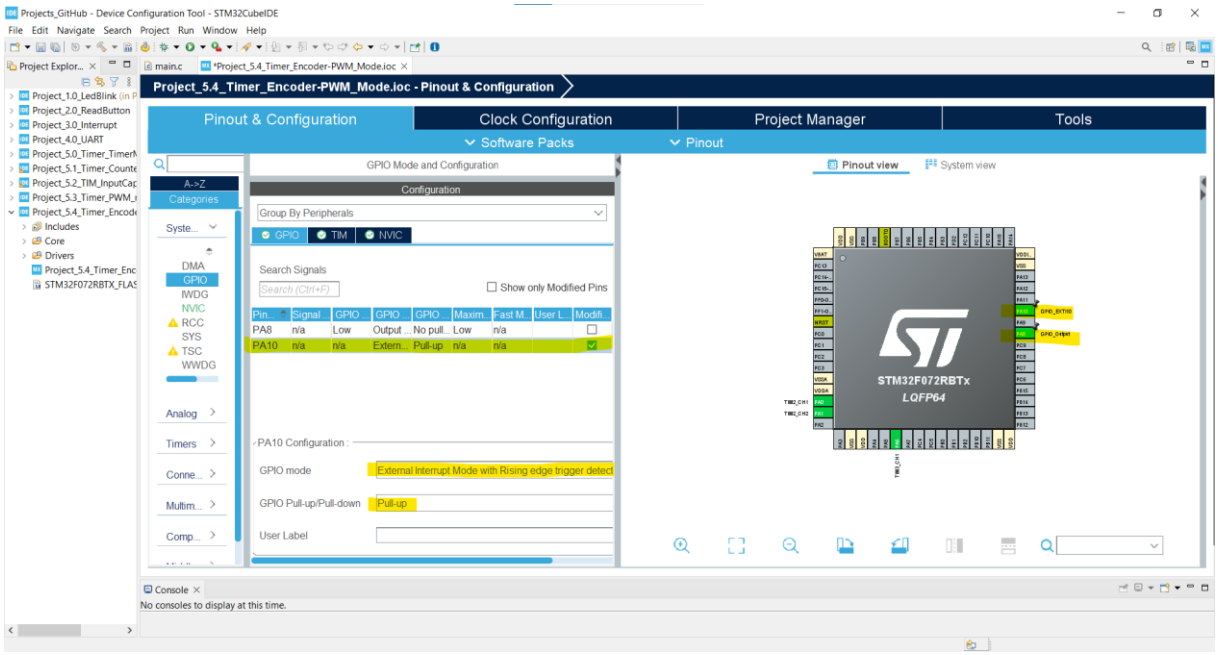


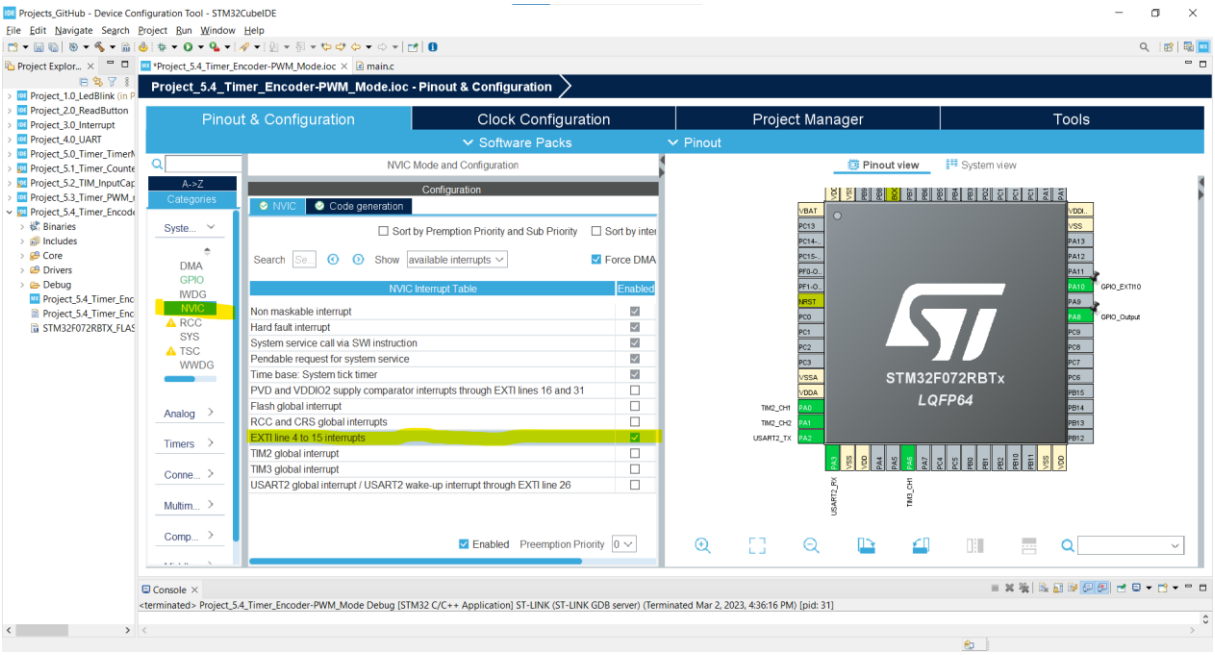


Step 4 : Configure Timer 3:



STEP 5: We will configure a GPIO pin for the button of the encoder. We will create an interrupt with this button and use this interrupt to turn the timers on and off, i.e. to use it to turn the system on and off. Therefore, we configure any pin as GPIOx_EXTIx. At the same time, let's add a LED to see if the system is working or not, and configure an output pin for it.





STEP 6: We configure the UART module.

STEP 7 : Set the RCC External Clock Source and then CTRL + S to generate the project code. And we open our main.c file in the project files.

STEP 8: Code :

```
/* USER CODE END PFP */

/* Private user code ---------------------------------------------------------*/
/* USER CODE BEGIN 0 */
uint8_t start_msg[50] = {"Encoder is starting\n\r"};
uint8_t stop_msg[50] = {"Encoder is stopped\n\r"};
uint8_t status = 0;

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin){

    if(GPIO_Pin == GPIO_PIN_10){

        HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_8);
        switch (status){
        case 0:
              HAL_TIM_Encoder_Start(&htim2, TIM_CHANNEL_ALL);
              HAL_TIM_PWM_Start(&htim3, TIM_CHANNEL_1);
              HAL_UART_Transmit(&huart2, start_msg, sizeof(start_msg), 100);
              status = 1;
              break;
        case 1:
              HAL_TIM_Encoder_Stop(&htim2, TIM_CHANNEL_ALL);
              HAL_TIM_PWM_Stop(&htim3, TIM_CHANNEL_1);
              HAL_UART_Transmit(&huart2, stop_msg, sizeof(stop_msg), 100);
              status=0;
              break;


        }

    }
}
/* USER CODE END 0 */
```

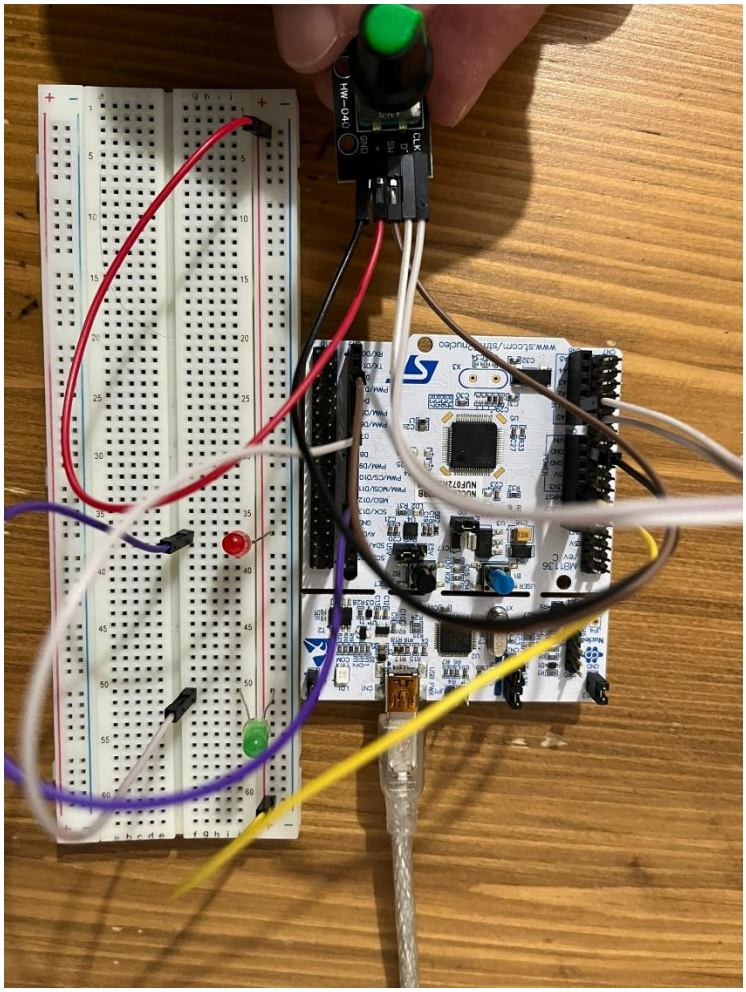
```
int main(void)
{
  /* USER CODE BEGIN 1 */
    uint8_t msg[50] = {'\0'};
    uint16_t LED_Duty_Cycle = 0;
  /* USER CODE END 1 */
```

```
  while (1)
  {
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
        if (status) {

            sprintf(msg, "Brightness of the LED = %d\n\r", ((TIM2->CNT)) >> 1);
            HAL_UART_Transmit(&huart2, msg, sizeof(msg), 100);

        }
        HAL_Delay(100);
        LED_Duty_Cycle = ((TIM2->CNT));
        TIM3->CCR1 = LED_Duty_Cycle;
  }
  /* USER CODE END 3 */
```



https://youtube.com/shorts/TGn5H9yMi9A?feature=share