

Министерство образования и науки
Российской Федерации

Федеральное государственное бюджетное образовательное
учреждение высшего образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра конструирования узлов и деталей радиоэлектронной аппаратуры
(КУДР)

А.А. Бомбизов, А.Г. Лоцилов

ВЫВОД ИНФОРМАЦИИ

Методические указания к выполнению
лабораторной и самостоятельной работы
по дисциплине «Программирование микроконтроллеров»

Томск 2017

1 Введение

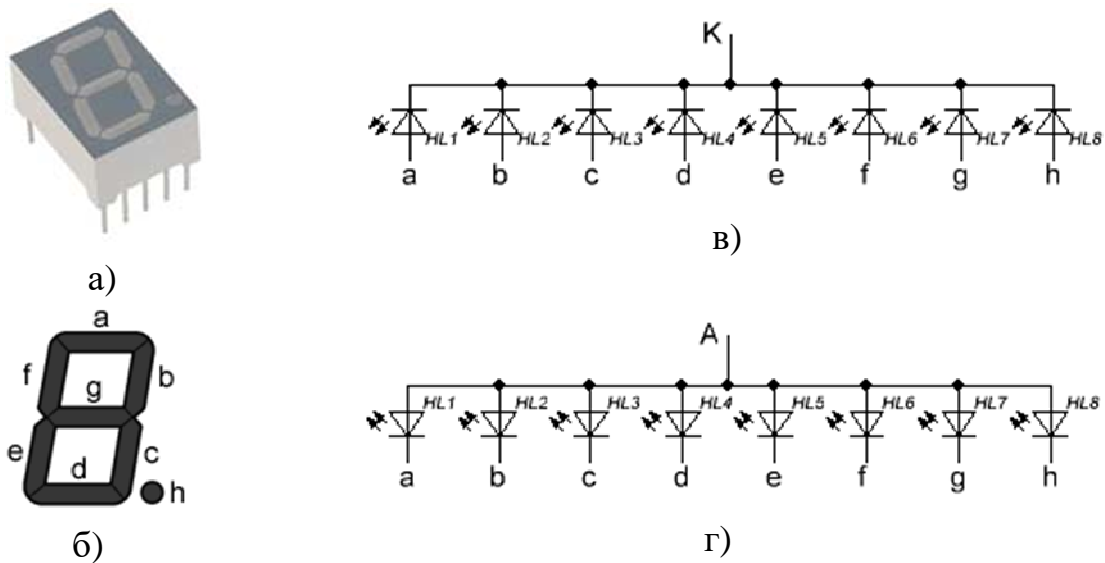
Регистрация и обработка внешних данных подразумевает визуализацию полученной информации. В реальной практике для индикации полученных данных могут использоваться звуковые, светодиодные устройства, дисплеи и др.

Целью настоящей работы является освоение работы семисегментного индикатора, управляемого сдвигowymi регистрами.

2 Краткая теория

Семисегментный индикатор представляет собой матрицу из семи светодиодов, размещенных таким образом, чтобы зажигая их в разных сочетаниях, можно было бы отобразить любую десятичную цифру, а также специальные символы. Кроме этого индикатор дополняется еще одним сегментом, который предназначен для отображения десятичной точки. На рисунке 1(а) изображен внешний вид индикатора. Принято каждый сегмент индикатора обозначать латинской буквой: *a*, *b*, *c*, *d*, *e*, *f*, *g* (рисунок 1(б)). Точка обозначается буквой *h* или *dp*.

По схеме включения семисегментные индикаторы подразделяются на индикаторы с общим катодом и с общим анодом (рисунок 1 (в–г)).



а) внешний вид; б) обозначения сегментов;

в) схема индикатора с общим катодом;

г) схема индикатора с общим анодом.

Рисунок 1 – Семисегментный индикатор

Для подключения одного семисегментного индикатора к микроконтроллеру достаточно его сегменты подключить через токоограничивающие резисторы к выводам порта. Общий вывод подключить к линии другого порта микроконтроллера. В зависимости от того какую

цифру надо вывести, в порт выводим двоичный код этой цифры, ссылаясь на тип подключенного индикатора (с общим анодом или катодом) на общий провод подаем плюс или минус. Для удобства можно сделать таблицу кодов для индикатора. Если подключение такое: $PD7 - h$, $PD6 - g$, $PD5 - f$, $PD4 - e$, $PD3 - d$, $PD2 - c$, $PD1 - b$, $PD0 - a$, то для отображения цифры 1 нужно зажечь сегменты b и c , а для этого в порт D нужно вывести такой двоичный код $0b00000110$, то есть зажечь сегменты b и c .

Для отображения цифровых данных одного семисегментного индикатора обычно недостаточно. В таких случаях к микроконтроллеру подключают сразу несколько индикаторов. Однако, из-за отсутствия достаточного количества выводов у микроконтроллера применяют специальные методы. Один из таких методов это динамическая индикация. Режим динамической индикации применяют для построения многоразрядных индикаторов. При таком режиме разряды индикатора работают не одновременно, а по очереди. Переключение разрядов происходит с большой скоростью (50 и более Гц), из-за этого человеческий глаз не замечает, что индикаторы работают по очереди. Так как у светодиодов очень малая инерционность, сменяющиеся разряды сливаются в одно изображение. В этом режиме в каждый момент времени работает только один разряд, включаются по очереди, начиная с первого, заканчивая последним, затем все начинается сначала.

Одним из вариантов такого подключения является счетверенный индикатор, изображенный на рисунке 2.



Рисунок 2 – Внешний вид счетверенного индикатора

Для определения способа вывода информации на такой вид индикатора, необходимо обратиться к его электрической принципиальной схеме (рисунок 3) и обозначениям активных элементов индикатора (рисунок 4). Здесь можно отметить, что аноды в пределах каждой цифры объединены и подключены к выводам 6, 8, 9, 12. В то же время катоды сегментов цифр соединены между собой по иному признаку: сегменты объединены по четырем цифрам. Это дает возможность для зажигания каждой цифры, используя один и тот же алгоритм и схему включения.

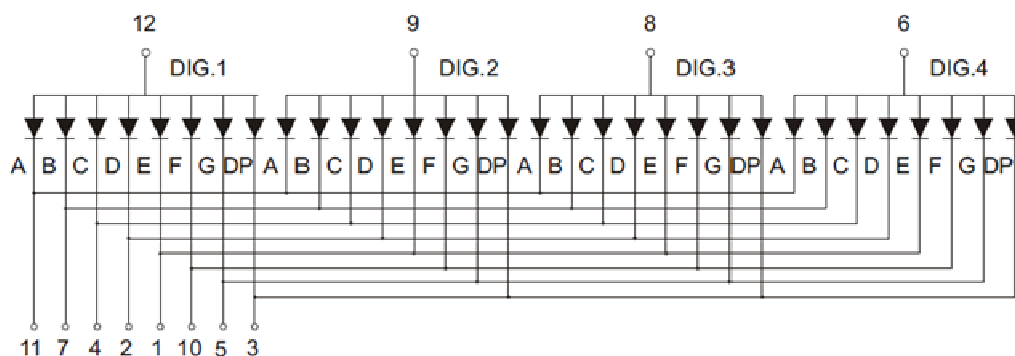


Рисунок 3 – Схема индикатора [1]

Например, опираясь на схему и обозначения активных элементов, высветить на индикаторе для цифры DIG.1 единицу можно следующим образом:

- 1) подать логическую единицу на вывод 12;
- 2) подать на выводы 3, 5, 10, 1, 2, 11 логическую единицу;
- 3) подать на выводы 7, 4 логический ноль.

Такая комбинация будет содействовать протеканию электрического тока только через светодиоды *B* и *C*, что обеспечит их свечение. Далее, если быстро менять комбинации по очереди для отображения каждой цифры, то можно вывести статичное для глаза изображение.

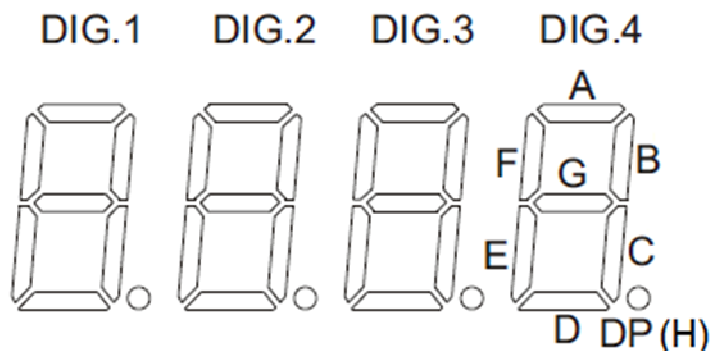


Рисунок 4 – Обозначения активных элементов счетверенного индикатора

Такой подход позволяет сократить используемое количество выводов с 32х (четыре одиночных индикатора) до 12, при этом индикатор используется счетверенный. Но в условиях ограниченного количества портов ввода-вывода микроконтроллера часто возникает необходимость еще сократить количество управляющих индикатором выводов. Для этих целей, как правило, используется сдвиговый регистр.

Микросхема 74НС595 [2] – один из самых распространённых сдвиговых регистров: синхронный, с регистром данных (*latch clock*). Она позволяет увеличивать количество выходов микроконтроллера.

Структура микросхемы изображена на рисунке 5. Чип преобразует входной последовательный сигнал на одном входе *Serial Data Input A* (номер

вывода 14) в выходной параллельный на восьми выводах $QA-QH$. Последовательная передача синхронна и использует тактовый сигнал. Для этого применяется дополнительный вывод *Shift Clock* (вывод 11), по которому передается тактовый сигнал. То есть для передачи логического нуля в сдвиговый регистр требуется на вывод 11 подать логическую единицу, а на вывод 14 логический ноль. Для передачи в сдвиговый регистр логической единицы на оба вывода должна быть подана единица. Регистр представляет собой цепочку последовательно связанных между собой триггеров $SRa-SRh$, с каждым тактовым сигналом значения из предыдущего $SR(x)$ передаются в следующий $SR(x+1)$. Таким образом, передавая последовательность из восьми бит данных синхронно с тактовым сигналом, будут заполнены значениями все триггеры $SR(x)$. Далее данные из триггеров SRx необходимо направить в параллельный регистр, состоящий из триггеров LRx . Для этого достаточно подать логическую единицу на вывод *Latch Clock* (вывод 12). После этих действий входная последовательность будет выведена на выводы $QA-QH$.

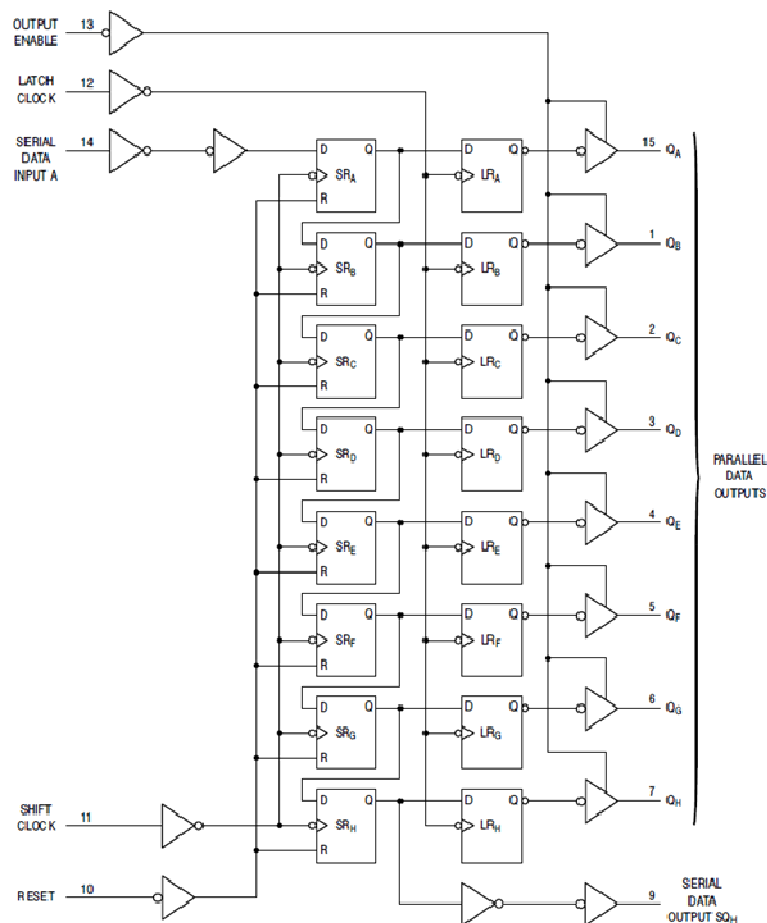


Рисунок 5 – Структурная схема сдвигового регистра 74НС595 [2]

Временная диаграмма работы сдвигового регистра изображена на рисунке 6, по которой можно проследить каким образом последовательные данные преобразуются в параллельные.

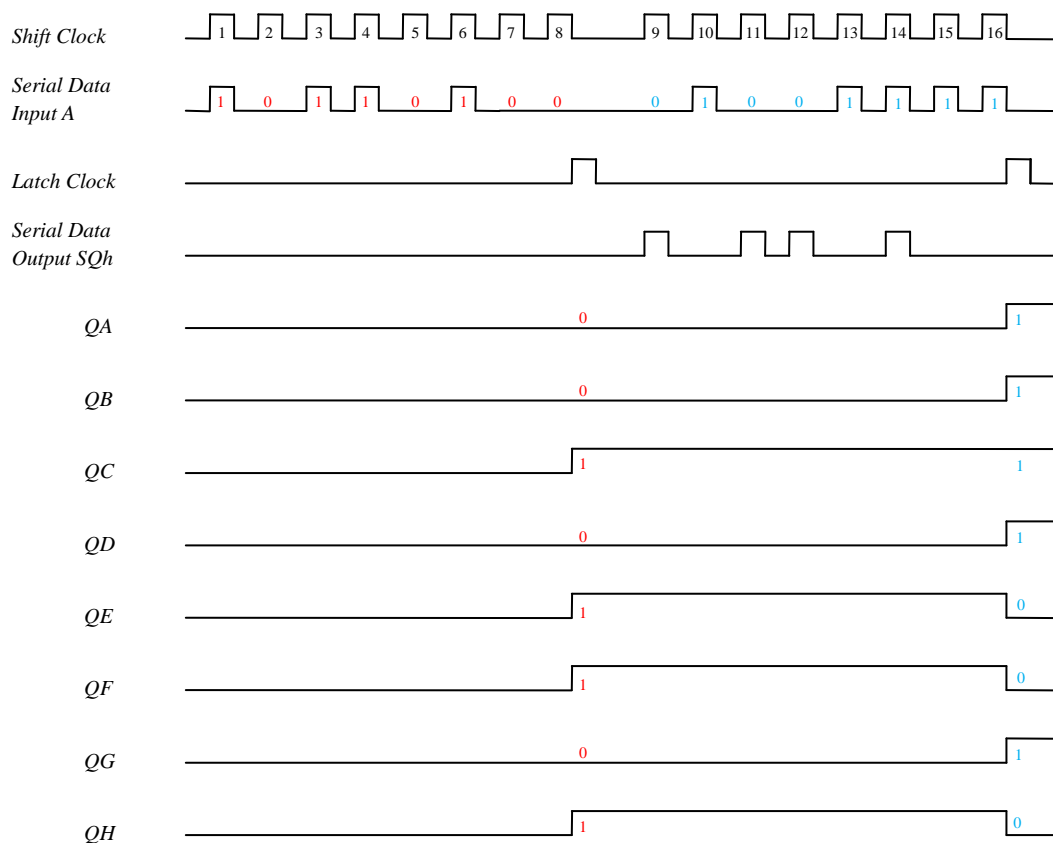


Рисунок 6 – Временная диаграмма передачи данных на сдвиговый регистр

Из регистров 74HC595 можно делать каскады, подключая один за другим (через вывод *Serial Data Output SQh*), и таким образом из тех же трех управляющих линий получать 16, 24, 32 и т.д. цифровых выходов. Таким образом выполнено подключение сдвиговых регистров в *Arduino UNO Multifunction Shield* (рисунок 7).

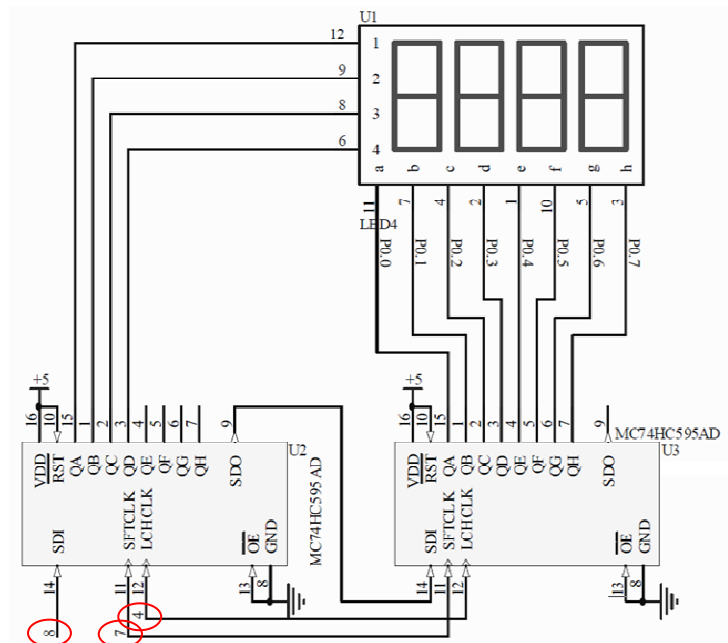


Рисунок 7 – Принципиальная схема подключения индикатора [3]

Согласно данной схеме входные данные поступают с восьмого (*PB0*) вывода микроконтроллера, тактовый сигнал с седьмого (*PD7*), а *Latch Clock* с четвертого (*PD4*). Первый регистр отвечает за выбор текущей выводимой цифры, второй – за значение цифры. Таким образом, для отображения единицы на второй цифре достаточно начиная с младшего бита отправить две восьмибитные последовательности *0b10011111* и *0b01000000*. Значения для цифры формируются согласно рисунку 4. Для вывода двойки на третьей цифре – *0b00100101* и *0b00100000*. Исходя из этого для каждой цифры должен быть сформирован определенный код, который должен передаваться в сдвиговые регистры. Наиболее простой вариант – это формирование массива с таблицей кодов символов:

```
char table[10]={ 0b00000011, 0b10011111, 0b00100101,... ,... ,... ,... };
```

где нулевой элемент соответствует цифре 0, первый – 1, второй – 2 и т.д.

3 Порядок выполнения работы

В ходе данной работы будет освоен вывод данных на счетверенный светодиодный семисегментный индикатор, подключенный к микроконтроллеру через сдвиговые регистры.

Для этого необходимо выполнить следующие действия:

3.1 Изучите предложенный в п. 2 теоретический материал.

3.2 Создайте новый проект.

3.3 В функции *setup()* сконфигурируйте выводы микроконтроллера, необходимые для управления сдвиговыми регистрами (см. рисунок 7).

3.4 В функции *loop()* с использованием портов ввода-вывода подайте на вывод *SDI* сдвигового регистра логическую единицу.

3.5 В функции *loop()* с использованием портов ввода-вывода подайте на вывод *SFTCLK* сдвигового регистра логическую единицу.

3.6 В функции *loop()* с использованием портов ввода-вывода подайте на вывод *SDI* сдвигового регистра логический ноль.

3.7 В функции *loop()* с использованием портов ввода-вывода подайте на вывод *SFTCLK* сдвигового регистра логический ноль.

3.8 В функции *loop()* с использованием портов ввода-вывода подайте на вывод *LCHCLK* сдвигового регистра логическую единицу, а затем сразу логический ноль.

3.9 Откомпилируйте и запустите программу.

Таким образом, будет осуществлена передача одного единичного бита в сдвиговый регистр (рисунок 8). При этом за передачу данных отвечали пункты 3.4–3.7, а за вывод данных на пины Q0–Q7 – пункт 3.9. Если в пункте

пункты 3.4 вместо логической единицы подать логический ноль, то в сдвиговый регистр будет передан ноль.

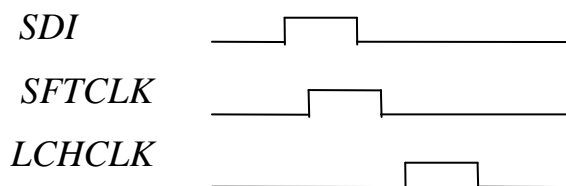


Рисунок 8 – Передача бита в сдвиговый регистр

3.10 В функции *loop()* организуйте вывод следующей последовательности *0b0100000010011111*. Вывод должен осуществляться справа налево.

3.11 Откомпилируйте и запустите программу. В результате во второй цифре должна быть отображена единица.

3.12 Опишите функцию *SHRSendBit(int bit)* для передачи одного бита в сдвиговый регистр (пункты 3.4–3.7). Параметр *bit* должен отвечать за значение бита.

3.13 Замените в пункте 3.10 строки, отвечающие за вывод битов в сдвиговый регистр на вызовы функции *SHRSendBit*, с соответствующими параметрами.

3.14 Откомпилируйте и запустите программу. В результате во второй цифре должна быть отображена единица.

3.15 Опишите функцию *SHRSendByte(byte data)* для передачи одного байта.

```
void SHRSendByte(byte data)
{
    for(int i=0;i<8;++i)
        SHRSendBit((data&(1<<i))>>i);
}
```

3.16 Выполните вывод последовательности *0b0100000010011111* с использованием последовательных вызовов функции *SHRSendByte*.

3.17 Сформируйте массив *char table[10]*, содержащий коды всех цифр.

3.18 Выполните вывод каждой цифры.

3.19 Выполните вывод одновременно двух цифр.

3.20 Выполните вывод одновременно трех цифр.

3.21 Выполните вывод одновременно четырех цифр.

3.22 Оформите отчет, содержащий титульный лист и разделы: введение, ход выполнения работы, ответы на контрольные вопросы и выводы.

3.23 Защитите отчет у преподавателя.

4 Контрольные вопросы

4.1 Что такое сдвиговый регистр?

4.2 Какой алгоритм необходимо выполнить для передачи цифры на семисегментный индикатор?

4.3 Какая последовательность битов должна быть отправлена в сдвиговый регистр, чтобы засветить сегмент F в 3й цифре.

4.4 Возможно ли одновременно засветить 4 цифры сдвигового регистра?

Список литературы

1. NFD-3641. URL:\\http:\\vsort.org.ua\\wp-content\\uploads\\2016\\03\\NFD-3641.pdf (Дата обращения 05.05.2017).

2. MC74HC595A. 8-Bit Serial-Input/Serial or Parallel-Output Shift Register with Latched 3-State Outputs. URL://http://www.onsemi.ru.com/pub/Collateral/MC74HC595A-D.PDF (дата обращения: 05.05.2017).

3. Using an Arduino Multi-function Shield.– URL: https:\\www.mpja.com\\download\\hacktronics-arduino-multi-function-shield.pdf (дата обращения: 10.01.2017).

4. ATMEL 8-BIT MICROCONTROLLER WITH 4/8/16/32KBYTES, IN-SYSTEM PROGRAMMABLE FLASH.– URL: http:\\www.atmel.com\\images\\Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf (дата обращения: 10.01.2017).

5. UNO Schematic – Arduino.– URL: https:\\www.arduino.cc\\en\\uploads\\Main\\arduino-uno-schematic.pdf (дата обращения: 10.01.2017).