#include <MKL25Z4.H>

void UART0\_init(void);

void delayMs(int n);

void ledR (void);

void ledRoff(void);

void ledG (void);

void ledGoff(void);

void ledB (void);

void ledBoff(void);

void ledA (void);

void ledAoff(void);

void UART0\_Tx(uint8\_t data);

uint8\_t UART0\_Rx (void) ;

void tab (void);

void lect (void);

void inter (void);

uint8\_t c;

char msg[] = "Hello World!\r\n";

int i,j,n,p;

char tabl [100] ;

char esp [] = "\r\n";

int main (void) {

UART0\_init();

for (i = 0; i < 16 ; i++) {

UART0\_Tx(msg[i]);

}

delayMs(10); /\* leave a gap between messages \*/

while (1) {

tab();

}

}

void tab (void)

{

for (i = 0; i < 20; i++)

{

UART0\_Tx(c);

c = UART0\_Rx();

if (c != 0x0D)

{

tabl [i] = c;

}

else

{

for (n = 0; n < 4; n++)

{

UART0\_Tx(esp[n]);

}

if (tabl[3] == 0x6E)

{

switch (tabl[0])

{

case 0x67 :

ledG();

break;

case 0x72 :

ledR();

break;

case 0x62 :

ledB();

break;

case 0x61 :

ledA();

break;

default :

break;

}

}else

if(tabl[3] == 0x66)

{

switch(tabl[0])

{

case 0x67 :

ledGoff();

break;

case 0x72 :

ledRoff();

break;

case 0x62 :

ledBoff();

break;

case 0x61 :

ledAoff();

break;

default :

break;

}

}

for (n = 0; n < 4; n++)

{

UART0\_Tx(esp[n]);

}

i = -1;

n = -1;

j = -1;

}

}

}

void ledR (void){

SIM->SCGC5 |= 0x0400; // enable clock to Port B

PORTB->PCR[18] = 0x0100; // make pin PTB18 as GPIO

FPTB->PDOR = 0x40000; // switch Red/Green LED off

FPTB->PDDR = 0x40000; // enable PTB18/19 as Output

FPTB->PCOR = 0x40000; // make the pin output low

}

void ledRoff(void)

{

FPTB->PSOR = 0x40000; // make the pin output high

}

void ledG (void)

{

SIM->SCGC5 |= 0x0400; // enable clock to Port B

PORTB->PCR[19] = 0x0100; // make pin PTB18 as GPIO

FPTB->PDOR = 0x80000; // switch Red/Green LED off

FPTB->PDDR = 0x80000; // enable PTB18/19 as Output

FPTB->PCOR = 0x80000; // make the pin output low

}

void ledGoff(void)

{

FPTB->PSOR = 0x80000; // make the pin output high

}

void ledB (void)

{

SIM->SCGC5 |= 0x1000; // enable clock to Port B

PORTD->PCR[1] = 0x0100; // make pin PTB18 as GPIO

FPTD->PDOR = 0x02; // switch Red/Green LED off

FPTD->PDDR = 0x02; // enable PTB18/19 as Output

FPTD->PCOR = 0x02; // make the pin output low

}

void ledBoff(void)

{

FPTD->PSOR = 0x02; // make the pin output high

}

void ledA (void)

{

}

void ledAoff(void)

{

FPTD->PSOR = 0x02; // make the pin output high

}

/\* initialize UART2 to transmit and receive at 57600 Baud \*/

void UART0\_init(void) {

SIM->SCGC4 |= 0x0400; /\* enable clock for UART0 \*/

SIM->SOPT2 |= 0x04000000; /\* use FLL output for UART Baud rate generator \*/

UART0->C2 = 0; /\* turn off UART0 while changing configurations \*/

UART0->BDH = 0x00;

UART0->BDL = 0x17; /\* 57600 Baud \*/

UART0->C4 = 0x0F; /\* Over Sampling Ratio 16 \*/

UART0->C1 = 0x00; /\* normal 8-bit, no parity \*/

UART0->C3 = 0x00; /\* no fault interrupt \*/

// UART0->C2 = 0x08; /\* enable transmit \*/

// UART0->C2 = 0x04; /\* enable receive \*/

UART0->C2 = 0x0C; // enable Tx and Rx

SIM->SCGC5 |= 0x0200; /\* enable clock for PORTA \*/

PORTA->PCR[2] = 0x0200; /\* make PTA2 UART0\_Tx pin \*/

PORTA->PCR[1] = 0x0200; /\* make PTA1 UART0\_Rx pin \*/

}

void UART0\_Tx (uint8\_t data)

{

// wait until Tx data register is empty

while (!(UART0->S1 & 0x80)){ };

// Tx data register now empty

UART0->D = data;

}

uint8\_t UART0\_Rx (void) {

// wait until Rx data register is full

while (!(UART0->S1 & 0x20)){ };

return UART0->D;

}

/\* Delay n milliseconds \*/

/\* The CPU core clock is set to MCGFLLCLK at 20.97152 MHz in SystemInit(). \*/

void delayMs(int n)

{

int i;

int j;

for(i = 0 ; i < n; i++)

for (j = 0; j < 3500; j++) {}

}