/\* p4\_2.c: Read data from UART0 and display it at the tri-color LEDs. \*/

/\* The LEDs are connected to Port F 3-1. \*/

/\* Press any A-z, a-z, 0-9 key at the terminal emulator \*/

/\* and see ASCII value in binary is displayed on LEDs of PORTF. \*/

#include <stdint.h>

#include "tm4c123gh6pm.h"

char UART0Rx(void);

void delayMs(int n);

int main(void)

{

char c;

SYSCTL->RCGCUART |= 1; /\* provide clock to UART0 \*/

SYSCTL->RCGCGPIO |= 1; /\* enable clock to PORTA \*/

SYSCTL->RCGCGPIO |= 0x20; /\* enable clock to PORTF \*/

/\* UART0 initialization \*/

UART0->CTL = 0; /\* disable UART0 \*/

UART0->IBRD = 104; /\* 16MHz/16=1MHz, 1MHz/104=9600 baud rate \*/

UART0->FBRD = 11; /\* fraction part, see Example 4-4 \*/

UART0->CC = 0; /\* use system clock \*/

UART0->LCRH = 0x60; /\* 8-bit, no parity, 1-stop bit, no FIFO \*/

UART0->CTL = 0x301; /\* enable UART0, TXE, RXE \*/

/\* UART0 TX0 and RX0 use PA0 and PA1. Set them up. \*/

GPIOA->DEN = 0x03; /\* Make PA0 and PA1 as digital \*/

GPIOA->AFSEL = 0x03; /\* Use PA0,PA1 alternate function \*/

GPIOA->PCTL = 0x11; /\* configure PA0 and PA1 for UART \*/

GPIOF->DIR = 0x0E; /\* configure Port F to control the LEDs \*/

GPIOF->DEN = 0x0E;

GPIOF->DATA = 0;

for(;;)

{

c = UART0Rx(); /\* get a character from UART \*/

GPIOF->DATA = c << 1; /\* shift left and write it to LEDs \*/

}

}

/\* UART0 Receive \*/

/\* This function waits until a character is received then returns it. \*/

char UART0Rx(void)

{

char c;

while((UART0->FR & 0x10) != 0); /\* wait until the buffer is not empty \*/

c = UART0->DR; /\* read the received data \*/

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

}

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