```c
#include "defs.h"
#include "usart.h"
#include "io.h"


USART_TypeDef * usart_handle[2u] = {USART1, USART2};
PortConfig_TypeDef port_config[2u];
PortRegister_TypeDef port_register[2u];

uint8_t TxState = USART_STATE_IDLE;
uint8_t RespondWaitingFlag = false;
uint8_t NewMessageFlag = false;

char* ptrPrimaryRxBuffer = port_register[PRIMARY_PORT].RxBuffer;
char* ptrPrimaryTxBuffer = port_register[PRIMARY_PORT].TxBuffer;
char* ptrSecondaryRxBuffer = port_register[SECONDARY_PORT].RxBuffer;
char* ptrSecondaryTxBuffer = port_register[SECONDARY_PORT].TxBuffer;

const uint32_t baudrates[6u] = { 2400u, 4800u, 9600u, 19200u, 38400u, 57600u };


/*  */
void USART_Config(uint8_t ucPORT, uint32_t ulBaudRate, uint32_t ulDataBits,  uint8_t ulParity ) {

    LL_USART_InitTypeDef USART_InitStruct = {
        .BaudRate = 19200,
        .DataWidth = LL_USART_DATAWIDTH_8B,
        .StopBits = LL_USART_STOPBITS_1,
        .Parity = LL_USART_PARITY_NONE,
        .TransferDirection = LL_USART_DIRECTION_TX_RX,
        .HardwareFlowControl = LL_USART_HWCONTROL_NONE,
        .OverSampling = LL_USART_OVERSAMPLING_16
    };

    UNUSED(ulDataBits);

    do {
        LL_USART_Disable(usart_handle[ucPORT]);
    } while( LL_USART_IsEnabled(usart_handle[ucPORT]) );

    /* cia reiketu patikrinti baudreito reiksme - ar ji standartine? */
    USART_InitStruct.BaudRate = ulBaudRate;

    switch(ulParity) {
    case USART_PAR_ODD:
        USART_InitStruct.Parity = LL_USART_PARITY_ODD;
    case USART_PAR_EVEN:
        USART_InitStruct.Parity = LL_USART_PARITY_EVEN;
        USART_InitStruct.DataWidth = LL_USART_DATAWIDTH_9B;
        break;
    default:
        USART_InitStruct.Parity = LL_USART_PARITY_NONE;
        USART_InitStruct.DataWidth = LL_USART_DATAWIDTH_8B;
    }

    LL_USART_Init(usart_handle[ucPORT], &USART_InitStruct);

    do {
        LL_USART_Enable(usart_handle[ucPORT]);
    } while( !LL_USART_IsEnabled(usart_handle[ucPORT]) );

    LL_USART_EnableIT_RXNE(usart_handle[ucPORT]);
    LL_USART_DisableIT_TC(usart_handle[ucPORT]);
}

```

```c

/*  */
void USART_Send( uint8_t ucPORT, void* data, size_t len ) {

    while(len--) {
        while(!LL_USART_IsActiveFlag_TC(usart_handle[ucPORT]));
        LL_USART_TransmitData8(usart_handle[ucPORT], *((uint8_t*)data++));
    }
}


/*  */
void USART_Send_DMA(size_t len){
    LL_DMA_SetDataLength(DMA1, LL_DMA_CHANNEL_4, len);
    LL_DMA_EnableChannel(DMA1, LL_DMA_CHANNEL_4);
}


/*  */
void USART_SendByte(uint8_t ucPORT, char data) {
    LL_USART_TransmitData8(usart_handle[ucPORT], data);
}

/*  */
void USART_SendString( uint8_t ucPORT, const char* str ) {

    uint8_t i = 0;

    while( *(str+i) ) {
        while(!LL_USART_IsActiveFlag_TC(usart_handle[ucPORT]));
        LL_USART_TransmitData8(usart_handle[ucPORT], *(str+i));
        i++;
    }
}


/*  */
void USART_ClearRxBuffer(uint8_t ucPORT) {

    uint8_t i = 0;

    while(i < RX_BUFFER_SIZE) {
        port_register[ucPORT].RxBuffer[i++] = 0;
    }

    port_register[ucPORT].RxBufferIndex = 0;
    port_register[ucPORT].ReceivedData = 0;
}


/*  */
uint8_t CheckBaudrate( uint32_t baudrate) {

    uint8_t i = 0;


    while( i < sizeof(baudrates)/sizeof(baudrate) ) {

        if( baudrate == baudrates[i] ) return i;
        i++;
    }

    return 0xFF;
}


```

```
132  /*  */
133  void USART_IRQ_Handler(uint8_t port) {
134
135      if( LL_USART_IsActiveFlag_RXNE(usart_handle[port]) && LL_USART_IsEnabledIT_RXNE(
usart_handle[port]) ) {
136
137          port_register[port].ReceivedData = LL_USART_ReceiveData8(usart_handle[port
]);
138
139          *(port_register[port].RxBuffer + port_register[port].RxBufferIndex) =
port_register[port].ReceivedData;
140
141          port_register[port].RxBufferIndex++;
142
143          port_register[port].PortState = USART_STATE_ANSWER_WAITING;
144
145          port_register[port].PortTimer = 10;
146      }
147  }
148
149
150  /*  */
151  void USART_TimerHandler(void) {
152
153      if(port_register[PRIMARY_PORT].PortTimer > 0) port_register[PRIMARY_PORT].
PortTimer--;
154      else {
155          if(port_register[PRIMARY_PORT].PortState == USART_STATE_ANSWER_WAITING) {
156              port_register[PRIMARY_PORT].PortState = USART_STATE_IDLE;
157              port_register[PRIMARY_PORT].RxBufferIndex = 0;
158
159              RespondWaitingFlag = false;
160              NewMessageFlag = true;
161          }
162      }
163  }
164
```