1. SDR = $(1001101)_2 = (77)_{10}$ → BR = F_{MCK} / ((SDR + 1) * 2) → BR = 3_{x10}^{6} / ((77 + 1) * 2) = 19230.769, we want 19200. Thus %error is = $\frac{19230.769-19200}{19200} = 0.16026\%\ error$
2.
    a. Done above
    b. SDR = 37.8601 → 0.358709% error
    c. SDR = 12.020833 → 0.1602564% error
3. Max SDR = 87.57028 ≈ 87; Min SDR = 76.7493990 ≈ 77.
4. And 5.

```
#include "r_cg_macrodriver.h"
#include "r_cg_cgc.h"
#include "r_cg_port.h"
#include "r_cg_serial.h"
/* Start user code for include. Do not edit comment generated here */
/* End user code. Do not edit comment generated here */
#include "r_cg_userdefine.h"

uint8_t uart2RxBuf[RX_BUF_LEN];
uint16_t uart2RxCnt;
uint8_t uart2RxFlag;
uint8_t uart2TxBuf[TX_BUF_LEN];
uint16_t uart2TxCnt;
uint8_t uart2TxFlag;
uint8_t     uart2RxErrFlag;        // UART2 Receive Error Flag
uint8_t uart2RxOvrFlag;            // UART2 Receive Overrun Flag
MD_STATUS uart2Status;

void R_MAIN_UserInit(void);
void sendHello();

static int sendHelloFlag = 0;
void sendHello()
{
        uart2TxBuf[0] = 'H';
        uart2TxBuf[1] = 'e';
        uart2TxBuf[2] = 'l';
        uart2TxBuf[3] = 'l';
        uart2TxBuf[4] = 'o';

        return;
}

void main(void)
{
        R_MAIN_UserInit();
        /* Start user code. Do not edit comment generated here */

        //Create and initialize the UART
        R_UART2_Create();
        R_UART2_Start();
        uart2Status = R_UART2_Receive(&uart2RxBuf[0],1);  // Prime UART2 Rx

        while (1U)
        {
                sendHelloFlag = 0;

                //Check if byte received on UART
                if (uart2RxFlag)
                {
                        // clear rx flag
                        uart2RxFlag = 0U;

                        if(uart2RxBuf[0] == 'l')
                                P7=(P7^0b10000000);
                        else if(uart2RxBuf[0] == 'h')
                        {
                                sendHello();
                sendHelloFlag = 1;
            }

            if(!sendHelloFlag)
            {
                //Echo back one higher character
                uart2TxBuf[0] = uart2RxBuf[0]+1;
            }
            //Send TX buffer, and specify how many characters to write
            uart2Status = R_UART2_Send(uart2TxBuf, ((sendHelloFlag) ? 5 : 1));

            // Prime UART2 Rx
            uart2Status = R_UART2_Receive(uart2RxBuf, 1);
        }


        //If a character has been transmitted
        if (uart2TxFlag)
        {
            // End of UART2 transmit
            uart2TxFlag = 0U;   // clear tx flag
        }
    }
}

void R_MAIN_UserInit(void)
{
    EI();
}
```