```c
/* Private define ------------------------------------------------------------*/
/* USER CODE BEGIN PD */
uint32_t WwdgStatus = 0;

/* USER CODE END PD */

/* Private macro -------------------------------------------------------------*/
/* USER CODE BEGIN PM */

static uint32_t TimeoutCalculation(uint32_t timevalue);

#define WWDG_WINDOW    0x50
#define WWDG_COUNTER   0x7F

// In Main()

  /* USER CODE BEGIN 1 */
uint32_t delay = 0;
MX_WWDG_Init();

  /*##-1- Check if the system has resumed from WWDG reset ####################*/
  if (__HAL_RCC_GET_FLAG(RCC_FLAG_WWDGRST) != 0x00u)
  {
    /* WWDGRST flag set: Turn LED4 on and set WWDGStatus */
    WwdgStatus = 1;
 //   BSP_LED_On(LED4);

    /* Insert 4s delay */
     HAL_Delay(4000);

    /* Prior to clear WWDGRST flag: Turn LED4 off */
  //  BSP_LED_Off(LED4);
  }

  /* Clear reset flags in any case */
  __HAL_RCC_CLEAR_RESET_FLAGS();
  WwdgStatus = 0;

  delay = TimeoutCalculation((hwwdg.Init.Counter-hwwdg.Init.Window) + 1) + 1;

  while (1)
  {
              HAL_Delay( delay );
      //     ApplicationCode
            if (HAL_WWDG_Refresh(&hwwdg) != HAL_OK)
            {
              Error_Handler();
            }
 }
```

```c
static void MX_WWDG_Init(void)
{

  /* USER CODE BEGIN WWDG_Init 0 */
    /*  Default WWDG Configuration:
     1] Set WWDG counter to 0x7F  and window to 0x50
     2] Set Prescaler to WWDG_PRESCALER_8

     Timing calculation:
     a) WWDG clock counter period (in ms) = (4096 * WWDG_PRESCALER_8) / (PCLK1 /
1000)
                                          = 0,512 ms
     b) WWDG timeout (in ms) = (0x7F + 1) * 0,512
                             ~= 65,53 ms
     => After refresh, WWDG will expires after 65,53 ms and generate reset if
     counter is not reloaded.
     c) Time to enter inside window
     Window timeout (in ms) = (127 - 80 + 1) * 0,512
                            = 24,58 ms */

  /* USER CODE END WWDG_Init 0 */

  /* USER CODE BEGIN WWDG_Init 1 */

  /* USER CODE END WWDG_Init 1 */
  hwwdg.Instance = WWDG;
  hwwdg.Init.Prescaler = WWDG_PRESCALER_8;
  hwwdg.Init.Window =  WWDG_WINDOW;
  hwwdg.Init.Counter = WWDG_COUNTER;
  hwwdg.Init.EWIMode = WWDG_EWI_DISABLE;
  if (HAL_WWDG_Init(&hwwdg) != HAL_OK)
  {
    Error_Handler();
  }
  /* USER CODE BEGIN WWDG_Init 2 */

  /* USER CODE END WWDG_Init 2 */

}

static uint32_t TimeoutCalculation(uint32_t timevalue)
{
  uint32_t timeoutvalue = 0;
  uint32_t pclk1 = 0;
  uint32_t wdgtb = 0;

  /* considering APB divider is still 1, use HCLK value */
  pclk1 = HAL_RCC_GetPCLK1Freq();

  /* get prescaler */
  wdgtb = (1 << ((hwwdg.Init.Prescaler) >> WWDG_CFR_WDGTB_Pos)); /* 2^WDGTB[1:0]
*/

  /* calculate timeout */
  timeoutvalue = ((4096 * wdgtb * timevalue) / (pclk1 / 1000));

  return timeoutvalue;
}
/* USER CODE END 4 */
```