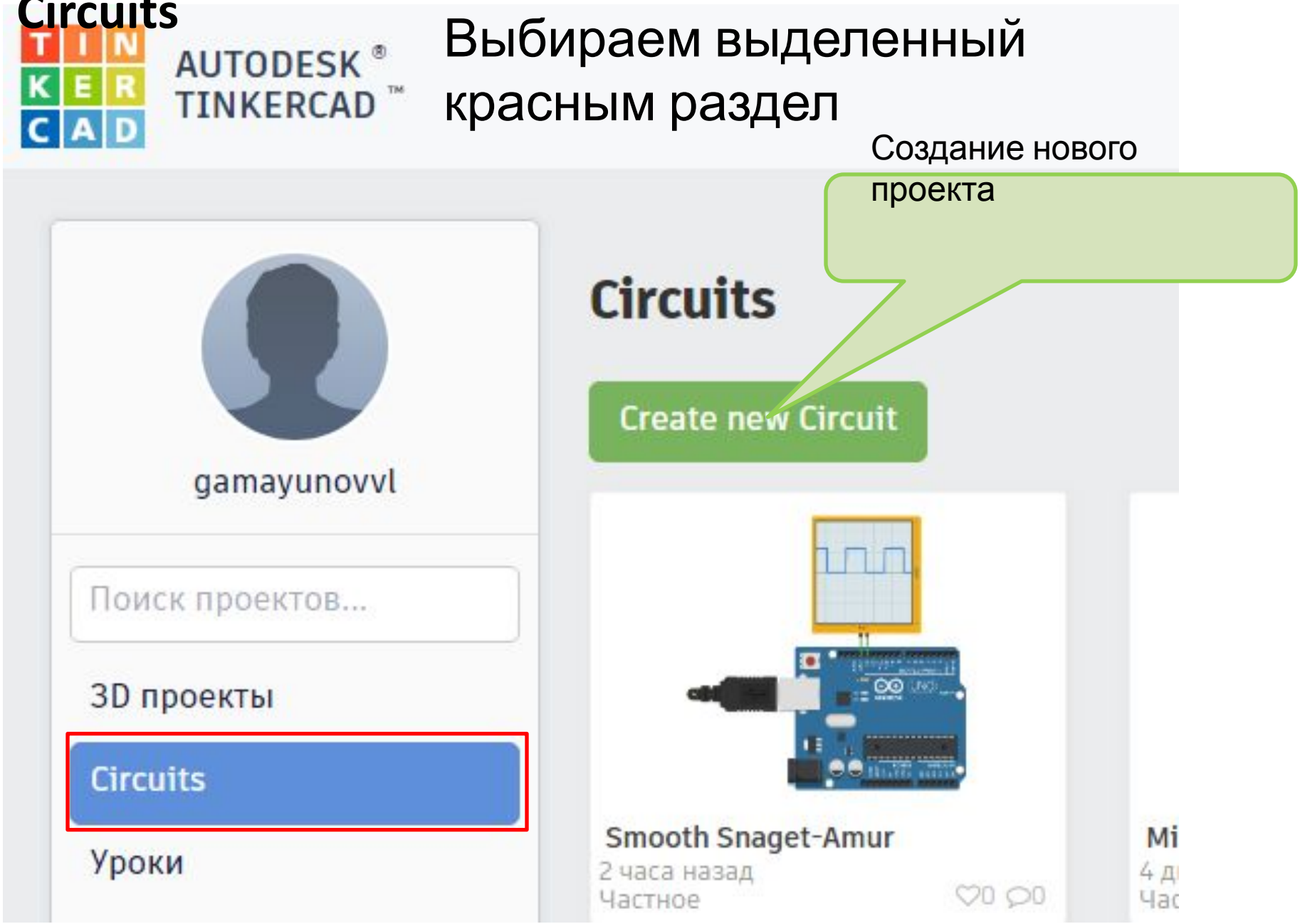
Эксперимент 6. «Ночной светильник»

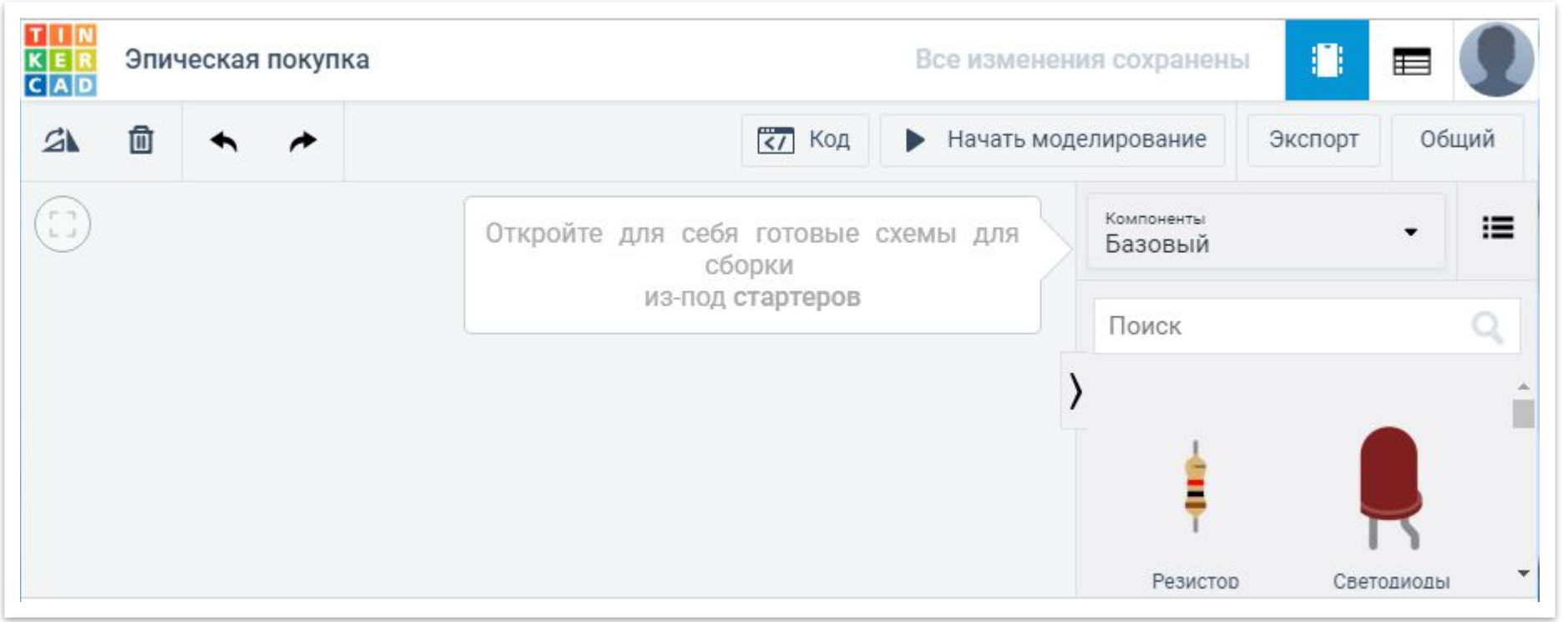
Работа с программой Tinkercad

Circuits

Выбираем выделенный красным раздел

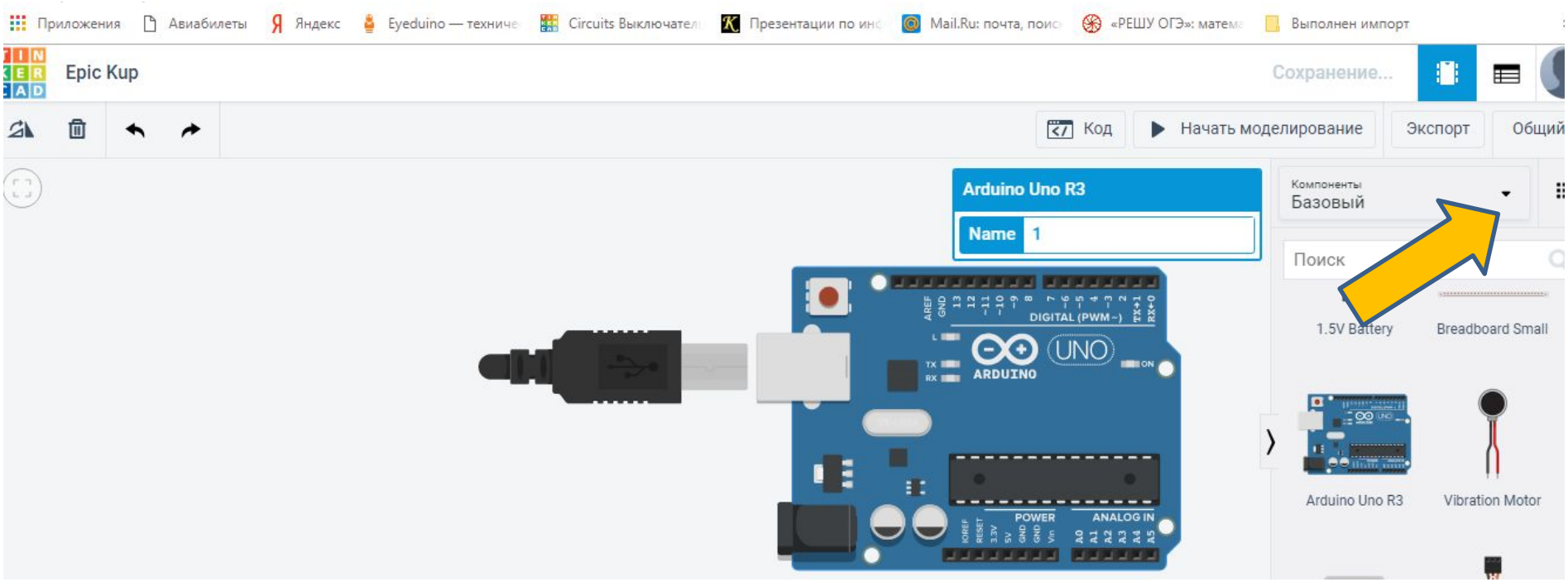


Окно нового проекта (название проекта выбирается случайно, не обращаем на это внимание)



Система готова для работы

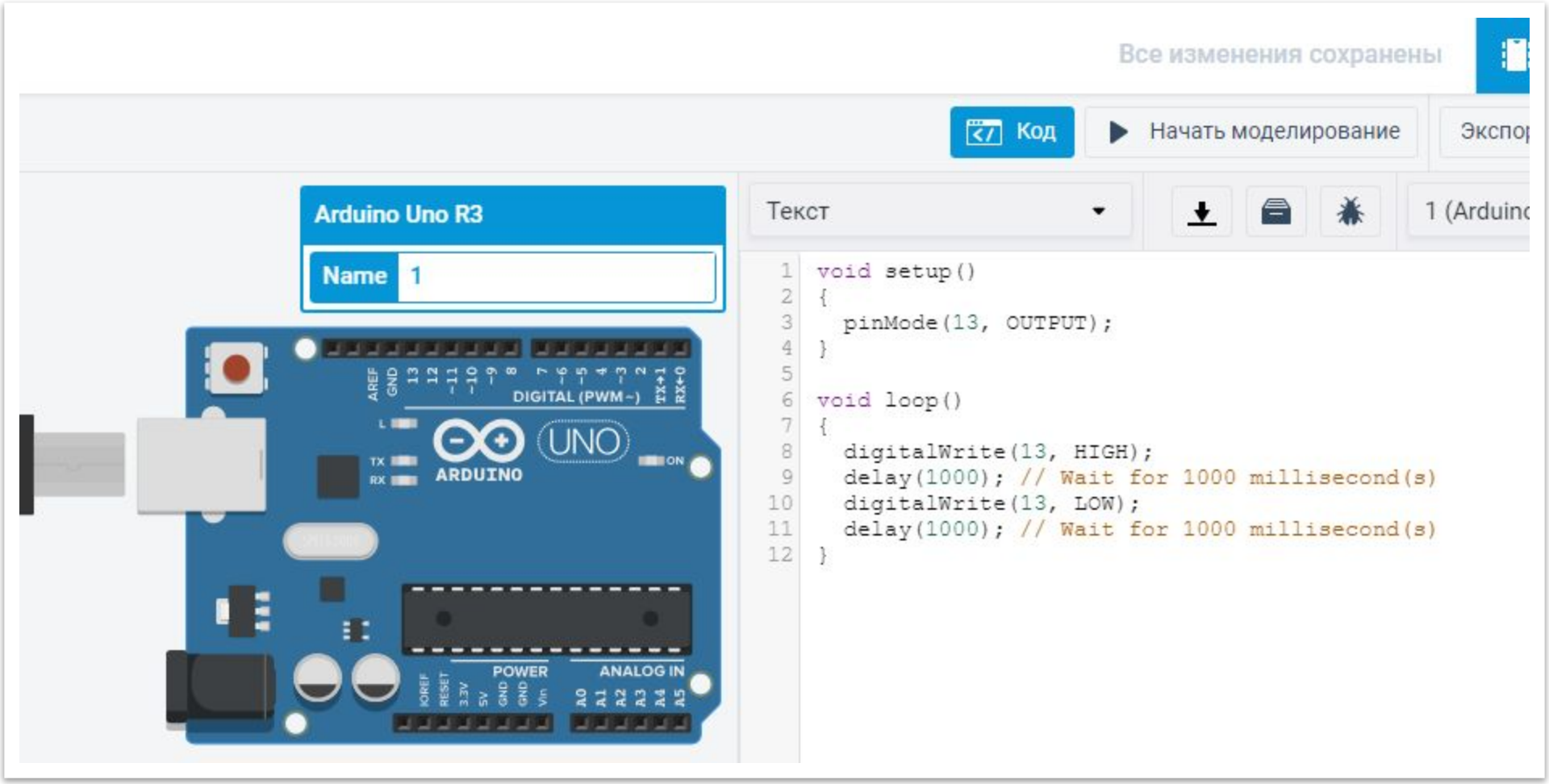
Работа с платой Ардуино в программе Tinkercad



Из базовых компонентов выбираем плату Arduino UNO R3 и перетаскиваем на рабочее поле.

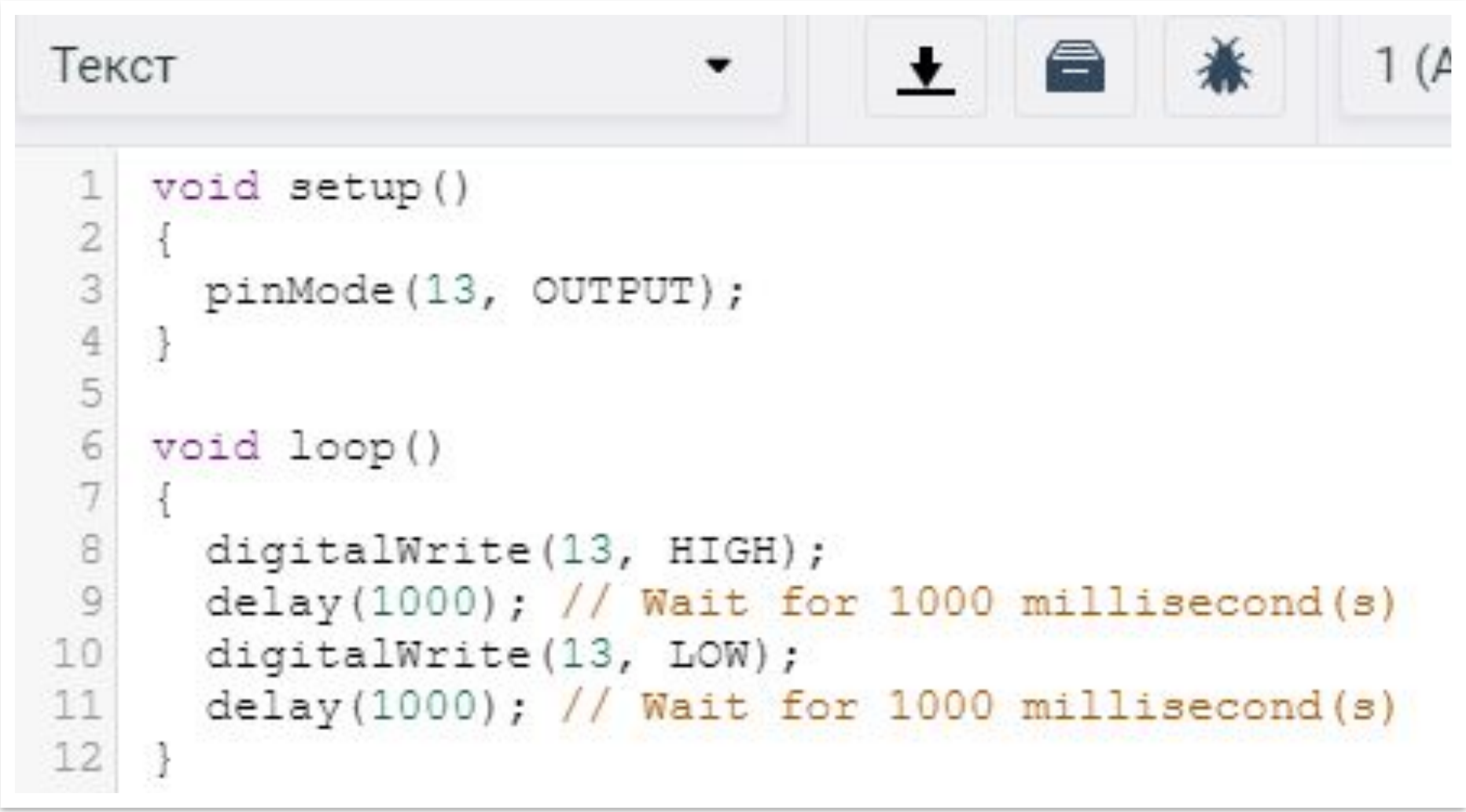
В стандартной комплектации в нее уже скопирована (залита) простейшая программа для демонстрации работы.

Нажимаем кнопку «Код» для просмотра программы и выбираем отображение кода в виде текста.



Мы будем работать в этом режиме

Код программы загруженный изначально



Контакт выхода 13. Подать высокий потенциал, ждать секунду, подать низкий потенциал ждать секунду. Повторять.

Эксперимент 6. Ночной светильник

В этом эксперименте светодиод должен включаться при падении уровня освещенности ниже порога, заданного потенциометром.

Список деталей для эксперимента

- 1 плата Arduino Uno
- 1 беспаечная макетная плата
- 1 светодиод
- 1 фоторезистор
- 1 резистор номиналом 220 Ом
- 1 резистор номиналом 10 кОм
- 1 переменный резистор (потенциометр)
- 7 проводов «папа-папа»

Принципиальная схема

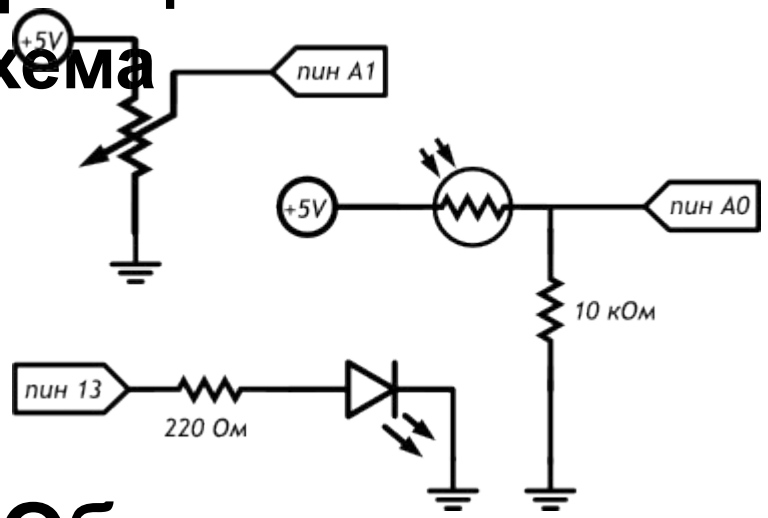
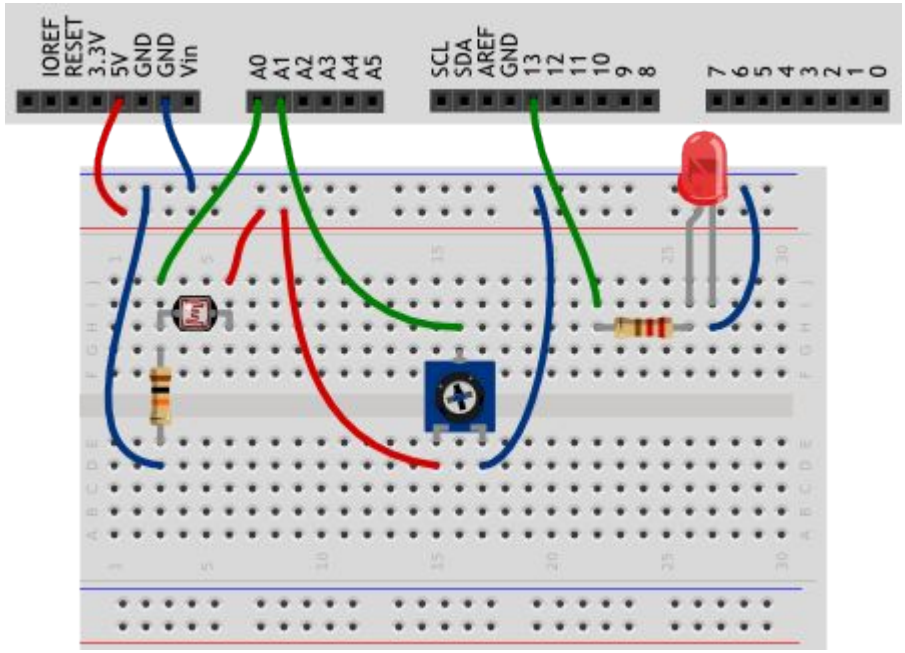


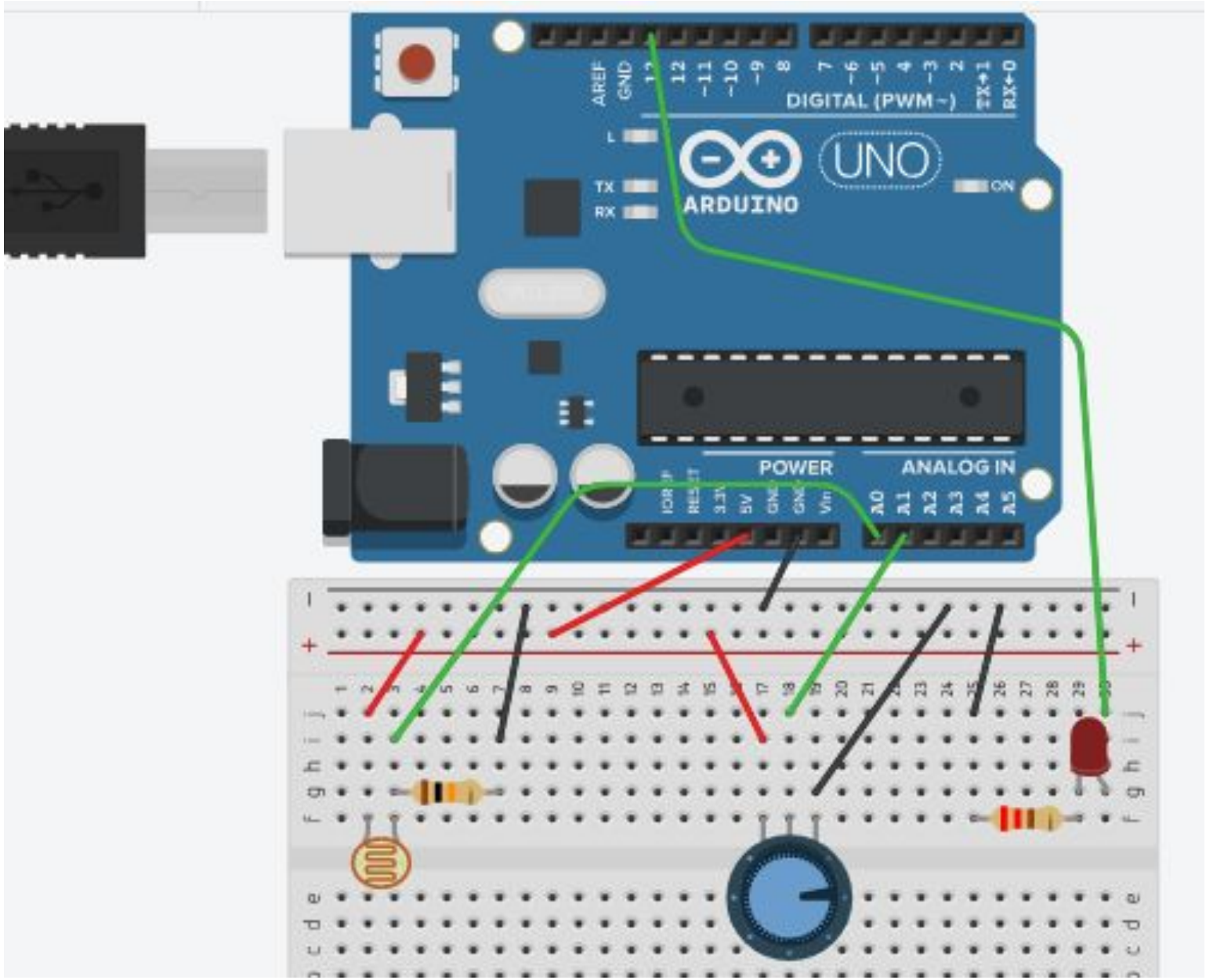
Схема на



Обратите внимание

- В этом эксперименте мы устанавливаем фоторезистор между питанием и аналоговым входом, т.е. в позицию R1 в схеме делителя напряжения. Это нам нужно для того, чтобы при уменьшении освещенности мы получали меньшее напряжение на аналоговом входе.
- Постарайтесь разместить компоненты так, чтобы светодиод не засвечивал фоторезистор.

Схема в программе Tinkercad



R – 10 кОм

R – 220 Ом

Код программы в Tinkercad Circuits (без комментариев)

```
#define LED_PIN  13
#define LDR_PIN  A0
#define POT_PIN  A1

void setup()
{
  pinMode(LED_PIN, OUTPUT);
}
void loop()
{
 int lightness = analogRead(LDR_PIN);
 int threshold = analogRead(POT_PIN);
 boolean tooDark = (lightness < threshold);
  if (tooDark) {
    digitalWrite(LED_PIN, HIGH);
  } else {
    digitalWrite(LED_PIN, LOW);
  }
}
```

Скетч (первая часть кода рабочей программы) с комментариями

После проверки работы программы добавляем комментарии

```
// даём имена для пинов
// LED_PIN для светодиода
// LDR_PIN для фоторезистора
// POT_PIN для переменного резистора
#define LED_PIN  13
#define LDR_PIN  A0
#define POT_PIN  A1
```

```
void setup()
{
  pinMode(LED_PIN, OUTPUT);
}
// LED_PIN или 13 пин определяем как выходной
// для светодиода
```

Скетч (код рабочей программы продолжение)

```
void loop()
{
  // считываем уровень освещённости. Кстати, объявлять
  // переменную и присваивать ей значение можно разом
  int lightness = analogRead(LDR_PIN);

  // считываем значение с потенциометра, которым мы
регулируем
  // пороговое значение между условными темнотой и светом
  int threshold = analogRead(POT_PIN);

  // объявляем логическую переменную и назначаем ей
значение
  // «темно ли сейчас». Логические переменные, в отличие от
  // целочисленных, могут содержать лишь одно из двух
значений:
  // истину (англ. true) или ложь (англ. false). Такие значения
  // ещё называют булевыми (англ. boolean).
  boolean tooDark = (lightness < threshold);

  // используем ветвление программы: процессор исполнит
один из
  // двух блоков кода в зависимости от исполнения условия.
  // Если (англ. «if») слишком темно...
  if (tooDark) {
    // ...включаем освещение
    digitalWrite(LED_PIN, HIGH);
  } else {
    // ...иначе свет не нужен — выключаем его
    digitalWrite(LED_PIN, LOW);
```

Пояснения к коду

- Мы используем новый тип переменных — boolean, которые хранят только значения true (истина, 1) или false (ложь, 0). Эти значения являются результатом вычисления логических выражений. В данном примере логическое выражение — это lightness < threshold. На человеческом языке это звучит как: «освещенность ниже порогового уровня». Такое высказывание будет истинным, когда освещенность ниже порогового уровня. Микроконтроллер может сравнить значения переменных lightness и threshold, которые, в свою очередь, являются результатами измерений, и вычислить истинность логического выражения.
- Мы взяли это логическое выражение в скобки только для наглядности. Всегда лучше писать читабельный код. В других случаях скобки могут влиять на порядок действий, как в обычной арифметике.
- В нашем эксперименте логическое выражение будет истинным, когда значение lightness меньше значения threshold, потому что мы использовали оператор <. Мы можем использовать операторы >, <=, >=, ==, !=, которые значат «больше», «меньше или равно», «больше или равно», «равно», «не равно»

Пояснения к коду (продолжение 1)

- Будьте особенно внимательны с логическим оператором == и не путайте его с оператором присваивания =. В первом случае мы сравниваем значения выражений и получаем логическое значение (истина или ложь), а во втором случае присваиваем левому операнду значение правого. Компилятор не знает наших намерений и ошибку не выдаст, а мы можем нечаянно изменить значение какой-нибудь переменной и затем долго разыскивать ошибку.
- Условный оператор if («если») — один из ключевых в большинстве языков программирования. С его помощью мы можем выполнять не только жестко заданную последовательность действий, но принимать решения, по какой ветви алгоритма идти, в зависимости от неких условий.
- У логического выражения lightness < threshold есть значение: true или false. Мы вычислили его и поместили в булеву переменную tooDark («слишком темно»). Таким образом мы как бы говорим «если слишком темно, то включить светодиод»
- С таким же успехом мы могли бы сказать «если освещенность меньше порогового уровня, то включить светодиод», т.е. передать в if всё логическое выражение:

Пояснения к коду (продолжение 2)

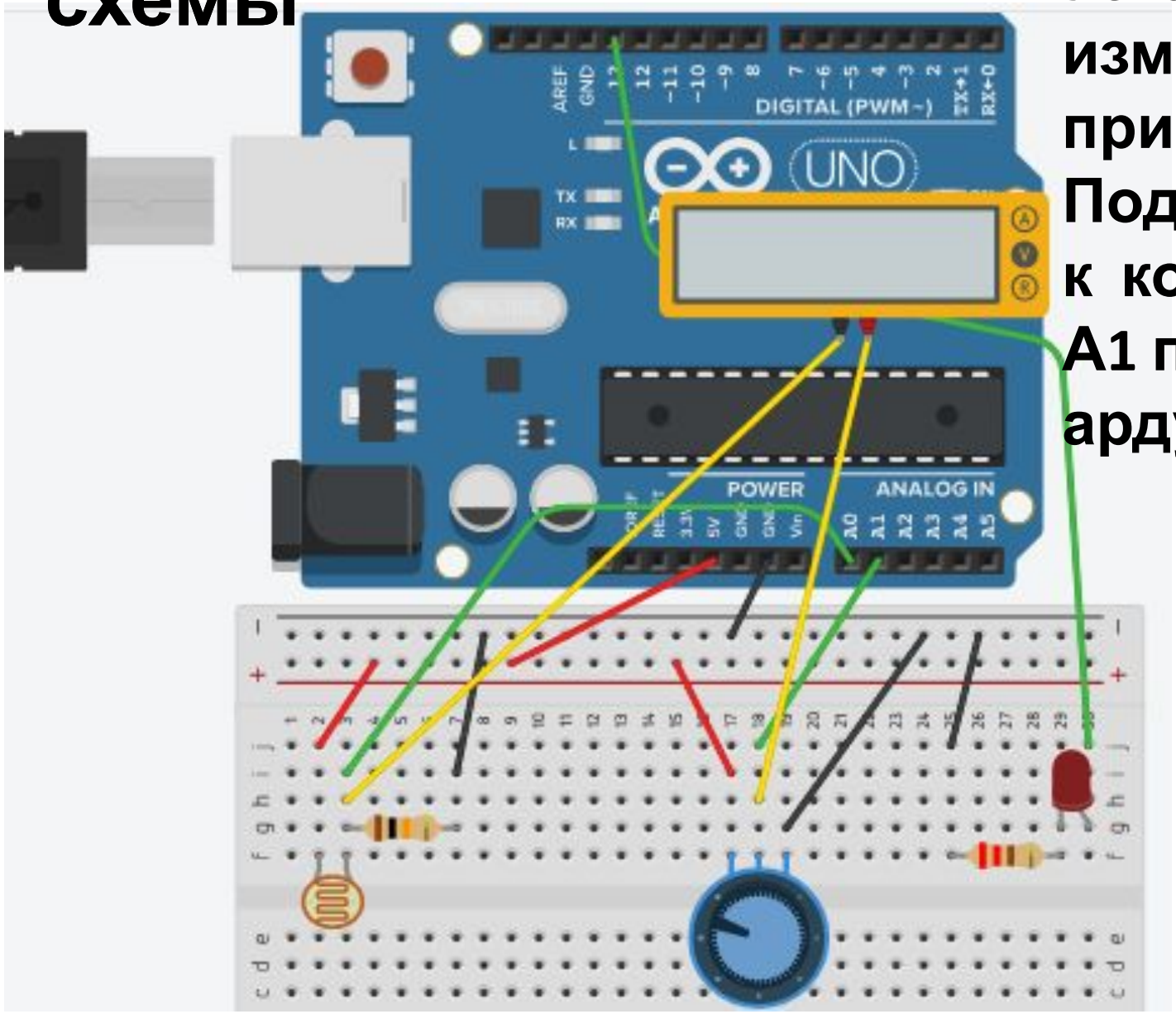
```
if (lightness < threshold) {
  // ...
}
```

- **За условным оператором if обязательно следует блок кода, который выполняется в случае истинности логического выражения. Не забывайте про обе фигурные скобки {}!**
- **Если в случае истинности выражения нам нужно выполнить только *одну* инструкцию, ее можно написать сразу после if (…) без фигурных скобок:**

```
if (lightness < threshold)
  digitalWrite(LED_PIN, HIGH);
```

- **Оператор if может быть расширен конструкцией else («иначе»). Блок кода или единственная инструкция, следующий за ней, будет выполнен только если логическое выражение в if имеет значение false, «ложь». Правила, касающиеся фигурных скобок, такие же. В нашем эксперименте мы написали «если слишком темно, включить светодиод, иначе выключить светодиод».**

Настройка и регулировка схемы



Устанавливаем измерительный прибор
Подключаем его к контактам A0 и A1 платы ардуино

Настройка и регулировка схемы (продолжение 1)

Измерительный прибор показывает значения логической переменной boolean tooDark

boolean tooDark = (lightness < threshold)

Если измерительный прибор показывает положительные значения, то значения логической переменной boolean tooDark **=1 или «Да» (светодиод горит)**

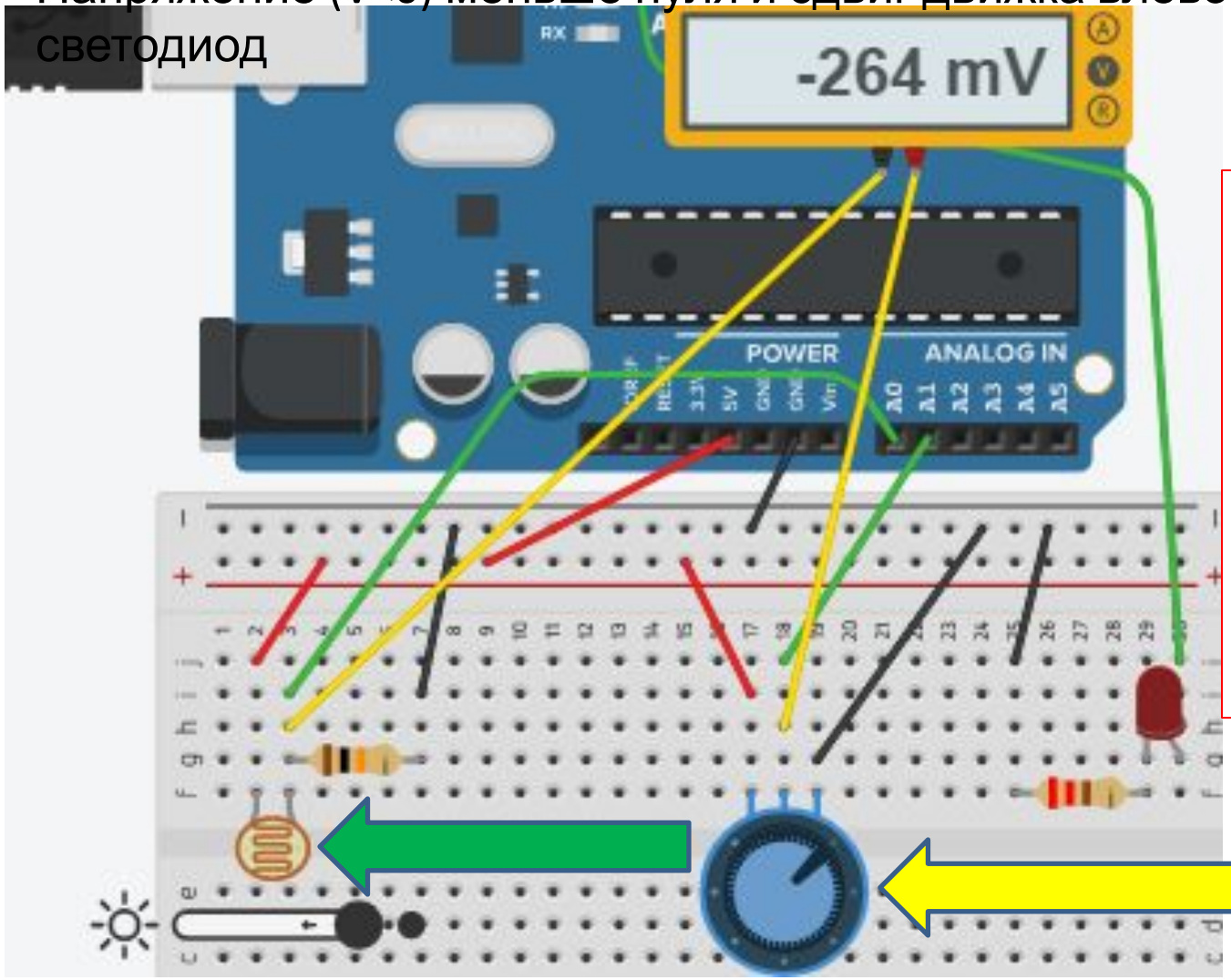
Если измерительный прибор показывает отрицательные значения, то значения логической переменной boolean tooDark **= - 1 или «Нет» (светодиод не горит)**

Настройка и регулировка схемы (продолжение 2)

Устанавливаем переменный резистор на первое деление (желтая стрелка)

Устанавливаем движок фоторезистора вправо до упора (зеленая стрелка)

Напряжение (V<0) меньше нуля и сдвиг движка влево не зажигает светодиод



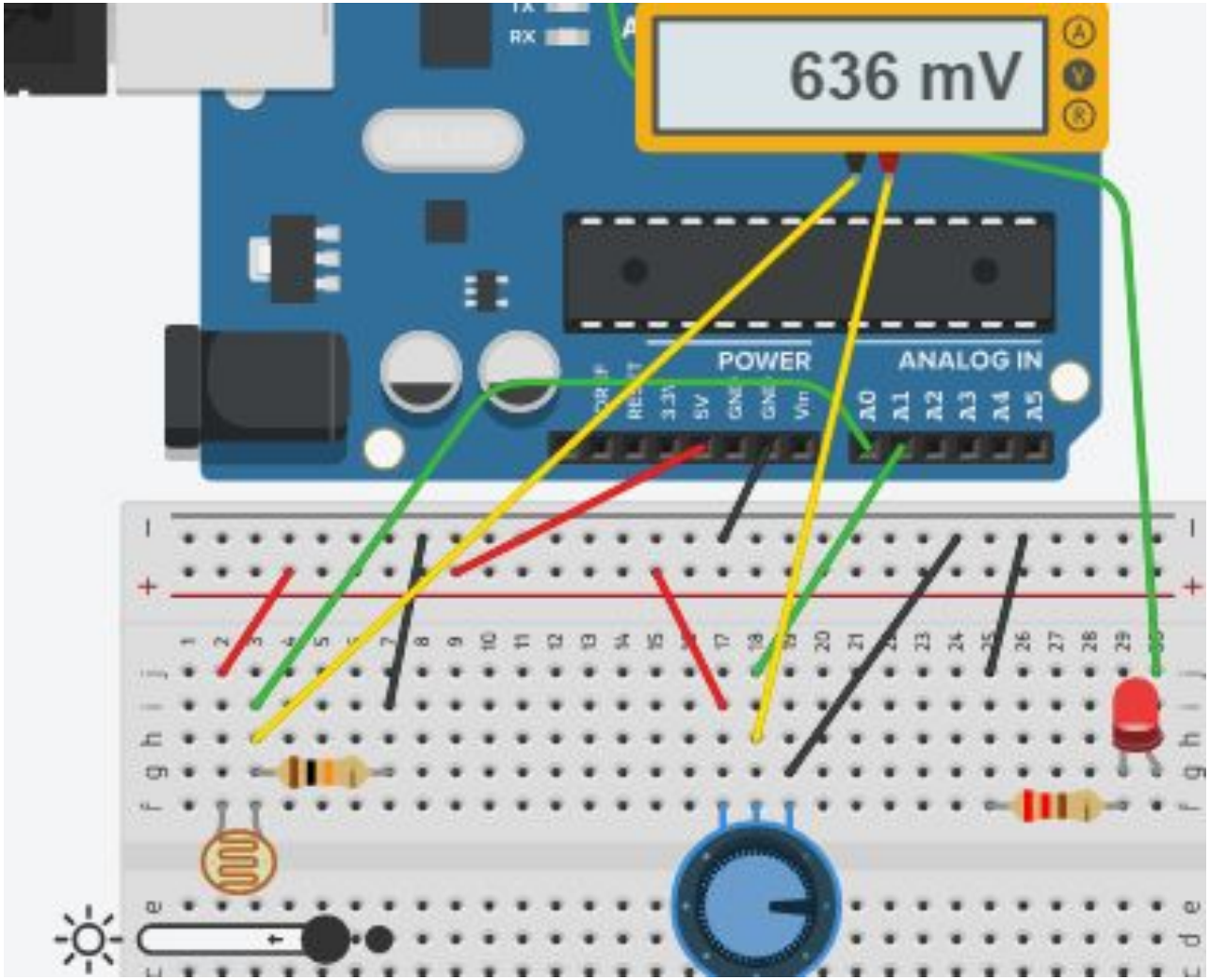
Вывод:

Свет не загорится при любом освещении фоторезистора

Настройка и регулировка схемы (продолжение 3)

Устанавливаем переменный резистор на второе деление, светодиод горит и на приборе положительное значение напряжения

Перемещаем движок фоторезистора влево и добиваемся угасания светодиода и перехода напряжения через V=0

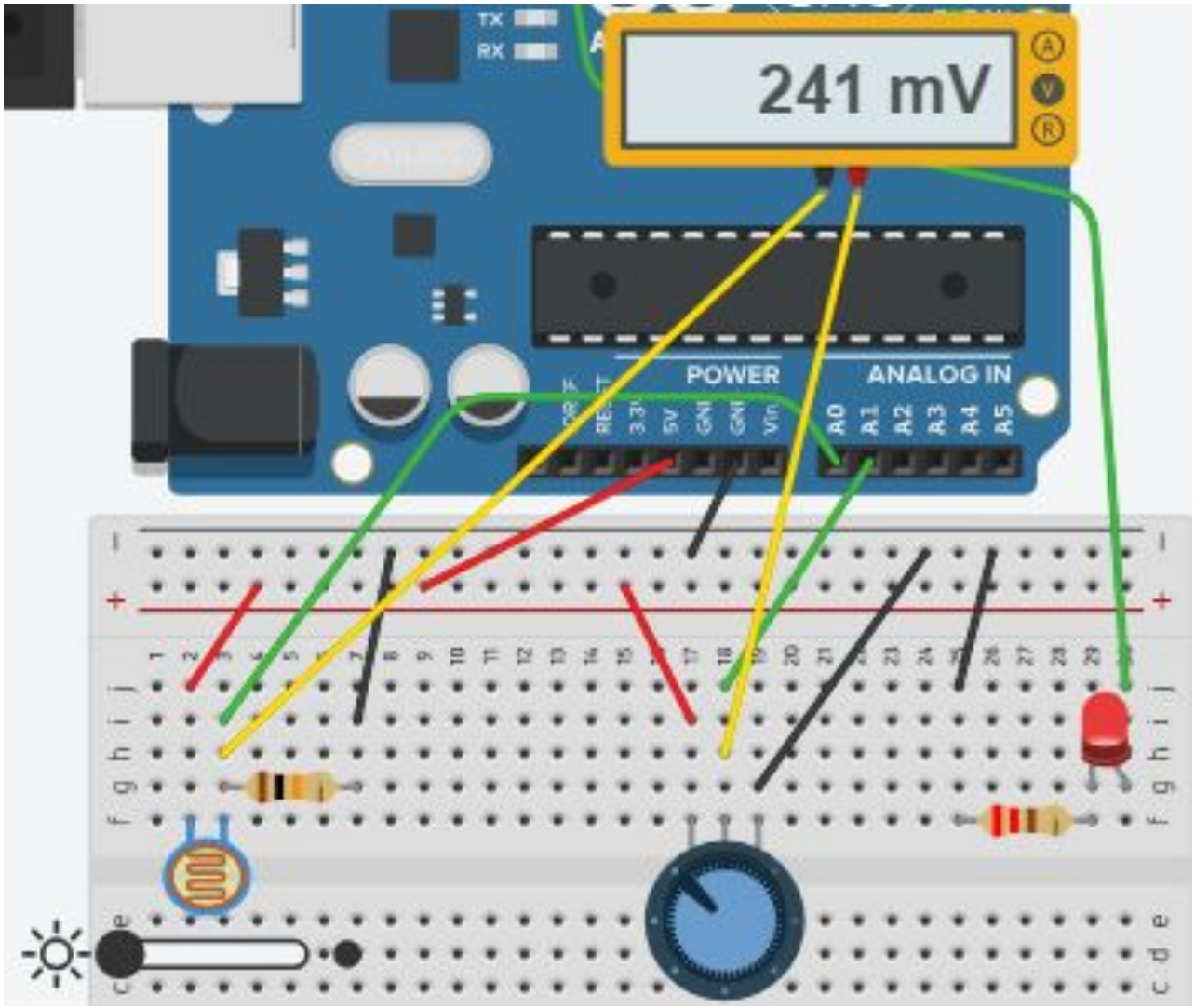


Вывод:

Освещение фоторезистора влияет на зажигание светодиода

Настройка и регулировка схемы (продолжение 4)

Устанавливаем переменный резистор на последнее деление
Устанавливаем движок фоторезистора влево до упора или двигаем
Напряжение (V>0) больше нуля и сдвиг движка вправо не гасит светодиод



Вывод:

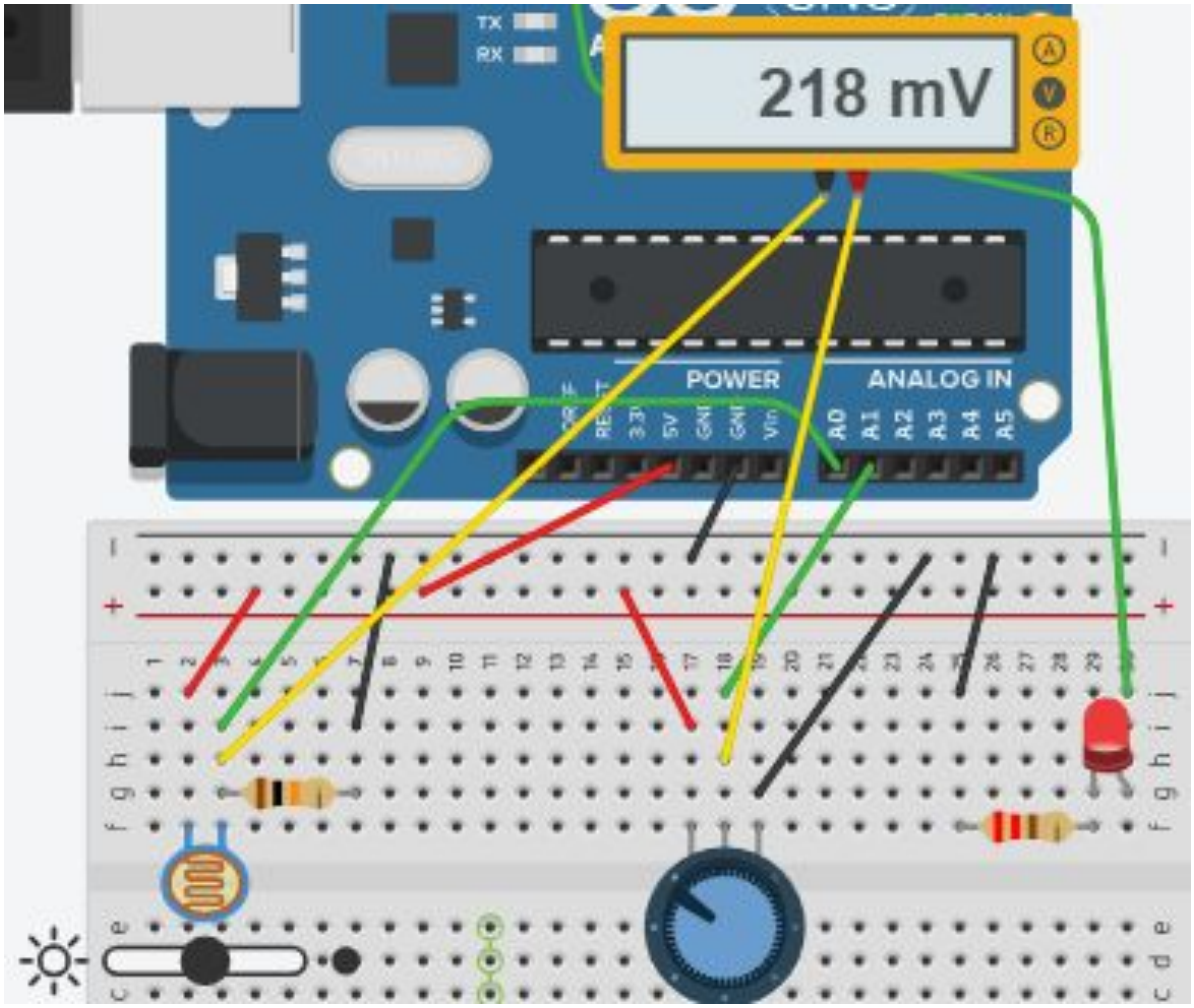
Свет не погаснет при любом освещении фоторезистора

Настройка и регулировка схемы (продолжение 5)

Устанавливаем движок фоторезистора посередине

Подбираем положение движка переменного резистора влево до зажигания светодиода

Большее затемнение не погасит светодиод



Вывод:

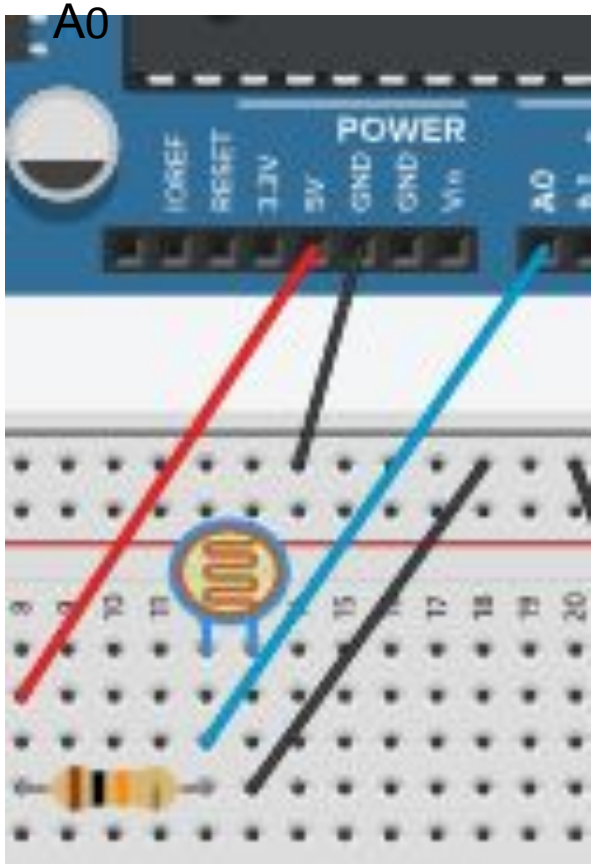
С наступлением темноты светодиод зажигается автоматически

Добавляем Serial в код рабочей программы

В раздел «void setup()» вводим

```
void setup()
{
  // пин с пьезопищалкой — выход...
  pinMode(BUZZER_PIN, OUTPUT);
  Serial.begin(9600);
  // ...а все остальные пины являются
  // всякий раз при подаче питания и
```

Сигнал снимаем с A0



В раздел «void loop()» вводим

```
  // потенциометра: в виде значения от
  val = analogRead(LDR_PIN);
  Serial.print(val);
  Serial.println(" svet");
  // рассчитываем частоту звучания пища
```

Включаем монитор порта

```
// частоту от 3,5 до 4,5 кГц.
```

Монитор последовательного интерфейса

```
98 svet
98 svet
98 svet
```

Вопросы для проверки себя

1. Если мы установим фоторезистор между аналоговым входом и землей, наше устройство будет работать наоборот: светодиод будет включаться при увеличении количества света. Почему?
2. Какой результат работы устройства мы получим, если свет от светодиода будет падать на фоторезистор?
3. Если мы все же установили фоторезистор так, как сказано в предыдущем вопросе, как нам нужно изменить программу, чтобы устройство работало верно?
4. Допустим, у нас есть код if (условие) {действие;}. В каких случаях будет выполнено действие?
5. При каких значениях y выражение x + y > 0 будет истинным, если x > 0?
6. Обязательно ли указывать, какие инструкции выполнять, если условие в операторе if ложно?
7. Чем отличается оператор == от оператора =?
8. Если мы используем конструкцию if (условие) действие1; else действие2;, может ли быть ситуация, когда ни одно из действий не выполнится? Почему?

Задания для самостоятельного решения

Для дополнительного задания

- еще 1 светодиод
- еще 1 резистор номиналом 220 Ом
- еще 2 провода

1. Перепишите программу без использования переменной tooDark с сохранением функционала устройства.
2. Добавьте в схему еще один светодиод. Дополните программу так, чтобы при падении освещенности ниже порогового значения включался один светодиод, а при падении освещенности ниже половины от порогового значения включались оба светодиода.
3. Измените схему и программу так, чтобы светодиоды включались по прежнему принципу, но светились тем сильнее, чем меньше света падает на фоторезистор.