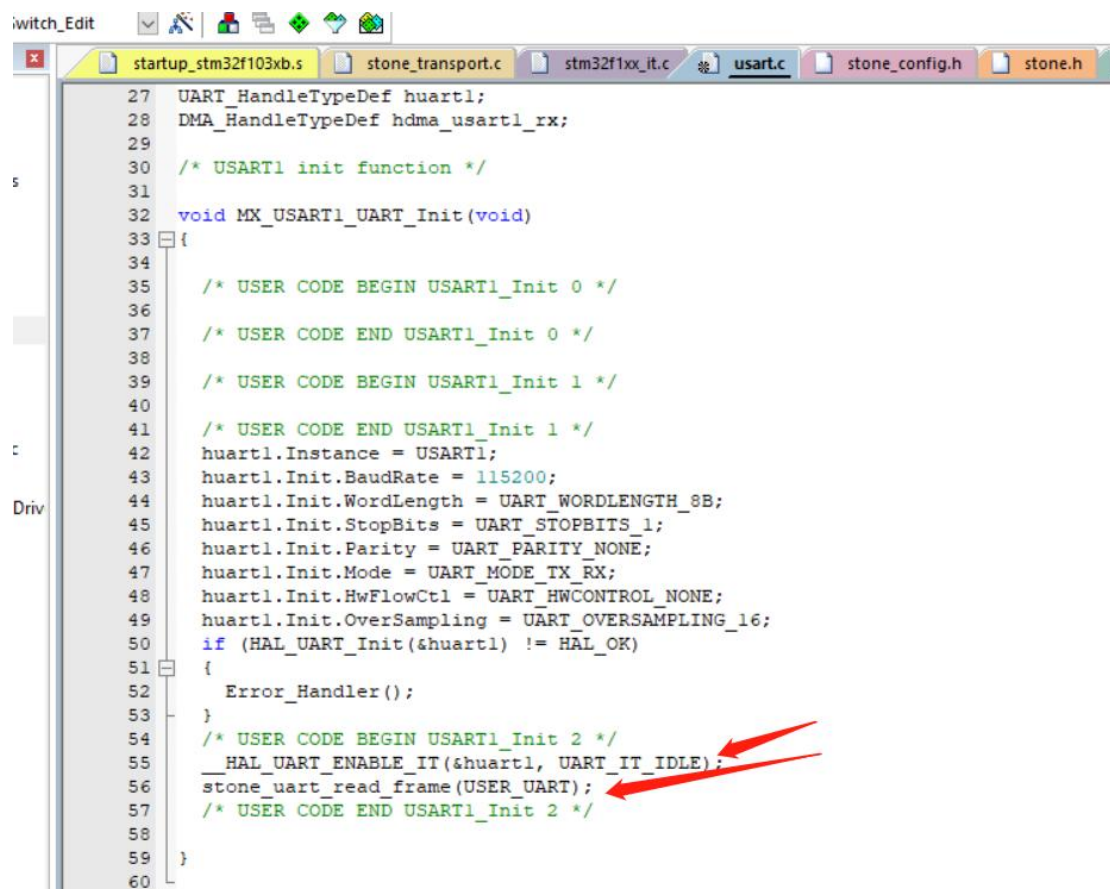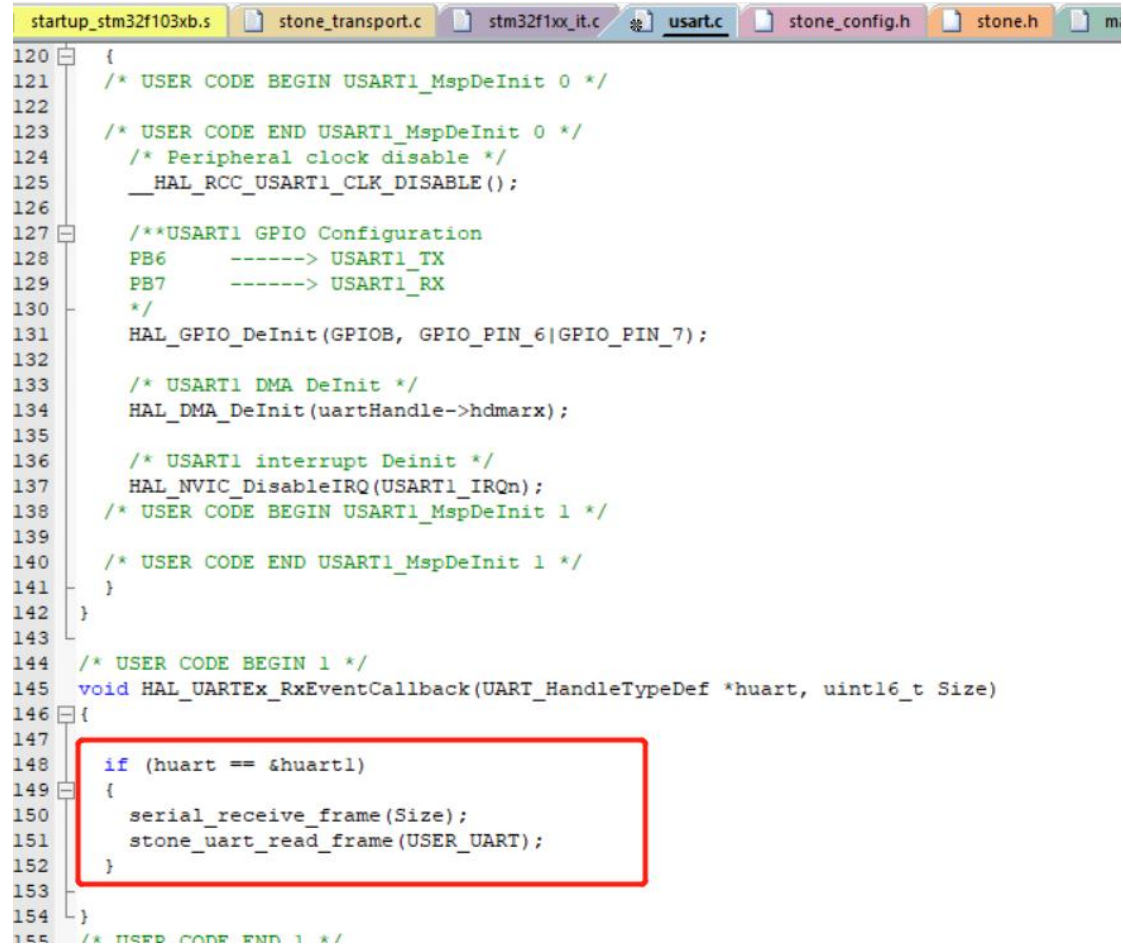# Method for the STM32HAL library to receive the return value of the STONEHMI instruction throughout the frame

1.Enable idle serial port reception in the corresponding serial port configuration function

2.Write a serial port callback function, call the whole frame instruction parsing function in the callback function, and then turn on idle receive again on the next line

```
startup_stm32f103xb.s    stone_transport.c    stm32f1xx_it.c    usart.c    stone_config.h    stone.h    m
120     {
121       /* USER CODE BEGIN USART1_MspDeInit 0 */
122
123       /* USER CODE END USART1_MspDeInit 0 */
124         /* Peripheral clock disable */
125         __HAL_RCC_USART1_CLK_DISABLE();
126
127         /**USART1 GPIO Configuration
128         PB6      ------> USART1_TX
129         PB7      ------> USART1_RX
130         */
131         HAL_GPIO_DeInit(GPIOB, GPIO_PIN_6|GPIO_PIN_7);
132
133         /* USART1 DMA DeInit */
134         HAL_DMA_DeInit(uartHandle->hdmarx);
135
136         /* USART1 interrupt Deinit */
137         HAL_NVIC_DisableIRQ(USART1_IRQn);
138       /* USER CODE BEGIN USART1_MspDeInit 1 */
139
140       /* USER CODE END USART1_MspDeInit 1 */
141       }
142     }
143
144    /* USER CODE BEGIN 1 */
145    void HAL_UARTEx_RxEventCallback(UART_HandleTypeDef *huart, uint16_t Size)
146    {
147
148       if (huart == &huart1)
149       {
150          serial_receive_frame(Size);
151          stone_uart_read_frame(USER_UART);
152       }
153
154    }
155    /* USER CODE END 1 */
```

3. Add frame list initializers and custom node Pointers to the first node of the initialized list in the main function or in the RTOS system task

4.In the last step, as shown in the figure below, the code in the red box is written in a fixed way and put into the tasks of the main loop or RTOS system (note the position of the node pointer in the previous step).

The code in the green box is user-defined. Users can add their own functions here