|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное 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**

**Отчет**

|  |  |
| --- | --- |
| **по лабораторной работе №** | 5 |

**Название:**

Работа последовательного канала SPI.

**Дисциплина:** Микропроцессорные системы.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-62Б |  |  | Ашуров Д.Н. Марчук И. С. |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Бычков Б. И. |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2022

**Вариант 2.**

**Цели работы:**

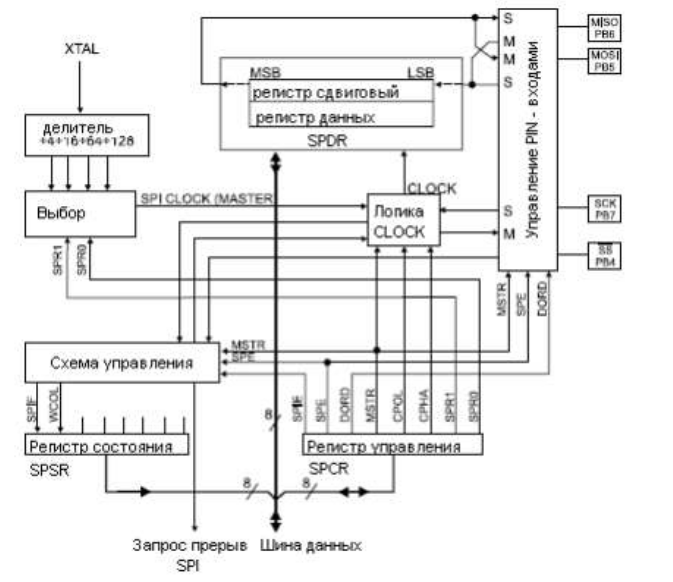
* изучение структуры канала последовательного интерфейса SPI (Serial Peripheral Interface);
* программирование приема-передачи данных по интерфейсу SPI;
* исследование опций (настроек) последовательного канала;
* моделирование и проверка работы канала в режиме обмена ведущий-ведомый.

**Ход работы.**

**Задание 1.**

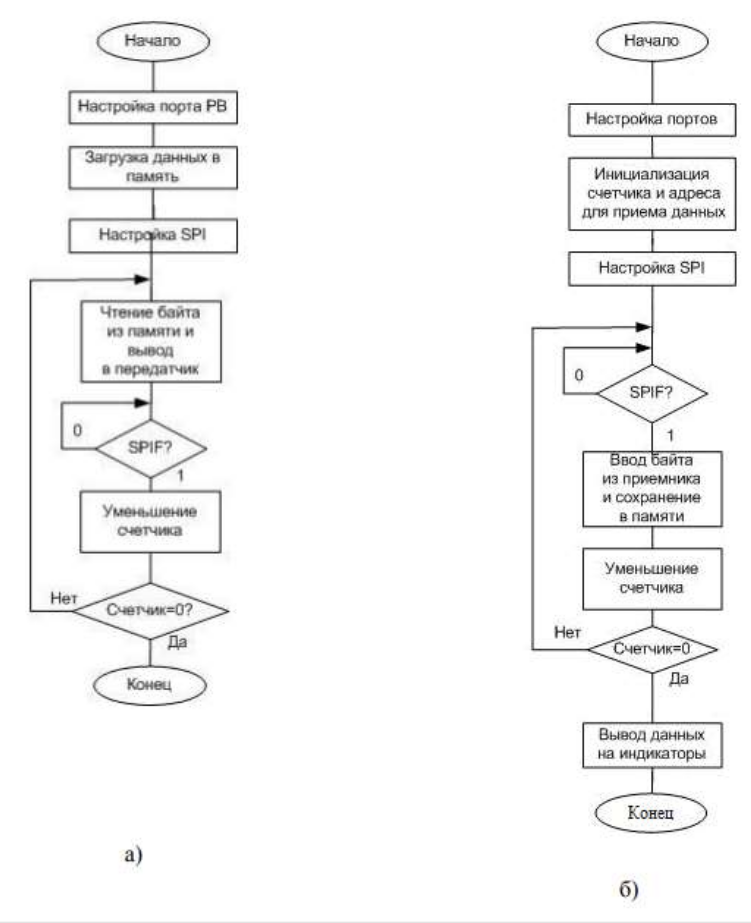
Создать в AVR Studio 4 проект для передачи данных с помощью программы. Проверить работу программы в режиме симуляции. Измерить длительность одного бита данных на линии PB5 и сравнить скорость передачи с запрограммированной.

Структурная схема порта SPI представлена на рисунке 1.



**Рисунок 1 — структурная схема порта SPI**

Схемы алгоритмов передачи и приема по каналу SPI представлены на рисунке 2.



**Рисунок 2 — Схема алгоритма передачи (а) и приема (б)**

Исходный код программы передачи:

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;Программа 5.1 для демонстрации работы канала SPI

;для передающего микроконтроллера ATx8515 в режиме MASTER.

;После сброса МК1 происходит передача трёх байтов,

;считываемых из ячеек SRAM по адресам из регистра Z

;Соединения: PB5мк1-PB5мк2, PB7мк1-PB7мк2, PB0мк1-PB4мк2

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;.include "8515def.inc" ;файл определений AT90S8515

.include "m8515def.inc" ;файл определений ATMEGA8515

.equ DD\_MOSI = 5

.equ DD\_SCK = 7

.def temp = r16 ;временный буфер

.def count = r17 ;счётчик

.org $000

rjmp init

;\*\*\*Инициализация МК

INIT: ldi temp,0xB1 ;DD\_MOSI, DD\_SCK, SS, PB0 для вывода

out DDRB,temp

ldi ZL,0x70 ;загрузка

ldi ZH,0x01 ; данных в

ldi temp,0x41 ; память

st Z+,temp ; данных

ldi temp,0x56 ; с использованием

st Z+,temp ; косвенной

ldi temp,0x52 ; адресации с

st Z+,temp ; постинкрементом

ldi ZL,0x70

ldi count,0x03 ;установка счётчика передач

;\*\*\*Настройка SPI в режиме MASTER на передачу данных

ldi temp,(1<<SPE)|(1<<MSTR)

out SPCR,temp

OUTPUT: sbi PORTB,0 ;переключение

nop ;PB0 из 1 в 0

cbi PORTB,0

ld temp,Z+ ;считывание байта из памяти

out SPDR,temp ;вывод байта в передатчик

Wait\_Transmit:

sbis SPSR,SPIF ; проверка флага передачи

rjmp Wait\_Transmit

dec count ;уменьшение счётчика на 1

brne OUTPUT

loop: rjmp loop

Исходный код программы приема:

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;Программа 5.2 для демонстрация работы канала SPI

;микроконтроллера ATx8515 в режиме SLAVE.

;После сброса МК2 происходит прием трёх байтов, записываемых в SRAM

;по адресам из регистра X.

;По окончании приёма загораются все светодиоды.

;При последовательном нажатии на SW5 (SHOW) происходит чтение данных

;и вывод их на светодиоды.

;Cоединения: SW5-PD5, шлейфом порт PC-LED

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;.include "8515def.inc" ;файл определений AT90S8515

.include "m8515def.inc" ;файл определений ATMEGA8515

.equ DD\_MISO = 6

.def temp = r16 ;временный буфер

.def count = r17 ;счётчик

.equ SHOW = 5 ;5-й вывод порта PD

.org $000

rjmp init

;\*\*\*Инициализация МК

INIT:

ldi temp,low(RAMEND) ;установка

out SPL,temp ; указателя стека

ldi temp,high(RAMEND) ; на последнюю

out SPH,temp ; ячейку ОЗУ

ldi temp,(1<<DD\_MISO)

out DDRB,temp

ldi temp,0xB0

out PORTB,temp

clr temp ;настройка

out DDRD,temp ; вывода

sbi PORTD,SHOW ; порта PD5 на ввод

ser temp ;настройка

out DDRC,temp ; выводов порта PC

out PORTC,temp ; на вывод

ldi count,3 ;установка счётчика байтов

ldi XL,0x80 ;в регистре X адрес, по которому

ldi XH,0x01 ; происходит запись принятых данных

;\*\*\*Настройка SPI в режиме SLAVE на приём данных

ldi temp,(1<<SPE)

out SPCR,temp

INPUT: sbis SPSR,SPIF ;проверка флага приема

rjmp INPUT

in temp,SPDR ;ввод байта из приёмника

st X+,temp ;сохранение байта в памяти

dec count

brne INPUT ;уменьшение счётчика на 1

rcall OUTLED ;вывод на индикацию

loop: rjmp loop

;\*\*\*Вывод на индикаторы\*\*\*

OUTLED: clr temp ;сигнализация - передача и

out PORTC,temp ; приём завершены

ldi XL,0x80 ;установка начального адреса

ldi count,3 ;установка счётчика байтов

WAIT\_SHOW: sbic PIND,SHOW ;ожидание нажатия

rjmp WAIT\_SHOW ; кнопки SHOW

ld temp,X+ ;считывание байта из памяти

com temp ;инвертирование и

out PORTC,temp ;вывод на светодиоды

rcall DELAY ;задержка

dec count ;если показаны не все данные,

brne WAIT\_SHOW ; то продолжение по нажатию SHOW

ret

;\*\*\*Задержка\*\*\*

DELAY: ldi r19,10

ldi r20,255

ldi r21,255

dd: dec r21

brne dd

dec r20

brne dd

dec r19

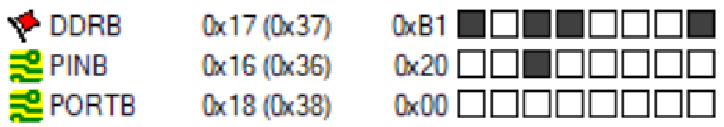
brne dd

ret

На рисунках 3-6 приведены показания часов и состояние порта B, зафиксированные с целью вычисления длительности одного бита данных на линии PB5.



**Рисунок 3 – показания часов в первый момент времени (установка PB5=1)**



**Рисунок 4 — состояние порта B в первый момент времени (установка PB5=1)**



**Рисунок 5 — показания часов во второй момент времени (PB5=0)**



**Рисунок 6 — состояние порта B во второй момент времени (PB5=0)**

Исходя из рисунков 3-6 *Vпередачи = 106 / (7.75 - 6.75) = 106 б/c.* При тактовой частоте 4 МГц и значении предделителя K=4: *V передачи теор = 4 \* 106/ 4 = 106 б/c.*

**Задание 2.**

Для совместной отладки программ и симуляции передачи/приема воспользоваться демонстрационной версией программы ISIS 6 Professional из пакета Proteus 6 Professional.

Схема для отладки программ в Proteus приведена на рисунке 7.

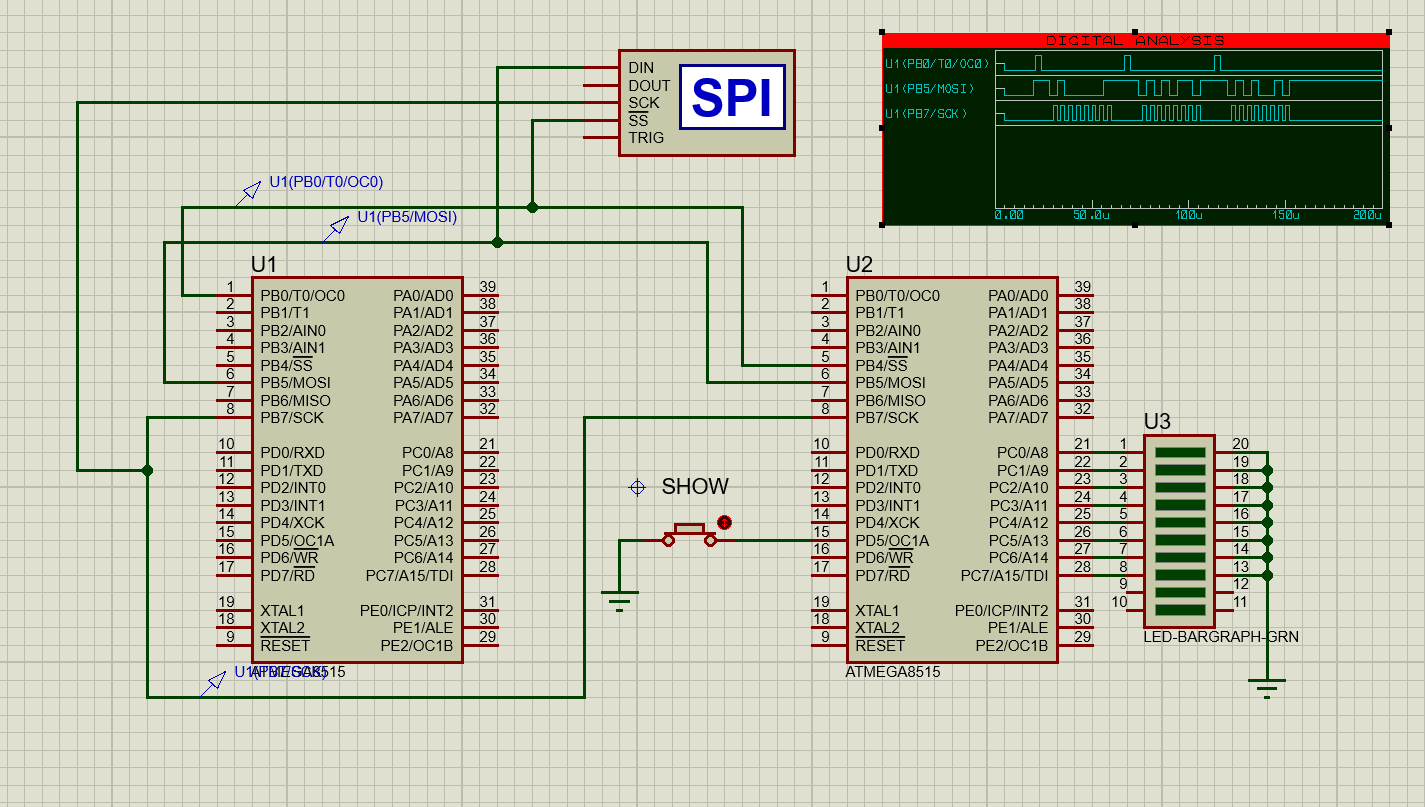


Рисунок 7 — схема для работы с SPI в Proteus

Временная диаграмма работы SPI представлена на рисунке 8.

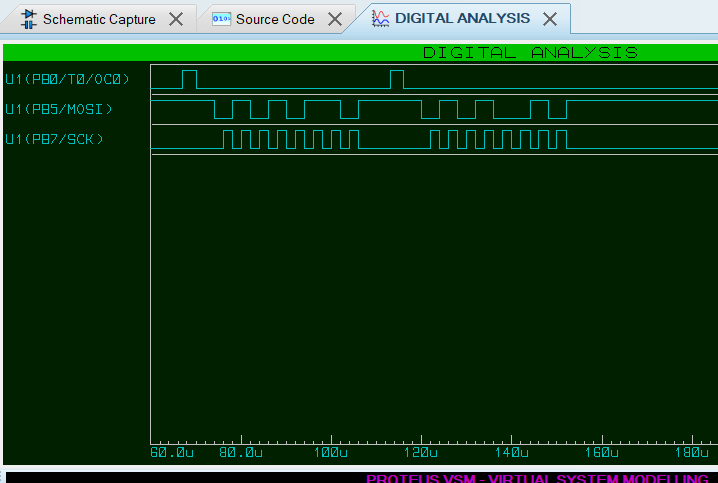


Рисунок 8 – временная диаграмма работы SPI

По временной диаграмме можно проверить корректность передачи данных:

1. Первый переданный байт – 01000001, соответствует числу 0х41
2. Второй переданный байт – 01010110, соответствует числу 0х56
3. Третий переданный байт – 01010010, соответствует числу 0х52

Окно SPI Debugger представлено на рисунке 9.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 9 — SPI Debugger

**Задание 3.**

Изменить обе программы, задав CPOL = 1. Проверить работу интерфейса на модели путем симуляции.

Для задания CPOL = 1 заменим строчки программы master-а

ldi temp,(1<<SPE)|(1<<MSTR)

out SPCR,temp

На

ldi temp,(1<<SPE)|(1<<MSTR)|(1<<CPOL)

out SPCR,temp

И строчки программы slave-a

ldi temp,(1<<SPE)

out SPCR,temp

На

ldi temp,(1<<SPE)|(1<<CPOL)

out SPCR,temp

Временная диаграмма работы интерфейса приведена на рисунке 10.

Изображение выглядит как текст, снимок экрана, электроника, дисплей

Автоматически созданное описание**Рисунок 10 — временная диаграмма работы SPI при CPOL = 1**

По временной диаграмме можно проверить корректность передачи данных:

1. Первый переданный байт – 01000001, соответствует числу 0х41
2. Второй переданный байт – 01010110, соответствует числу 0х56
3. Третий переданный байт – 01010010, соответствует числу 0х52

Сигнал SCK стал инверсным, остальные сигналы не изменились.

**Задание 4.**

Изменить обе программы, задав DORD =1. Проверить работу интерфейса на модели.

Для задания DORD = 1 заменим строчки программы master-а

ldi temp,(1<<SPE)|(1<<MSTR)

out SPCR,temp

На

ldi temp,(1<<SPE)|(1<<MSTR)|(1<<DORD)

out SPCR,temp

И строчки программы slave-a

ldi temp,(1<<SPE)

out SPCR,temp

На

ldi temp,(1<<SPE)|(1<<DORD)

out SPCR,temp

Временная диаграмма работы интерфейса приведена на рисунке 11. Изображение выглядит как текст, снимок экрана, электроника, дисплей

Автоматически созданное описание

**Рисунок 11 — временная диаграмма работы SPI при DORD = 1**

По временной диаграмме можно проверить корректность передачи данных:

1. Первый переданный байт – 10000010, соответствует числу 0х82
2. Второй переданный байт – 01101010, соответствует числу 0х6А
3. Третий переданный байт – 01001010, соответствует числу 0х4А

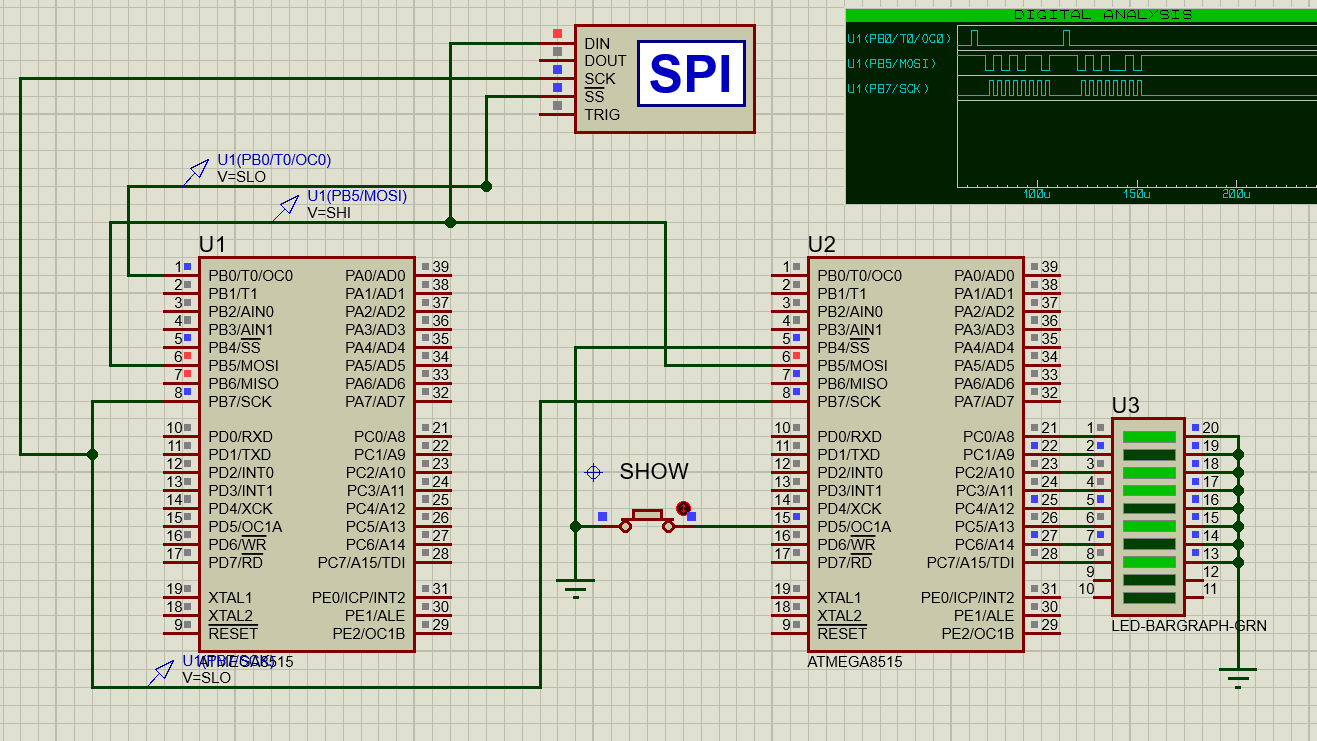
Биты каждого из чисел стали передаваться по линии MOSI в обратном

порядке.

**Задание 5.**

Изменить проект, подключив к входу PB4 МК2 уровень GND (логический «0»). Проверить работу интерфейса на модели.

В результате модификации схемы данные были переданы в МК2. Результаты представлены на рисунке 12.



**Рисунок 12 — работа схемы при PB4 = 0**

**Задание 6.**

Убедившись в правильной работе программ, можно перейти к экспериментальной проверке в STK500.

**Задание 7.**

Изменить программы для передачи-приема символьного набора 1234, хранимого в памяти программ передающего микроконтроллера, используя механизм выборки, использованный в работе 3. Проверить работу канала SPI.

Соединив вход PB4 приемного микроконтроллера с выводом GND, повторить передачу. Сравнить наблюдаемые результаты с результатами симуляции.

Код программы:

.include "m8515def.inc"

.equ DD\_MOSI = 5

.equ DD\_SCK = 7 ;мк1-PB7мк2, PB0мк1-PB4мк2

.def res = r18 ;результат операции (сумма, разность,

.def temp = r16

.def count = r17 ;мк1-PB7мк2, PB0мк1-PB4мк2

.macro vvod ;ввод операнда

lpm

mov @0,r0

mov res, r0 ; и пересылка в регистр операнда

adiw zl, 1 ;увеличение указателя адреса на 1

.endmacro

.org $000

rjmp init

INIT: ldi temp,0xB1

out DDRB,temp

ldi temp,low(RAMEND)

out SPL,temp

ldi temp,high(RAMEND)

out SPH,temp

ldi ZL,low(tabl\_op\*2)

ldi ZH,high(tabl\_op\*2)

ldi count,4

clr temp

ldi temp,(1<<SPE)|(1<<MSTR)

out SPCR,temp

OUTPUT:

sbi PORTB,0

nop

cbi PORTB,0

vvod temp

out SPDR,temp

Wait\_Transmit:

sbis SPSR,SPIF

rjmp Wait\_Transmit

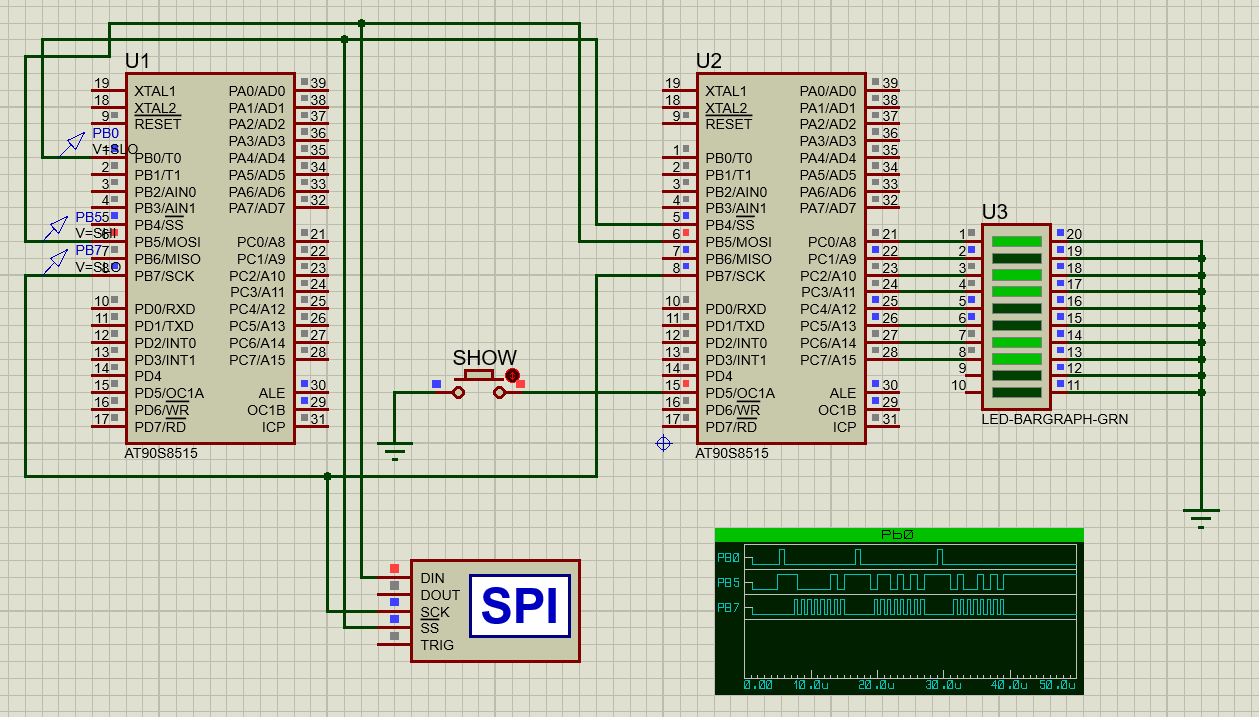
dec count

brne OUTPUT

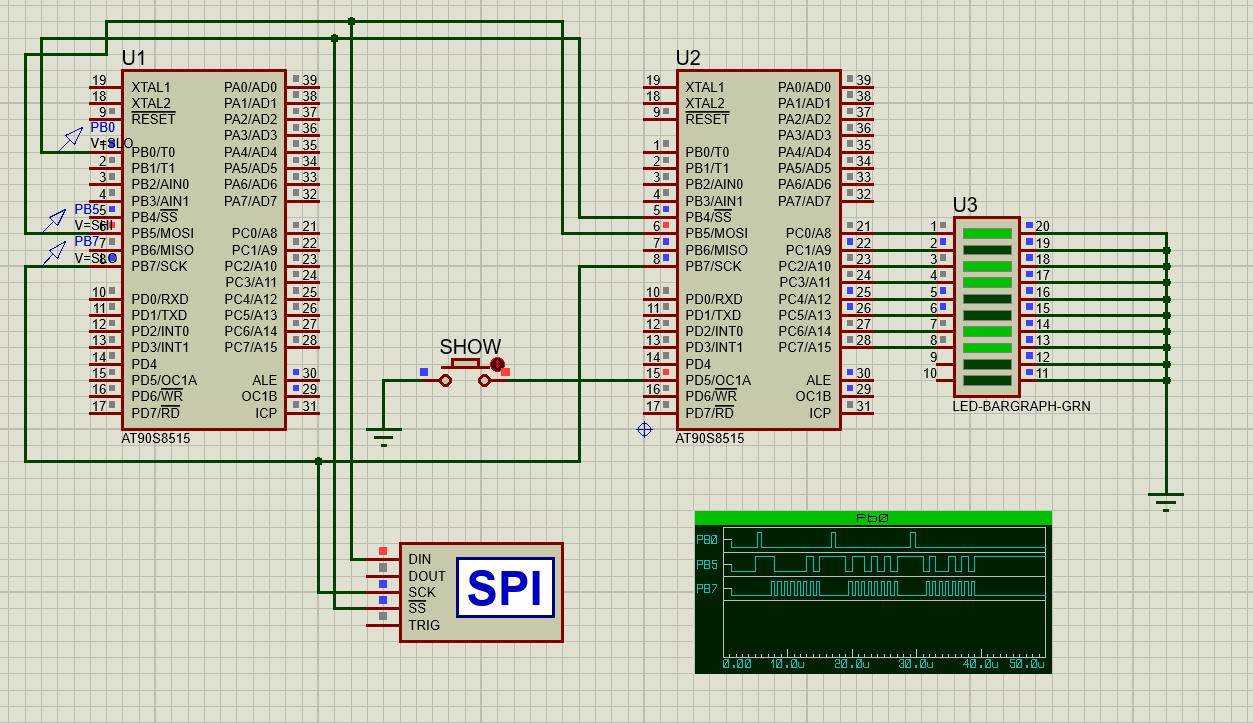
loop: rjmp loop

tabl\_op: .db 0x31, 0x32,0x33,0x34

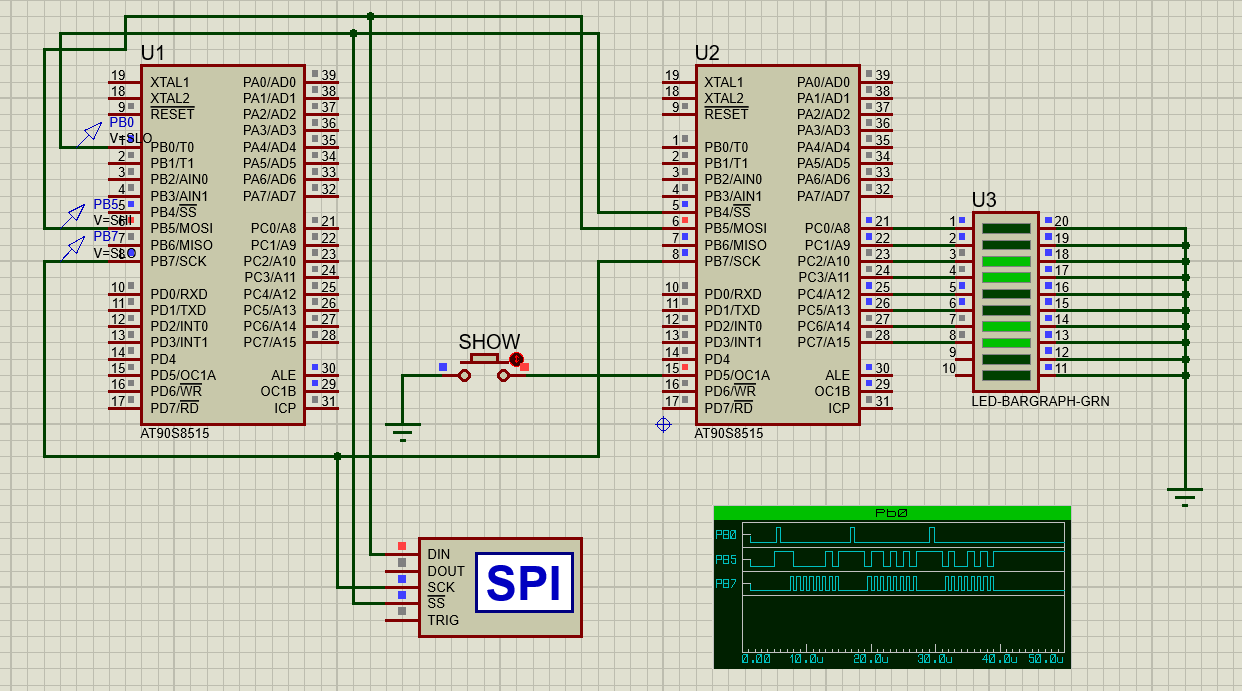
На рисунках 13,14,15 и 16 продемонстрирована передача символов цифр 1,2,3 и 4 соответственно.



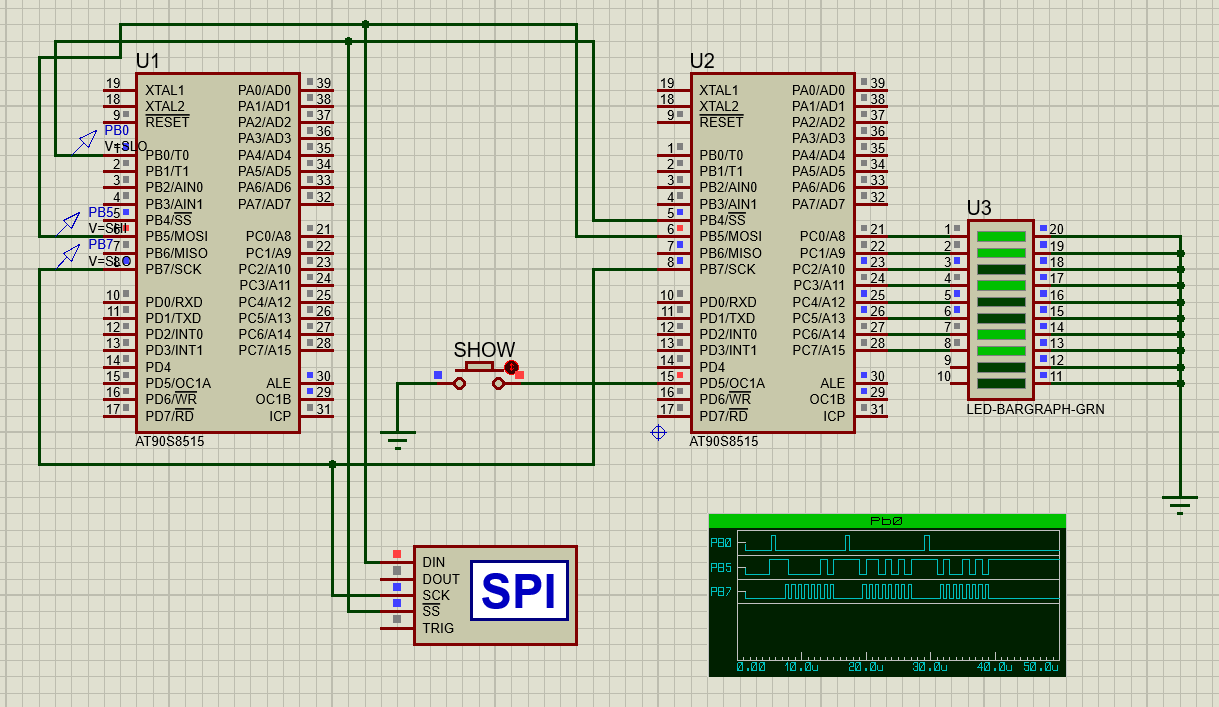
**Рисунок 13 — передача символа цифры 1**



**Рисунок 14 — передача символа цифры 2**



**Рисунок 15 — передача символа цифры 3**



**Рисунок 16 — передача символа цифры 4**

При соединении входа PB4 приемного микроконтроллера с выводом GND результаты аналогичны пункту 5.

**Вывод:** в ходе данной лабораторной работы были изучены теоретические основы и получены практические навыки работы с каналом SPI. По линии порта MOSI осуществляется передача полезных данных, по линии SCK передаются импульсы, по которым устройство-приемник считывает данные с линии MOSI, по линии SS производится выбор устройства-приемника и сигнализируется о начале приема очередного байта полезных данных. При установке бита CPOL сигнал на линии SCK становится инверсным, то есть полезные данные считываются во время нулевых импульсов. При установке бита DORD биты полезных данных передаются в обратном порядке.