

```

#include "soft.h"

SC_MODULE_EXPORT (soft_sc_module);

void soft_sc_module :: skip_space (char ** str)
{
    bool flag = true;

    while ( flag ) {
        if ( !strlen((*str)) )
            //if ( (*str)[0] == '\0' ) //EOS
            flag = false;
        else
            if ((*str)[0]==' ')
                (*str) += 1;
            else
                flag = false;
    }
}

void soft_sc_module :: wait_start_answer_timer ()
{
    while(1){
        wait (f_start_timer.value_changed_event()); //event
        wait (1,SC_MS);
        //f_timer.write(f_start_timer.read()&&(!f_start_get_answer.read()));
        f_timer=f_start_timer &(~f_start_get_answer);
        cout<< "wait_start_answer_timer work\n";
    }
}

void soft_sc_module :: wait_byte_timer ()
{
    next_trigger (0.5,SC_MS);
    if ((sc_time_stamp()-tmd) > sc_time(1, SC_MS) )
        if (f_start_get_answer)
            {f_byte_timer=true;
             next_trigger (0.1,SC_MS);
             f_byte_timer=false;
            }
    //next_trigger (f_start_get_answer.value_changed_event()); //event
    //f_byte_timer = f_start_get_answer;
    cout<< "wait_byte_timer work\n";
}

void soft_sc_module :: p_get ()
{
    cout<<"work p_get\n";
    int cnt_answer_byte=0;
    int num_answer_byte=0;
    bool f_get_first_byte=false;
    while (1) {
        wait ( rx_valid.posedge_event() | f_byte_timer.posedge_event() );
        if (f_byte_timer){cout<<"превышено время в одной посылке мужду байтами
1мс\n";sc_stop();}
        f_get_answer = false;
        tmd=sc_time_stamp();//now

        cout << "получен сигнал rx_valid\n";
        if ( comand == "0x01" ||
            comand == "0x02" ||
            comand == "0x04" )
            if (cnt_answer_byte == 0){

```

```

        if(data_out.read() != "00000000"){
            cout<<"ошибка обработки данных код ошибки:"<<data_out<<endl; sc_stop();
        }
        f_get_answer=true;
        wait (1,SC_US);
        f_get_answer=false;
    }
    if (comand == "0x03") {
        if (cnt_answer_byte == 0){
            if(data_out.read() != "00000000")
                {cout<<"ошибка обработки данных код ошибки:"<<data_out.read()<<endl;
sc_stop();
            }
            f_start_get_answer = true;
            cnt_answer_byte += 1;
        }else
        if (cnt_answer_byte == 1){
            num_answer_byte = data_out.read().to_int();
            cnt_answer_byte += 1;
        }else
        if (cnt_answer_byte == (num_answer_byte + 1)){
            f_get_answer = true;
            wait (1, SC_US);
            f_get_answer = false;
            f_start_get_answer = false;
            cnt_answer_byte = 0;
        }else
            cnt_answer_byte += 1;
    }
}
}
}

```

```

void soft_sc_module :: p_send ()
{
    // fin.open("comfile.txt",ios::out); //open file read
    fin.open(comand_file.c_str(),ios::out);
    cout << "work4\n";
    bool f_first_byte=false;
    if (!fin){
        std::cout << "File invalide open";
        exit (0);
    }

    while (!fin.eof()){

        fin.getline (str,1000);
        skip_space (&(str)); //пропускаем пробелы в начале строки
        if (strlen(str)){ //пропускаем пустые строки
            if (str[0] == '#')
                cout << "koment: " << str <<endl;
            else {
                cout << "data: " << str <<endl;
                f_first_byte=true;
                //while ( str[0] != '\0' ){
                while ( strlen(str) ){
                    if ( str[0] == ' ' )
                        skip_space(&(str));
                    else {
                        //cout << "1/2\n";
                        byte_b.range(7,4)=str[0];
                        str++;
                        skip_space(&(str));
                        if (!strlen(str)){

```

```

        cout<<endl<<"ошибка чтения число символов в строке команд должно
быть четным";
        sc_stop();
    }
    else {
        //cout << "2/2\n";
        byte_b.range(3,0)=str[0];
        str++;
        //
    }
    if (f_first_byte){
        f_first_byte=false;
        comand=byte_b;
    }
    //cout << byte_b << endl;
    tx_valid_i.write(SC_LOGIC_1);
    data_in.write(byte_b);
    wait (tx_busy.posedge_event());
}
}
f_start_timer = true;
tx_valid_i.write(SC_LOGIC_0);
wait ( f_get_answer.posedge_event() | f_timer.posedge_event() );
if ( f_timer ) {cout << "превышено время ожидания ответа 1 мс";sc_stop();}
f_start_timer = false;
}
}
str=str_b;//начало указателя так как в процессе обработки значение менялось
}
cout << "конец командного файла";
fin.close();
sc_stop();
}

void soft_sc_module :: reset_gen ()
{
    rst_n_i.write( SC_LOGIC_0 );
    wait (100, SC_NS);
    cout << "run reset";
    rst_n_i.write( SC_LOGIC_1 );
}

void soft_sc_module :: clk_gen ()
{
    clk_i.write(!clk_i.read());
    next_trigger (period_ns, SC_NS);
}

void soft_sc_module :: rst_n_i2rst_n ()
{
    rst_n.write( rst_n_i.read());
}

void soft_sc_module :: clk_i2clk ()
{
    clk.write(clk_i.read());
}

void soft_sc_module :: data_in_i2data_in ()
{
    data_in.write( data_in_i.read());
}

```

```
void soft_sc_module :: tx_valid_i2tx_valid ()
{
    tx_valid.write( tx_valid_i.read());
}
```