QUIZ 4

1.

- Generate interrupts from pins PA0 and PA2. The priority of the interrupt received from the PA1 pin can be higher than the PA0 pin. For these, make the necessary adjustments in CubeMx.
- When there is no interrupt, the LED connected to the 15th pin lights up continuously using ODR. (Write the relevant code inside the while loop in main.c).
- When the interrupt is received from the PA0 pin, the value of the i variable increases by 1. Reset all pins connected to port D using BSRR. After the LEDs are connected to the PD13&PD14 pins lights for 2 seconds, all the pins connected to the D port are reset again using BSRR. In addition, check if the number i is an even number. Use a n variable to control how many times i is in an even state. When n is greater than 3, mask the PA0 pin using the Interrupt Mask Register. Write the relevant code inside the EXTI0_IRQHandler function. Observe the cahanges of i and n variables using debug.
- When the interrupt is received from the PA2 pin, the value of the a variable increases by 1. Reset all pins connected to port D using BSRR. After the LEDs are connected to the PD12&PD15 pins lights for 2 seconds, all the pins connected to the D port are reset again using BSRR.
- Give necessary interrupts in order to observe the Late Arrival'. Explain Late Arrival'.

```
while (1)
{
GPIOD->ODR=0x8000;
 HAL_Delay(100);
}


void EXTI0_IRQHandler(void)
{
  /* USER CODE BEGIN EXTI0_IRQn 0 */

  /* USER CODE END EXTI0_IRQn 0 */
  HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_0);
  /* USER CODE BEGIN EXTI0_IRQn 1 */
           i=i+1;
           GPIOD->BSRR=0xFFFF0000;
           HAL_Delay(2000);
           GPIOD->BSRR=0x6000;
           HAL_Delay(2000);
           GPIOD->BSRR=0xFFFF0000;
           if (i%2==0)
               n=n+1;
```

```
            if(n>3)
               EXTI->IMR=0x7FFFFE;
  /* USER CODE END EXTI0_IRQn 1 */
}

void EXTI2_IRQHandler(void)
{
  /* USER CODE BEGIN EXTI2_IRQn 0 */

  /* USER CODE END EXTI2_IRQn 0 */
  HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_2);
  /* USER CODE BEGIN EXTI2_IRQn 1 */
  /* USER CODE BEGIN EXTI1_IRQn 1 */
a=a+1;
GPIOD->BSRR=0xFFFF0000;
GPIOD->BSRR=0x9000;
HAL_Delay(2000);
GPIOD->BSRR=0xFFFF0000;
  /* USER CODE END EXTI2_IRQn 1 */
}
```

2.

- Generate interrupts from pins PA0 and PA3. The priority of the interrupt received from the PA3 pin can be higher than the PA0 pin. For these, make the necessary adjustments in CubeMx.
- When there is no interrupt, the LED connected to the 13th pin lights up continuously using ODR. (Write the relevant code inside the while loop in main.c).
- When the interrupt is received from the PA0 pin, the value of the i variable increases by 1. Reset all pins connected to port D using BSRR. After the LEDs are connected to the PD12&PD15 pins lights for 2 seconds, all the pins connected to the D port are reset again using BSRR. In addition, check if the number i is an odd number. Use a n variable to control how many times i is in an odd state. When n is greater than 2, mask the PA0 pin using the Interrupt Mask Register. Write the relevant code inside the EXTI0_IRQHandler function. Observe the cahanges of i and n variable using debug.
- When the interrupt is received from the PA2 pin, the value of the a variable increases by 1. Reset all pins connected to port D using BSRR. After the LEDs are connected to the PD13&PD14 pins lights for 2 seconds, all the pins connected to the D port are reset again using BSRR.
- Give necessary interrupts in order to observe the ‘Tail Chaining’. Explain Tail Chaining.

```
while (1)

    {
      GPIOD->ODR=0x2000;
      HAL_Delay(100); //Wait 100 ms
     }



void EXTI0_IRQHandler(void)
{
  /* USER CODE BEGIN EXTI0_IRQn 0 */

  /* USER CODE END EXTI0_IRQn 0 */
  HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_0);
  /* USER CODE BEGIN EXTI0_IRQn 1 */
            i=i+1;
           GPIOD->BSRR=0xFFFF0000;
           HAL_Delay(2000);
           GPIOD->BSRR=0x9000;
           HAL_Delay(2000);
```

```
            GPIOD->BSRR=0xFFFF0000;
            if (i%2!=0)
                  n=n+1;
            if(n>3)
                  EXTI->IMR=0x7FFFFE;
  /* USER CODE END EXTI0_IRQn 1 */
}


void EXTI3_IRQHandler(void)
{
  /* USER CODE BEGIN EXTI3_IRQn 0 */
  /* USER CODE END EXTI3_IRQn 0 */
  HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_3);
  /* USER CODE BEGIN EXTI3_IRQn 1 */
a=a+1;
GPIOD->BSRR=0xFFFF0000;
HAL_Delay(2000);
GPIOD->BSRR=0x6000;
HAL_Delay(2000);
GPIOD->BSRR=0xFFFF0000;
  /* USER CODE END EXTI3_IRQn 1 */
}
```