

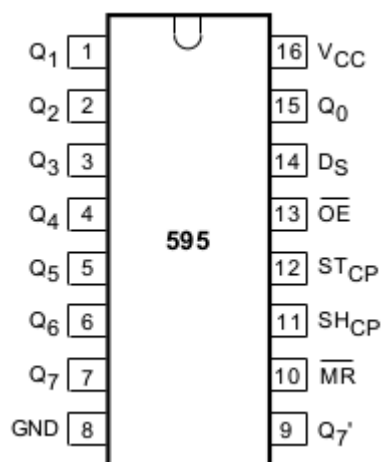
Практическая работа 14

Счетчик нажатий

В этом эксперименте мы выводим на семисегментный индикатор количество нажатий на кнопку (единицы).

Микросхема 74НС595 — один из самых распространённых сдвиговых регистров: синхронный, с регистром данных (latch). Она позволяет увеличивать количество выходов микроконтроллера.

Чип преобразовывает входящий последовательный сигнал на 1 пине (Ds) в выходной параллельный на 8 пинах (Qx). Последовательная передача синхронна: для такта используется дополнительный пин (SHcp). Также отдельным пином управляется регистр данных (STcp), что позволяет изменять сигнал на 8 выходах одновременно, когда все данные переданы.



Таким образом образом, из трёх пинов микроконтроллера, такого как Arduino, можно получить 8 цифровых выходов. Из регистров 74НС595 можно делать каскады, подключая один за другим (через пин Q7'), и таким образом из всё тех же 3 входящих линий получать 16, 24, 32 и т.д. цифровых выходов.

Используйте сдвиговый регистр для увеличения количества выходов микроконтроллера. Например, для управления большим количеством светодиодов.

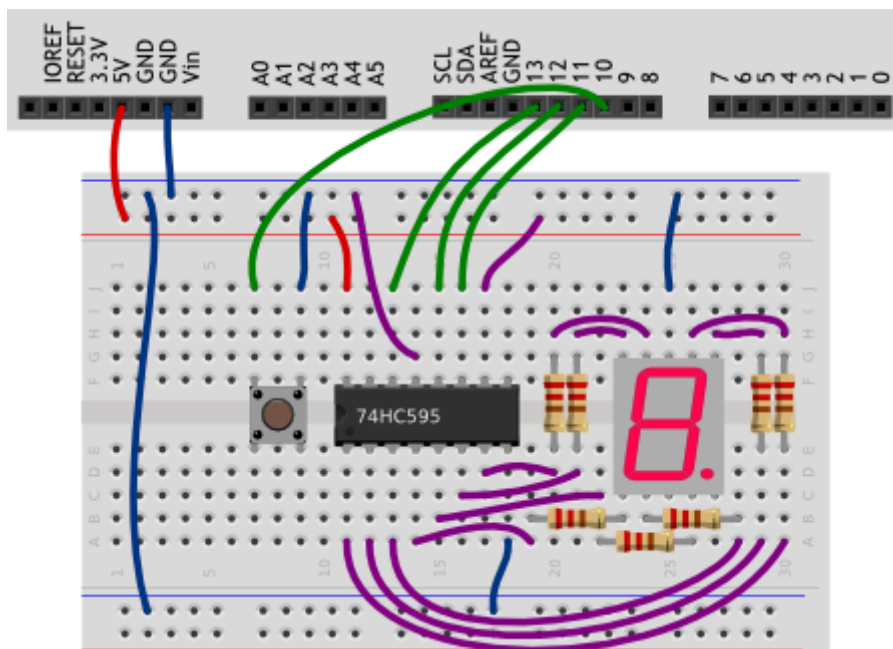
Задание 1. Ответить на вопросы

1. Зачем нужен сдвиговый регистр?
2. На сколько пинов увеличится плата, если подключить 3 сдвиговых резистора?

Задание 2. Список деталей для эксперимента

1. 1 плата Arduino Uno
2. 1 беспаячная макетная плата
3. 1 семисегментный индикатор
4. 1 тактовая кнопка
5. 1 выходной сдвиговый регистр 74HC595
6. 7 резисторов 220 Ом
7. 25 проводов «папа-папа»
8. 1 фоторезистор
9. 1 резистор 10 Ком

Схема на макетной плате:



1. Зарисуйте принципиальную схему установки.

Обратите внимание

✓ В этом эксперименте мы впервые используем микросхему, в данном случае — выходной сдвиговый регистр 74НС595. Микросхемы полезны тем, что позволяют решать определенную задачу, не собирая каждый раз стандартную схему.

✓ Выходной сдвиговый регистр дает нам возможность «сэкономить» цифровые выходы, использовав всего 3 вместо 8. Каскад регистров позволил бы давать 16 и т.д. сигналов через те же три пина.

✓ Перед использованием микросхемы нужно внимательно изучить схему ее подключения в datasheet'е. Для того, чтобы понять, откуда считать ножки микросхемы, на них с одной стороны есть полукруглая выемка. Если мы расположим нашу 74НС595 выемкой влево, то в нижнем ряду будут ножки 1—8, а в верхнем 16—9.

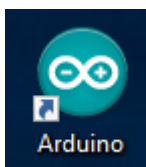
✓ На принципиальной схеме нашего эксперимента ножки расположены в другом порядке, чтобы не вышло путаницы в соединениях. Назначения выводов согласно datasheet'у подписаны внутри изображения микросхемы, номера ножек — снаружи.

✓ Напомним, что на изображении семисегментного индикатора подписаны номера его ножек и их соответствие сегментам.

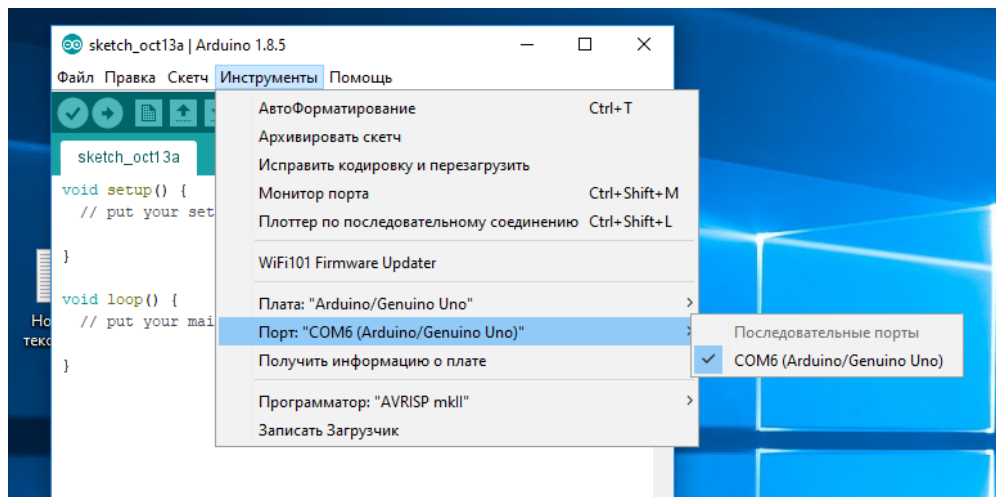
✓ Сегменты китайского индикатора могут отличаться от представленного

Задание 3. Программирование микроконтроллера

1. Запустите приложение



2. Убедитесь, что выбран нужный порт



3. Наберите в редакторе кода следующий код программы:

```

sketch_nov20a $
#define DATA_PIN    13 // пин данных (англ. data)
#define LATCH_PIN    12 // пин строба (англ. latch)
#define CLOCK_PIN    11 // пин такта (англ. clock)
#define BUTTON_PIN   10

int clicks = 0;
boolean buttonWasUp = true;
byte segments[10] = {
    0b01111101, 0b00100100, 0b01111010, 0b01110110, 0b00100111,
    0b01010111, 0b01011111, 0b01100100, 0b01111111, 0b01110111
};

void setup()
{
    pinMode(DATA_PIN, OUTPUT);
    pinMode(CLOCK_PIN, OUTPUT);
    pinMode(LATCH_PIN, OUTPUT);
    pinMode(BUTTON_PIN, INPUT_PULLUP);
}

void loop()
{
    // считаем клики кнопки, как уже делали это раньше
    if (buttonWasUp && !digitalRead(BUTTON_PIN)) {
        delay(10);
        if (!digitalRead(BUTTON_PIN))
            clicks = (clicks + 1) % 10;
    }
    buttonWasUp = digitalRead(BUTTON_PIN);
    // для записи в 74HC595 нужно притянуть пин строба к земле
    digitalWrite(LATCH_PIN, LOW);
    // задвигаем (англ. shift out) байт-маску бит за битом,
    // начиная с младшего (англ. Least Significant Bit first)
    shiftOut(DATA_PIN, CLOCK_PIN, LSBFIRST, segments[clicks]);
    // чтобы переданный байт отразился на выходах Qх, нужно
    // подать на пин строба высокий сигнал
    digitalWrite(LATCH_PIN, HIGH);
}

```

Пояснения к коду

- ✓ Обратите внимание, что в этом эксперименте кодировки символов отличаются от кодировок из эксперимента «Секундомер».
- ✓ Для того, чтобы передать порцию данных, которые будут отправлены через сдвиговый регистр далее, нам нужно подать LOW на latch pin (вход STcp микросхемы), затем передать данные, а затем отправить HIGH на latch pin, после чего на соответствующих выходах 74HC595 появится переданная комбинация высоких и низких уровней сигнала.

✓ Для передачи данных мы использовали функцию `shiftOut(dataPin, clockPin, bitOrder, value)`. Функция ничего не возвращает, а в качестве параметров ей нужно сообщить

- пин Arduino, который подключен ко входу DS микросхемы (data pin),
- пин Arduino, соединенный со входом SHcp (clock pin),
- порядок записи битов: LSBFIRST (least significant bit first) — начиная с младшего, или MSBFIRST (most significant bit first) — начиная со старшего,
- байт данных, который нужно передать. Функция работает с порциями данных в один байт, так что если вам нужно передать больше, придется вызывать ее несколько раз.

Задание 4. Ответьте на следующие вопросы

1. Для чего нужны микросхемы? Для чего нужен выходной сдвиговый регистр?
2. Как найти ножку микросхемы, на которую отправляются данные?
3. Что нужно сделать до и после отправки собственно данных на 74НС595?
4. Сколько данных можно передать с помощью `shiftOut()` и как управлять порядком их передачи?

Задание 5. Самостоятельно измените существующую программу и схему

1. Заставьте `shiftOut()` отправлять биты, начиная со старшего, и измените код так, чтобы счетчик по-прежнему показывал арабские цифры.
2. Замените кнопку на датчик света (фоторезистор в схеме делителя напряжения) и переделайте программу так, чтобы индикатор цифрой показывал уровень освещенности.