UESTC2004 Embedded Processors
Written Assignment (2020-2021)

Overview

In this assignment, you will use internet and on-line slides to answer the following question. You are not allowed to exchange your note with your fellow students. All the answers will go through automatically similarity checking by plagiarism software.

You need submit your answers with the file name as "FirstName_Surname_GUID.pdf" and submit to Moodle gateway by 5th July.

Your Task

Question 1 [20 points]: Complete the following matrix.

binary	hexadecimal	decimal
10101100	0xAC	172
11100011	0xE3	227
11011011	0xD8	-37
01001111	0x4F	79
00101011	0X2B	43

Question 2 [15 points]: Give an approximation of sqrt(2) using the decimal fixed-point (0.01) format.

$$\sqrt{2} == 1.41 = 141 \times 0.01$$

Question 3 [15 points]: In order to set only bit 8 of 32-bit variable n to 0 using a logical AND operation with variable MASK, what value MASK should be used?

MASK= 11111111 11111111 11111111 01111111

Q4. [50 points]

(a) Describe the main differences between UART and SPI. [12 points]

UART is asynchronous, while SPI is synchronous.

To be more specific, UART uses start bits and stop bits to indicate beginning and ending of communications since its clock is asynchronous. On the contrary, the sampling time of SPI is determined by clock phase and clock polarity.

(b) Describe the main differences between UART and RS232. [12 points]

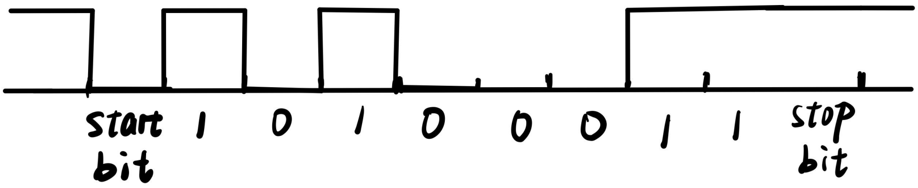
UART takes charge of sending and receiving a sequence of bits. Whereas, RS232 is a standard in physical level, defining voltage levels, connectors and pinouts to assure hardware can identify such bits.

(c) Sketch the output waveform of pin 9 of the UART port when both switches are pressed, based on the following program. [13 points]

```
/* Sending its own switch positions, and displaying those of the
other. */

#include “mbed.h"
Serial async_port(p9, p10);    //set up TX and RX
DigitalOut red_led(p25);
DigitalOut green_led(p26);
DigitalOut strobe(p7);  //a strobe to trigger the scope
DigitalIn switch_ip1(p5);
DigitalIn switch_ip2(p6);
char switch_word ;             //the word we will send
char recd_val;               //the received value

int main() {
  async_port.baud(9600);         //set baud rate to 9600
  while (1){
       if (switch_ip1==1)
       switch_word=switch_word|0x01;  //OR in lsb
if (switch_ip2==1)
      switch_word=switch_word|0x02;  //OR in next lsb
      strobe =1;          //short strobe pulse
      wait_us(10);
      strobe=0;
      async_port.putc(switch_word);    //transmit switch_word
     if (async_port.readable()==1)    //is there a character to be
     read?
        recd_val=async_port.getc();  //if yes, then read it
        //set LEDs according to incoming word
       red_led = 0;      //preset both to 0
        green_led = 0;
        recd_val=recd_val&0x03; //AND out unwanted bits
       if (recd_val==1)
            red_led=1;
      if (recd_val==2)
            green_led=1;
      if (recd_val==3){
         red_led=1;
            green_led=1;}
```



(d) Sketch the output of pin 11 and pin 13 of the SPI port when both switches are pressed, based on the following program. Show their relationship. [13 points]

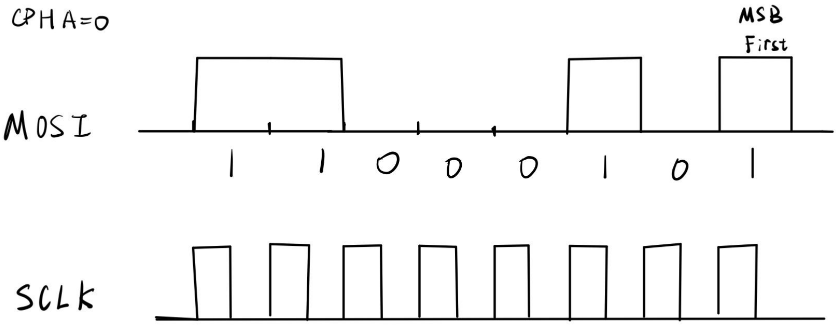
```
    #include “mbed.h”
    SPI   ser_port(p11,p12,p13);  //mosi, miso, sclk
DigitalOut  red_led(p25);           //red led
DigitalOut  green_led(P26);         //green led
    DigitalOut cs(p14);                  //this acts as
    “slave select” DigitalIn switch_ip1(p5);
    DigitalIn  switch_ip2(p6);
    char switch_word;             //word we will send
    char recd_val;                      //value return from slave

    int main(){
          while(1){
```

```
//default settings from SPI Master chosen, no need for further
configuration
            //Set up the word to be sent, by testing switch inputs
            switch_word = 0xa0;     //set up a recognizable output
            pattern
            if(switch_ip1==1)
                  switch_word = switch_word|0X01;     //OR in lsb
            if(switch_ip2==1)
                  switch_word = switch_word/0x02;     //OR in next
                  1sb
            cs = 0;
            recd_val = ser_port.write(switch_word);
                  //send switch_word and receive data
            cs = 1;
            wait(0.01);

            //set leds according to incoming word
            from slave red_led =0; //preset both to
            0
            green_led = 0;
            recd_val = recd_val & 0x03;   //And out unwanted bits
            if(recd_val ==1)  red_led = 1;
            if(recd_val == 2)   green_led = 1;
            if(recd_Val ==3)
            {
                  red_led = 1;
                  green_led = 1;
            }
      }
}
```

Assume CPOL=0
CPHA=0


MSB First
MOSI
1 1 0 0 0 1 0 1
SCLK