

Board : Nucleo-F446RE  
 IDE : TrueStudio 5.4.0 and STM32CubeMX  
 Purpose : Using USART1,USART2, USART3

Status : USART1 Tx/Rx -----> Work Tx / Work Rx  
 USART2 Tx/Rx -----> Not Work Tx / Not Work Rx  
 USART3 Tx/Rx -----> Work Tx / Nor Work Rx(board down)

Please, Red Color.... (USART3 setup)

\*Attach source( main.c , stm32f4xx\_hal\_msp.c )

\* HAL\_UART\_MspInit??? USART Port? ????? ???.

```
void HAL_UART_MspInit(UART_HandleTypeDef* huart)
{
```

```
  GPIO_InitTypeDef GPIO_InitStructure;
```

```
  if(huart->Instance==USART1)
```

```
  {
```

```
    /* USER CODE BEGIN USART1_MspInit 0 */
```

```
    /* USER CODE END USART1_MspInit 0 */
```

```
    /* Peripheral clock enable */
```

```
    __USART1_CLK_ENABLE();
```

```
    /**USART1 GPIO Configuration
```

```
    PA9 -----> USART1_TX
```

```
    PA10 -----> USART1_RX
```

```
    */
```

```
    GPIO_InitStructure.Pin = GPIO_PIN_9|GPIO_PIN_10;
```

```
    GPIO_InitStructure.Mode = GPIO_MODE_AF_PP;
```

```
    GPIO_InitStructure.Pull = GPIO_PULLUP;
```

```
    GPIO_InitStructure.Speed = GPIO_SPEED_HIGH;
```

```
    GPIO_InitStructure.Alternate = GPIO_AF7_USART1;
```

```
    HAL_GPIO_Init(GPIOA, &GPIO_InitStructure);
```

```
    /* Peripheral interrupt init*/
```

```
    HAL_NVIC_SetPriority(USART1_IRQn, 1, 0); // original 0,0
```

```
    HAL_NVIC_EnableIRQ(USART1_IRQn);
```

```
    /* USER CODE BEGIN USART1_MspInit 1 */
```

```
    /* USER CODE END USART1_MspInit 1 */
```

```
  }
```

```
  else if(huart->Instance==USART2)
```

```
  {
```

```
    /* USER CODE BEGIN USART2_MspInit 0 */
```

```
    /* USER CODE END USART2_MspInit 0 */
```

```
    /* Peripheral clock enable */
```

```
    __USART2_CLK_ENABLE();
```

```
    /**USART2 GPIO Configuration
```

```
    PA2 -----> USART2_TX
```

```
    PA3 -----> USART2_RX
```

```
    */
```

```
    GPIO_InitStructure.Pin = GPIO_PIN_2|GPIO_PIN_3;
```

```

GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
GPIO_InitStruct.Pull = GPIO_PULLUP;
GPIO_InitStruct.Speed = GPIO_SPEED_HIGH;
GPIO_InitStruct.Alternate = GPIO_AF7_USART2;
HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

/* Peripheral interrupt init*/
HAL_NVIC_SetPriority(USART2_IRQn, 1, 1); // original 0,0
HAL_NVIC_EnableIRQ(USART2_IRQn);
/* USER CODE BEGIN USART2_MspInit 1 */

/* USER CODE END USART2_MspInit 1 */
}
else if(huart->Instance==USART3)
{
/* USER CODE BEGIN USART3_MspInit 0 */

/* USER CODE END USART3_MspInit 0 */
/* Peripheral clock enable */
__USART3_CLK_ENABLE();

/**USART3 GPIO Configuration
PC5 -----> USART3_RX
PB10 -----> USART3_TX
*/
GPIO_InitStruct.Pin = GPIO_PIN_5;
GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
GPIO_InitStruct.Pull = GPIO_PULLUP;
GPIO_InitStruct.Speed = GPIO_SPEED_HIGH;
GPIO_InitStruct.Alternate = GPIO_AF7_USART3;
HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);

GPIO_InitStruct.Pin = GPIO_PIN_10;
GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
GPIO_InitStruct.Pull = GPIO_PULLUP;
GPIO_InitStruct.Speed = GPIO_SPEED_HIGH;
GPIO_InitStruct.Alternate = GPIO_AF7_USART3;
HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

/* USER CODE BEGIN USART3_MspInit 1 */
HAL_NVIC_SetPriority(USART3_IRQn, 1, 2); // original 0,0
HAL_NVIC_EnableIRQ(USART3_IRQn);

/* USER CODE END USART3_MspInit 1 */
}

*MX_USART3_UART_Init? USART3 ??? ???? ??
void MX_USART3_UART_Init(void)
{
huart3.Instance = USART3;
huart3.Init.BaudRate = 115200;
huart3.Init.WordLength = UART_WORDLENGTH_8B;
huart3.Init.StopBits = UART_STOPBITS_1;
huart3.Init.Parity = UART_PARITY_NONE;
huart3.Init.Mode = UART_MODE_TX_RX;
huart3.Init.HwFlowCtl = UART_HWCONTROL_NONE;
huart3.Init.OverSampling = UART_OVERSAMPLING_16;

```

```

HAL_UART_Init(&huart3);
}

*HAL_UART_RxCpltCallback? Rx??? ??? ???? ???? ??
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
{
uint8_t i;

HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);

if(huart->Instance==USART1)
{
if(Rx_Indx1==0)
{
for(i=0;i<DATA;i++) Rx_Buffer1[i] = 0;
}
if(Rx_Data1[0]!=0x0d)
{
Rx_Buffer1[Rx_Indx1++]=Rx_Data1[0];
}
else
{
Rx_Indx1 = 0;
Transfer_cplt1 = 1;
}
HAL_UART_Receive_IT(&huart1, Rx_Data1, 1);
HAL_UART_Transmit(&huart1, Rx_Data1, 1,1000);
}
else if(huart->Instance==USART2)
{
if(Rx_Indx2==0)
{
for(i=0;i<DATA;i++) Rx_Buffer2[i] = 0;
}
if(Rx_Data2[0]!=0x0d)
{
Rx_Buffer2[Rx_Indx2++]=Rx_Data2[0];
}
else
{
Rx_Indx2 = 0;
Transfer_cplt2 = 1;
}
HAL_UART_Receive_IT(&huart2, Rx_Data2, 1);
HAL_UART_Transmit(&huart2, Rx_Data2, 1,1000);
}
else if(huart->Instance==USART3)
{
if(Rx_Indx3==0)
{
for(i=0;i<DATA;i++) Rx_Buffer3[i] = 0;
}
if(Rx_Data3[0]!=0x0d)
{
Rx_Buffer3[Rx_Indx3++]=Rx_Data3[0];
}
else

```

```
{  
Rx_Indx3 = 0;  
Transfer_cplt3 = 1;  
}  
HAL_UART_Receive_IT(&huart3, Rx_Data3, 1);  
HAL_UART_Transmit(&huart3, Rx_Data3, 1,1000);  
}  
  
}
```

----- Reference -----