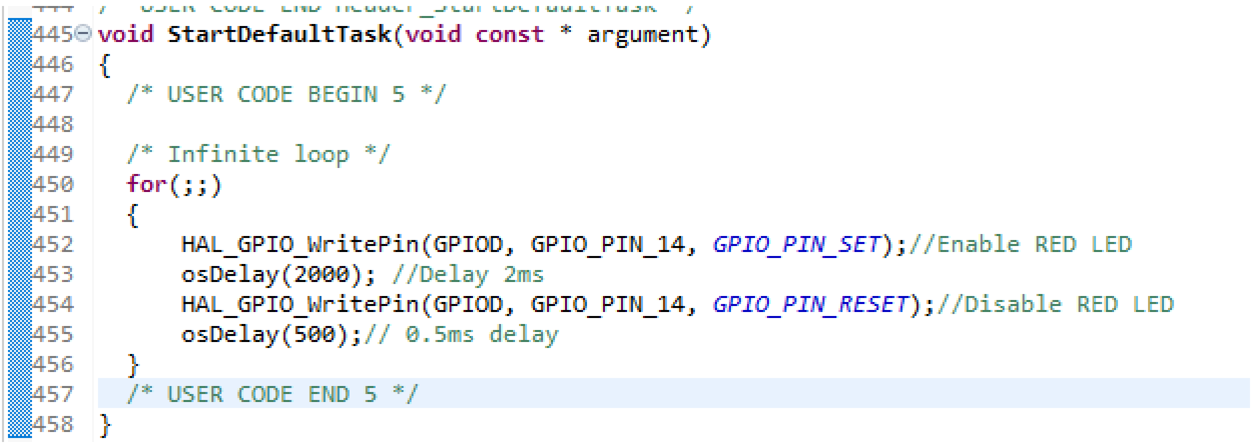
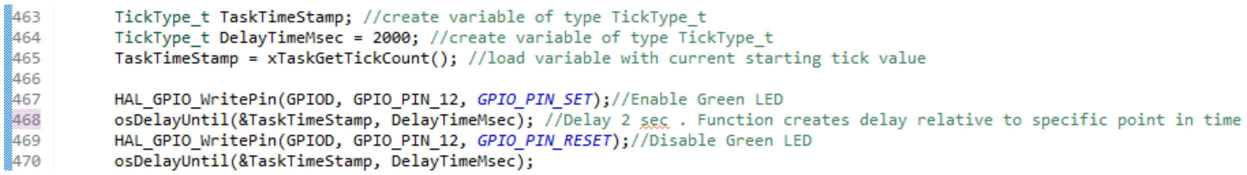
Farhan Ali Experiment 2 2/3/24

Exercise 1)



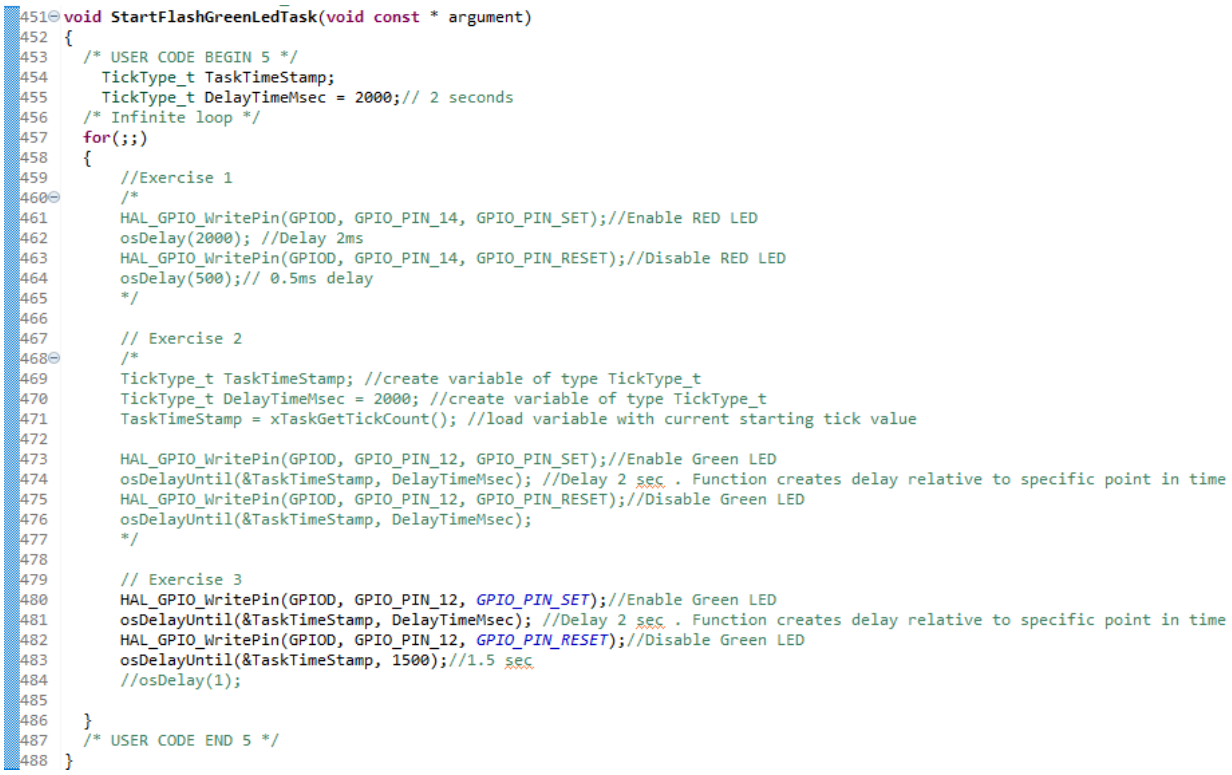
```c
void StartDefaultTask(void const * argument)
{
  /* USER CODE BEGIN 5 */

  /* Infinite loop */
  for(;;)
  {
      HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_SET);//Enable RED LED
      osDelay(2000); //Delay 2ms
      HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_RESET);//Disable RED LED
      osDelay(500);// 0.5ms delay
  }
  /* USER CODE END 5 */
}
```

Exercise 2)

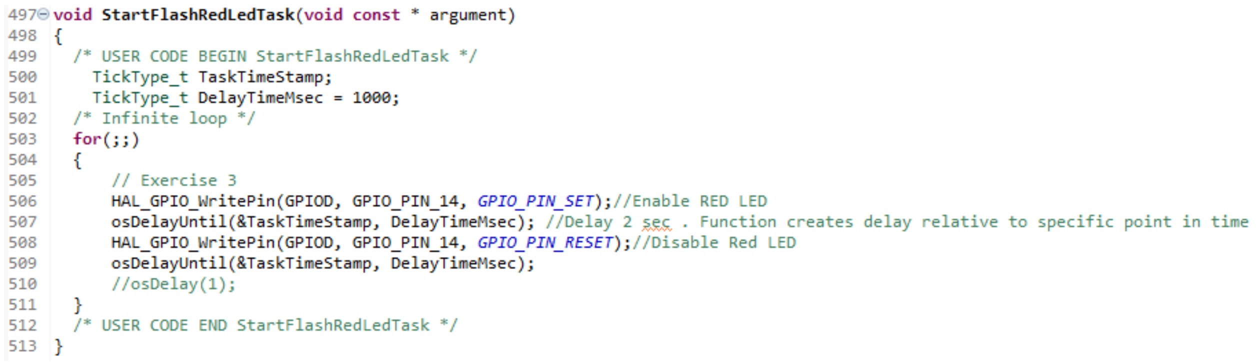


```c
      TickType_t TaskTimeStamp; //create variable of type TickType_t
      TickType_t DelayTimeMsec = 2000; //create variable of type TickType_t
      TaskTimeStamp = xTaskGetTickCount(); //load variable with current starting tick value

      HAL_GPIO_WritePin(GPIOD, GPIO_PIN_12, GPIO_PIN_SET);//Enable Green LED
      osDelayUntil(&TaskTimeStamp, DelayTimeMsec); //Delay 2 sec . Function creates delay relative to specific point in time
      HAL_GPIO_WritePin(GPIOD, GPIO_PIN_12, GPIO_PIN_RESET);//Disable Green LED
      osDelayUntil(&TaskTimeStamp, DelayTimeMsec);
```

Exercise 3)

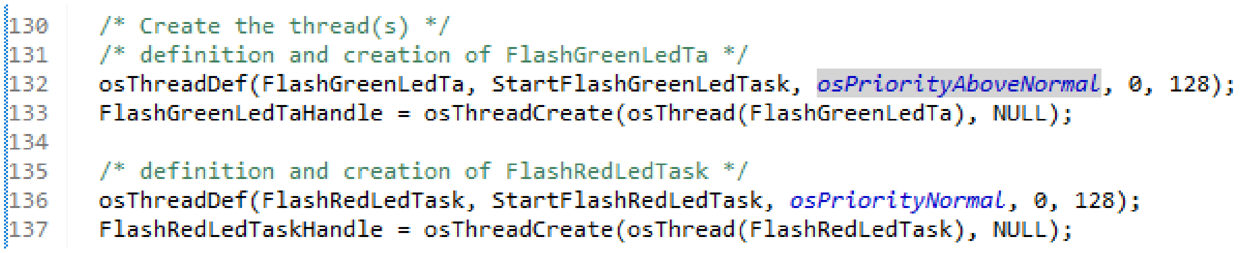


```c
void StartFlashGreenLedTask(void const * argument)
{
  /* USER CODE BEGIN 5 */
    TickType_t TaskTimeStamp;
    TickType_t DelayTimeMsec = 2000;// 2 seconds
  /* Infinite loop */
  for(;;)
  {
      //Exercise 1
      /*
      HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_SET);//Enable RED LED
      osDelay(2000); //Delay 2ms
      HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_RESET);//Disable RED LED
      osDelay(500);// 0.5ms delay
      */

      // Exercise 2
      /*
      TickType_t TaskTimeStamp; //create variable of type TickType_t
      TickType_t DelayTimeMsec = 2000; //create variable of type TickType_t
      TaskTimeStamp = xTaskGetTickCount(); //load variable with current starting tick value

      HAL_GPIO_WritePin(GPIOD, GPIO_PIN_12, GPIO_PIN_SET);//Enable Green LED
      osDelayUntil(&TaskTimeStamp, DelayTimeMsec); //Delay 2 sec . Function creates delay relative to specific point in time
      HAL_GPIO_WritePin(GPIOD, GPIO_PIN_12, GPIO_PIN_RESET);//Disable Green LED
      osDelayUntil(&TaskTimeStamp, DelayTimeMsec);
      */

      // Exercise 3
      HAL_GPIO_WritePin(GPIOD, GPIO_PIN_12, GPIO_PIN_SET);//Enable Green LED
      osDelayUntil(&TaskTimeStamp, DelayTimeMsec); //Delay 2 sec . Function creates delay relative to specific point in time
      HAL_GPIO_WritePin(GPIOD, GPIO_PIN_12, GPIO_PIN_RESET);//Disable Green LED
      osDelayUntil(&TaskTimeStamp, 1500);//1.5 sec
      //osDelay(1);

  }
  /* USER CODE END 5 */
}
```

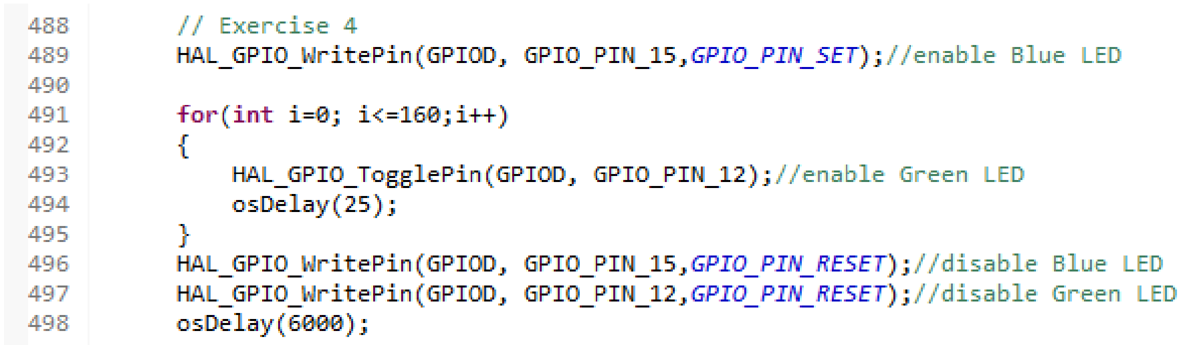


```
void StartFlashRedLedTask(void const * argument)
{
  /* USER CODE BEGIN StartFlashRedLedTask */
    TickType_t TaskTimeStamp;
    TickType_t DelayTimeMsec = 1000;
  /* Infinite loop */
  for(;;)
  {
      // Exercise 3
      HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_SET);//Enable RED LED
      osDelayUntil(&TaskTimeStamp, DelayTimeMsec); //Delay 2 sec . Function creates delay relative to specific point in time
      HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_RESET);//Disable Red LED
      osDelayUntil(&TaskTimeStamp, DelayTimeMsec);
      //osDelay(1);
  }
  /* USER CODE END StartFlashRedLedTask */
}
```

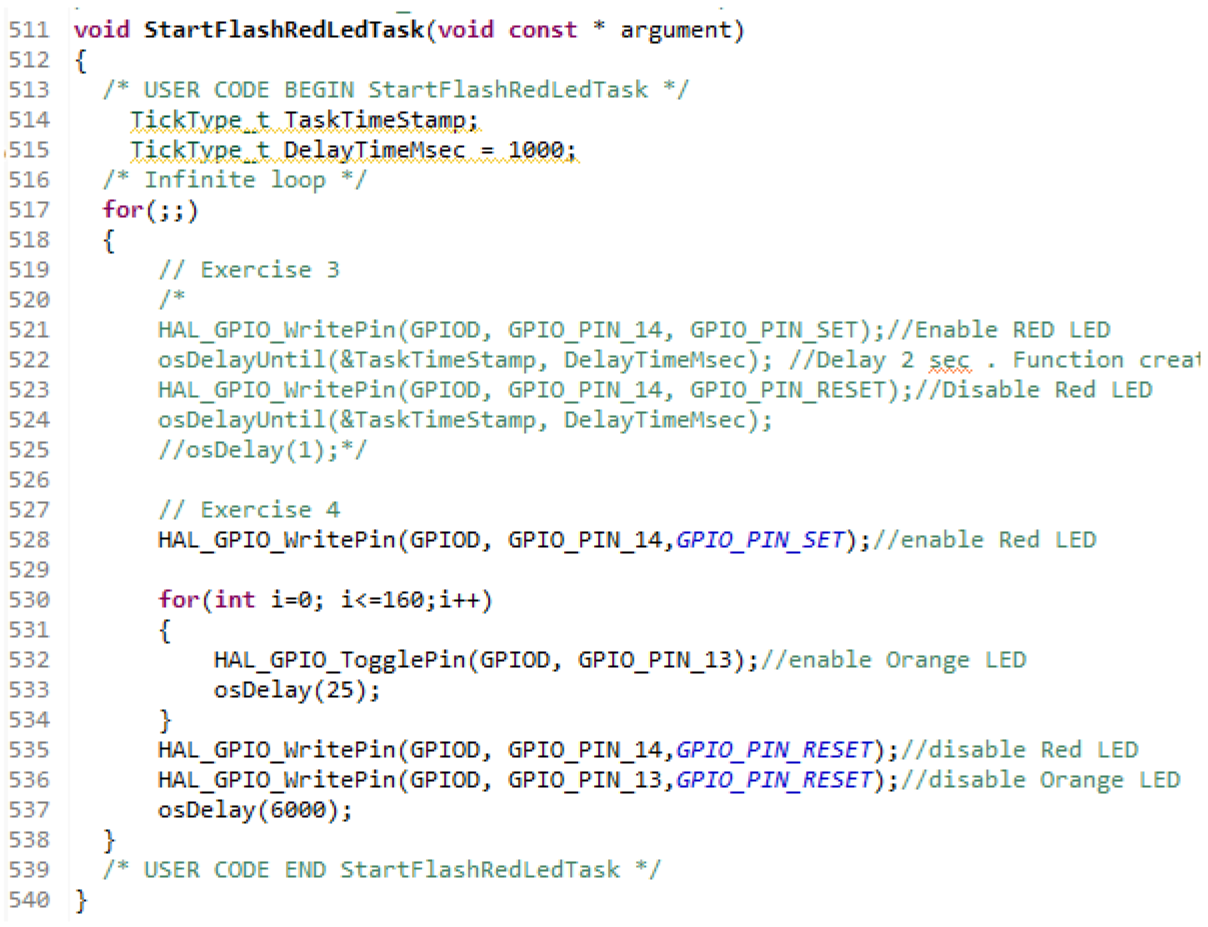
Exercise 4)



```
  /* Create the thread(s) */
  /* definition and creation of FlashGreenLedTa */
  osThreadDef(FlashGreenLedTa, StartFlashGreenLedTask, osPriorityAboveNormal, 0, 128);
  FlashGreenLedTaHandle = osThreadCreate(osThread(FlashGreenLedTa), NULL);

  /* definition and creation of FlashRedLedTask */
  osThreadDef(FlashRedLedTask, StartFlashRedLedTask, osPriorityNormal, 0, 128);
  FlashRedLedTaskHandle = osThreadCreate(osThread(FlashRedLedTask), NULL);
```



```
      // Exercise 4
      HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15,GPIO_PIN_SET);//enable Blue LED

      for(int i=0; i<=160;i++)
      {
          HAL_GPIO_TogglePin(GPIOD, GPIO_PIN_12);//enable Green LED
          osDelay(25);
      }
      HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15,GPIO_PIN_RESET);//disable Blue LED
      HAL_GPIO_WritePin(GPIOD, GPIO_PIN_12,GPIO_PIN_RESET);//disable Green LED
      osDelay(6000);
```



```c
void StartFlashRedLedTask(void const * argument)
{
  /* USER CODE BEGIN StartFlashRedLedTask */
    TickType_t TaskTimeStamp;
    TickType_t DelayTimeMsec = 1000;
  /* Infinite loop */
  for(;;)
  {
      // Exercise 3
      /*
      HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_SET);//Enable RED LED
      osDelayUntil(&TaskTimeStamp, DelayTimeMsec); //Delay 2 sec . Function creat
      HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_RESET);//Disable Red LED
      osDelayUntil(&TaskTimeStamp, DelayTimeMsec);
      //osDelay(1);*/

      // Exercise 4
      HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14,GPIO_PIN_SET);//enable Red LED

      for(int i=0; i<=160;i++)
      {
          HAL_GPIO_TogglePin(GPIOD, GPIO_PIN_13);//enable Orange LED
          osDelay(25);
      }
      HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14,GPIO_PIN_RESET);//disable Red LED
      HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13,GPIO_PIN_RESET);//disable Orange LED
      osDelay(6000);
  }
  /* USER CODE END StartFlashRedLedTask */
}
```