Experiment 2 Code

```
466 void StartDefaultTask(void const * argument)
467 {
468   /* USER CODE BEGIN 5 */
469   /* Infinite loop */
470   for(;;)
471   {
472     HAL_GPIO_WritePin(GPIOD,GPIO_PIN_14,GPIO_PIN_SET);//Turn On LED
473     osDelay(2000);//2 sec Delay
474     HAL_GPIO_WritePin(GPIOD,GPIO_PIN_14,GPIO_PIN_RESET);//Turn Off LED
475     osDelay(500);//0.5 sec Delay
476   }
```

Exercise 1

```
466 void StartDefaultTask(void const * argument)
467 {
468   /* USER CODE BEGIN 5 */
469   /* Infinite loop */
470  TickType_t TaskTimeStamp;
471  TickType_t DelayTimeMsec = 2000;
472  TaskTimeStamp = xTaskGetTickCount();
473   for(;;)
474   {
475     HAL_GPIO_WritePin(GPIOD,GPIO_PIN_14,GPIO_PIN_SET);//Turn On LED
476     osDelay(1000);//1 sec Delay
477     osDelayUntil(&TaskTimeStamp,DelayTimeMsec);
478     HAL_GPIO_WritePin(GPIOD,GPIO_PIN_14,GPIO_PIN_RESET);//Turn off LED
479     osDelayUntil(&TaskTimeStamp,500);
480   }
```

Exercise 2

```
472 void StartFlashGreenLedTask(void const * argument) {
473   /* USER CODE BEGIN 5 */
474   /* Infinite loop */
475  TickType_t TaskTimeStamp;
476  TickType_t DelayTimeMsec = 4000;
477  TaskTimeStamp = xTaskGetTickCount();
478   for(;;)
479   {
480     HAL_GPIO_WritePin(GPIOD,GPIO_PIN_15,GPIO_PIN_SET);//Turn On Blue LED
481     HAL_GPIO_TogglePin(GPIOD,GPIO_PIN_12);//Toggle Green LED
482     osDelayUntil(&TaskTimeStamp,DelayTimeMsec);//toggle every 4 seconds
483     HAL_GPIO_WritePin(GPIOD,GPIO_PIN_15,GPIO_PIN_RESET);//Turn Off Blue LED
484     osDelay(6000);
485   }
486   /* USER CODE END 5 */
487 }
488 /* USER CODE BEGIN Header_StartRedFlashLedTask */
489 /**
490 * @brief Function implementing the FlashRedLedTask thread.
491 * @param argument: Not used
492 * @retval None
493 */
494 /* USER CODE END Header_StartRedFlashLedTask */
495 void StartRedFlashLedTask(void const * argument) {
496   /* USER CODE BEGIN StartRedFlashLedTask */
497   /* Infinite loop */
498     TickType_t TaskTimeStamp;
499      TaskTimeStamp = xTaskGetTickCount();
500   for(;;)
501   {
502       HAL_GPIO_WritePin(GPIOD,GPIO_PIN_13,GPIO_PIN_SET);//Turn On Orange LED
503           HAL_GPIO_TogglePin(GPIOD,GPIO_PIN_14);//Toggle Red LED
504           osDelayUntil(&TaskTimeStamp,500);//toggle every 0.5 sec
505           HAL_GPIO_WritePin(GPIOD,GPIO_PIN_13,GPIO_PIN_RESET);//Turn Off Orange LED
506           osDelay(1500);//delay for 1.5 sec
507   }
```

Exercise 3

```
void StartFlashGreenLedTask(void const * argument)
{
  /* USER CODE BEGIN 5 */
  /* Infinite loop */
 TickType_t TaskTimeStamp;
 TickType_t DelayTimeMsec = 2000;
 TaskTimeStamp = xTaskGetTickCount();
  for(;;)
  {
    HAL_GPIO_TogglePin(GPIOD,GPIO_PIN_12);//Turn On LED
    //osDelay(2000);//2 sec Delay
    osDelayUntil(&TaskTimeStamp,DelayTimeMsec);
  }
  /* USER CODE END 5 */
}

/* USER CODE BEGIN Header_StartRedFlashLedTask */
/**
* @brief Function implementing the FlashRedLedTask thread.
* @param argument: Not used
* @retval None
*/
/* USER CODE END Header_StartRedFlashLedTask */
void StartRedFlashLedTask(void const * argument)
{
  /* USER CODE BEGIN StartRedFlashLedTask */
  /* Infinite loop */
    TickType_t TaskTimeStamp;
     TaskTimeStamp = xTaskGetTickCount();
  for(;;)
  {
      HAL_GPIO_TogglePin(GPIOD,GPIO_PIN_14);//Turn On LED
      //osDelay(1000);//1 sec Delay
      osDelayUntil(&TaskTimeStamp,1000);
  }
  /* USER CODE END StartRedFlashLedTask */
```

Exercise 4