Подключение семисегментных индикаторов с помощью сдвигового регистра 74HC164. Делаем простой вольтметр на Attiny13.

Цель: Изучить методику использования сдвиговых регистров с микроконтроллерами.

Применение сдвиговых регистров таких как **74HC164** обусловлено возможностью управления большим количеством выходных линий с помощью 3-х линий управления, например, эти регистры можно использовать для подключения гирлянд или матриц из светодиодов. Они широко используются в бытовой и промышленной аппаратуре для вывода данных на индикаторы или опросе матричной клавиатуры. Из управляющих входов в этой микросхеме есть: DATA, RESET, CLK и есть 8-выходов, получается эта микросхема преобразует последовательный код в параллельный. Вход RESET который сбрасывает установленные значения на выходе обычно не используется, так как занимает лишний вывод у микроконтроллера, он должен всегда быть поднятым к Vcc. Обнулять значения можно посылкой из 8бит логических единиц. Вход CLK продвигает значение по регистру. На вход DATA поступают данные в последовательном режиме.

Для заполнения регистра нужно выполнить такую последовательность:

1. Сравниваем старший бит передаваемого байта с нулем, если он равен нулю передаем в DATA ноль, если не равен передаем единицу;
2. Сдвигаем передаваемое значение на 1 разряд;
3. На тактовый вход CLK подаем ноль;
4. На тактовый вход CLK подаем единицу;
5. Проделываем эти операции пока не передадим весь байт нашего значения.

Подробнее со сдвиговым регистром разберемся на примере вольтметра постоянного напряжения до 55 Вольт. Микроконтроллер на этот раз будет **Attiny13**, ведь у него всего 5 рабочих линий порта PB он как раз нам подходит. Работает он от внутреннего генератора частотой 4,8 MHz. АЦП настроен стандартно, вход ADC1. Каждый сегмент индикаторов подключается через резистор 220 Ом, индикаторы используются с общим катодом. Для предотвращения мерцания сегментов индикаторов их общие катоды подключаются через транзистор, который включается только тогда, когда все значение будет передано в регистр. Схема вольтметра для индикаторов с общим катодом показана ниже:

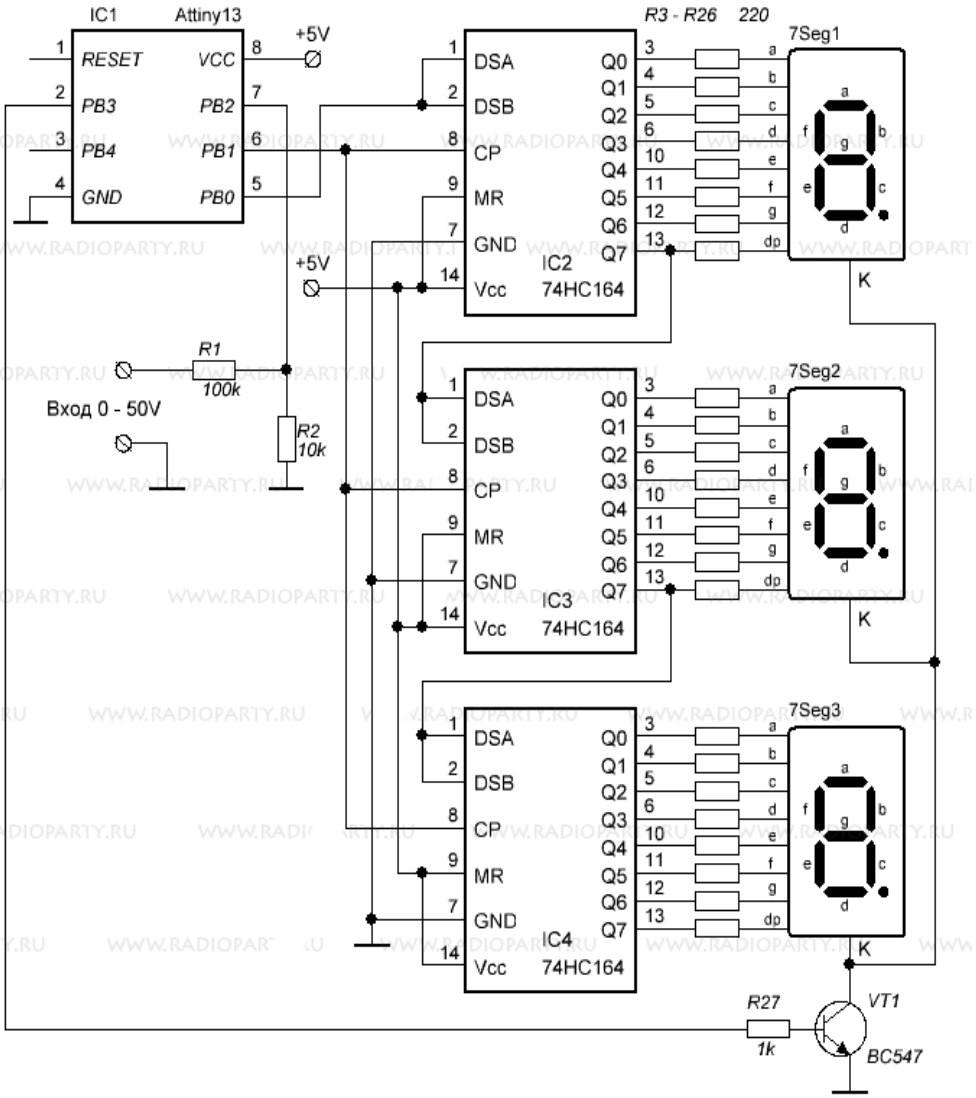


Рис. 41

Исходный код примера для индикаторов с общим катодом с подробными комментариями:

```
// Управление семисегментными индикаторами через регистр сдвига 74HC164(OK)
#include <avr/io.h>
#include <util/delay.h>
// Массив комбинаций сегментов
unsigned char SEGMENTE[] =
{
0x3F, // 0
0x06, // 1
```

```
0x5B, // 2
0x4F, // 3
0x66, // 4
0x6D, // 5
0x7D, // 6
0x07, // 7
0x7F, // 8
0x6F  // 9
};
unsigned int volt;
// Функция сдвига
void write_byte(unsigned char data)
{
  for(unsigned char i = 0; i < 8 ; i++)
  {
    if((data & 0x80) != 0) // Сравниваем 8-й бит с нулем
    PORTB |= (1 << PB0); // DATA 1
    else
    PORTB &= ~(1 << PB0); // DATA 0
    data = data << 1; // Сдвигаем биты
    PORTB |= (1 << PB1); // CLK 1
    PORTB &= ~(1 << PB1); // CLK 0
  }
}
int main(void)
{
DDRB |= (1 << PB3)|(1 << PB1)|(1 << PB0);
PORTB = 0x00;
ACSR |= (1 << ACD); // Выключаем аналоговый компаратор
DIDR0 |= (1 << ADC1D); // Отключаем неиспользуемые цифровые входы
ADMUX |= (1 << MUX0); // Вход ADC1
ADCSRA |= (1 << ADEN) // Разрешение АЦП
      |(1 << ADPS2)|(1 << ADPS1); // Предделитель на 64
```

```
while(1)
{
// Рассчитаем максимальное входное напряжение на делителе
// Umax = Uin*(R1+R2)/R2
// Umax = 5*(100k+10k)/10k = 55V
// Рассчитаем коэффициент делителя напряжения
// K = (R1+R2)/R2
// K = (100k + 10k)/10k = 11
// Рассчитаем результат преобразования в мВ
// U = ADC*Uref*K*100/1024
ADCSRA |= (1 << ADSC); // Начинаем преобразование
while (ADCSRA & (1 << ADSC)){} // Ждем завершения преобразования
volt = ((unsigned long)ADC*5*11*100)/1024;
PORTB &= ~(1 << PB3); // Выключаем индикатор
write_byte(SEGMENTE[volt%100/10]); // Выводим 1 разряд
write_byte((SEGMENTE[volt%1000/100])|0x80); // Выводим 2 разряд
write_byte(SEGMENTE[volt%10000/1000]); // Выводим 3 разряд
PORTB |= (1 << PB3); // Включаем индикатор
_delay_ms(100);
}
}
```

Схема вольтметра для индикаторов с общим анодом:

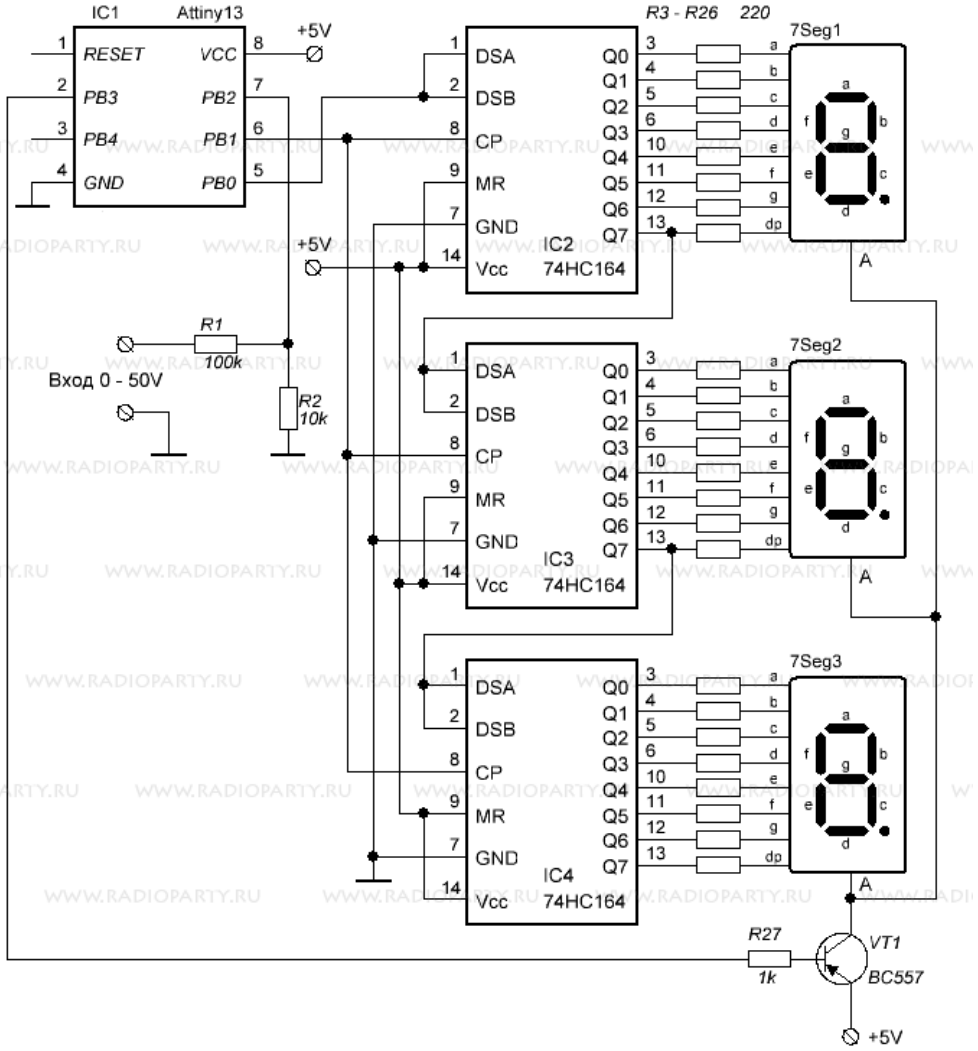


Рис. 42

Исходный код примера для индикаторов с общим анодом с подробными комментариями:

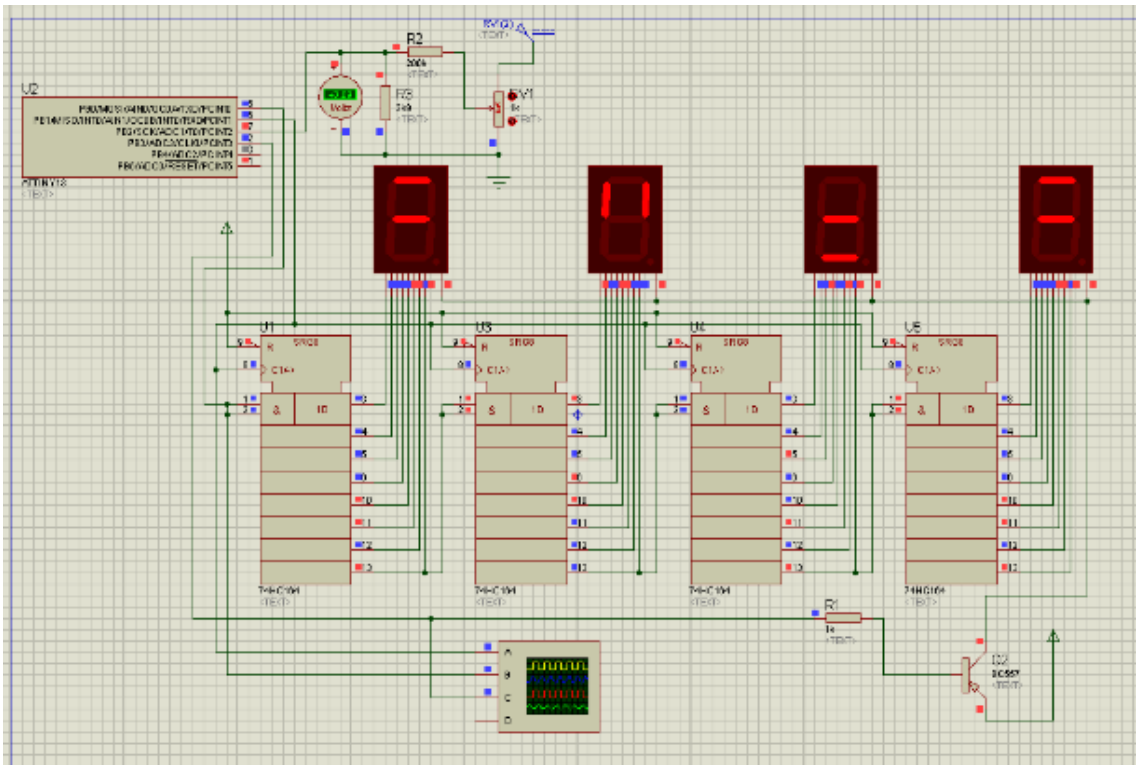
```
// Управление семисегментными индикаторами через регистр сдвига 74HC164(OA)
#include <avr/io.h>
#include <util/delay.h>
// Массив комбинаций сегментов
unsigned char SEGMENTE[] =
{
0x3F, // 0
```

```
0x06, // 1
0x5B, // 2
0x4F, // 3
0x66, // 4
0x6D, // 5
0x7D, // 6
0x07, // 7
0x7F, // 8
0x6F  // 9
};
unsigned int volt;
// Функция сдвига
void write_byte(unsigned char data)
{
  for(unsigned char i = 0; i < 8 ; i++)
  {
    if((data & 0x80) != 0) // Сравниваем 8-й бит с нулем
    PORTB |= (1 << PB0); // DATA 1
    else
    PORTB &= ~(1 << PB0); // DATA 0
    data = data << 1; // Сдвигаем биты
    PORTB |= (1 << PB1); // CLK 1
    PORTB &= ~(1 << PB1); // CLK 0
  }
}
int main(void)
{
DDRB |= (1 << PB3)|(1 << PB1)|(1 << PB0);
PORTB = 0x00;
ACSR |= (1 << ACD); // Выключаем аналоговый компаратор
DIDR0 |= (1 << ADC1D); // Отключаем неиспользуемые цифровые входы

ADMUX |= (1 << MUX0); // Вход ADC1
```

```
ADCSRA |= (1 << ADEN) // Разрешение АЦП
      |(1 << ADPS2)|(1 << ADPS1); // Предделитель на 64
while(1)
{
// Рассчитаем максимальное входное напряжение на делителе
// Umax = Uin*(R1+R2)/R2
// Umax = 5*(100k+10k)/10k = 55V
// Рассчитаем коэффициент делителя напряжения
// K = (R1+R2)/R2
// K = (100k + 10k)/10k = 11
// Рассчитаем результат преобразования в мВ
// U = (ADC*Uref*K*100)/1024
ADCSRA |= (1 << ADSC); // Начинаем преобразование
while (ADCSRA & (1 << ADSC)){} // Ждем завершения преобразования
volt = ((unsigned long)ADC*5*11*100)/1024;
PORTB |= (1 << PB3); // Выключаем индикатор
write_byte(~SEGMENTE[volt%100/10]); // Выводим 1 разряд
write_byte(~((SEGMENTE[volt%1000/100])|0x80)); // Выводим 2 разряд
write_byte(~SEGMENTE[volt%10000/1000]); // Выводим 3 разряд
PORTB &= ~(1 << PB3); // Включаем индикатор
_delay_ms(100);
}
}
```

Пример выполнения в среде Proteus:



Задания для выполнения:

1. Изучить методику использования сдвиговых регистров с микроконтроллерами.
2. Смоделировать схему вольтметра в среде Proteus.

Результаты работы отправить на e-mail: rasov@rambler.ru с темой Вольтметр_ФИО