



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

**О Т Ч Е Т**

по лабораторной работе № 5

Название: Обмен данными по интерфейсу SPI

Дисциплина: Микропроцессорные системы

Студент

ИУ6-63Б

\_\_\_\_\_  
(Группа)

В.К. Залыгин

Р.В. Дорохов

\_\_\_\_\_  
(И.О. Фамилия)

\_\_\_\_\_  
(Подпись, дата)

Преподаватель

Е.Ю. Гаврилова

\_\_\_\_\_  
(И.О. Фамилия)

\_\_\_\_\_  
(Подпись, дата)

## Цель работы:

- изучение структуры модуля SPI в микроконтроллере AVR,
- программирование передачи и приема данных по SPI.

## Задание 1. Передача данных в симплексном режиме в Proteus

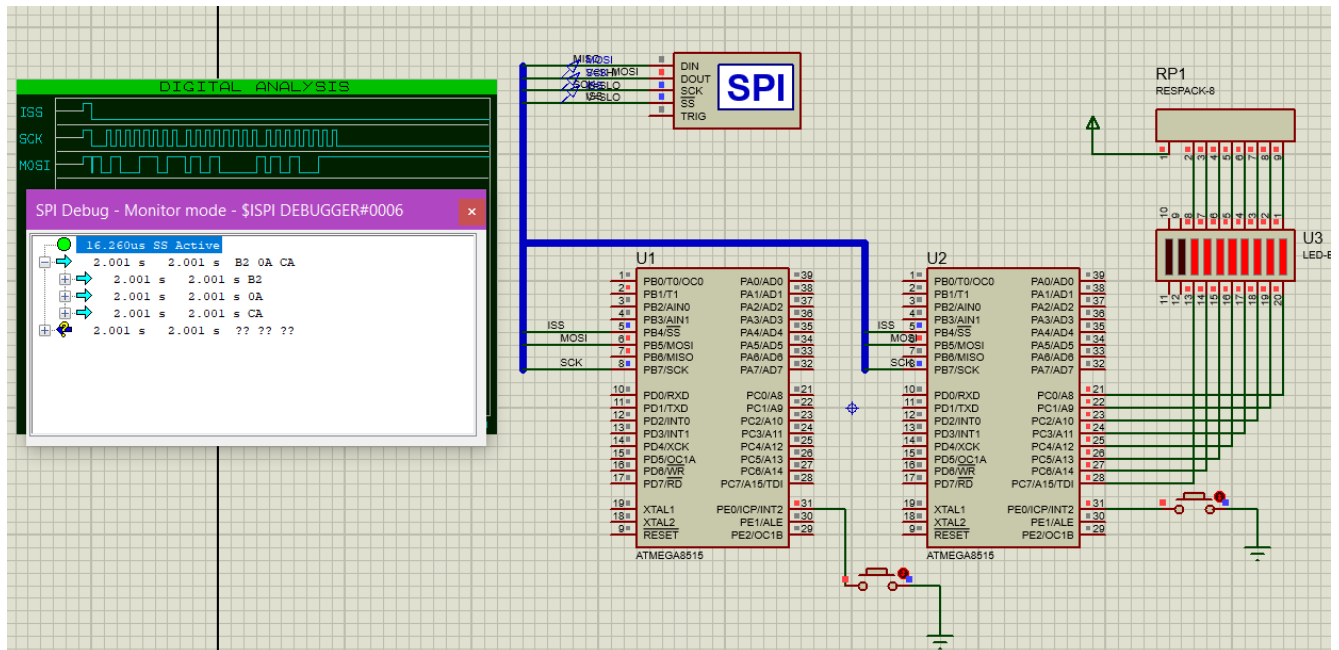


Рисунок 1 - Схема с открытым окном SPI Debugger

AVR\Variables - U1			
AVR\Variables - U2			
AVR\Variables - U2			
Name	Address	Type	Value
data	00800060	byte[3]	byte[3]
data[0]	00800060	byte	'M'
data[1]	00800061	byte	'P'
data[2]	00800062	byte	'S'
received	00800063	byte	0x03

AVR\Data Memory - U1	AVR\Variables - U1	AVR\Data Memory - U2	AVR\Variables - U2
AVR\Data Memory - U2			
0060	00 00 00 00	00 00 00 00	00 00 00 00
0061	00 00 00 00	00 00 00 00	00 00 00 00
0062	00 00 00 00	00 00 00 00	00 00 00 00
0063	00 00 00 00	00 00 00 00	00 00 00 00
0064	00 00 00 00	00 00 00 00	00 00 00 00
0065	00 00 00 00	00 00 00 00	00 00 00 00
0066	00 00 00 00	00 00 00 00	00 00 00 00
0067	00 00 00 00	00 00 00 00	00 00 00 00
0068	00 00 00 00	00 00 00 00	00 00 00 00
0069	00 00 00 00	00 00 00 00	00 00 00 00
0070	00 00 00 00	00 00 00 00	00 00 00 00
0071	00 00 00 00	00 00 00 00	00 00 00 00
0072	00 00 00 00	00 00 00 00	00 00 00 00
0073	00 00 00 00	00 00 00 00	00 00 00 00
0074	00 00 00 00	00 00 00 00	00 00 00 00
0075	00 00 00 00	00 00 00 00	00 00 00 00
0076	00 00 00 00	00 00 00 00	00 00 00 00
0077	00 00 00 00	00 00 00 00	00 00 00 00
0078	00 00 00 00	00 00 00 00	00 00 00 00
0079	00 00 00 00	00 00 00 00	00 00 00 00
0080	00 00 00 00	00 00 00 00	00 00 00 00
0081	00 00 00 00	00 00 00 00	00 00 00 00
0082	00 00 00 00	00 00 00 00	00 00 00 00
0083	00 00 00 00	00 00 00 00	00 00 00 00
0084	00 00 00 00	00 00 00 00	00 00 00 00
0085	00 00 00 00	00 00 00 00	00 00 00 00
0086	00 00 00 00	00 00 00 00	00 00 00 00
0087	00 00 00 00	00 00 00 00	00 00 00 00
0088	00 00 00 00	00 00 00 00	00 00 00 00
0089	00 00 00 00	00 00 00 00	00 00 00 00
0090	00 00 00 00	00 00 00 00	00 00 00 00
0091	00 00 00 00	00 00 00 00	00 00 00 00
0092	00 00 00 00	00 00 00 00	00 00 00 00
0093	00 00 00 00	00 00 00 00	00 00 00 00
0094	00 00 00 00	00 00 00 00	00 00 00 00
0095	00 00 00 00	00 00 00 00	00 00 00 00
0096	00 00 00 00	00 00 00 00	00 00 00 00
0097	00 00 00 00	00 00 00 00	00 00 00 00
0098	00 00 00 00	00 00 00 00	00 00 00 00
0099	00 00 00 00	00 00 00 00	00 00 00 00
00A0	00 00 00 00	00 00 00 00	00 00 00 00
00A1	00 00 00 00	00 00 00 00	00 00 00 00
00A2	00 00 00 00	00 00 00 00	00 00 00 00
00A3	00 00 00 00	00 00 00 00	00 00 00 00
00A4	00 00 00 00	00 00 00 00	00 00 00 00
00A5	00 00 00 00	00 00 00 00	00 00 00 00
00A6	00 00 00 00	00 00 00 00	00 00 00 00
00A7	00 00 00 00	00 00 00 00	00 00 00 00
00A8	00 00 00 00	00 00 00 00	00 00 00 00
00A9	00 00 00 00	00 00 00 00	00 00 00 00
00AA	00 00 00 00	00 00 00 00	00 00 00 00
00AB	00 00 00 00	00 00 00 00	00 00 00 00
00AC	00 00 00 00	00 00 00 00	00 00 00 00
00AD	00 00 00 00	00 00 00 00	00 00 00 00
00AE	00 00 00 00	00 00 00 00	00 00 00 00
00AF	00 00 00 00	00 00 00 00	00 00 00 00
00B0	00 00 00 00	00 00 00 00	00 00 00 00
00B1	00 00 00 00	00 00 00 00	00 00 00 00
00B2	00 00 00 00	00 00 00 00	00 00 00 00
00B3	00 00 00 00	00 00 00 00	00 00 00 00
00B4	00 00 00 00	00 00 00 00	00 00 00 00
00B5	00 00 00 00	00 00 00 00	00 00 00 00
00B6	00 00 00 00	00 00 00 00	00 00 00 00
00B7	00 00 00 00	00 00 00 00	00 00 00 00
00B8	00 00 00 00	00 00 00 00	00 00 00 00
00B9	00 00 00 00	00 00 00 00	00 00 00 00
00BA	00 00 00 00	00 00 00 00	00 00 00 00
00BB	00 00 00 00	00 00 00 00	00 00 00 00
00BC	00 00 00 00	00 00 00 00	00 00 00 00
00BD	00 00 00 00	00 00 00 00	00 00 00 00
00BE	00 00 00 00	00 00 00 00	00 00 00 00
00BF	00 00 00 00	00 00 00 00	00 00 00 00
00C0	00 00 00 00	00 00 00 00	00 00 00 00
00C1	00 00 00 00	00 00 00 00	00 00 00 00
00C2	00 00 00 00	00 00 00 00	00 00 00 00
00C3	00 00 00 00	00 00 00 00	00 00 00 00
00C4	00 00 00 00	00 00 00 00	00 00 00 00
00C5	00 00 00 00	00 00 00 00	00 00 00 00
00C6	00 00 00 00	00 00 00 00	00 00 00 00
00C7	00 00 00 00	00 00 00 00	00 00 00 00
00C8	00 00 00 00	00 00 00 00	00 00 00 00
00C9	00 00 00 00	00 00 00 00	00 00 00 00
00CA	00 00 00 00	00 00 00 00	00 00 00 00
00CB	00 00 00 00	00 00 00 00	00 00 00 00
00CC	00 00 00 00	00 00 00 00	00 00 00 00
00CD	00 00 00 00	00 00 00 00	00 00 00 00
00CE	00 00 00 00	00 00 00 00	00 00 00 00
00CF	00 00 00 00	00 00 00 00	00 00 00 00
00D0	00 00 00 00	00 00 00 00	00 00 00 00
00D1	00 00 00 00	00 00 00 00	00 00 00 00
00D2	00 00 00 00	00 00 00 00	00 00 00 00
00D3	00 00 00 00	00 00 00 00	00 00 00 00
00D4	00 00 00 00	00 00 00 00	00 00 00 00
00D5	00 00 00 00	00 00 00 00	00 00 00 00
00D6	00 00 00 00	00 00 00 00	00 00 00 00
00D7	00 00 00 00	00 00 00 00	00 00 00 00
00D8	00 00 00 00	00 00 00 00	00 00 00 00
00D9	00 00 00 00	00 00 00 00	00 00 00 00
00DA	00 00 00 00	00 00 00 00	00 00 00 00
00DB	00 00 00 00	00 00 00 00	00 00 00 00
00DC	00 00 00 00	00 00 00 00	00 00 00 00
00DD	00 00 00 00	00 00 00 00	00 00 00 00
00DE	00 00 00 00	00 00 00 00	00 00 00 00
00DF	00 00 00 00	00 00 00 00	00 00 00 00
00E0	00 00 00 00	00 00 00 00	00 00 00 00
00E1	00 00 00 00	00 00 00 00	00 00 00 00
00E2	00 00 00 00	00 00 00 00	00 00 00 00
00E3	00 00 00 00	00 00 00 00	00 00 00 00
00E4	00 00 00 00	00 00 00 00	00 00 00 00
00E5	00 00 00 00	00 00 00 00	00 00 00 00
00E6	00 00 00 00	00 00 00 00	00 00 00 00
00E7	00 00 00 00	00 00 00 00	00 00 00 00
00E8	00 00 00 00	00 00 00 00	00 00 00 00
00E9	00 00 00 00	00 00 00 00	00 00 00 00
00EA	00 00 00 00	00 00 00 00	00 00 00 00
00EB	00 00 00 00	00 00 00 00	00 00 00 00
00EC	00 00 00 00	00 00 00 00	00 00 00 00
00ED	00 00 00 00	00 00 00 00	00 00 00 00
00EE	00 00 00 00	00 00 00 00	00 00 00 00
00EF	00 00 00 00	00 00 00 00	00 00 00 00
00F0	00 00 00 00	00 00 00 00	00 00 00 00
00F1	00 00 00 00	00 00 00 00	00 00 00 00
00F2	00 00 00 00	00 00 00 00	00 00 00 00
00F3	00 00 00 00	00 00 00 00	00 00 00 00
00F4	00 00 00 00	00 00 00 00	00 00 00 00
00F5	00 00 00 00	00 00 00 00	00 00 00 00
00F6	00 00 00 00	00 00 00 00	00 00 00 00
00F7	00 00 00 00	00 00 00 00	00 00 00 00
00F8	00 00 00 00	00 00 00 00	00 00 00 00
00F9	00 00 00 00	00 00 00 00	00 00 00 00
00FA	00 00 00 00	00 00 00 00	00 00 00 00
00FB	00 00 00 00	00 00 00 00	00 00 00 00
00FC	00 00 00 00	00 00 00 00	00 00 00 00
00FD	00 00 00 00	00 00 00 00	00 00 00 00
00FE	00 00 00 00	00 00 00 00	00 00 00 00
00FF	00 00 00 00	00 00 00 00	00 00 00 00

Рисунок 2 - Содержимое памяти микроконтроллера-приёмника с выделенными байтами, которые были получены по UART

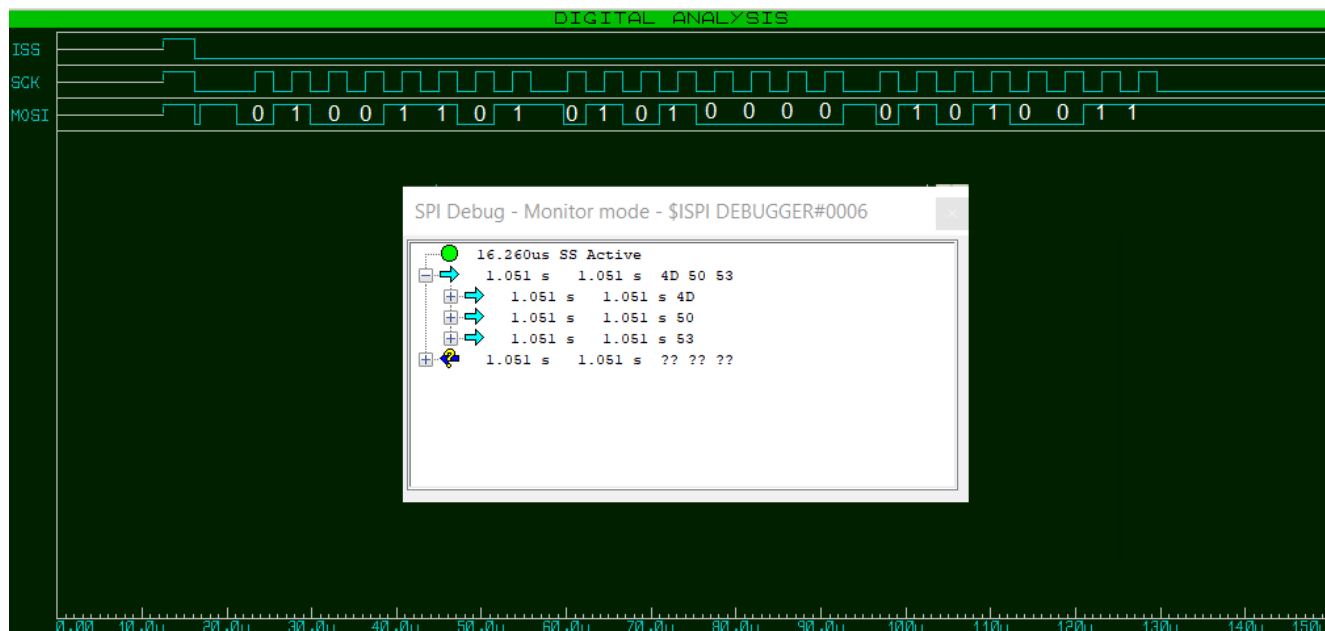


Рисунок 3 - Временная диаграмма с расшифровкой

## Задание 2. Обработка прерывания SPI

### Листинг 1 - Программа ведущего

```
#include <avr/io.h>
/* Кнопка START - PE0 */
#define BUTTON_START 0
#include <avr/interrupt.h>
/* Выводы SPI - PB */
#define PIN_SS 4
#define PIN_SS1 1
#define PIN_MOSI 5
#define PIN_SCK 7
/* Передаваемые данные */
#define DATA_LENGTH 3
const unsigned char data[DATA_LENGTH] = {'M', 'P', 'S'};
uint8_t i = 0;
uint8_t sending = 0;
ISR(SPI_STC_vect) {
    PORTB |= (1<<PIN_SS1);
    if (i < DATA_LENGTH) {
        PORTB &= ~(1<<PIN_SS1);
        SPDR = data[i++];
    }
    if (i == DATA_LENGTH) {
        sending = 0;
    }
}
```

### Листинг 1 - Программа ведущего

```
int main() {
    /* Инициализация SPI */
    /* Настройка выводов MOSI, SCK, SS на вывод */
    DDRB = (1<<PIN_MOSI) | (1<<PIN_SCK) | (1<<PIN_SS);
    /* Включение SPI в режиме ведущего, частота передачи f_clk/16 */
```

```

SPCR = (1<<SPE) | (1<<MSTR) | (1<<SPR0) | (1<<SPIE);
/* Инициализация портов ввода-вывода */
/* Настройка PIN_SS1 на вывод */
DDRB |= (1<<PIN_SS1);
PORTB |= (1<<PIN_SS1);
/* Настройка PE0 на ввод с подтягивающим резистором */
PORTE = (1<<BUTTON_START);
sei();
/* Бесконечный цикл */
while(1) {
/* Проверка нажатия кнопки */
if (!(PINE & (1<<BUTTON_START))) {
/* Ожидание отпускания кнопки */
while (!(PINE & (1<<BUTTON_START)))
;
while (sending);
i = 0;
PORTB &= ~(1<<PIN_SS1);
SPDR = data[i];
i++;
}
}
return 0;
}

```

### Задание 3. Передача данных в симплексном режиме на макете

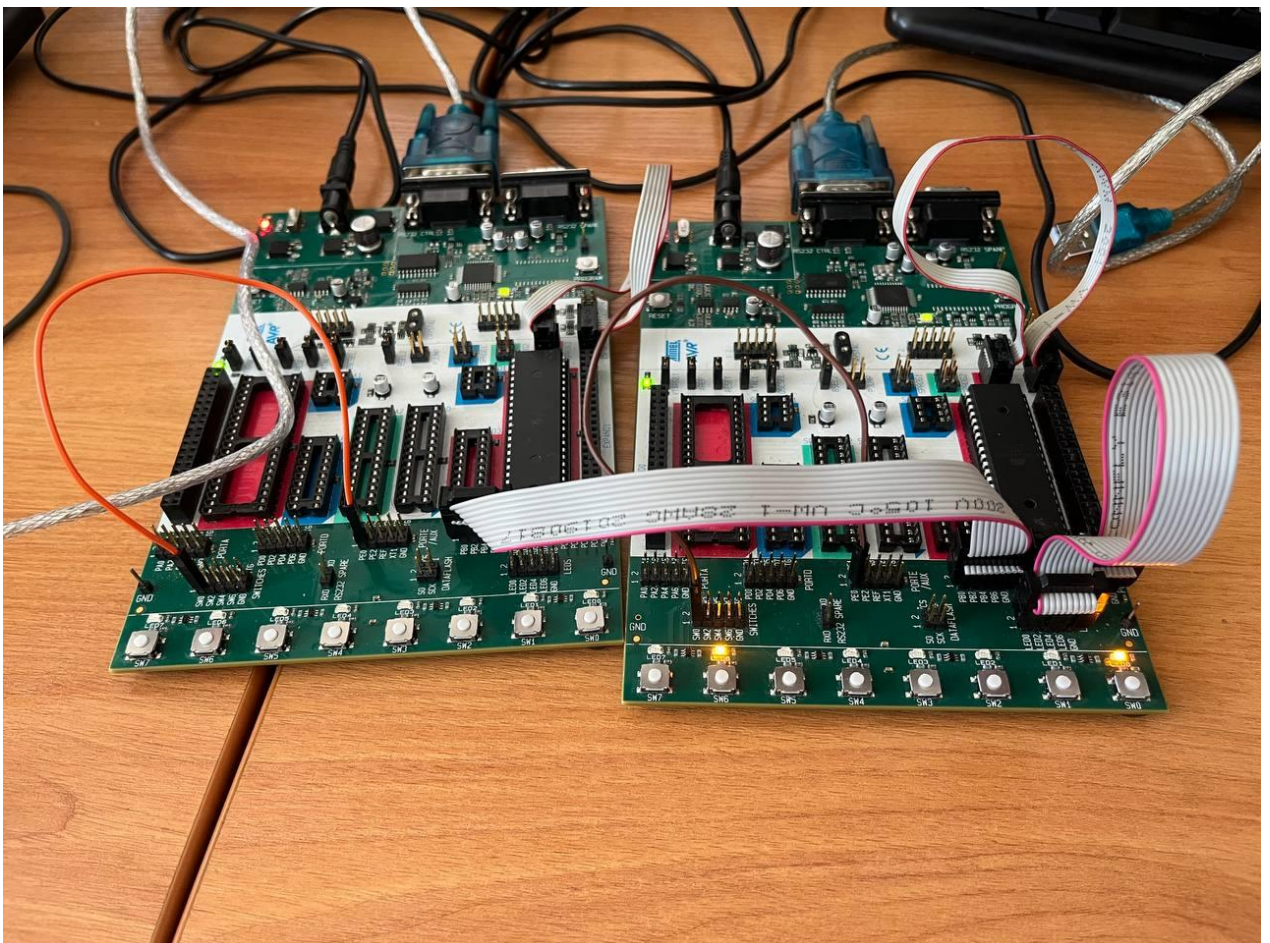


Рисунок 4 - Фотография макета с принятым байтом на светодиодах



## Задание 4. Передача данных в дуплексном режиме в Proteus

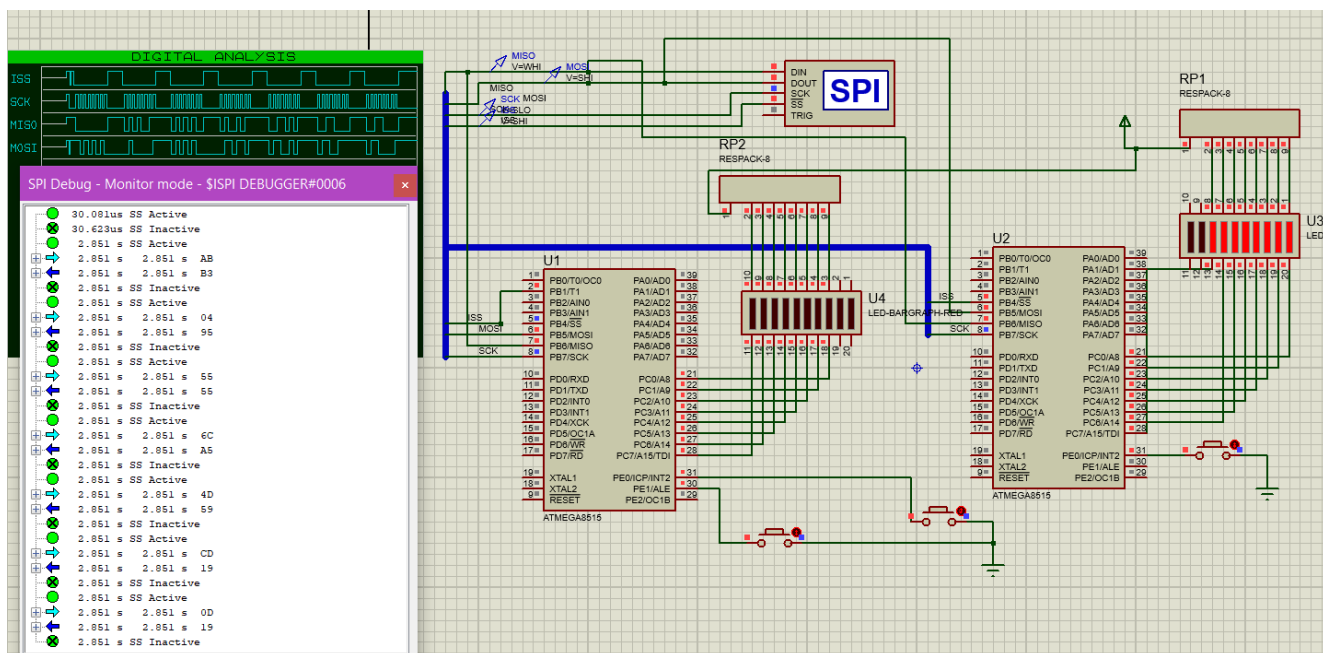


Рисунок 5 - Схема с открытым окном SPI Debugger

AVR\Variables - U1				AVR\Variables - U2			
AVR\Variables - U1				AVR\Variables - U2			
Name	Address	Type	Value	Name	Address	Type	Value
data	00800060	byte[6]	byte[6]	data	00800060	byte[6]	byte[6]
data[0]	00800060	byte	'S'	data[0]	00800060	byte	'A'
data[1]	00800061	byte	'T'	data[1]	00800061	byte	'T'
data[2]	00800062	byte	'K'	data[2]	00800062	byte	'm'
data[3]	00800063	byte	'S'	data[3]	00800063	byte	'e'
data[4]	00800064	byte	'0'	data[4]	00800064	byte	'g'
data[5]	00800065	byte	'0'	data[5]	00800065	byte	'a'
currentBy	00800066	byte	'\0'	currentBy	00800067	byte	'\0'
				received	00800066	byte	0x06

AVR\Data Memory - U1				AVR\Data Memory - U2			
AVR\Data Memory - U1				AVR\Data Memory - U2			
0060	00	00	00	0060	00	00	00
0061	00	00	00	0061	00	00	00
0062	00	00	00	0062	00	00	00
0063	00	00	00	0063	00	00	00
0064	00	00	00	0064	00	00	00
0065	00	00	00	0065	00	00	00
0066	00	00	00	0066	00	00	00
0067	00	00	00	0067	00	00	00
0068	00	00	00	0068	00	00	00
0069	00	00	00	0069	00	00	00
0070	00	00	00	0070	00	00	00
0071	00	00	00	0071	00	00	00
0072	00	00	00	0072	00	00	00
0073	00	00	00	0073	00	00	00
0074	00	00	00	0074	00	00	00
0075	00	00	00	0075	00	00	00
0076	00	00	00	0076	00	00	00
0077	00	00	00	0077	00	00	00
0078	00	00	00	0078	00	00	00
0079	00	00	00	0079	00	00	00
0080	00	00	00	0080	00	00	00
0081	00	00	00	0081	00	00	00
0082	00	00	00	0082	00	00	00
0083	00	00	00	0083	00	00	00
0084	00	00	00	0084	00	00	00
0085	00	00	00	0085	00	00	00
0086	00	00	00	0086	00	00	00
0087	00	00	00	0087	00	00	00
0088	00	00	00	0088	00	00	00
0089	00	00	00	0089	00	00	00
0090	00	00	00	0090	00	00	00
0091	00	00	00	0091	00	00	00
0092	00	00	00	0092	00	00	00
0093	00	00	00	0093	00	00	00
0094	00	00	00	0094	00	00	00
0095	00	00	00	0095	00	00	00
0096	00	00	00	0096	00	00	00
0097	00	00	00	0097	00	00	00
0098	00	00	00	0098	00	00	00
0099	00	00	00	0099	00	00	00
00A0	00	00	00	00A0	00	00	00
00A1	00	00	00	00A1	00	00	00
00A2	00	00	00	00A2	00	00	00
00A3	00	00	00	00A3	00	00	00
00A4	00	00	00	00A4	00	00	00
00A5	00	00	00	00A5	00	00	00
00A6	00	00	00	00A6	00	00	00
00A7	00	00	00	00A7	00	00	00
00A8	00	00	00	00A8	00	00	00
00A9	00	00	00	00A9	00	00	00
00AA	00	00	00	00AA	00	00	00
00AB	00	00	00	00AB	00	00	00
00AC	00	00	00	00AC	00	00	00
00AD	00	00	00	00AD	00	00	00
00AE	00	00	00	00AE	00	00	00
00AF	00	00	00	00AF	00	00	00
00B0	00	00	00	00B0	00	00	00
00B1	00	00	00	00B1	00	00	00
00B2	00	00	00	00B2	00	00	00
00B3	00	00	00	00B3	00	00	00
00B4	00	00	00	00B4	00	00	00
00B5	00	00	00	00B5	00	00	00
00B6	00	00	00	00B6	00	00	00
00B7	00	00	00	00B7	00	00	00
00B8	00	00	00	00B8	00	00	00
00B9	00	00	00	00B9	00	00	00
00BA	00	00	00	00BA	00	00	00
00BB	00	00	00	00BB	00	00	00
00BC	00	00	00	00BC	00	00	00
00BD	00	00	00	00BD	00	00	00
00BE	00	00	00	00BE	00	00	00
00BF	00	00	00	00BF	00	00	00
00C0	00	00	00	00C0	00	00	00
00C1	00	00	00	00C1	00	00	00
00C2	00	00	00	00C2	00	00	00
00C3	00	00	00	00C3	00	00	00
00C4	00	00	00	00C4	00	00	00
00C5	00	00	00	00C5	00	00	00
00C6	00	00	00	00C6	00	00	00
00C7	00	00	00	00C7	00	00	00
00C8	00	00	00	00C8	00	00	00
00C9	00	00	00	00C9	00	00	00
00CA	00	00	00	00CA	00	00	00
00CB	00	00	00	00CB	00	00	00
00CC	00	00	00	00CC	00	00	00
00CD	00	00	00	00CD	00	00	00
00CE	00	00	00	00CE	00	00	00
00CF	00	00	00	00CF	00	00	00
00D0	00	00	00	00D0	00	00	00
00D1	00	00	00	00D1	00	00	00
00D2	00	00	00	00D2	00	00	00
00D3	00	00	00	00D3	00	00	00
00D4	00	00	00	00D4	00	00	00
00D5	00	00	00	00D5	00	00	00
00D6	00	00	00	00D6	00	00	00
00D7	00	00	00	00D7	00	00	00
00D8	00	00	00	00D8	00	00	00
00D9	00	00	00	00D9	00	00	00
00DA	00	00	00	00DA	00	00	00
00DB	00	00	00	00DB	00	00	00
00DC	00	00	00	00DC	00	00	00
00DD	00	00	00	00DD	00	00	00
00DE	00	00	00	00DE	00	00	00
00DF	00	00	00	00DF	00	00	00
00E0	00	00	00	00E0	00	00	00
00E1	00	00	00	00E1	00	00	00
00E2	00	00	00	00E2	00	00	00
00E3	00	00	00	00E3	00	00	00
00E4	00	00	00	00E4	00	00	00
00E5	00	00	00	00E5	00	00	00
00E6	00	00	00	00E6	00	00	00
00E7	00	00	00	00E7	00	00	00
00E8	00	00	00	00E8	00	00	00
00E9	00	00	00	00E9	00	00	00
00EA	00	00	00	00EA	00	00	00
00EB	00	00	00	00EB	00	00	00
00EC	00	00	00	00EC	00	00	00
00ED	00	00	00	00ED	00	00	00
00EE	00	00	00	00EE	00	00	00
00EF	00	00	00	00EF	00	00	00
00F0	00	00	00	00F0	00	00	00
00F1	00	00	00	00F1	00	00	00
00F2	00	00	00	00F2	00	00	00
00F3	00	00	00	00F3	00	00	00
00F4	00	00	00	00F4	00	00	00
00F5	00	00	00	00F5	00	00	00
00F6	00	00	00	00F6	00	00	00
00F7	00	00	00	00F7	00	00	00
00F8	00	00	00	00F8	00	00	00
00F9	00	00	00	00F9	00	00	00
00FA	00	00	00	00FA	00	00	00
00FB	00	00	00	00FB	00	00	00
00FC	00	00	00	00FC	00	00	00
00FD	00	00	00	00FD	00	00	00
00FE	00	00	00	00FE	00	00	00
00FF	00	00	00	00FF	00	00	00

Рисунок 6 - Содержимое памяти обоих МК с выделенными байтами, которые были получены по SPI

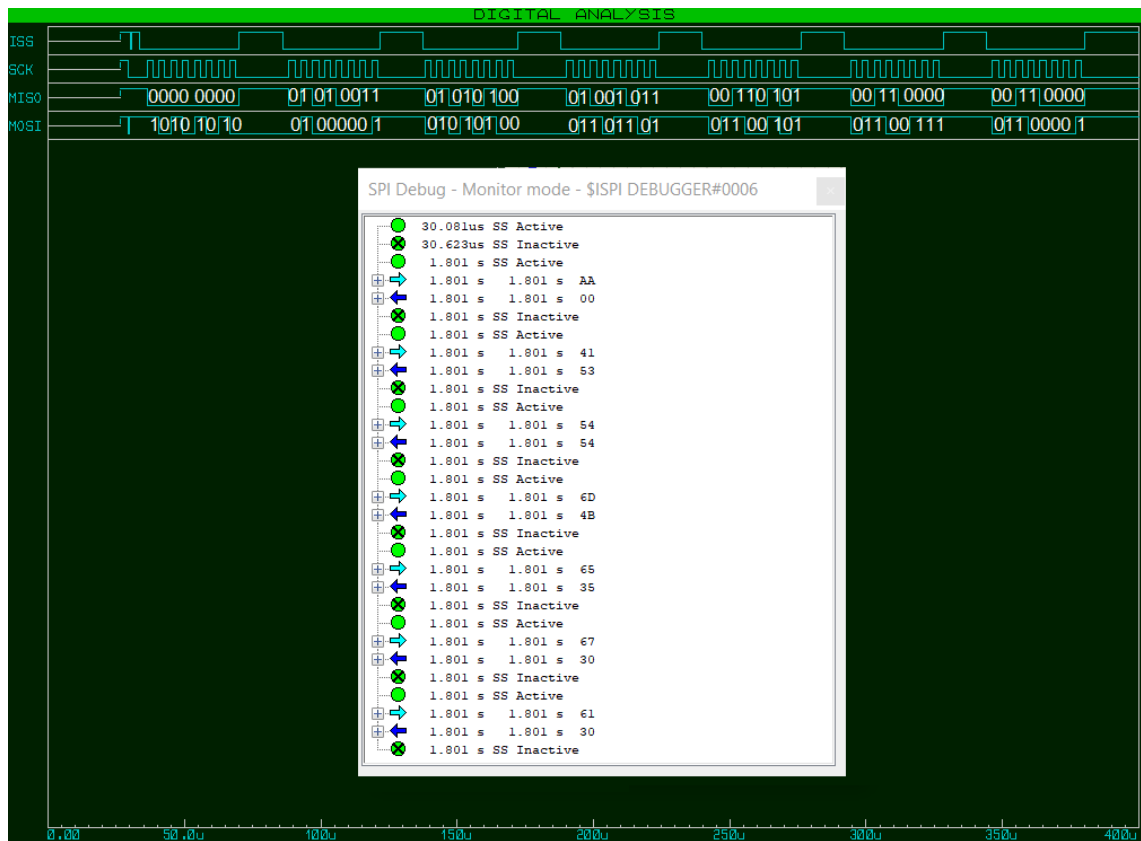


Рисунок 7 - Временная диаграмма с расшифровкой

### Задание 5. Передача данных в дуплексном режиме на макете

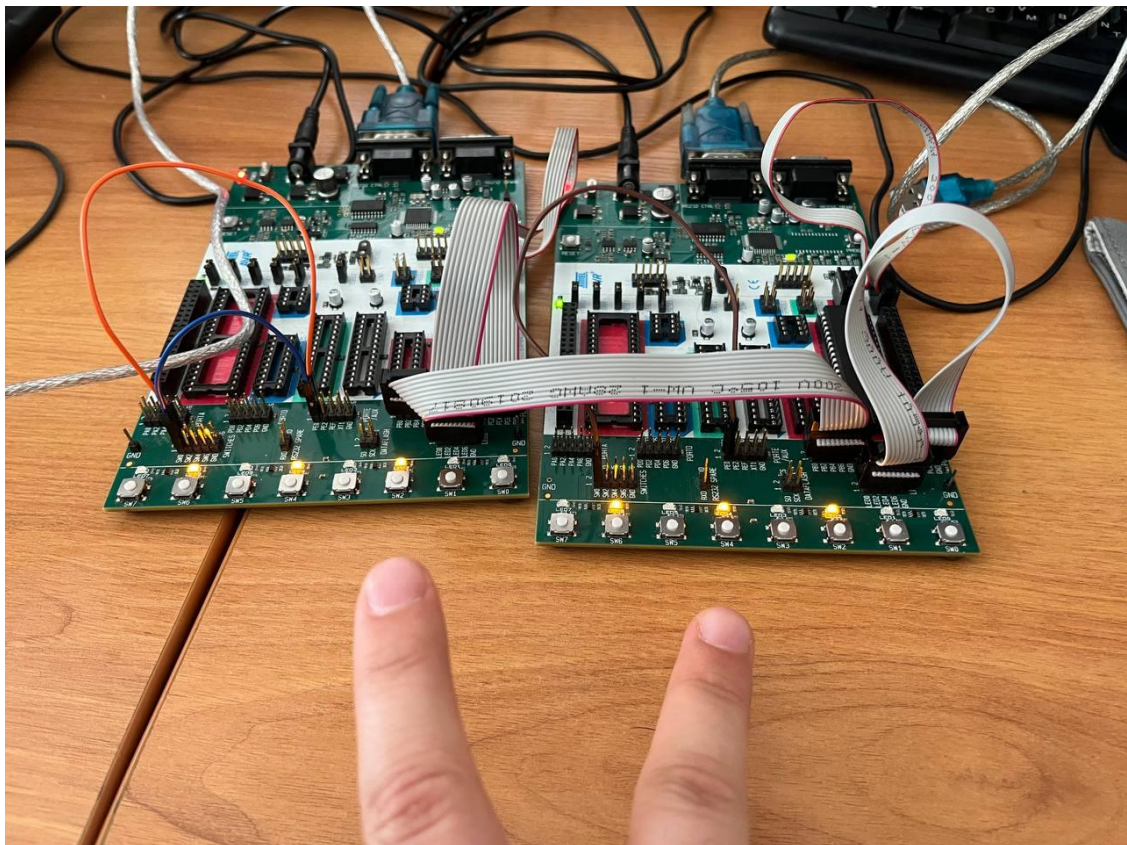


Рисунок 8 - Фотография макета с принятыми байтами на светодиодах

## Задание 6. Передача произвольного сообщения

### Листинг 2 - Программа ведущего

```
#include <avr/io.h>
#define F_CPU 3690000
#include <util/delay.h>
/* Кнопки управления - PE */
#define BUTTON_START 0
#define BUTTON_SHOW 1
#define PIN_SS 4
#define PIN_SS1 1
#define PIN_MOSI 5
#define PIN_SCK 7
/* Массив данных */
#define DATA_LENGTH 5
unsigned char data[DATA_LENGTH] = {'H', 'e', 'l', 'l', 'o'};
uint8_t currentByte = 0; /* Выводимый байт */
/* Байт для подачи сигнала о начале обмена */
#define START_SYM 0xAA
char SpiSendReceive(char c) {
    /* Выбор ведомого */
    PORTB &= ~(1<<PIN_SS1);
    /* Отправка байта */
    SPDR = c;
    /* Ожидание освобождения буфера */
    while( !(SPSR & (1<<SPIF)) )
        ;
    /* Завершение обмена с ведомым */
    PORTB |= (1<<PIN_SS1);
    /* Чтение байта */
    return SPDR;
}
int main() {
    /* Инициализация SPI */
    /* Настройка выводов MOSI, SCK, SS на вывод */
    DDRB = (1<<PIN_MOSI) | (1<<PIN_SCK) | (1<<PIN_SS);
    /* Включение SPI в режиме ведущего, частота передачи f_clk/16 */
    SPCR = (1<<SPE) | (1<<MSTR) | (1<<SPR0) | (1<<CPHA) | (1<<DORD);
    /* Инициализация портов ввода-вывода */
    /* Настройка PIN_SS1 на вывод */
    DDRB |= (1<<PIN_SS1);
    PORTB |= (1<<PIN_SS1);
    /* Настройка PE0, PE1 на ввод с подтягивающим резистором */
    PORTE = (1<<BUTTON_START) | (1<<BUTTON_SHOW);
    /* Настройка PC на вывод */
    DDRC = 0xFF;
    /* Погасить светодиоды, подключенные к PC */
    PORTC = 0xFF;
    /* Бесконечный цикл */
    while(1) {
```

```

/* Проверка нажатия кнопки START */
if (!(PINE & (1<<BUTTON_START))) {
/* Ожидание отпускания кнопки */
while (!(PINE & (1<<BUTTON_START)));
/* Отправка сигнала о начале обмена */
SpiSendReceive(START_SYM);
/* Задержка 10 мкс */
_delay_us(10);
/* Цикл передачи данных */
uint8_t i;
for (i = 0; i < DATA_LENGTH; i++) {
/* Отправка и прием i-го байта */
data[i] = SpiSendReceive(data[i]);
/* Задержка 10 мкс */
_delay_us(10);
}
}
/* Проверка нажатия кнопки SHOW */
if (!(PINE & (1<<BUTTON_SHOW))) {
/* Ожидание отпускания кнопки */
while (!(PINE & (1<<BUTTON_SHOW)))
;
/* Вывод данных с инверсией битов для светодиодов */
PORTC = ~data[currentByte];
currentByte = (currentByte + 1) % DATA_LENGTH;
}
}
return 0;
}

```

### Листинг 3 - Программа ведомого

```

#include <avr/io.h>
#include <avr/interrupt.h>
/* Кнопка SHOW - PE0 */
#define BUTTON_SHOW 0
/* SPI MISO - PB6 */
#define PIN_MISO 6
/* Массив данных */
#define DATA_LENGTH 5
unsigned char data[DATA_LENGTH] = {'w', 'o', 'r', 'l', 'd'};
uint8_t receivedBytes = 0; /* Счетчик принятых байтов */
uint8_t currentByte = 0; /* Выводимый байт */
/* Байт для подачи сигнала о начале обмена */
#define START_SYM 0xAA
/* Обработчик прерывания SPI_STC */
ISR(SPI_STC_vect) {
    unsigned char incomingByte = SPDR;
    if (incomingByte == START_SYM) {8

```



```

/* Загрузка 0-го байта для дальнейшей отправки */
SPDR = data[0];
}
else if (receivedBytes < DATA_LENGTH) {
SPDR = data[receivedBytes + 1];
data[receivedBytes] = incomingByte;
receivedBytes++;
/* Включить светодиоды по завершении приема */
if (receivedBytes == DATA_LENGTH) {
PORTC = 0x00;
}
}
}
int main() {
/* Инициализация SPI */
/* Включение SPI в режиме ведомого */
SPCR = (1<<SPE) | (1<<SPIE) | (1<<CPHA) | (1<<DORD);
/* Инициализация портов ввода-вывода */
/* Настройка MISO на вывод */
DDRB = (1<<PIN_MISO);
/* Настройка PE0 на ввод с подтягивающим резистором */
PORTE = (1<<BUTTON_SHOW);
/* Настройка PC на вывод */
DDRC = 0xFF;
/* Погасить светодиоды, подключенные к PC */
PORTC = 0xFF;
/* Глобальное разрешение прерываний */
sei();
/* Вывод полученных данных на светодиоды, байт за байтом */
/* Бесконечный цикл */
while(1) {
/* Проверка нажатия кнопки */
if (!(PINE & (1<<BUTTON_SHOW))) {
/* Ожидание отпускания кнопки */
while (!(PINE & (1<<BUTTON_SHOW)))
;

/* Вывод данных с инверсией битов для светодиодов */
PORTC = ~data[currentByte];
currentByte = (currentByte + 1) % DATA_LENGTH;
}
}
return 0;
}

```

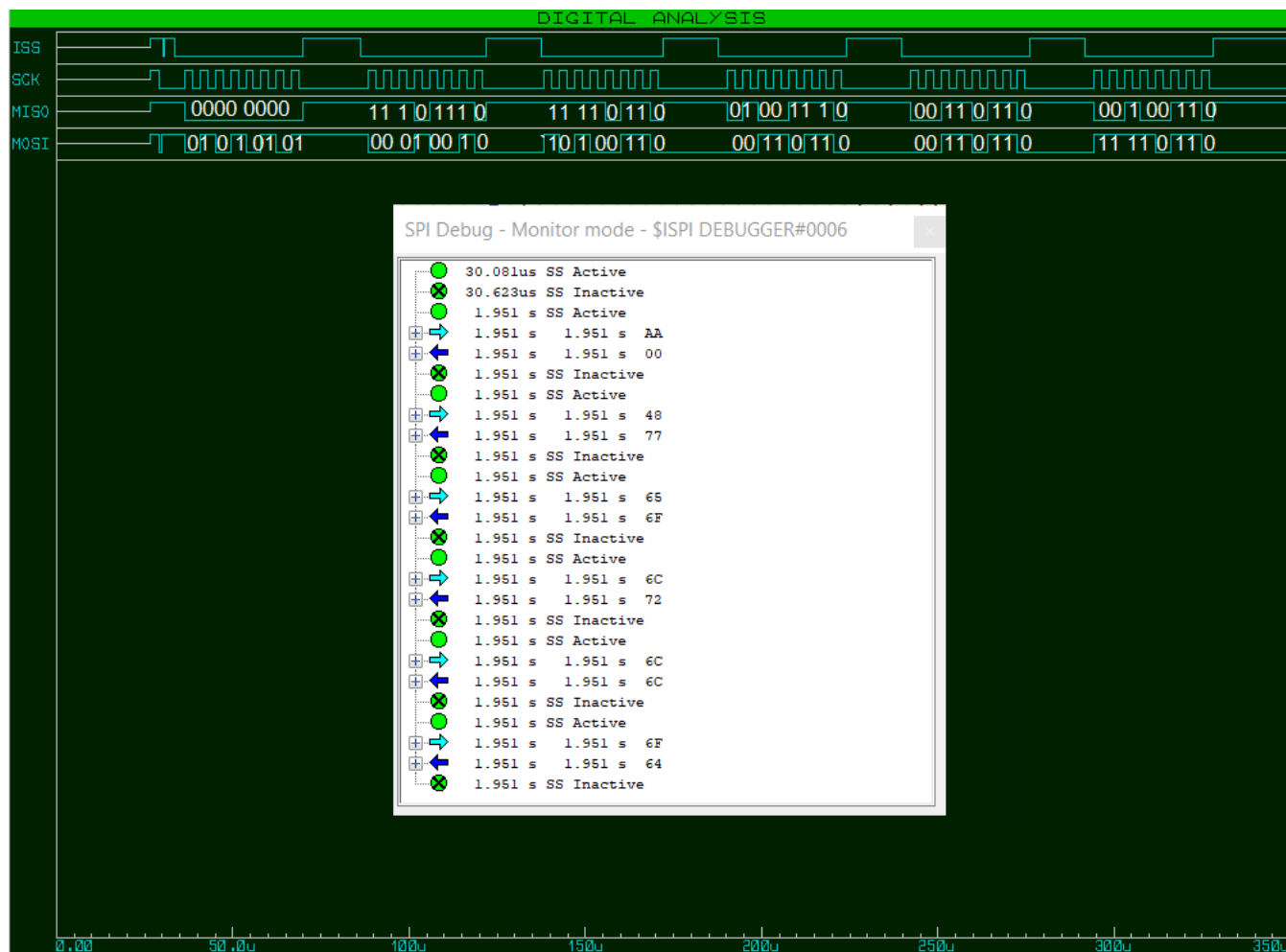


Рисунок 9 - Временная диаграмма с расшифровкой

## Вывод

Изучил структуры модуля SPI в микроконтроллере AVR, а так же программировал передачу и прием данных по протоколу SPI.