

Практическое занятие № Тема: «Программирование системы “Сигнализация”»

Цель работы: приобрести практические навыки по подключению и программированию лазера на платформе Arduino.

Последовательность выполнения работы:

- Собрать схемы на макетной плате, иначе при отсутствии набора Arduino в web-приложениях (<https://wokwi.com/projects/new/arduino-uno> или <https://www.tinkercad.com/>) для приведенных примеров.
- Запрограммировать микроконтроллер согласно заданию в примере.
- Выполнить задание для самостоятельной работы.

Содержание отчета:

- Название практического занятия, его цель.
- Фото или скриншоты собранной схемы.
- Написанный программный код вставить текстом, Courier New, 12 кегль, одинарный отступ без абзацев.
- Вывод о проделанной работе.
- Файл Fritzing с принципиальной и монтажной схемой.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

KY-008 — это популярный и недорогой модуль лазерного излучателя для микроконтроллеров, таких как Arduino. Он состоит из электронного модуля, содержащий **лазерный диод** с длиной волны примерно **650 нм** (видимый красный свет) и мощностью обычно около **5 мВт**. Он предназначен для создания четкой точки или линии (с линзой) и используется в проектах, требующих точного указания, сигнализации, оптических экспериментов или в качестве части датчиков (например, лазерных охранных систем).

Важное предупреждение по безопасности: Лазер класса 3R/IIIa. Никогда не направляйте луч в глаза людям или животным. Избегайте отражений от зеркальных поверхностей. Мощности в 5 мВт достаточно для мгновенного повреждения сетчатки.



Рисунок 1 – Лазерный модуль ky-008

РАСПИНОВКА МОДУЛЯ (3 КОНТАКТА)

Модуль обычно имеет три вывода (иногда два, без среднего):

S (Сигнал) — Вход управления. Подключается к цифровому выходу микроконтроллера (например, пину Arduino). Активный низкий уровень (Active LOW) — лазер включается при подаче **LOW (0V/GND)**.

Middle (Центральный) — Питание. Подключается к **+5V**.

- (Минус) — Земля. Подключается к **GND**.

Некоторые версии модуля имеют только два контакта: **GND** и **Сигнал**, а питание (+5V) берется через внутреннюю подтяжку со стороны сигнального пина (что не всегда надежно). Лучше использовать трехпиновую версию.

ВНУТРЕННЯЯ СХЕМА И ПРИНЦИП РАБОТЫ

Лазерный диод: Генерирует лазерное излучение. Чрезвычайно чувствителен к перегрузкам по току.

Транзистор (NPN, например, S8050): Выполняет роль **ключа**. Управляется сигналом с микроконтроллера и включает/выключает питание лазерного диода, защищая чувствительный выход МК.

Токоограничивающий резистор: Встроен в модуль (обычно около 20-30 Ом). **Крайне важен!** Он ограничивает ток через лазерный диод, предотвращая его мгновенное сгорание. **При подключении «голого» лазерного диода всегда необходим внешний резистор, но в KY-008 он уже есть.**

Входной резистор (~220 Ом): Также встроен между сигнальным входом (S) и базой транзистора. Ограничивает ток базы.

Принцип: Когда на контакт **S** подается **LOW (0V)**, через резистор и переход база-эмиттер транзистора течет ток. Транзистор открывается, замыкая цепь лазерного диода на землю. Ток от **+5V** через встроенный резистор и лазерный диод устремляется на землю через транзистор — лазер светится. При подаче **HIGH (+5V)** транзистор закрыт, цепь разорвана — лазер выключен.

ХАРАКТЕРИСТИКИ

Напряжение питания: 5 В (постоянного тока).

Рабочий ток: ~40 мА (зависит от встроенного ограничительного резистора).

Длина волны: ~650 нм (красный).

Мощность лазера: ~5 мВт (класс 3R/IIIa).

Управляющий сигнал: Цифровой, **Active LOW**.

Расходимость луча: Очень малая (луч почти не расходится на больших расстояниях).

ПОДКЛЮЧЕНИЕ К ARDUINO (ПРИМЕР)

Схема подключения проста:

Pin S (KY-008) → Цифровой пин Arduino (например, **D9**).

Средний Pin (KY-008) → **5V** Arduino.

Pin “—“ (KY-008) → **GND** Arduino.

ПРИМЕР СКЕТЧА ДЛЯ ARDUINO

```
int laserPin = 9; // Пин, к которому подключен сигнал (S)
лазера

void setup() {
  pinMode(laserPin, OUTPUT);
  digitalWrite(laserPin, HIGH); // Устанавливаем HIGH для
  ВЫКЛЮЧЕНИЯ лазера (Active LOW!)
  // Лазер должен быть выключен при старте
}

void loop() {
  // Мигание: 1 секунда ВКЛ, 1 секунда ВЫКЛ
  digitalWrite(laserPin, LOW); // ВКЛ лазер (Active LOW)
  delay(1000);
  digitalWrite(laserPin, HIGH); // ВЫКЛ лазер
  delay(1000);

  // Или можно управлять с помощью ШИМ для регулировки яркости
  (на некоторых модулях работает)
  // analogWrite(laserPin, 50); // Яркость ~20%
}
```

ТИПОВЫЕ ПРИМЕНЕНИЯ

Охранные системы: В паре с фотодиодом или фоторезистором (например, модуль KY-018). Прерывание луча регистрируется как тревога.

Лазерная связь: Модуляция сигнала (через ШИМ или цифрово) для передачи данных на небольшие расстояния (требуется приемник — фотодиод).

Указатели и уровни: Для точного позиционирования в станках, строителях плоскостей (с рассеивающей линзой).

Эксперименты по оптике: Демонстрация отражения, преломления, интерференции.

Кот-развлечение: Лазерная указка для игр с питомцем (только под присмотром!).

ПРЕИМУЩЕСТВА И НЕДОСТАТКИ

Преимущества:

Очень низкая стоимость.

Простота подключения и управления.

Четкий, яркий луч, видимый на больших расстояниях.

Наличие встроенной защиты (резистор, транзистор).

Недостатки:

Опасность для зрения. Главный и критический недостаток.

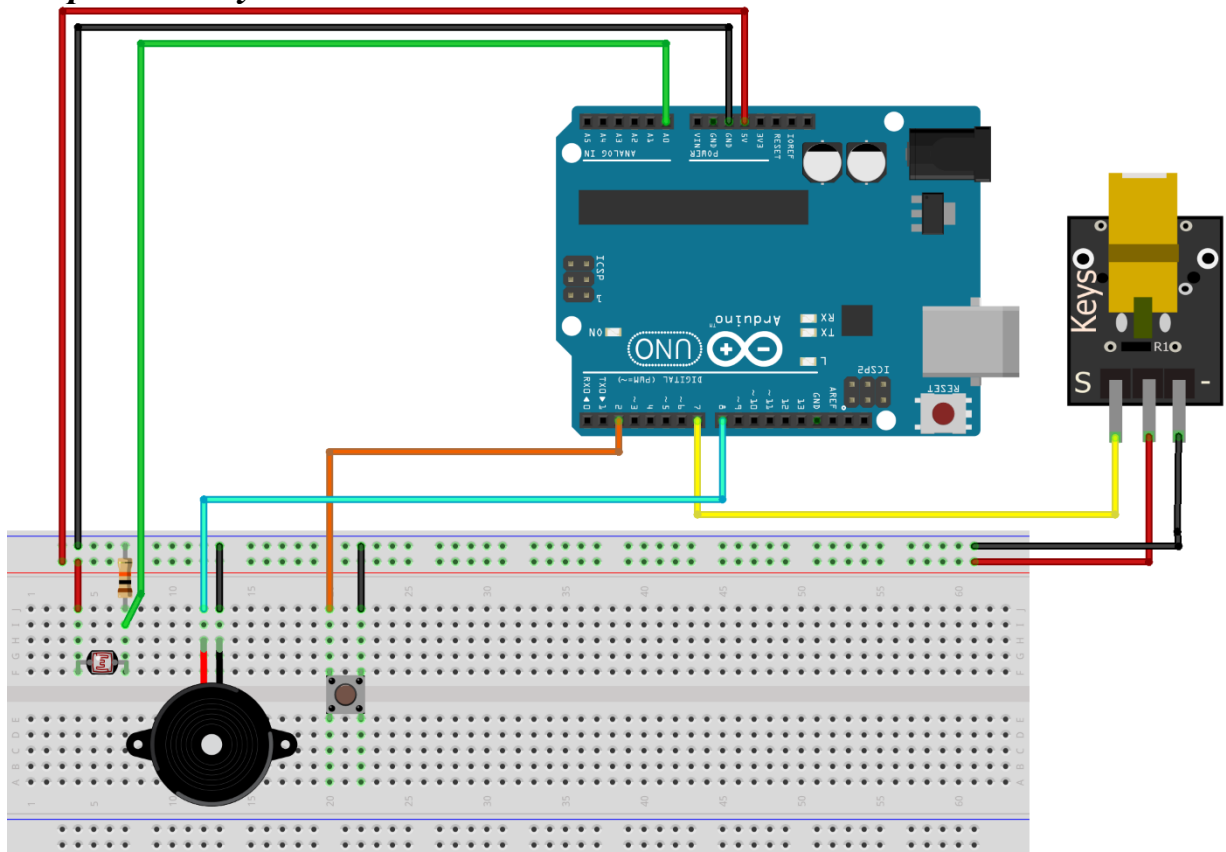
Активный низкий уровень (может быть неочевидно для новичков).

Невозможность фокусировки или изменения формы луча без дополнительных оптических элементов.

Мощность может варьироваться от модуля к модулю.

ЗАДАНИЯ

Собрать схему:



fritzing

Написать программный код:

В данной программе модуль ky-008 используется в качестве постоянного источника воздействия на фоторезистор, в случае если луч будет прерван сработает сигнализация. Отключить её можно при нажатии на кнопку.

```
// Определение пинов
const int laserPin = 7;           // Пин для лазера
const int buzzerPin = 8;          // Пин для зуммера
const int buttonPin = 2;          // Пин для кнопки сброса
const int photoResistorPin = A0;  // Пин для фоторезистора

// Переменные
int photoValue;                   // Текущее значение с
фоторезистора
int threshold;                    // Порог срабатывания
bool alarmActive = false;         // Флаг активной тревоги
unsigned long lastBuzzTime = 0;   // Время последнего звука
зуммера
const int buzzInterval = 500;    // Интервал звучания
зуммера (мс)

// Настройка начальных параметров
void setup() {
  Serial.begin(9600);             // Инициализация Serial
  Monitor
  Serial.println("Система лазерной сигнализации запущена");

  // Настройка пинов
  pinMode(laserPin, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
  pinMode(buttonPin, INPUT_PULLUP); // Используем внутренний
  подтягивающий резистор

  // Включение лазера
  digitalWrite(laserPin, HIGH);
  Serial.println("Лазер включен");

  // Калибровка: запоминаем начальный уровень освещенности
  delay(1000); // Даем время на стабилизацию
  threshold = analogRead(photoResistorPin);
  Serial.print("Порог срабатывания установлен: ");
  Serial.println(threshold);
}
```

```

    // Предупреждение о калибровке
    Serial.println("Убедитесь, что луч попадает на
фоторезистор!");
    Serial.println("=====
");
}

// Основной цикл
void loop() {
    // 1. Считываем значение с фоторезистора
    photoValue = analogRead(photoResistorPin);

    // 2. Проверяем, прерван ли луч (значение упало
значительно ниже порога)
    // -50 добавляет гистерезис для предотвращения ложных
срабатываний
    if (photoValue < threshold - 50 && !alarmActive) {
        // Луч прерван - активируем тревогу
        alarmActive = true;
        Serial.println("ALARM! Луч прерван!");
        Serial.print("Текущее значение: ");
        Serial.println(photoValue);
    }

    // 3. Если тревога активна - включаем зуммер
    if (alarmActive) {
        // Прерывистое звучание зуммера
        if (millis() - lastBuzzTime > buzzInterval) {
            digitalWrite(buzzerPin, !digitalRead(buzzerPin)); //
Переключаем состояние
            lastBuzzTime = millis();
        }

        // 4. Проверяем кнопку сброса
        if (digitalRead(buttonPin) == LOW) { // Кнопка нажата
(LOW, так как используется PULLUP)
            delay(50); // Антидребезг
            if (digitalRead(buttonPin) == LOW) { // Повторная
проверка
                // Сброс тревоги
                alarmActive = false;
                digitalWrite(buzzerPin, LOW); // Выключаем зуммер
                Serial.println("Тревога сброшена кнопкой");
                Serial.println("Ожидание восстановления луча...");
            }
        }
    }
}

```

```
        // Ждем восстановления луча
        while (analogRead(photoResistorPin) < threshold -
50) {
            delay(100);
            Serial.print(".");
        }
        Serial.println("\nЛуч восстановлен! Система
активирована.");
    }
}
} else {
    // Тревога не активна - выключаем зуммер
    digitalWrite(buzzerPin, LOW);
}

// Для отладки: выводим значение с фоторезистора
static unsigned long lastPrintTime = 0;
if (millis() - lastPrintTime > 1000) {
    Serial.print("Значение датчика: ");
    Serial.println(photoValue);
    lastPrintTime = millis();
}

delay(50); // Небольшая задержка для стабильности
}
```