/\* p4\_4.c: This program sets up UART5 on TI ARM LaunchPad (TM4C123GH6PM) to do terminal echo. When a key is pressed at the terminal emulator of the PC, the character is received by UART5 and it is sent out of UART5 back to the terminal. \*/

/\* UART5 Tx is on PE5, Rx is on PE4. \*/

/\* Use TeraTerm to see that the keys are echoed. \*/

#include <stdint.h>

#include "tm4c123gh6pm.h"

char UART5Rx(void);

void UART5Tx(char c);

void delayMs(int n);

int main(void)

{

char c;

SYSCTL->RCGCUART |= 0x20; /\* provide clock to UART5 \*/

SYSCTL->RCGCGPIO |= 0x10; /\* Enable clock to PORTE \*/

/\* UART5 initialization \*/

UART5->CTL = 0; /\* disable UART5 \*/

UART5->IBRD = 104; /\* 16MHz/16=1MHz, 1MHz/104=9600 baud rate \*/

UART5->FBRD = 11; /\* fraction part, see Example 4-4 \*/

UART5->CC = 0; /\* use system clock \*/

UART5->LCRH = 0x60; /\* 8-bit, no parity, 1-stop bit \*/

UART5->CTL = 0x301; /\* enable UART5, TXE, RXE \*/

/\* UART5 TX5 and RX5 use PE5 and PE4. Set them up. \*/

GPIOE->DEN = 0x30; /\* make PE5, PE4 as digital \*/

GPIOE->AMSEL = 0; /\* turn off analog function \*/

GPIOE->AFSEL = 0x30; /\* use PE5, PE4 alternate function \*/

GPIOE->PCTL = 0x00110000; /\* configure PE5, PE4 for UART5 \*/

delayMs(1); /\* wait for output line to stabilize \*/

UART5Tx('>'); /\* send the prompt character \*/

for(;;)

{

c = UART5Rx(); /\* get a character from UART \*/

UART5Tx(c); /\* echo it back to the terminal \*/

}

}

/\* UART0 Receive \*/

char UART5Rx(void)

{

char c;

while((UART5->FR & 0x10) != 0); /\* wait until the buffer is not empty \*/

c = UART5->DR; /\* read the received data \*/

return c; /\* and return it \*/

}

/\* UART5 Transmit \*/

void UART5Tx(char c)

{

while((UART5->FR & 0x20) != 0); /\* wait until Tx buffer not full \*/

UART5->DR = c; /\* before giving it another byte \*/

}

/\* Append delay functions and SystemInit() here \*/