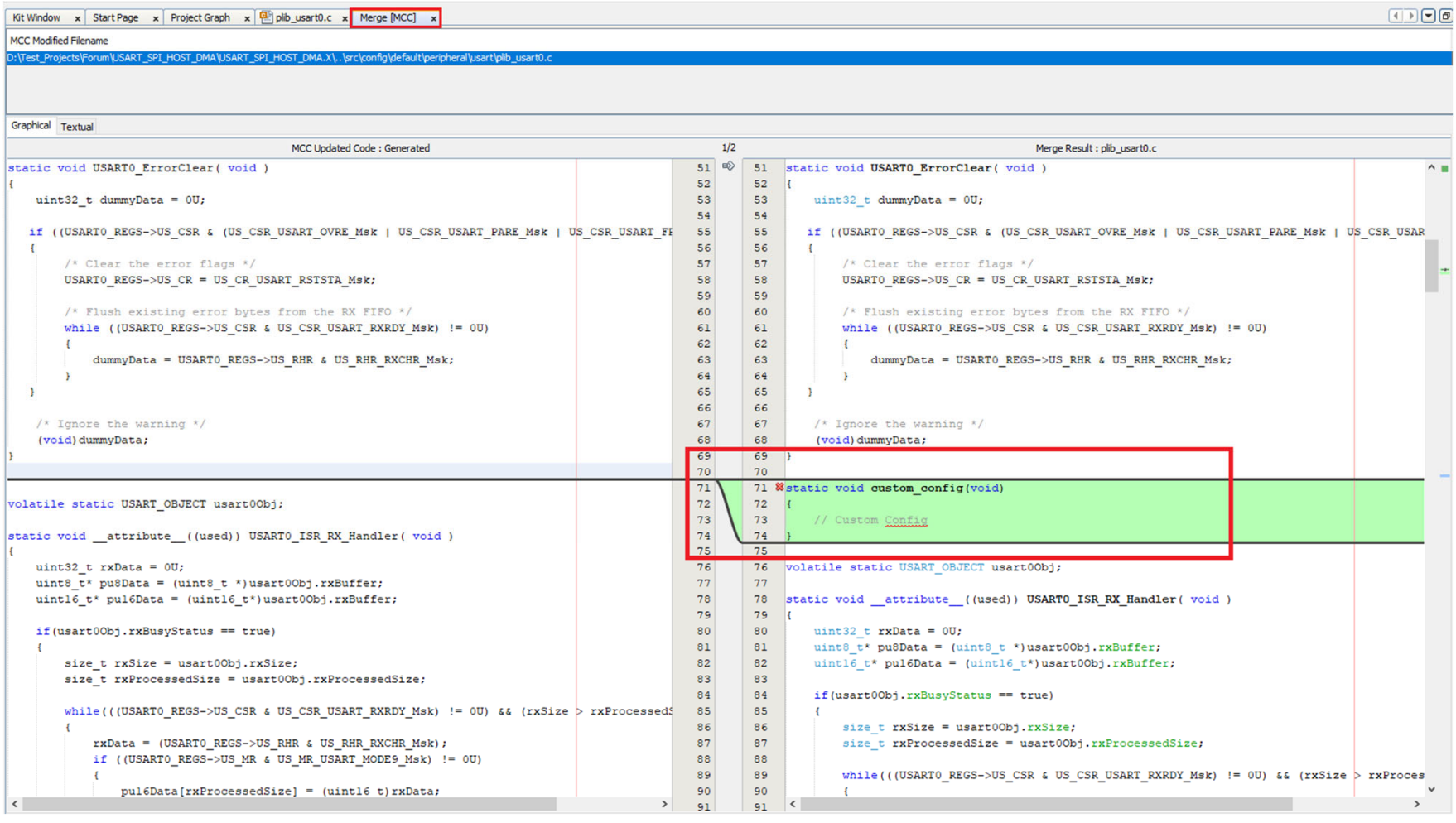
Kit Window
Start Page
Project Graph
plib_usart0.c
Merge [MCC]
MCC Modified Filename
D:\Test_Projects\Forum\USART_SPI_HOST_DMA\USART_SPI_HOST_DMA.X\..\src\config\default\peripheral\usart\plib_usart0.c
Graphical
Textual
MCC Updated Code : Generated
1/2
Merge Result : plib_usart0.c
static void USART0_ErrorClear(void)
{
uint32_t dummyData = 0U;
if ((USART0_REGS->US_CSR & (US_CSR_USART_OVRE_Msk | US_CSR_USART_PARE_Msk | US_CSR_USART_FR
{
/* Clear the error flags */
USART0_REGS->US_CR = US_CR_USART_RSTSTA_Msk;
/* Flush existing error bytes from the RX FIFO */
while ((USART0_REGS->US_CSR & US_CSR_USART_RXRDY_Msk) != 0U)
{
dummyData = USART0_REGS->US_RHR & US_RHR_RXCHR_Msk;
}
}
/* Ignore the warning */
(void)dummyData;
}
volatile static USART_OBJECT usart0Obj;
static void __attribute__((used)) USART0_ISR_RX_Handler(void)
{
uint32_t rxData = 0U;
uint8_t* pu8Data = (uint8_t *)usart0Obj.rxBuffer;
uint16_t* pu16Data = (uint16_t*)usart0Obj.rxBuffer;
if(usart0Obj.rxBusyStatus == true)
{
size_t rxSize = usart0Obj.rxSize;
size_t rxProcessedSize = usart0Obj.rxProcessedSize;
while(((USART0_REGS->US_CSR & US_CSR_USART_RXRDY_Msk) != 0U) && (rxSize > rxProcessedS
{
rxData = (USART0_REGS->US_RHR & US_RHR_RXCHR_Msk);
if ((USART0_REGS->US_MR & US_MR_USART_MODE9_Msk) != 0U)
{
pu16Data[rxProcessedSize] = (uint16_t)rxData;
static void USART0_ErrorClear(void)
{
uint32_t dummyData = 0U;
if ((USART0_REGS->US_CSR & (US_CSR_USART_OVRE_Msk | US_CSR_USART_PARE_Msk | US_CSR_USAR
{
/* Clear the error flags */
USART0_REGS->US_CR = US_CR_USART_RSTSTA_Msk;
/* Flush existing error bytes from the RX FIFO */
while ((USART0_REGS->US_CSR & US_CSR_USART_RXRDY_Msk) != 0U)
{
dummyData = USART0_REGS->US_RHR & US_RHR_RXCHR_Msk;
}
}
/* Ignore the warning */
(void)dummyData;
}
static void custom_config(void)
{
// Custom Config
}
volatile static USART_OBJECT usart0Obj;
static void __attribute__((used)) USART0_ISR_RX_Handler(void)
{
uint32_t rxData = 0U;
uint8_t* pu8Data = (uint8_t *)usart0Obj.rxBuffer;
uint16_t* pu16Data = (uint16_t*)usart0Obj.rxBuffer;
if(usart0Obj.rxBusyStatus == true)
{
size_t rxSize = usart0Obj.rxSize;
size_t rxProcessedSize = usart0Obj.rxProcessedSize;
while(((USART0_REGS->US_CSR & US_CSR_USART_RXRDY_Msk) != 0U) && (rxSize > rxProces
{