

Hello,

I am setting up communication via USART2 Asynchronous to receive my data. Configuration for receive the data is 9600/7bit/1-bit stop/Parity Even/Mode Rx/Tx. The implementation works fine when no parity is implemented. But the specification dictates that even parity should be supported.

If added the parity, I received any value.I do not understand where is the problem.

My configuration is with STMCubeMx for STM32L031K6 nucleo.

```
void MX_USART2_UART_Init(void)
{
    huart2.Instance = USART2;
    huart2.Init.BaudRate = 9600;
    huart2.Init.WordLength = UART_WORDLENGTH_7B;
    huart2.Init.StopBits = UART_STOPBITS_1;
    huart2.Init.Parity = UART_PARITY_NONE;
    huart2.Init.Mode = UART_MODE_RX;
    huart2.Init.HwFlowCtl = UART_HWCONTROL_NONE;
    huart2.Init.OverSampling = UART_OVERSAMPLING_16;
    huart2.Init.OneBitSampling = UART_ONE_BIT_SAMPLE_DISABLE;
    huart2.AdvancedInit.AdvFeatureInit =
UART_ADVFEATURE_RXOVERRUNDISABLE_INIT|UART_ADVFEATURE_DMADISABLEONERROR_INIT;
    huart2.AdvancedInit.OverrunDisable = UART_ADVFEATURE_OVERRUN_DISABLE;
    huart2.AdvancedInit.DMADisableonRxError = UART_ADVFEATURE_DMA_DISABLEONRXERROR;
    if (HAL_UART_Init(&huart2) != HAL_OK)
    {
        Error_Handler();
    }
}

void HAL_UART_MspInit(UART_HandleTypeDef* uartHandle)
{
    GPIO_InitTypeDef GPIO_InitStruct;
    if(uartHandle->Instance==USART2)
    {
        /* Peripheral clock enable */
        __HAL_RCC_USART2_CLK_ENABLE();

        /**USART2 GPIO Configuration
        PA9      -----> USART2_TX
        PA10     -----> USART2_RX
        */
        GPIO_InitStruct.Pin = GPIO_PIN_9|GPIO_PIN_10;
        GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
        GPIO_InitStruct.Pull = GPIO_PULLUP;
        GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
        GPIO_InitStruct.Alternate = GPIO_AF4_USART2;
        HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

        /* Peripheral interrupt init */
        HAL_NVIC_SetPriority(USART2_IRQn, 0, 0);
        HAL_NVIC_EnableIRQ(USART2_IRQn);
    }
}

void HAL_UART_MspDeInit(UART_HandleTypeDef* uartHandle)
{
    if(uartHandle->Instance==USART2)
    {
        /* Peripheral clock disable */
        __HAL_RCC_USART2_CLK_DISABLE();
    }
}
```

```
/**USART2 GPIO Configuration
PA9      -----> USART2_TX
PA10     -----> USART2_RX
*/
HAL_GPIO_DeInit(GPIOA, GPIO_PIN_9|GPIO_PIN_10);

/* Peripheral interrupt Deinit*/
HAL_NVIC_DisableIRQ(USART2_IRQn);

}

}

void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStruct;

    /* GPIO Ports Clock Enable */
    __HAL_RCC_GPIOA_CLK_ENABLE();
    __HAL_RCC_GPIOB_CLK_ENABLE();

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_3, GPIO_PIN_RESET);

    /*Configure GPIO pin : PB3 */
    GPIO_InitStruct.Pin = GPIO_PIN_3;
    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

}

/**************************************** interruption *****/

void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
{
    if (huart->Instance == USART2) //current UART
    {
        Receive();
        HAL_UART_Receive_IT(&huart2, Rx_data, 1);
    }
}

/**************************************** Main *****/
int main(void)
{
    HW_Init();
    HAL_UART_Receive_IT(&huart2, Rx_data, 1);
    while (1)
    {
    }
}
```

