;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

; ED2022, Computer Technology I

; Lab 6, task 1

; Hardware: STK600, CPU ATmega2560

; Date: 2015-11-26

; Authors: Ivan Hussein

; Function:

; --------

; The program writes a character on the CyberTech Display. The display is connected to the serial port ; (RS232). Frame: [ START ] [ ADRESS ] [ COMMAND ] [ CHECKSUM ] [ END ]

; Used subroutine: Yes

; Global subroutines (that can be used from other programs):

; Other information:

;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

#include <avr/io.h>

#include <util/delay.h>

#include <stdio.h> // printf()

#include <avr/iom2560.h>

#include <avr/iomxx0\_1.h>

// value of USART Baud Rate Register (UBRR)

#define CLOCK 1843200

#define SPEED\_BAUD 2400

#define UBRR\_value (((CLOCK)/16UL/(SPEED\_BAUD)) - 1)

#define MS 5000

// function prototypes

void USART\_Init(unsigned int ubrr\_v); // initialize

void USART\_T(unsigned char character); // send

unsigned char USART\_R(void); // receive

unsigned int check\_sum(char\* rad);

void USART\_T\_printf(char c, FILE \*stream);

static FILE usart\_stdout = FDEV\_SETUP\_STREAM(USART\_T\_printf, NULL,

\_FDEV\_SETUP\_WRITE);

int START = 13;

void main( void ){

stdout = &usart\_stdout;

USART\_Init (UBRR\_value);

unsigned int temp;

unsigned char rad2[] ="BO0001 Jippy "; ; string array with Adress and COMMAND

USART\_T(0x0D); ; start

for(int i=0; i < sizeof(rad2); i++) ; checksum

{

USART\_T(rad2[i]);

}

temp = check\_sum(rad2);

printf("%X",temp); ; print the string

USART\_T(0x0A); ; end the frame

USART\_T(0x0D);

USART\_T('Z');

USART\_T('D');

USART\_T('0');

USART\_T('0');

USART\_T('1');

USART\_T('3');

USART\_T('C');

USART\_T(0x0A);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

unsigned int check\_sum(char\* rad)

{

unsigned int csum = START; // 13, 0x0D, CR (carriage return)

while (\*rad)

{

csum += \*rad;

rad++;

}

return csum %256; // modulo 256

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void USART\_Init(unsigned int ubrrv){

// Set baud rate

UBRR0H = (unsigned char)(ubrrv>>8); // load upper 8-bits of ubrrv

UBRR0L = (unsigned char)ubrrv; // load lower 8-bits of ubrrv

// Enable receiver and transmitter

UCSR0B = (1 << RXEN0) | (1 << TXEN0); // Receive Enable (RXEN) bit

// Transmit Enable (TXEN) bit

// Set frame format using USART Character SiZe (UCSZ) bits

UCSR0C = (1 << UCSZ00) | (1 << UCSZ01); // USART Control and Status

// Register C (UCSRnC)

}

void USART\_T(unsigned char data)

{

// Wait for empty transmit buffer

while ( ( UCSR0A & (1 << UDRE0)) == 0 ); // USART Data Register

// Empty(UDRE)flag

// Put data into buffer, sends the data

UDR0 = data; // USART Data Register (UDR)

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

unsigned char USART\_R(void)

{

// Wait for data to be received

while ( !(UCSR0A & (1 << RXC0)) ); //Receive Complete (RXCn) flag

// Return received data from buffer

return UDR0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void USART\_T\_printf(char c, FILE \*stream)

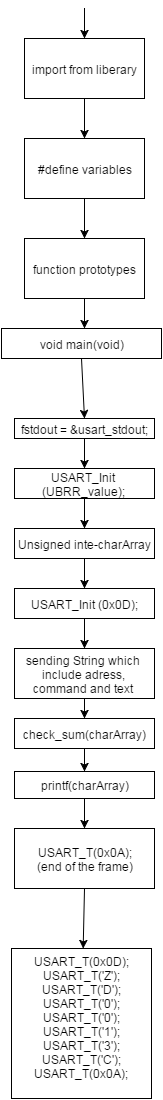
{

if (c == '\n') USART\_T('\r'); // replace \n (newline) with

// \r (carriage-return)

USART\_T(c);

}



#include <avr/io.h>

#include <util/delay.h>

#include <stdio.h> // printf()

#include <avr/iom2560.h>

#include <avr/iomxx0\_1.h>

// value of USART Baud Rate Register (UBRR)

#define CLOCK 1843200

;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

; ED2022, Computer Technology I

; Lab 6, task 2

; Hardware: STK600, CPU ATmega2560

; Date: 2015-11-26

; Authors: Ivan Hussein

; Function:

; --------

; The program writes characters on all text-lines on the CyberTech. The display is connected to the

; serial port (RS232). Frame: [ START ] [ ADRESS ] [ COMMAND ] [ CHECKSUM ] [ END ]

; Used subroutine:

; Global subroutines (that can be used from other programs):

; Other information:

;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

#define SPEED\_BAUD 2400

#define UBRR\_value (((CLOCK)/16UL/(SPEED\_BAUD)) - 1)

#define MS 5000

// function prototypes

void USART\_Init(unsigned int ubrr\_v); // initialize

void USART\_T(unsigned char character); // send

unsigned char USART\_R(void); // receive

unsigned int check\_sum(char\* rad);

void USART\_T\_printf(char c, FILE \*stream);

static FILE usart\_stdout = FDEV\_SETUP\_STREAM(USART\_T\_printf, NULL,

\_FDEV\_SETUP\_WRITE);

int START = 13;

void main( void ){

stdout = &usart\_stdout;

USART\_Init (UBRR\_value);

unsigned int temp;

unsigned char rad[] ="AO0001 task one is done lets celibrate my frinds lalalalal "; ; string array with Adress and COMMAND

unsigned char rad2[] ="BO0001 Jippy "; ; string array with Adress and COMMAND

USART\_T(0x0D); ; start

for(int i=0; i < sizeof(rad); i++) ; checksum

{

USART\_T(rad[i]);

}

temp = check\_sum(rad);

printf("%X",temp); ; print the string

USART\_T(0x0A); ; end the frame

USART\_T(0x0D);

USART\_T('Z');

USART\_T('D');

USART\_T('0');

USART\_T('0');

USART\_T('1');

USART\_T('3');

USART\_T('C');

USART\_T(0x0A);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

USART\_T(0x0D); ; start

for(int i=0; i < sizeof(rad2); i++) ; checksum

{

USART\_T(rad2[i]);

}

temp = check\_sum(rad2);

printf("%X",temp); ; print the string

USART\_T(0x0A); ; end the frame

USART\_T(0x0D);

USART\_T('Z');

USART\_T('D');

USART\_T('0');

USART\_T('0');

USART\_T('1');

USART\_T('3');

USART\_T('C');

USART\_T(0x0A);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

unsigned int check\_sum(char\* rad)

{

unsigned int csum = START; // 13, 0x0D, CR (carriage return)

while (\*rad)

{

csum += \*rad;

rad++;

}

return csum %256; // modulo 256

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void USART\_Init(unsigned int ubrrv){

// Set baud rate

UBRR0H = (unsigned char)(ubrrv>>8); // load upper 8-bits of ubrrv

UBRR0L = (unsigned char)ubrrv; // load lower 8-bits of ubrrv

// Enable receiver and transmitter

UCSR0B = (1 << RXEN0) | (1 << TXEN0); // Receive Enable (RXEN) bit

// Transmit Enable (TXEN) bit

// Set frame format using USART Character SiZe (UCSZ) bits

UCSR0C = (1 << UCSZ00) | (1 << UCSZ01); // USART Control and Status

// Register C (UCSRnC)

}

void USART\_T(unsigned char data)

{

// Wait for empty transmit buffer

while ( ( UCSR0A & (1 << UDRE0)) == 0 ); // USART Data Register

// Empty(UDRE)flag

// Put data into buffer, sends the data

UDR0 = data; // USART Data Register (UDR)

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

unsigned char USART\_R(void)

{

// Wait for data to be received

while ( !(UCSR0A & (1 << RXC0)) ); //Receive Complete (RXCn) flag

// Return received data from buffer

return UDR0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void USART\_T\_printf(char c, FILE \*stream)

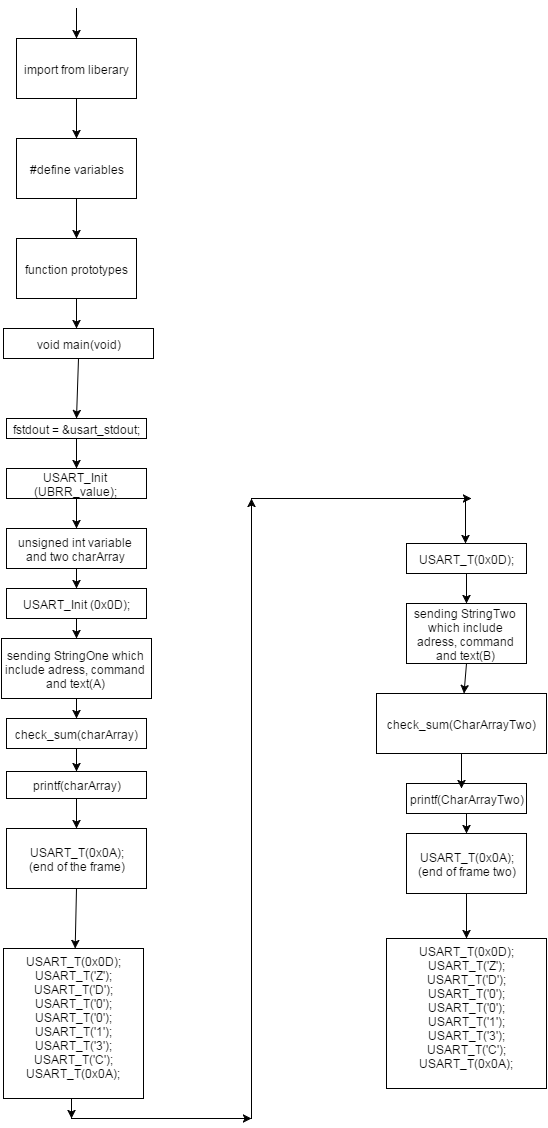
{

if (c == '\n') USART\_T('\r'); // replace \n (newline) with

// \r (carriage-return)

USART\_T(c);

}



;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

; ED2022, Computer Technology I

; Lab 6, task 3

; Hardware: STK600, CPU ATmega2560

; Date: 2015-11-26

; Authors: Ivan Hussein

; Function:

; --------

; The program change text strings on the display. Delays are between each sended frame.

; The display is connected to the serial port (RS232).

; Frame: [ START ] [ ADRESS ] [ COMMAND ] [ CHECKSUM ] [ END ]

; Used subroutine:

; Global subroutines (that can be used from other programs):

; Other information:

;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

#include <avr/io.h>

#include <util/delay.h>

#include <stdio.h> // printf()

#include <avr/iom2560.h>

#include <avr/iomxx0\_1.h>

// value of USART Baud Rate Register (UBRR)

#define CLOCK 1843200

#define SPEED\_BAUD 2400

#define UBRR\_value (((CLOCK)/16UL/(SPEED\_BAUD)) - 1)

#define MS 5000

// function prototypes

void USART\_Init(unsigned int ubrr\_v); // initialize

void USART\_T(unsigned char character); // send

unsigned char USART\_R(void); // receive

unsigned int check\_sum(char\* rad);

void USART\_T\_printf(char c, FILE \*stream);

static FILE usart\_stdout = FDEV\_SETUP\_STREAM(USART\_T\_printf, NULL,

\_FDEV\_SETUP\_WRITE);

int START = 13;

void main( void ){

while(1){ ;forever loop

stdout = &usart\_stdout;

USART\_Init (UBRR\_value);

unsigned int temp;

unsigned char rad[] ="AO0001 Hello "; ; string array with Adress and COMMAND

unsigned char rad2[] ="AO0001 Ivan ";

USART\_T(0x0D); ; start

for(int i=0; i < sizeof(rad); i++) ; checksum

{

USART\_T(rad[i]);

}

temp = check\_sum(rad);

printf("%X",temp); ; print the string

USART\_T(0x0A); ; end the frame

USART\_T(0x0D);

USART\_T('Z');

USART\_T('D');

USART\_T('0');

USART\_T('0');

USART\_T('1');

USART\_T('3');

USART\_T('C');

USART\_T(0x0A);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

\_delay\_ms(MS); ; after delay is the second frame

USART\_T(0x0D); ; start

for(int i=0; i < sizeof(rad2); i++) ; checksum

{

USART\_T(rad2[i]);

}

temp = check\_sum(rad2);

printf("%X",temp); ; print the string

USART\_T(0x0A); ; end the frame

USART\_T(0x0D);

USART\_T('Z');

USART\_T('D');

USART\_T('0');

USART\_T('0');

USART\_T('1');

USART\_T('3');

USART\_T('C');

USART\_T(0x0A);

\_delay\_ms(MS); ; delay

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

unsigned int check\_sum(char\* rad)

{

unsigned int csum = START; // 13, 0x0D, CR (carriage return)

while (\*rad)

{

csum += \*rad;

rad++;

}

return csum %256; // modulo 256

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void USART\_Init(unsigned int ubrrv){

// Set baud rate

UBRR0H = (unsigned char)(ubrrv>>8); // load upper 8-bits of ubrrv

UBRR0L = (unsigned char)ubrrv; // load lower 8-bits of ubrrv

// Enable receiver and transmitter

UCSR0B = (1 << RXEN0) | (1 << TXEN0); // Receive Enable (RXEN) bit

// Transmit Enable (TXEN) bit

// Set frame format using USART Character SiZe (UCSZ) bits

UCSR0C = (1 << UCSZ00) | (1 << UCSZ01); // USART Control and Status

// Register C (UCSRnC)

}

void USART\_T(unsigned char data)

{

// Wait for empty transmit buffer

while ( ( UCSR0A & (1 << UDRE0)) == 0 ); // USART Data Register

// Empty(UDRE)flag

// Put data into buffer, sends the data

UDR0 = data; // USART Data Register (UDR)

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

unsigned char USART\_R(void)

{

// Wait for data to be received

while ( !(UCSR0A & (1 << RXC0)) ); //Receive Complete (RXCn) flag

// Return received data from buffer

return UDR0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void USART\_T\_printf(char c, FILE \*stream)

{

if (c == '\n') USART\_T('\r'); // replace \n (newline) with

// \r (carriage-return)

USART\_T(c);

}

