```
#ifndef USART_H_INCLUDED
#define USART_H_INCLUDED

#include "defs.h"

#define PRIMARY_PORT    0
#define SECONDARY_PORT  1


enum { BR2400 = 0, BR4800, BR9600, BR19200, BR38400, BR57600 };
enum { USART_STATE_IDLE = 0, USART_STATE_BUSY, USART_STATE_ANSWER_WAITING, USART_STATE_DATA_TRANSMITTED, USART_STATE_DATA_RECEIVED };


#define USART_PAR_NONE  0   //MB_PAR_NONE
#define USART_PAR_ODD   1   //MB_PAR_ODD
#define USART_PAR_EVEN  2   //MB_PAR_EVEN

#define RX_BUFFER_SIZE  32
#define TX_BUFFER_SIZE  32

typedef struct{
    uint8_t             MbAddr;
    uint16_t            Baudrate;
    uint8_t             Parity;
    uint8_t             StopBits;
    uint8_t             DataBits;
}PortConfig_TypeDef;

typedef struct{
    uint8_t             PortState;                  // porto busena
    uint8_t             PortError;
    volatile uint8_t    PortTimer;                  //
    uint8_t             ReceivedData;               // priimtas baitas
    char                RxBuffer[RX_BUFFER_SIZE];   // porto RX buferis
    char                TxBuffer[TX_BUFFER_SIZE];   // porto TX buferis
    uint8_t             RxBufferIndex;              // porto RX buferio indeksas
    uint8_t             TxBufferIndex;              // porto TX buferio indeksas
    char*               ptrRxBuffer;
    char*               ptrTxBuffer;
}PortRegister_TypeDef;

extern USART_TypeDef * usart_handle[2u];
extern PortConfig_TypeDef port_config[2u];
extern PortRegister_TypeDef port_register[2u];
extern const uint32_t baudrates[6u];

extern uint8_t TxState;
uint8_t RespondWaitingFlag;
extern uint8_t NewMessageFlag;

extern char* ptrPrimaryRxBuffer;
extern char* ptrPrimaryTxBuffer;
extern char* ptrSecondaryRxBuffer;
extern char* ptrSecondaryTxBuffer;


void    USART_Config(uint8_t ucPORT, uint32_t ulBaudRate, uint32_t ulDataBits, uint8_t ulParity);
void    USART_Send( uint8_t ucPORT, void* buf, size_t size_of_data );
void    USART_Send_DMA(size_t len);
void    USART_SendByte(uint8_t ucPORT, char data);
void    USART_SendString( uint8_t ucPORT, const char* str );
void    USART_IRQ_Handler(uint8_t port);
void    USART_TimerHandler(void);

```

```c
void    USART_ClearRxBuffer(uint8_t ucPORT);
uint8_t CheckBaudrate( uint32_t baudrate);

#endif /* USART_H_INCLUDED */






/* pavizdys, kaip galima laikyti ir naudoti porto nustatymus (ir kita info) 32 bitu registre */
#if defined(QWERTY)

#define USART_BAUDRATE_MSK          0b00000000000000001111111111111111
#define USART_PARITY_MSK            0b00000000000000110000000000000000
#define USART_STOPBITS_MSK          0b00000000000011000000000000000000
#define USART_DATABITS_MSK          0b00000000011100000000000000000000
#define USART_BAUDRATE_IDX_MSK      0b00000011100000000000000000000000
#define USART_PORT_NUMBER_MSK       0b00001100000000000000000000000000

#define USART_GET_BAUDRATE          ( Usart_ConfigRegister & USART_BAUDRATE_MSK )
#define USART_SET_BAUDRATE(x)       ( Usart_ConfigRegister ^= (Usart_ConfigRegister & USART_BAUDRATE_MSK); Usart_ConfigRegister |= x )
#define USART_GET_PARITY            ( (Usart_ConfigRegister & USART_PARITY_MSK)>>16 )
#define USART_SET_PARITY(x)         ( Usart_ConfigRegister ^= (Usart_ConfigRegister & USART_PARITY_MSK); Usart_ConfigRegister |= (x<<16) )
#define USART_GET_STOPBITS          ( (Usart_ConfigRegister & USART_STOPBITS_MSK)>>18 )
#define USART_SET_STOPBITS(x)       ( Usart_ConfigRegister ^= (Usart_ConfigRegister & USART_STOPBITS_MSK); Usart_ConfigRegister |= (x<<18) )
#define USART_GET_DATABITS          ( (Usart_ConfigRegister & USART_DATABITS_MSK)>>20 )
#define USART_SET_DATABITS(x)       ( Usart_ConfigRegister ^= (Usart_ConfigRegister & USART_DATABITS_MSK); Usart_ConfigRegister |= (x<<20) )
#define USART_GET_BAUDRATE_IDX      ( (Usart_ConfigRegister & USART_BAUDRATE_IDX_MSK)>>23 )
#define USART_SET_BAUDRATE_IDX(x)   ( Usart_ConfigRegister ^= (Usart_ConfigRegister & USART_BAUDRATE_IDX_MSK); Usart_ConfigRegister |= (x<<23) )
#define USART_GET_PORT_NUMBER       ( (Usart_ConfigRegister & USART_PORT_NUMBER_MSK)>>26 )
#define USART_SET_PORT_NUMBER(x)    ( Usart_ConfigRegister ^= (Usart_ConfigRegister & USART_PORT_NUMBER_MSK); Usart_ConfigRegister |= (x<<26) )

/*
     0-15 - Baudrate
    16-17 - Parity
    18-19 - StopBits
    20-22 - DataBits
    23-25 - baudreito indeksas is bodreitu masyvo
    26-27 - naudojamo USART porto numeris
*/
extern uint32_t    Usart_ConfigRegister;
#endif

```