1. SDR = (1001101)2 = (77)10 🡪 BR = FMCK / ((SDR + 1) \* 2) 🡪 BR = 3x106 / ((77 + 1) \* 2) = 19230.769, we want 19200. Thus %error is =
   1. Done above
   2. SDR = 37.8601 🡪 0.358709% error
   3. SDR = 12.020833 🡪 0.1602564% error
2. Max SDR = 87.57028 ≈ 87; Min SDR = 76.7493990 ≈ 77.
3. 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();

}