**#include** "init.h"

**unsigned** **char** data;

**void** **main**(**void**) {

init\_device();

**while**(1) {

RCGCUART = 0x00000004; //activate clock for UART2

RCGCGPIO = 0x00000008; // GPIO Port D Run Mode Clock

GPIO\_PORTD\_AFSEL = 0x01; // enable alt funct on PD5

GPIO\_PORTD\_DEN = 0x01; // enable digital I/O on PD5

GPIODR2R = 0x01; // To enable GPIO Current level

GPIO\_PORTD\_PCTL = (GPIO\_PORTD\_PCTL & 0x0000FF00) + 0x00001100; //GPIO Port Control

GPIO\_PORTD\_AMSEL = 0x00; // disable analog functionality on PD

UART2\_CTL &= ~UART\_CTL\_UARTEN; // disable UART

UART2\_IBRD = 8; // IBRD = integer(16,000,000 / (16 \* 115,200)) = int(8.6805)

UART2\_FBRD = 44; // FBRD = integer(0.6805 \* 64+0.5) = int(44.052)

UART2\_LCRH = 0x60; // For Setting 8 bit word length (no parity bits, one stop bit)

UART2\_CC = (UART2\_CC &~ UART\_CC\_CS\_M)+UART\_CC\_CS\_PIOSC;

SYSCTL\_ALTCLKCFG = (SYSCTL\_ALTCLKCFG &~ SYSCTL\_ALTCLKCFG\_ALTCLK\_M)+SYSCTL\_ALTCLKCFG\_ALTCLK\_PIOSC;

UART2\_CTL &= ~UART\_CTL\_HSE; // high-speed disable; divide clock by 16 rather than 8 (default)

UART2\_CTL |= UART\_CTL\_UARTEN; // enable UART

UART\_Transmit(data);

**unsigned** **char** rxuart = UART\_Receive();

UART2DR |= rxuart;

}

}

**void** **UART\_Transmit**(**unsigned** **char** txdata)

{

txdata = 0xAC;

**while**(!UART2\_LCRH & UART2DR)

{

UART2DR |= txdata;

}

}

**unsigned** **char** **UART\_Receive**()

{

**while**(!UART2\_LCRH & UART2DR)

{

}

**return** UART2DR;

}