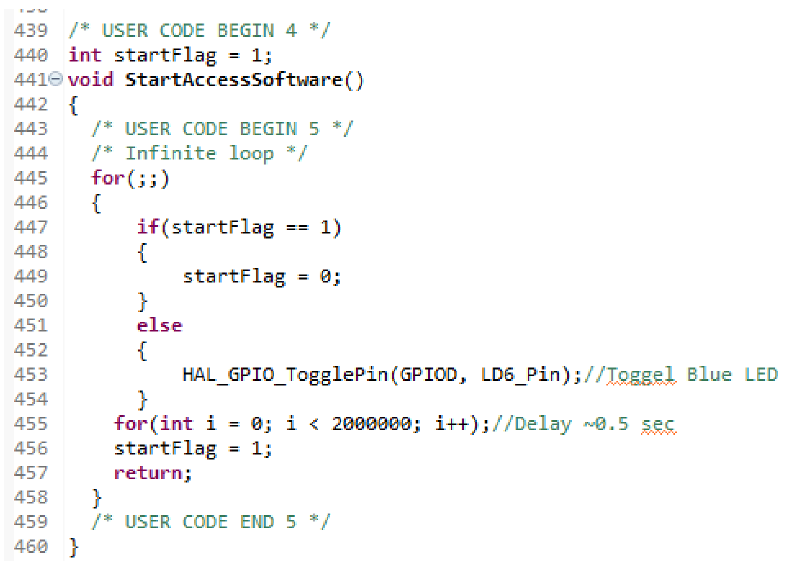
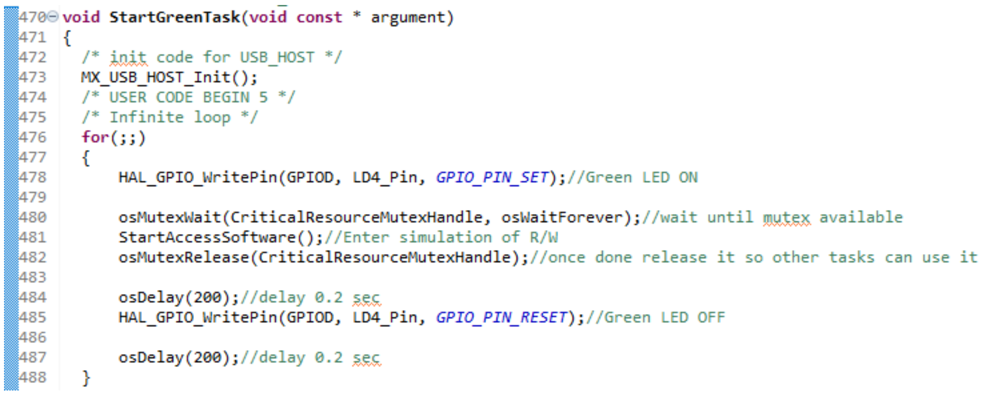
Farhan Ali   Experiment 4

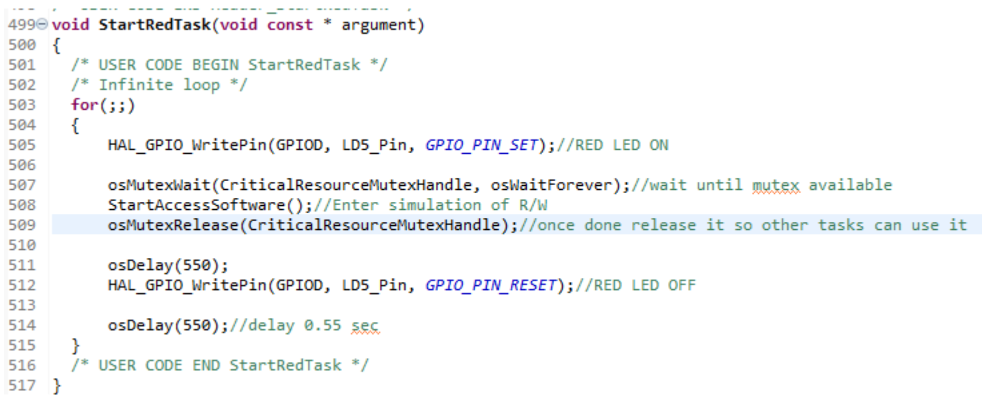
Exercise 9:



```
/* USER CODE BEGIN 4 */
int startFlag = 1;
void StartAccessSoftware()
{
  /* USER CODE BEGIN 5 */
  /* Infinite loop */
  for(;;)
  {
      if(startFlag == 1)
      {
          startFlag = 0;
      }
      else
      {
          HAL_GPIO_TogglePin(GPIOD, LD6_Pin);//Toggel Blue LED
      }
    for(int i = 0; i < 2000000; i++);//Delay ~0.5 sec
    startFlag = 1;
    return;
  }
  /* USER CODE END 5 */
}
```



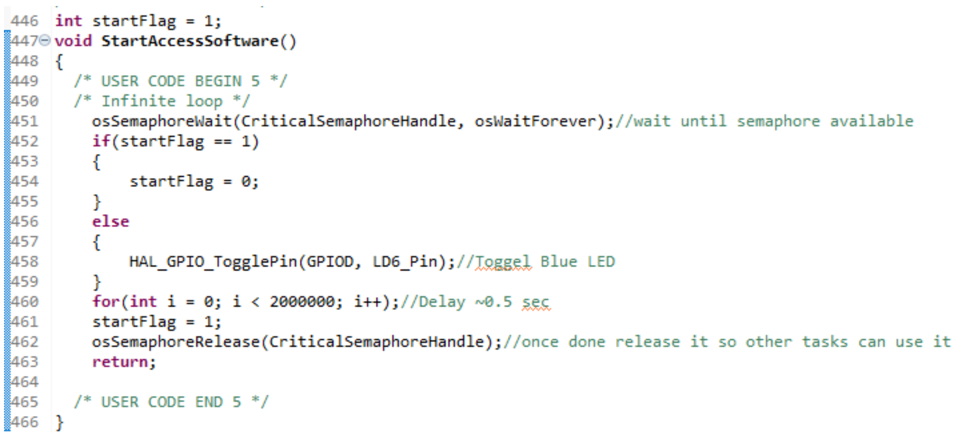
```
void StartGreenTask(void const * argument)
{
  /* init code for USB_HOST */
  MX_USB_HOST_Init();
  /* USER CODE BEGIN 5 */
  /* Infinite loop */
  for(;;)
  {
      HAL_GPIO_WritePin(GPIOD, LD4_Pin, GPIO_PIN_SET);//Green LED ON

      osMutexWait(CriticalResourceMutexHandle, osWaitForever);//wait until mutex available
      StartAccessSoftware();//Enter simulation of R/W
      osMutexRelease(CriticalResourceMutexHandle);//once done release it so other tasks can use it

      osDelay(200);//delay 0.2 sec
      HAL_GPIO_WritePin(GPIOD, LD4_Pin, GPIO_PIN_RESET);//Green LED OFF

      osDelay(200);//delay 0.2 sec
  }
```



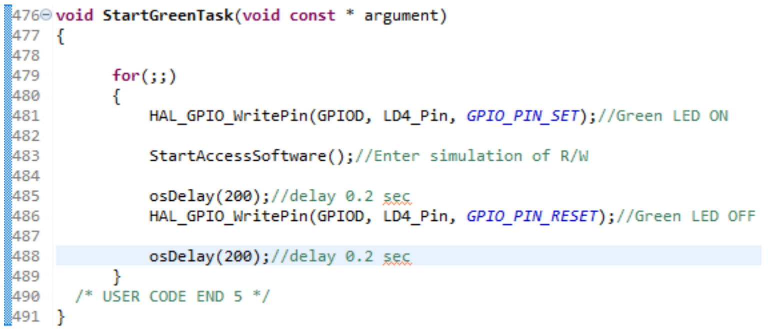
```
void StartRedTask(void const * argument)
{
  /* USER CODE BEGIN StartRedTask */
  /* Infinite loop */
  for(;;)
  {
      HAL_GPIO_WritePin(GPIOD, LD5_Pin, GPIO_PIN_SET);//RED LED ON

      osMutexWait(CriticalResourceMutexHandle, osWaitForever);//wait until mutex available
      StartAccessSoftware();//Enter simulation of R/W
      osMutexRelease(CriticalResourceMutexHandle);//once done release it so other tasks can use it

      osDelay(550);
      HAL_GPIO_WritePin(GPIOD, LD5_Pin, GPIO_PIN_RESET);//RED LED OFF

      osDelay(550);//delay 0.55 sec
  }
  /* USER CODE END StartRedTask */
}
```

Farhan Ali Experiment 4

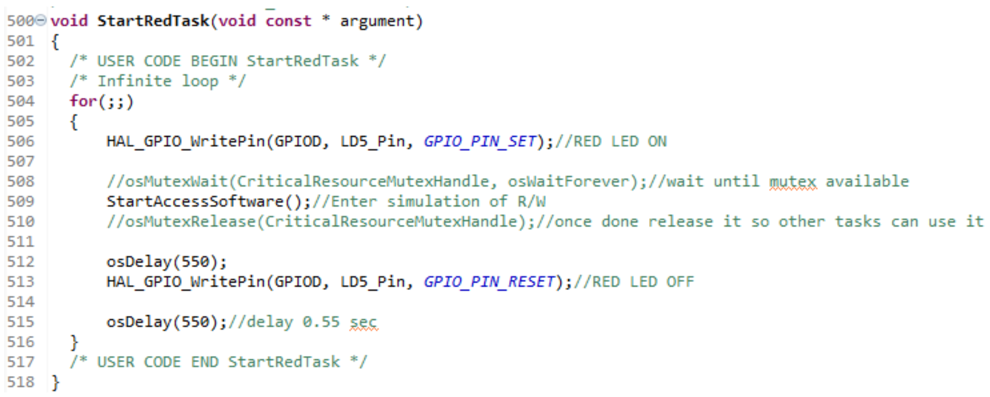
Exercise 10:



```c
int startFlag = 1;
void StartAccessSoftware()
{
  /* USER CODE BEGIN 5 */
  /* Infinite loop */
    osSemaphoreWait(CriticalSemaphoreHandle, osWaitForever);//wait until semaphore available
    if(startFlag == 1)
    {
        startFlag = 0;
    }
    else
    {
        HAL_GPIO_TogglePin(GPIOD, LD6_Pin);//Toggel Blue LED
    }
    for(int i = 0; i < 2000000; i++);//Delay ~0.5 sec
    startFlag = 1;
    osSemaphoreRelease(CriticalSemaphoreHandle);//once done release it so other tasks can use it
    return;

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



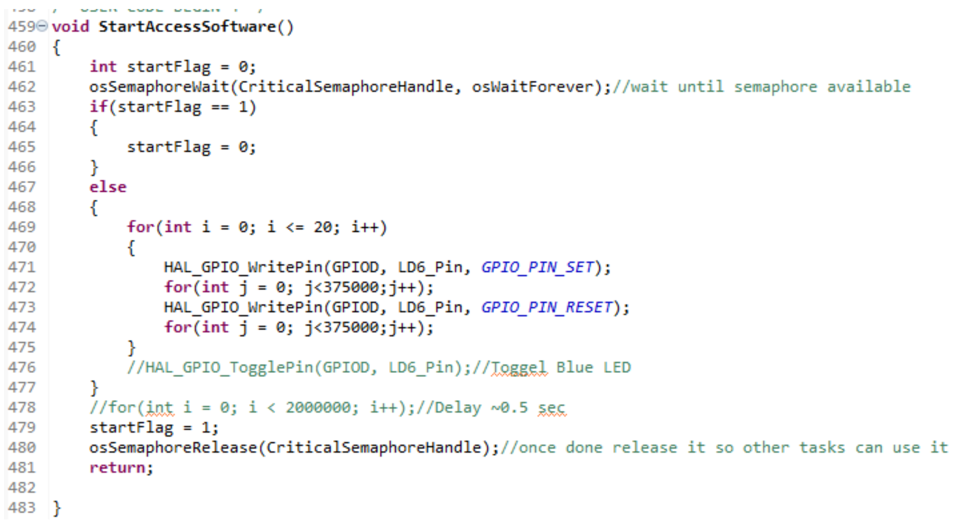
```c
void StartGreenTask(void const * argument)
{

      for(;;)
      {
          HAL_GPIO_WritePin(GPIOD, LD4_Pin, GPIO_PIN_SET);//Green LED ON

          StartAccessSoftware();//Enter simulation of R/W

          osDelay(200);//delay 0.2 sec
          HAL_GPIO_WritePin(GPIOD, LD4_Pin, GPIO_PIN_RESET);//Green LED OFF

          osDelay(200);//delay 0.2 sec
      }
  /* USER CODE END 5 */
}
```



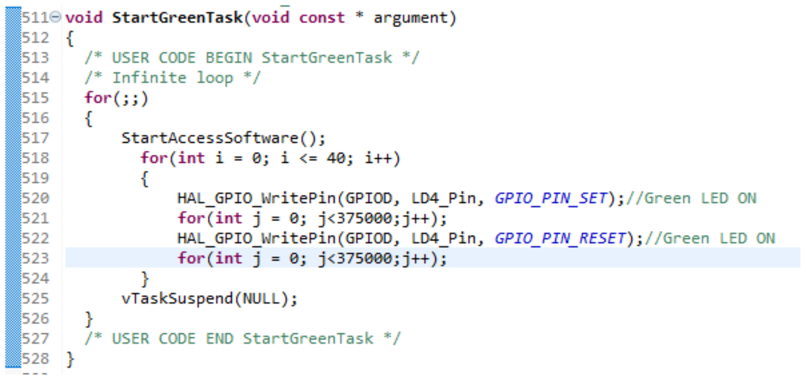
```c
void StartRedTask(void const * argument)
{
  /* USER CODE BEGIN StartRedTask */
  /* Infinite loop */
  for(;;)
  {
      HAL_GPIO_WritePin(GPIOD, LD5_Pin, GPIO_PIN_SET);//RED LED ON

      //osMutexWait(CriticalResourceMutexHandle, osWaitForever);//wait until mutex available
      StartAccessSoftware();//Enter simulation of R/W
      //osMutexRelease(CriticalResourceMutexHandle);//once done release it so other tasks can use it

      osDelay(550);
      HAL_GPIO_WritePin(GPIOD, LD5_Pin, GPIO_PIN_RESET);//RED LED OFF

      osDelay(550);//delay 0.55 sec
  }
  /* USER CODE END StartRedTask */
}
```

Farhan Ali Experiment 4

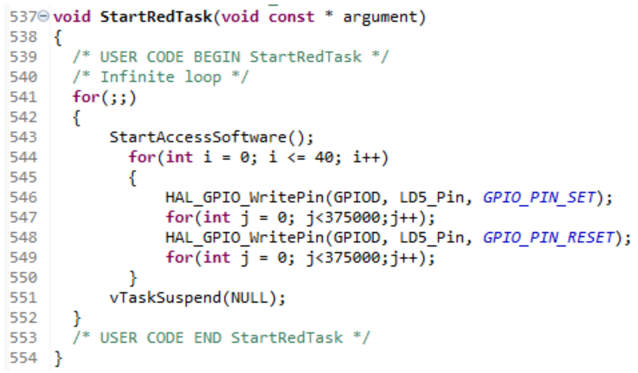
Exercise 11.1)



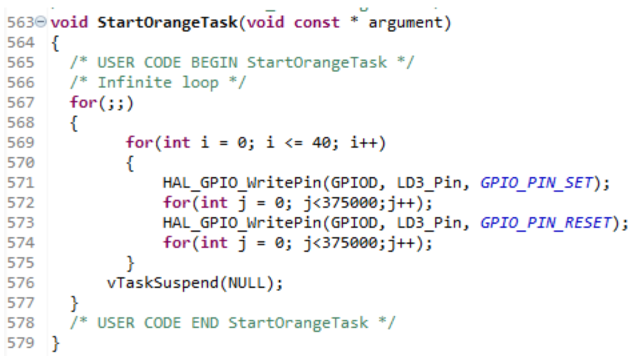
```c
void StartAccessSoftware()
{
    int startFlag = 0;
    osSemaphoreWait(CriticalSemaphoreHandle, osWaitForever);//wait until semaphore available
    if(startFlag == 1)
    {
        startFlag = 0;
    }
    else
    {
        for(int i = 0; i <= 20; i++)
        {
            HAL_GPIO_WritePin(GPIOD, LD6_Pin, GPIO_PIN_SET);
            for(int j = 0; j<375000;j++);
            HAL_GPIO_WritePin(GPIOD, LD6_Pin, GPIO_PIN_RESET);
            for(int j = 0; j<375000;j++);
        }
        //HAL_GPIO_TogglePin(GPIOD, LD6_Pin);//Toggel Blue LED
    }
    //for(int i = 0; i < 2000000; i++);//Delay ~0.5 sec
    startFlag = 1;
    osSemaphoreRelease(CriticalSemaphoreHandle);//once done release it so other tasks can use it
    return;

}
```



```c
void StartGreenTask(void const * argument)
{
  /* USER CODE BEGIN StartGreenTask */
  /* Infinite loop */
  for(;;)
  {
      StartAccessSoftware();
        for(int i = 0; i <= 40; i++)
        {
            HAL_GPIO_WritePin(GPIOD, LD4_Pin, GPIO_PIN_SET);//Green LED ON
            for(int j = 0; j<375000;j++);
            HAL_GPIO_WritePin(GPIOD, LD4_Pin, GPIO_PIN_RESET);//Green LED ON
            for(int j = 0; j<375000;j++);
        }
      vTaskSuspend(NULL);
  }
  /* USER CODE END StartGreenTask */
}
```

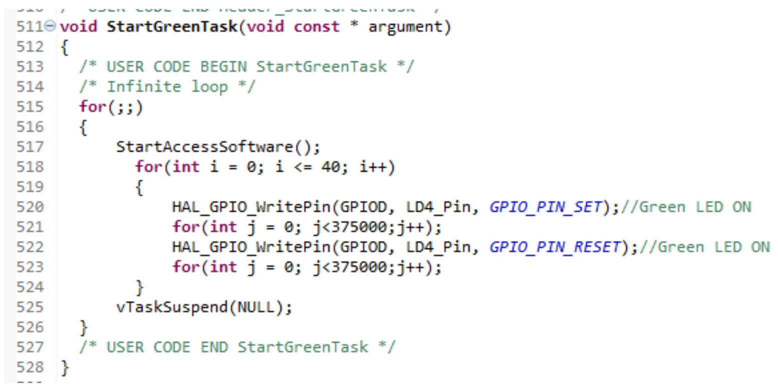


```c
void StartRedTask(void const * argument)
{
  /* USER CODE BEGIN StartRedTask */
  /* Infinite loop */
  for(;;)
  {
      StartAccessSoftware();
        for(int i = 0; i <= 40; i++)
        {
            HAL_GPIO_WritePin(GPIOD, LD5_Pin, GPIO_PIN_SET);
            for(int j = 0; j<375000;j++);
            HAL_GPIO_WritePin(GPIOD, LD5_Pin, GPIO_PIN_RESET);
            for(int j = 0; j<375000;j++);
        }
      vTaskSuspend(NULL);
  }
  /* USER CODE END StartRedTask */
}
```

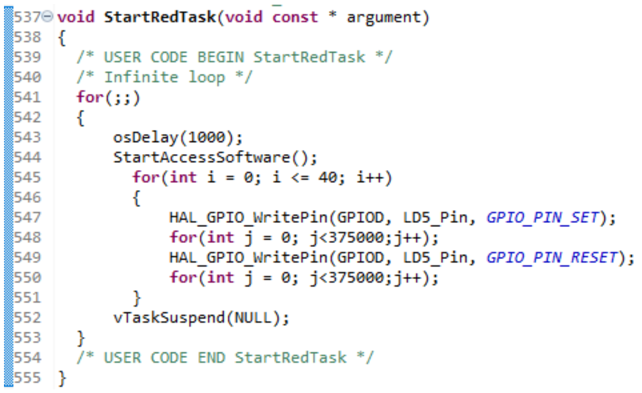


```c
void StartOrangeTask(void const * argument)
{
  /* USER CODE BEGIN StartOrangeTask */
  /* Infinite loop */
  for(;;)
  {
        for(int i = 0; i <= 40; i++)
        {
            HAL_GPIO_WritePin(GPIOD, LD3_Pin, GPIO_PIN_SET);
            for(int j = 0; j<375000;j++);
            HAL_GPIO_WritePin(GPIOD, LD3_Pin, GPIO_PIN_RESET);
            for(int j = 0; j<375000;j++);
        }
      vTaskSuspend(NULL);
  }
  /* USER CODE END StartOrangeTask */
}
```

11.2)

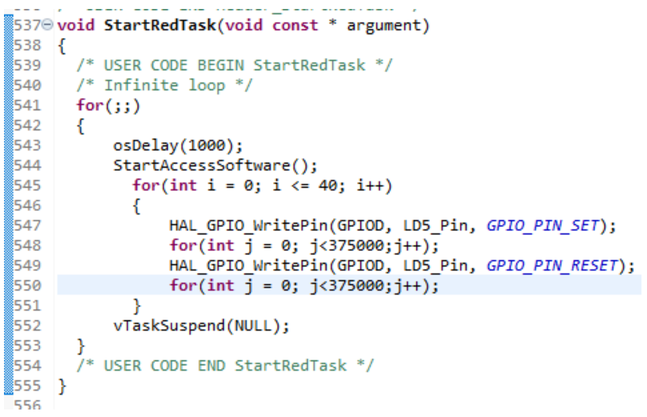


```
void StartGreenTask(void const * argument)
{
  /* USER CODE BEGIN StartGreenTask */
  /* Infinite loop */
  for(;;)
  {
      StartAccessSoftware();
        for(int i = 0; i <= 40; i++)
        {
            HAL_GPIO_WritePin(GPIOD, LD4_Pin, GPIO_PIN_SET);//Green LED ON
            for(int j = 0; j<375000;j++);
            HAL_GPIO_WritePin(GPIOD, LD4_Pin, GPIO_PIN_RESET);//Green LED ON
            for(int j = 0; j<375000;j++);
        }
      vTaskSuspend(NULL);
  }
  /* USER CODE END StartGreenTask */
}
```

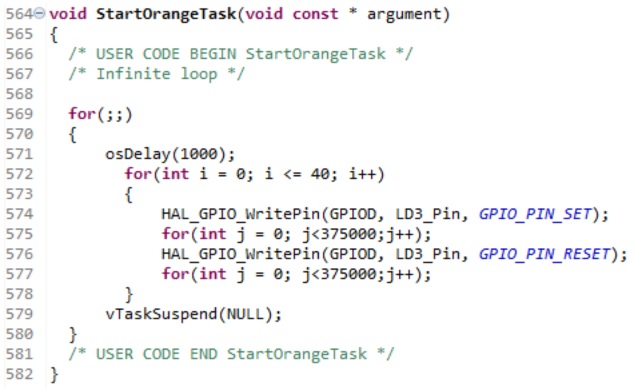


```
void StartRedTask(void const * argument)
{
  /* USER CODE BEGIN StartRedTask */
  /* Infinite loop */
  for(;;)
  {
      osDelay(1000);
      StartAccessSoftware();
        for(int i = 0; i <= 40; i++)
        {
            HAL_GPIO_WritePin(GPIOD, LD5_Pin, GPIO_PIN_SET);
            for(int j = 0; j<375000;j++);
            HAL_GPIO_WritePin(GPIOD, LD5_Pin, GPIO_PIN_RESET);
            for(int j = 0; j<375000;j++);
        }
      vTaskSuspend(NULL);
  }
  /* USER CODE END StartRedTask */
}
```

11.3)



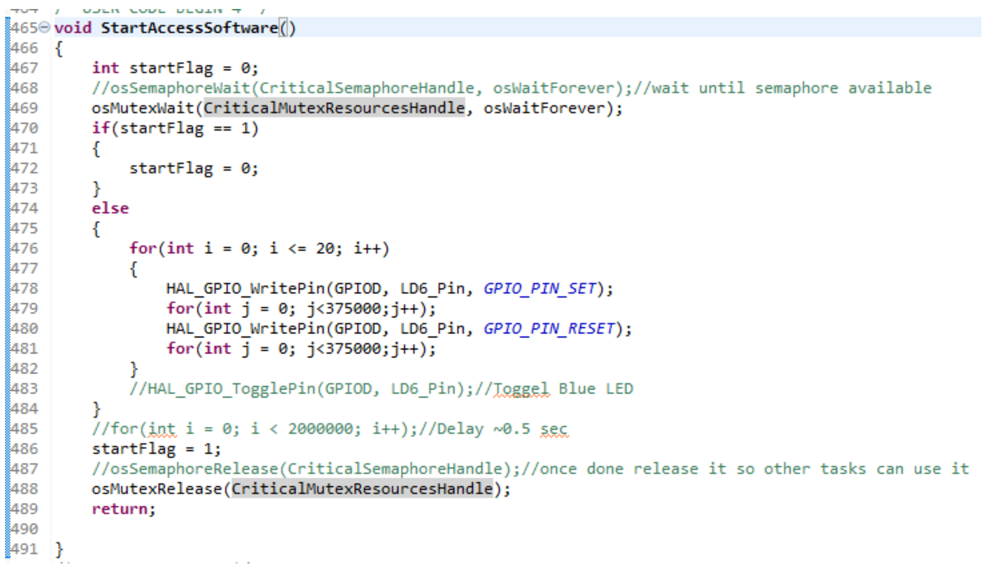
```
void StartRedTask(void const * argument)
{
  /* USER CODE BEGIN StartRedTask */
  /* Infinite loop */
  for(;;)
  {
      osDelay(1000);
      StartAccessSoftware();
        for(int i = 0; i <= 40; i++)
        {
            HAL_GPIO_WritePin(GPIOD, LD5_Pin, GPIO_PIN_SET);
            for(int j = 0; j<375000;j++);
            HAL_GPIO_WritePin(GPIOD, LD5_Pin, GPIO_PIN_RESET);
            for(int j = 0; j<375000;j++);
        }
      vTaskSuspend(NULL);
  }
  /* USER CODE END StartRedTask */
}
```



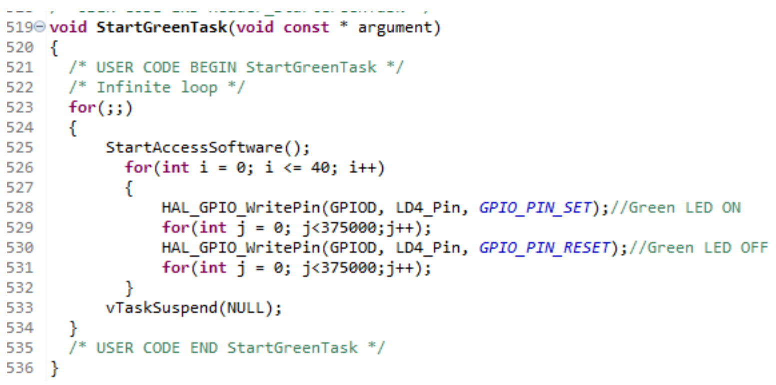
```
void StartOrangeTask(void const * argument)
{
  /* USER CODE BEGIN StartOrangeTask */
  /* Infinite loop */

  for(;;)
  {
      osDelay(1000);
        for(int i = 0; i <= 40; i++)
        {
            HAL_GPIO_WritePin(GPIOD, LD3_Pin, GPIO_PIN_SET);
            for(int j = 0; j<375000;j++);
            HAL_GPIO_WritePin(GPIOD, LD3_Pin, GPIO_PIN_RESET);
            for(int j = 0; j<375000;j++);
        }
      vTaskSuspend(NULL);
  }
  /* USER CODE END StartOrangeTask */
}
```

Farhan Ali Experiment 4

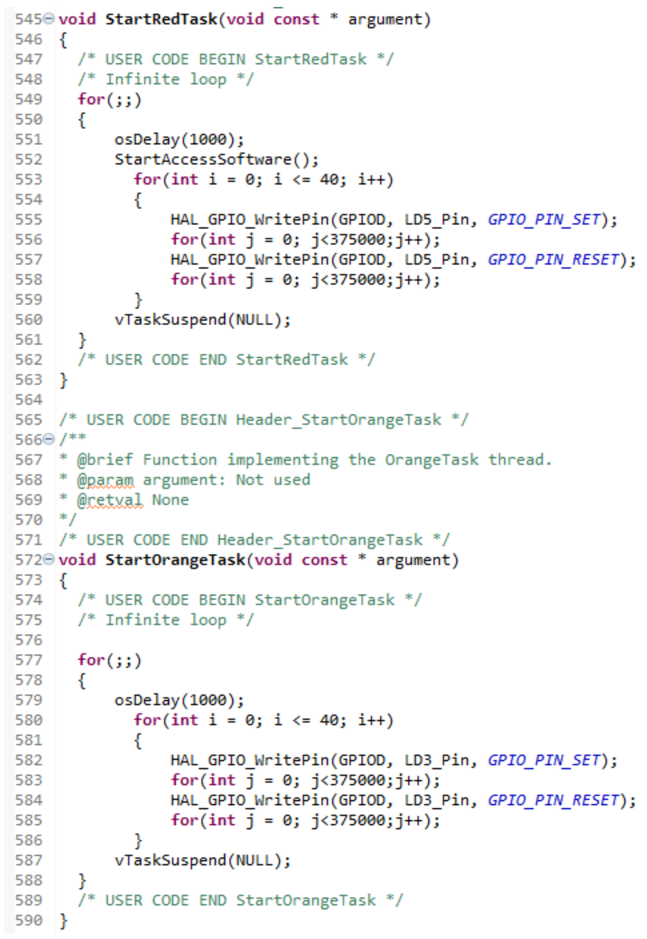
Exercise 12)



```
void StartAccessSoftware()
{
    int startFlag = 0;
    //osSemaphoreWait(CriticalSemaphoreHandle, osWaitForever);//wait until semaphore available
    osMutexWait(CriticalMutexResourcesHandle, osWaitForever);
    if(startFlag == 1)
    {
        startFlag = 0;
    }
    else
    {
        for(int i = 0; i <= 20; i++)
        {
            HAL_GPIO_WritePin(GPIOD, LD6_Pin, GPIO_PIN_SET);
            for(int j = 0; j<375000;j++);
            HAL_GPIO_WritePin(GPIOD, LD6_Pin, GPIO_PIN_RESET);
            for(int j = 0; j<375000;j++);
        }
        //HAL_GPIO_TogglePin(GPIOD, LD6_Pin);//Toggel Blue LED
    }
    //for(int i = 0; i < 2000000; i++);//Delay ~0.5 sec
    startFlag = 1;
    //osSemaphoreRelease(CriticalSemaphoreHandle);//once done release it so other tasks can use it
    osMutexRelease(CriticalMutexResourcesHandle);
    return;

}
```



```
void StartGreenTask(void const * argument)
{
  /* USER CODE BEGIN StartGreenTask */
  /* Infinite loop */
  for(;;)
  {
      StartAccessSoftware();
        for(int i = 0; i <= 40; i++)
        {
            HAL_GPIO_WritePin(GPIOD, LD4_Pin, GPIO_PIN_SET);//Green LED ON
            for(int j = 0; j<375000;j++);
            HAL_GPIO_WritePin(GPIOD, LD4_Pin, GPIO_PIN_RESET);//Green LED OFF
            for(int j = 0; j<375000;j++);
        }
      vTaskSuspend(NULL);
  }
  /* USER CODE END StartGreenTask */
}
```



```c
void StartRedTask(void const * argument)
{
  /* USER CODE BEGIN StartRedTask */
  /* Infinite loop */
  for(;;)
  {
      osDelay(1000);
      StartAccessSoftware();
        for(int i = 0; i <= 40; i++)
        {
            HAL_GPIO_WritePin(GPIOD, LD5_Pin, GPIO_PIN_SET);
            for(int j = 0; j<375000;j++);
            HAL_GPIO_WritePin(GPIOD, LD5_Pin, GPIO_PIN_RESET);
            for(int j = 0; j<375000;j++);
        }
      vTaskSuspend(NULL);
  }
  /* USER CODE END StartRedTask */
}

/* USER CODE BEGIN Header_StartOrangeTask */
/**
* @brief Function implementing the OrangeTask thread.
* @param argument: Not used
* @retval None
*/
/* USER CODE END Header_StartOrangeTask */
void StartOrangeTask(void const * argument)
{
  /* USER CODE BEGIN StartOrangeTask */
  /* Infinite loop */

  for(;;)
  {
      osDelay(1000);
        for(int i = 0; i <= 40; i++)
        {
            HAL_GPIO_WritePin(GPIOD, LD3_Pin, GPIO_PIN_SET);
            for(int j = 0; j<375000;j++);
            HAL_GPIO_WritePin(GPIOD, LD3_Pin, GPIO_PIN_RESET);
            for(int j = 0; j<375000;j++);
        }
      vTaskSuspend(NULL);
  }
  /* USER CODE END StartOrangeTask */
}
```