Farhan Ali Set 5

Exercise 13)

$f = \frac{1}{T}$

$\underline{10\,Hz}$

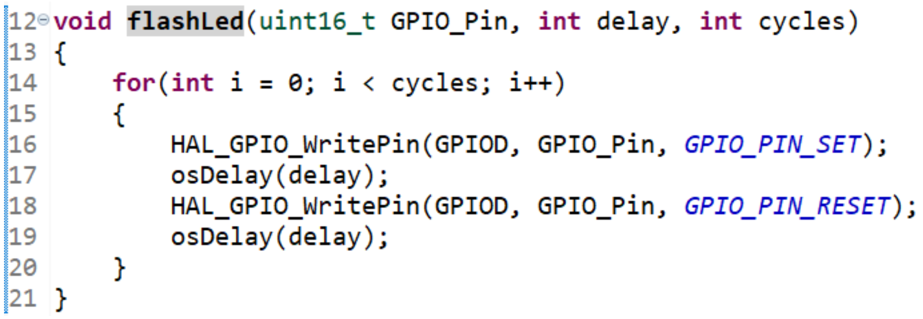
$10Hz = \frac{1}{T}$

$Cycles = f \cdot duration$

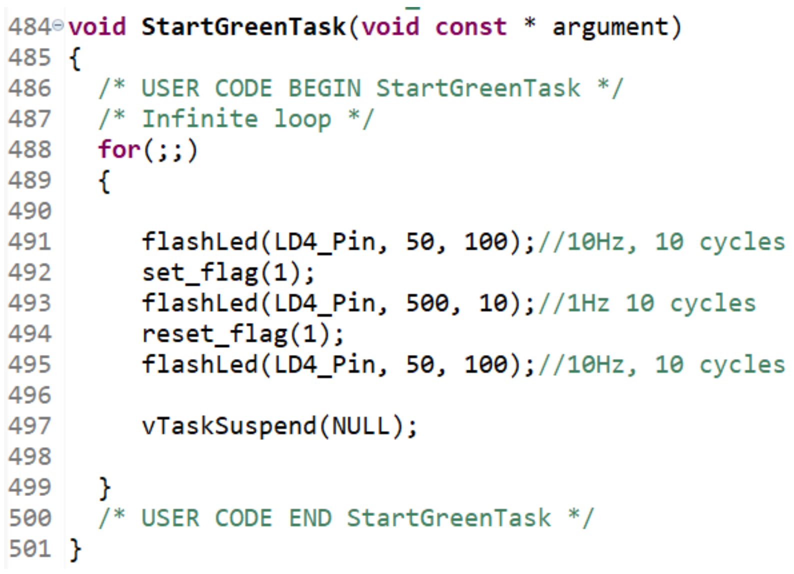
$100 = 10Hz \cdot 10$

$T = \frac{1}{10Hz} = 0.1$

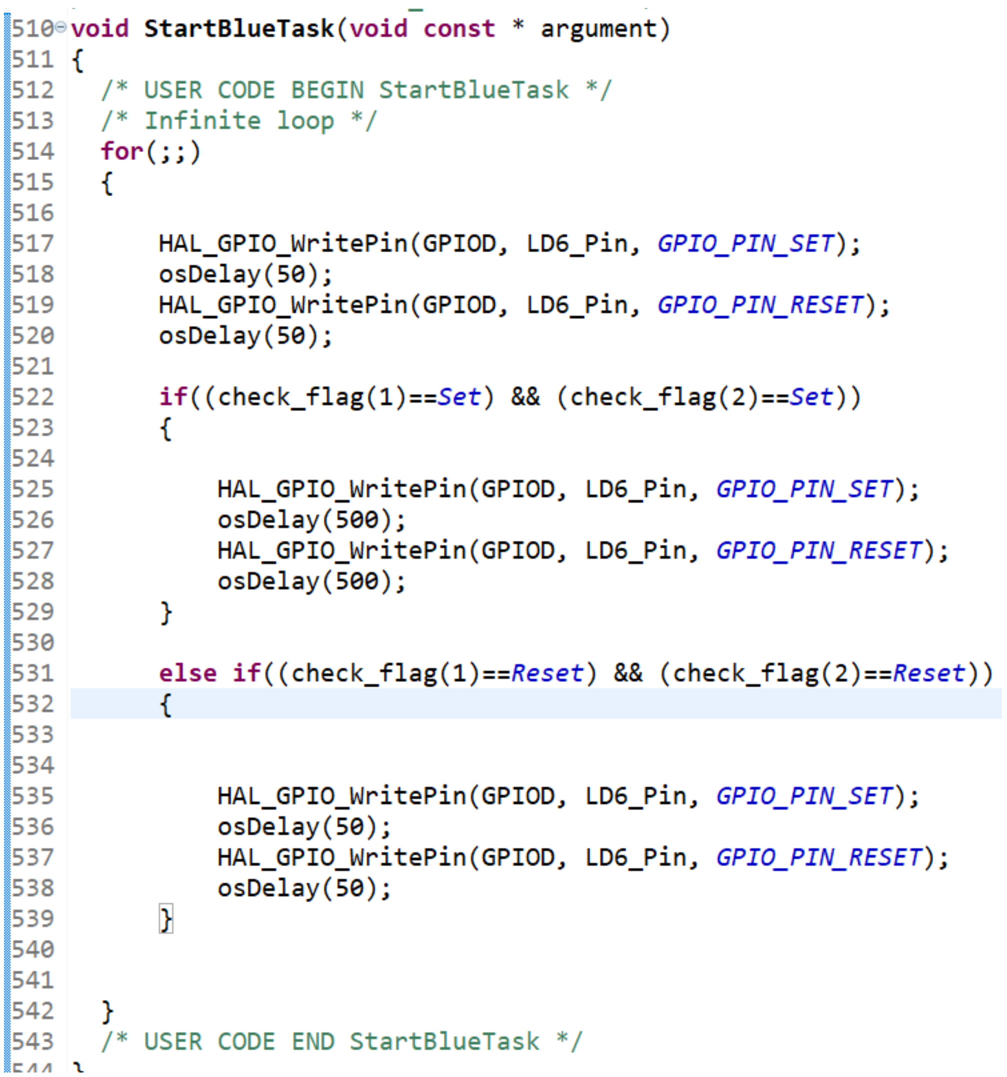
$\frac{0.1}{2} = 0.05 sec = 50m$



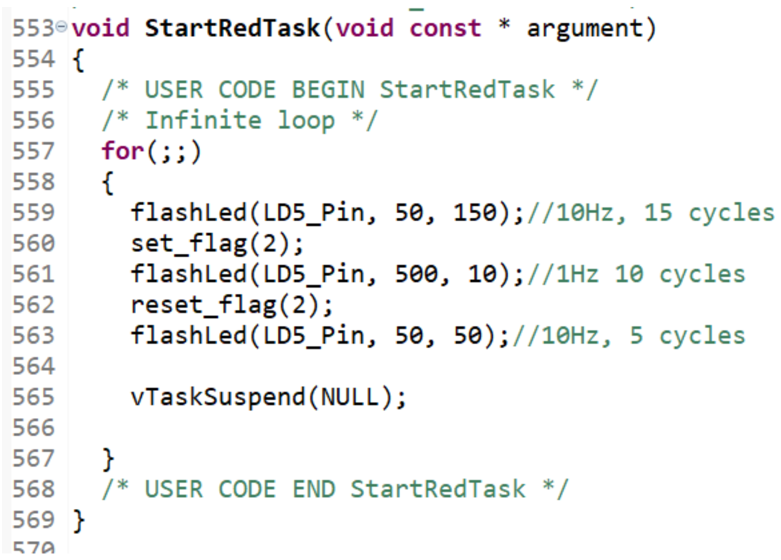
```
void flashLed(uint16_t GPIO_Pin, int delay, int cycles)
{
    for(int i = 0; i < cycles; i++)
    {
        HAL_GPIO_WritePin(GPIOD, GPIO_Pin, GPIO_PIN_SET);
        osDelay(delay);
        HAL_GPIO_WritePin(GPIOD, GPIO_Pin, GPIO_PIN_RESET);
        osDelay(delay);
    }
}
```



```c
void StartGreenTask(void const * argument)
{
  /* USER CODE BEGIN StartGreenTask */
  /* Infinite loop */
  for(;;)
  {

     flashLed(LD4_Pin, 50, 100);//10Hz, 10 cycles
     set_flag(1);
     flashLed(LD4_Pin, 500, 10);//1Hz 10 cycles
     reset_flag(1);
     flashLed(LD4_Pin, 50, 100);//10Hz, 10 cycles

     vTaskSuspend(NULL);

  }
  /* USER CODE END StartGreenTask */
}
```

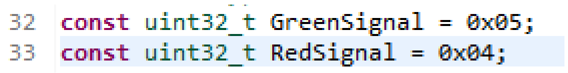


```c
void StartBlueTask(void const * argument)
{
  /* USER CODE BEGIN StartBlueTask */
  /* Infinite loop */
  for(;;)
  {

      HAL_GPIO_WritePin(GPIOD, LD6_Pin, GPIO_PIN_SET);
      osDelay(50);
      HAL_GPIO_WritePin(GPIOD, LD6_Pin, GPIO_PIN_RESET);
      osDelay(50);

      if((check_flag(1)==Set) && (check_flag(2)==Set))
      {

          HAL_GPIO_WritePin(GPIOD, LD6_Pin, GPIO_PIN_SET);
          osDelay(500);
          HAL_GPIO_WritePin(GPIOD, LD6_Pin, GPIO_PIN_RESET);
          osDelay(500);
      }

      else if((check_flag(1)==Reset) && (check_flag(2)==Reset))
      {


          HAL_GPIO_WritePin(GPIOD, LD6_Pin, GPIO_PIN_SET);
          osDelay(50);
          HAL_GPIO_WritePin(GPIOD, LD6_Pin, GPIO_PIN_RESET);
          osDelay(50);
      }


  }
  /* USER CODE END StartBlueTask */
}
```

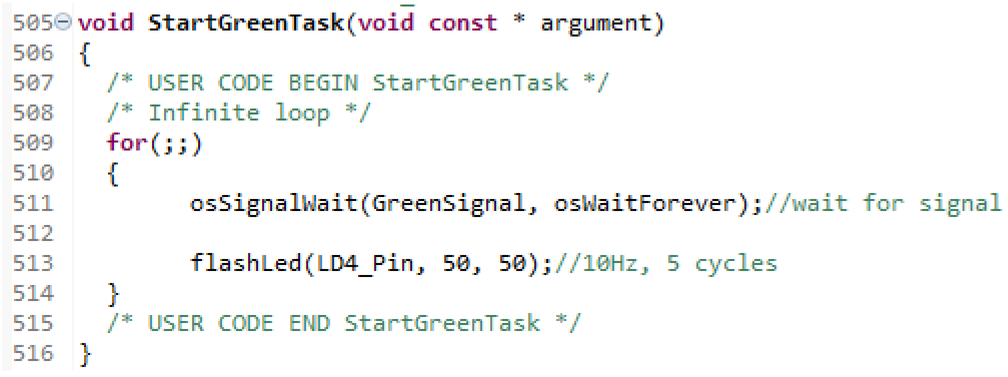


```c
void StartRedTask(void const * argument)
{
  /* USER CODE BEGIN StartRedTask */
  /* Infinite loop */
  for(;;)
  {
    flashLed(LD5_Pin, 50, 150);//10Hz, 15 cycles
    set_flag(2);
    flashLed(LD5_Pin, 500, 10);//1Hz 10 cycles
    reset_flag(2);
    flashLed(LD5_Pin, 50, 50);//10Hz, 5 cycles

    vTaskSuspend(NULL);

  }
  /* USER CODE END StartRedTask */
}
```

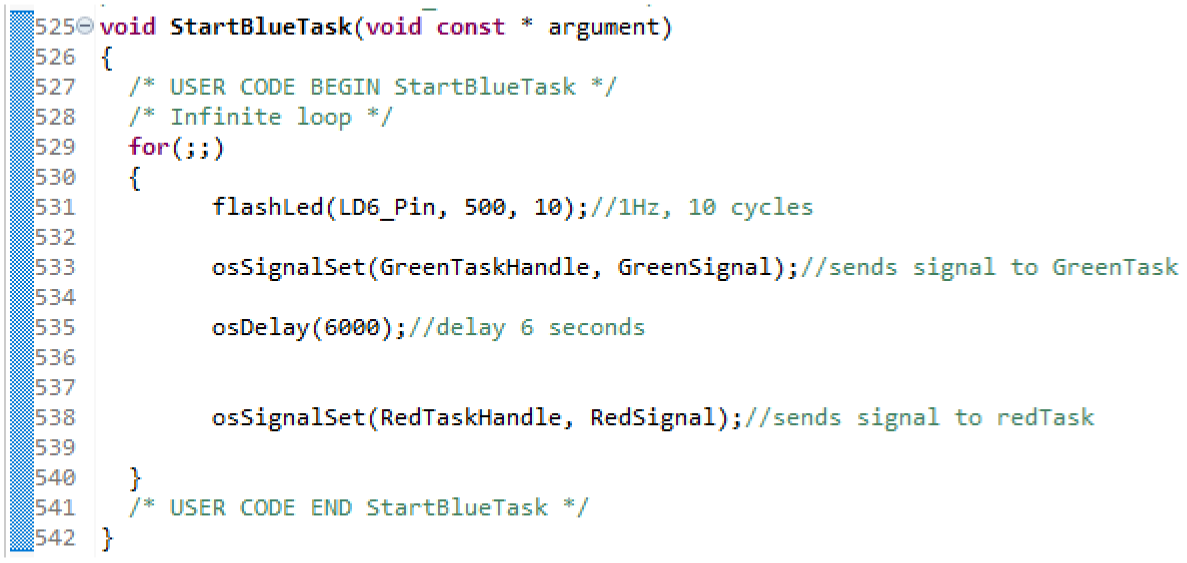
Exercise 14)



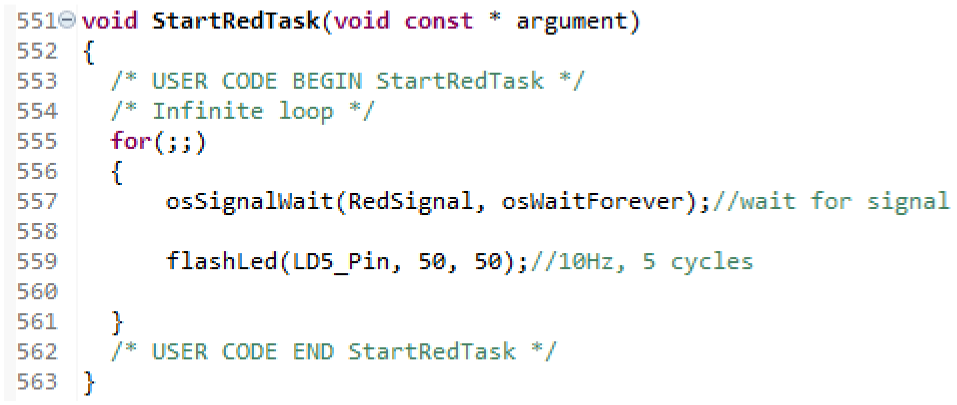
```c
const uint32_t GreenSignal = 0x05;
const uint32_t RedSignal = 0x04;
```



```c
void StartGreenTask(void const * argument)
{
  /* USER CODE BEGIN StartGreenTask */
  /* Infinite loop */
  for(;;)
  {
        osSignalWait(GreenSignal, osWaitForever);//wait for signal

        flashLed(LD4_Pin, 50, 50);//10Hz, 5 cycles
  }
  /* USER CODE END StartGreenTask */
}
```

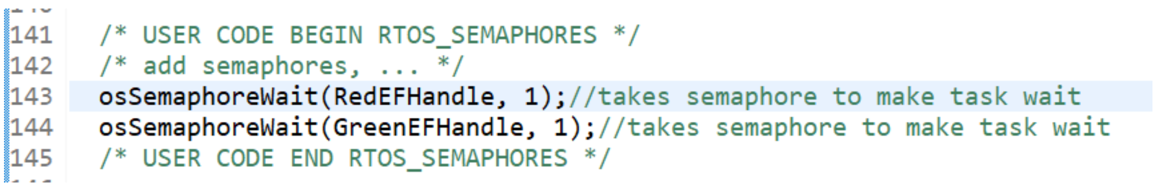


```c
void StartBlueTask(void const * argument)
{
  /* USER CODE BEGIN StartBlueTask */
  /* Infinite loop */
  for(;;)
  {
        flashLed(LD6_Pin, 500, 10);//1Hz, 10 cycles

        osSignalSet(GreenTaskHandle, GreenSignal);//sends signal to GreenTask

        osDelay(6000);//delay 6 seconds


        osSignalSet(RedTaskHandle, RedSignal);//sends signal to redTask

  }
  /* USER CODE END StartBlueTask */
}
```

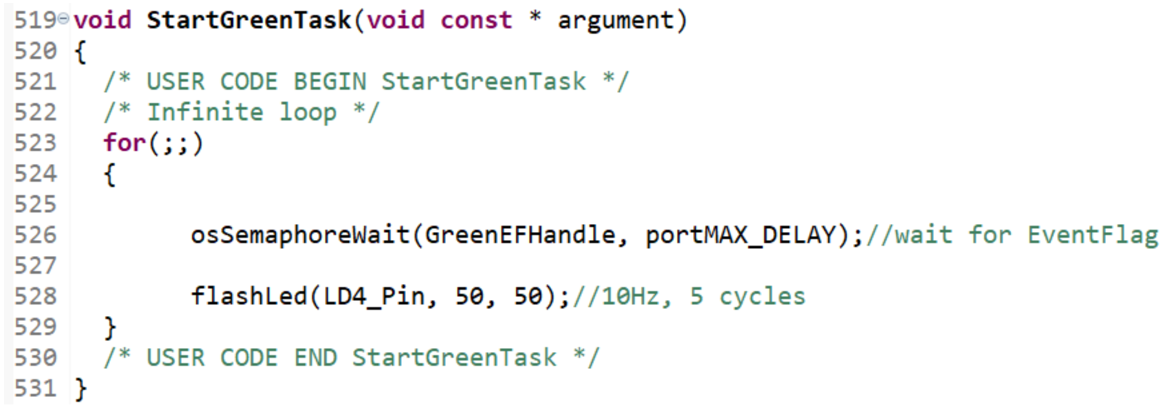


```
void StartRedTask(void const * argument)
{
  /* USER CODE BEGIN StartRedTask */
  /* Infinite loop */
  for(;;)
  {
      osSignalWait(RedSignal, osWaitForever);//wait for signal

      flashLed(LD5_Pin, 50, 50);//10Hz, 5 cycles

  }
  /* USER CODE END StartRedTask */
}
```

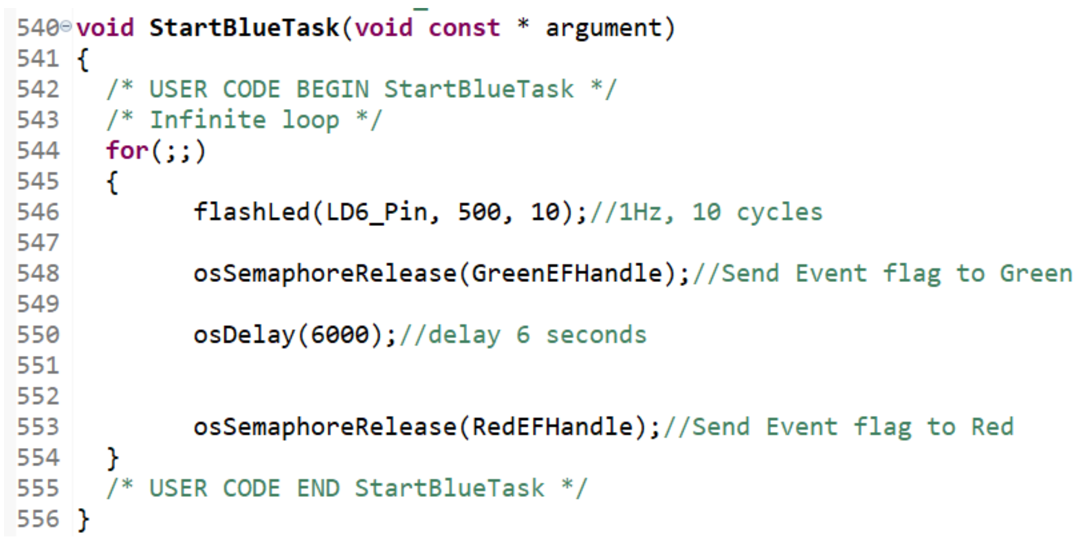
Exercise 15)



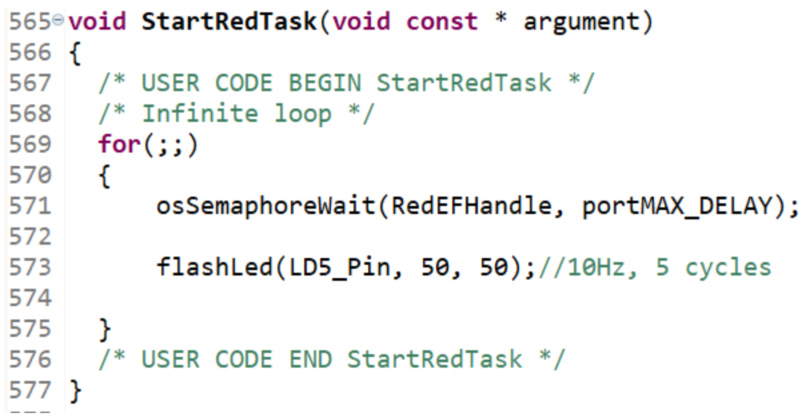
```
  /* USER CODE BEGIN RTOS_SEMAPHORES */
  /* add semaphores, ... */
  osSemaphoreWait(RedEFHandle, 1);//takes semaphore to make task wait
  osSemaphoreWait(GreenEFHandle, 1);//takes semaphore to make task wait
  /* USER CODE END RTOS_SEMAPHORES */
```



```
void StartGreenTask(void const * argument)
{
  /* USER CODE BEGIN StartGreenTask */
  /* Infinite loop */
  for(;;)
  {

        osSemaphoreWait(GreenEFHandle, portMAX_DELAY);//wait for EventFlag

        flashLed(LD4_Pin, 50, 50);//10Hz, 5 cycles
  }
  /* USER CODE END StartGreenTask */
}
```

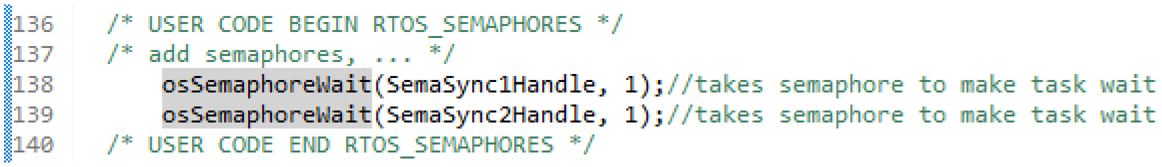


```
void StartBlueTask(void const * argument)
{
  /* USER CODE BEGIN StartBlueTask */
  /* Infinite loop */
  for(;;)
  {
        flashLed(LD6_Pin, 500, 10);//1Hz, 10 cycles

        osSemaphoreRelease(GreenEFHandle);//Send Event flag to Green

        osDelay(6000);//delay 6 seconds


        osSemaphoreRelease(RedEFHandle);//Send Event flag to Red
  }
  /* USER CODE END StartBlueTask */
}
```

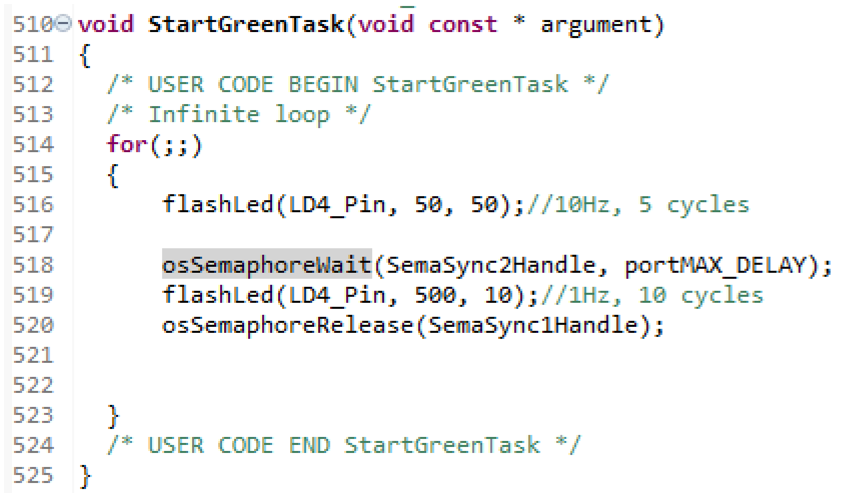


```c
void StartRedTask(void const * argument)
{
  /* USER CODE BEGIN StartRedTask */
  /* Infinite loop */
  for(;;)
  {
      osSemaphoreWait(RedEFHandle, portMAX_DELAY);

      flashLed(LD5_Pin, 50, 50);//10Hz, 5 cycles

  }
  /* USER CODE END StartRedTask */
}
```

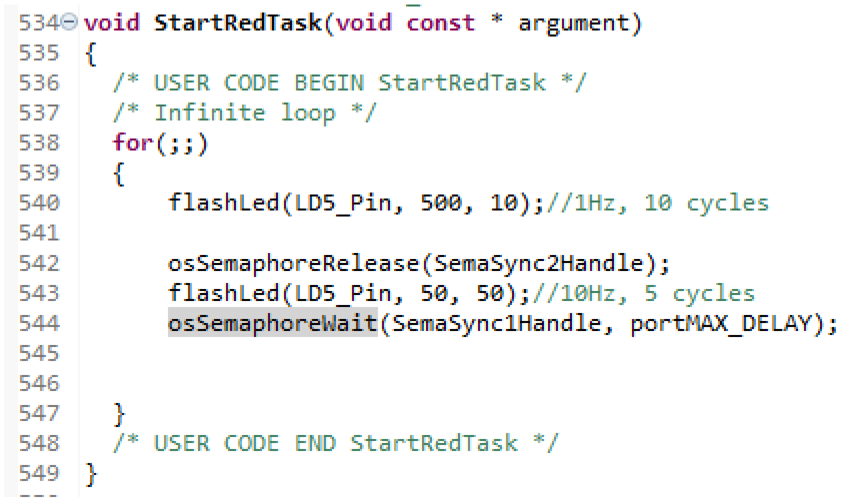
Exercise 16)



```c
  /* USER CODE BEGIN RTOS_SEMAPHORES */
  /* add semaphores, ... */
      osSemaphoreWait(SemaSync1Handle, 1);//takes semaphore to make task wait
      osSemaphoreWait(SemaSync2Handle, 1);//takes semaphore to make task wait
  /* USER CODE END RTOS_SEMAPHORES */
```



```c
void StartGreenTask(void const * argument)
{
  /* USER CODE BEGIN StartGreenTask */
  /* Infinite loop */
  for(;;)
  {
      flashLed(LD4_Pin, 50, 50);//10Hz, 5 cycles

      osSemaphoreWait(SemaSync2Handle, portMAX_DELAY);
      flashLed(LD4_Pin, 500, 10);//1Hz, 10 cycles
      osSemaphoreRelease(SemaSync1Handle);


  }
  /* USER CODE END StartGreenTask */
}
```



```c
void StartRedTask(void const * argument)
{
  /* USER CODE BEGIN StartRedTask */
  /* Infinite loop */
  for(;;)
  {
      flashLed(LD5_Pin, 500, 10);//1Hz, 10 cycles

      osSemaphoreRelease(SemaSync2Handle);
      flashLed(LD5_Pin, 50, 50);//10Hz, 5 cycles
      osSemaphoreWait(SemaSync1Handle, portMAX_DELAY);


  }
  /* USER CODE END StartRedTask */
}
```