Эксперимент № 4. «Секундомер». 7-сегментный индикатор.

Вам понадобится

1. Proteus 8.1 Professional
2. http://tinkercad.com

Список деталей для эксперимента

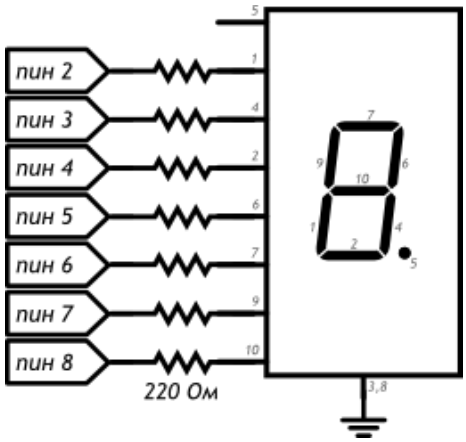
- 1 плата Arduino Uno
- 1 беспаечная макетная плата
- 1 7-ми сегментный индикатор
- 6 резисторов номиналом 100-220 Ом
- провода «папа-папа»

Повторение

1) Понятие резистора, светодиода
2) Семисегментный индикатор
3) Широтно-импульсная модуляция
4) Единицы измерения информации (бит, байт)

Пошаговое руководство для проведения эксперимента

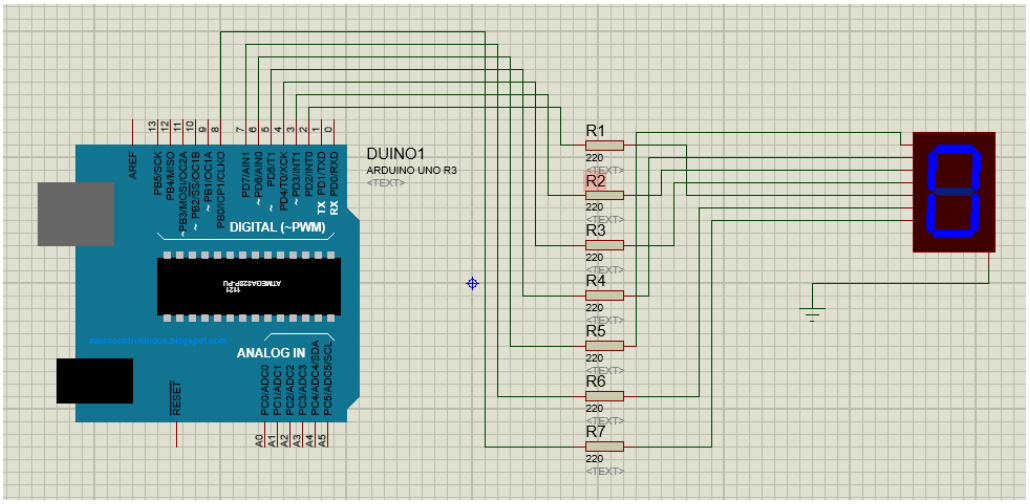
Принципиальная схема



Разработка эксперимента в среде Proteus

Создайте новый эксперимент в среде Proteus. Подтвердите создание нового проекта нажатием кнопки **Finish.** Сохраните проект, дайте имя файлу – **7Segment.** Добавление и размещение нужных компонентов выполняется аналогично примерам прошлого занятия. Выберите все необходимые детали. В списке устройств добавьте по ключевому слову следующие устройства: **Arduino UNO R3**, **7-SEG-COM-CAT-BLUE** и 7 резисторов на **200 Ohm.** Для выбора устройства произведите двойной щелчок на выбранном устройстве в списке устройств. В результате правильно выполненных операций в списке устройств Devices должны отобразиться все выбранные вами детали для эксперимента.

Соедените устройства. В результате правильно выполненных операций схема должна будет выглядеть следующим образом.



Обратите внимание. **8 PIN** -> R1 -> 5; **7 PIN** -> R6 -> 6; **6 PIN** -> R5 -> 1; **5 PIN** -> R4 -> 2; **4 PIN** -> R3 -> 4; **3 PIN** -> R2 -> 3; **2 PIN** -> R1 -> 7 (№ порта –> номер резистора –> контакт 7-сегментного индикатора).

- Внимательно рассмотрите схему, сопоставьте сегменты индикатора с номерами его ножек, а те, в свою очередь, с пинами Arduino, к которым мы их подключаем.
- Сегменты индикатора — просто светодиоды, поэтому мы используем резистор с каждым из них.

Список в Scratch. Вот так кодируются элементы цифр:

0	1	2	3	4	5	6	7	8	9
1	0	1	0	0	0	1	0	1	0
1	0	0	0	1	1	1	0	1	1
1	0	1	1	0	1	1	1	1	1
1	1	1	1	1	0	0	1	1	1
1	1	0	1	1	1	1	1	1	1
1	0	1	1	0	1	1	0	1	1
0	0	1	1	1	1	1	0	1	1
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

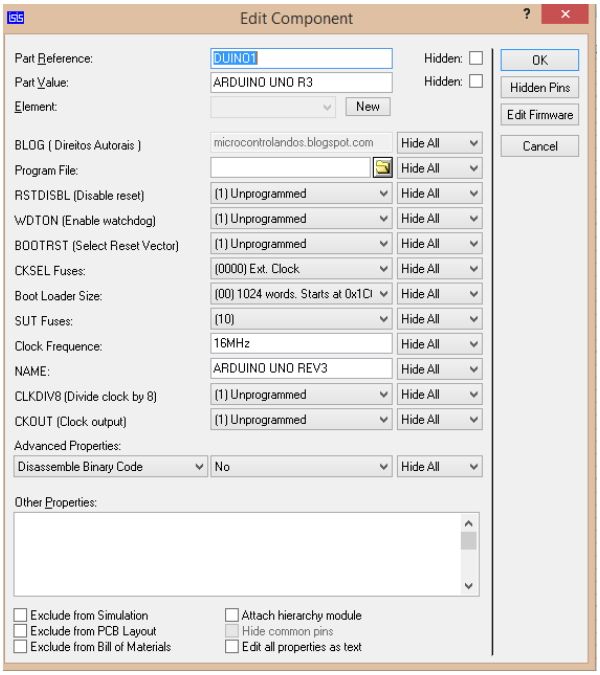
Приложение. Код программы «Секундомер» от 0-9 на основе 7-сегментного индикатора.

```
#define FIRST_SEGMENT_PIN   2
#define SEGMENT_COUNT       7

// префикс «0b» означает, что целое число за ним записано в
// в двоичном коде. Единицами мы обозначим номера сегментов
// индикатора, которые должны быть включены для отображения
// арабской цифры. Всего цифр 10, поэтому в массиве 10 чисел.
// Нам достаточно всего байта (англ. byte, 8 бит) для хранения
// комбинации сегментов для каждой из цифр.
byte numberSegments[10] = {
  0b00111111, 0b00001010, 0b01011101, 0b01011110, 0b01101010,
  0b01110110, 0b01110111, 0b00011010, 0b01111111, 0b01111110,
};
```

```
void setup()
{
 for (int i = 0; i < SEGMENT_COUNT; ++i)
   pinMode(i + FIRST_SEGMENT_PIN, OUTPUT);
}

void loop()
{
 // определяем число, которое собираемся отображать. Пусть им
 // будет номер текущей секунды, зацикленный на десятке
 int number = (millis() / 1000) % 10;
 // получаем код, в котором зашифрована арабская цифра
 int mask = numberSegments[number];
 // для каждого из 7 сегментов индикатора...
 for (int i = 0; i < SEGMENT_COUNT; ++i) {
   // ...определяем: должен ли он быть включён. Для этого
   // считываем бит (англ. read bit), соответствующий текущему
   // сегменту «i». Истина — он установлен (1), ложь — нет (0)
   boolean enableSegment = bitRead(mask, i);
   // включаем/выключаем сегмент на основе полученного значения
   digitalWrite(i + FIRST_SEGMENT_PIN, enableSegment);
 }
}
```

- Мы создали массив типа byte: каждый его элемент это 1 байт, 8 бит, может принимать значения от 0 до 255.
- Символы арабских цифр закодированы состоянием пинов, которые соединены с выводами соответствующих сегментов: 0, если сегмент должен быть выключен, и 1, если включен.
- В переменную mask мы помещаем тот элемент массива numberSegments, который соответствует текущей секунде, вычисленной в предыдущей инструкции.
- В цикле for мы пробегаем по всем сегментам, извлекая с помощью встроенной функции bitRead нужное состояние для текущего пина, в которое его и приводим с помощью digitalWrite и переменной enableSegment
- bitRead(x, n) возвращает boolean значение: **n**-ный бит *справа* в байте **x**

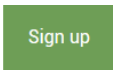
Не забывайте что для эмуляции в Proteus можно использовать только **HEX-файлы**! Скопируйте этот файл из временной папки в свою папку с проектами. Вернитесь в среду Proteus. Щелкните дважды на Arduino UNO R3 в среде Proteus для открытия окна следующего вида.



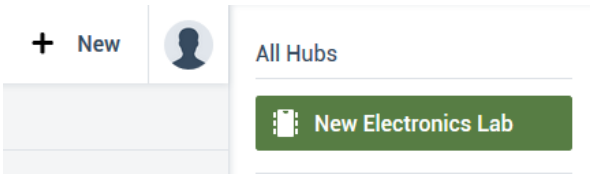
В окне нажмите на кнопке со значком  и выберите **HEX** файл из папки с проектом. Нажмите кнопку **OK**. Внизу экрана, в левой нижней части нажмите на кнопке **Run the simulation**  .

Эксперимент с **Proteus** завершен!

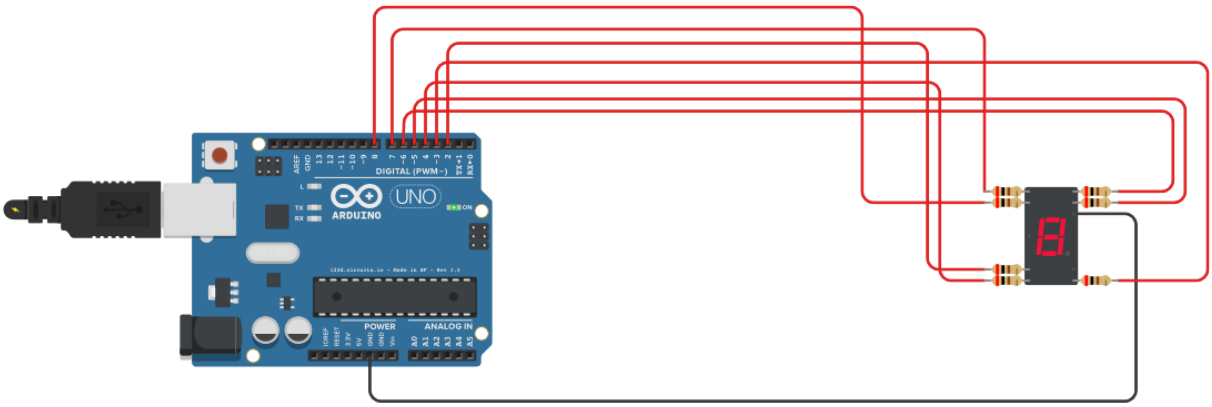
Приступим к эксперименту с использованием **HTPP://123D.CIRCUIT.IO**

Для продолжения работы нажмите кнопку 

и пройдите авторизацию используя учетную запись **Facebook.** После успешного прохождения авторизации, нажмите в верхней правой части сайта кнопку **NEW** и выберите **New Electronics Lab** для создания нового эксперимента.



Задание. Разместите на макетной плате компоненты как показано на рисунке ниже. Для соединения деталей между собой воспользуйтесь нажатиями левой кнопкой мыши. Нажмите на кнопке **Code Editor**. Скопируйте код проекта «Секундомер» в Code Editor. Нажмите на кнопке **Run Simulation**. Продолжите эксперименты с «7SEG-COM» проверив себя и выполнив **задания для самостотеятельного решения**.



Вопросы для проверки себя

1. К которой ножке нашего семисегментного индикатора нужно подключать землю?
2. Как мы храним закодированные символы цифр?
3. Каким образом мы выводим символ на индикатор?

Задания для проверки себя

1. Измените код, чтобы индикатор отсчитывал десятые секунды.
2. Поменяйте программу так, чтобы вместо символа «0» отображался символ «A».
3. Дополните схему и программу таким образом, чтобы сегмент-точка включался при прохождении четных чисел и выключался на нечетных