Устройства индикации

Проверка связи

Если у вас нет звука:

- убедитесь, что на вашем устройстве и на колонках включён звук
- обновите страницу вебинара (или закройте страницу и заново присоединитесь к вебинару)
- откройте вебинар в другом браузере
- перезагрузите компьютер (ноутбук) и заново попытайтесь зайти

Поставьте в чат:

\+ если меня видно и слышно

— если нет

Вспоминаем прошлое занятие

Вопрос: Какому напряжению соответствует максимальный код АЦП?

Вспоминаем прошлое занятие

Вопрос: Какому напряжению соответствует максимальный код АЦП?

Ответ: Максимальный код АЦП соответствует опорному напряжению

Вспоминаем прошлое занятие

Вопрос: Как выполнить программную фильтрацию помех, наводимых на аналоговый сигнал?

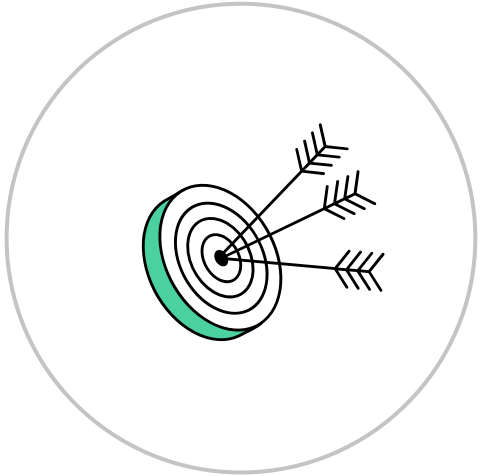
Вспоминаем прошлое занятие

Вопрос: Как выполнить программную фильтрацию помех, наводимых на аналоговый сигнал?

Ответ: самый простой способ - фильтр скользящего среднего, т.е. усреднение нескольких отсчетов

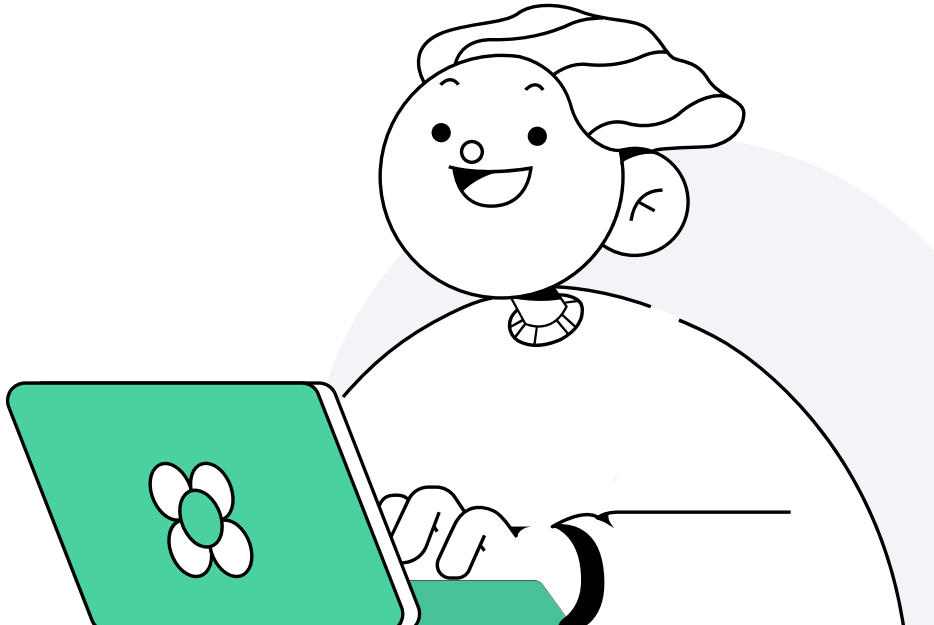
Цели занятия

- Узнаем, как подключать семисегментный индикатор
- Узнаем, как подключать многоразрядный семисегментный индикатор
- Научимся реализовывать принцип динамической индикации
- Узнаем, как подключать и использовать жидкокристаллический индикатор



План занятия

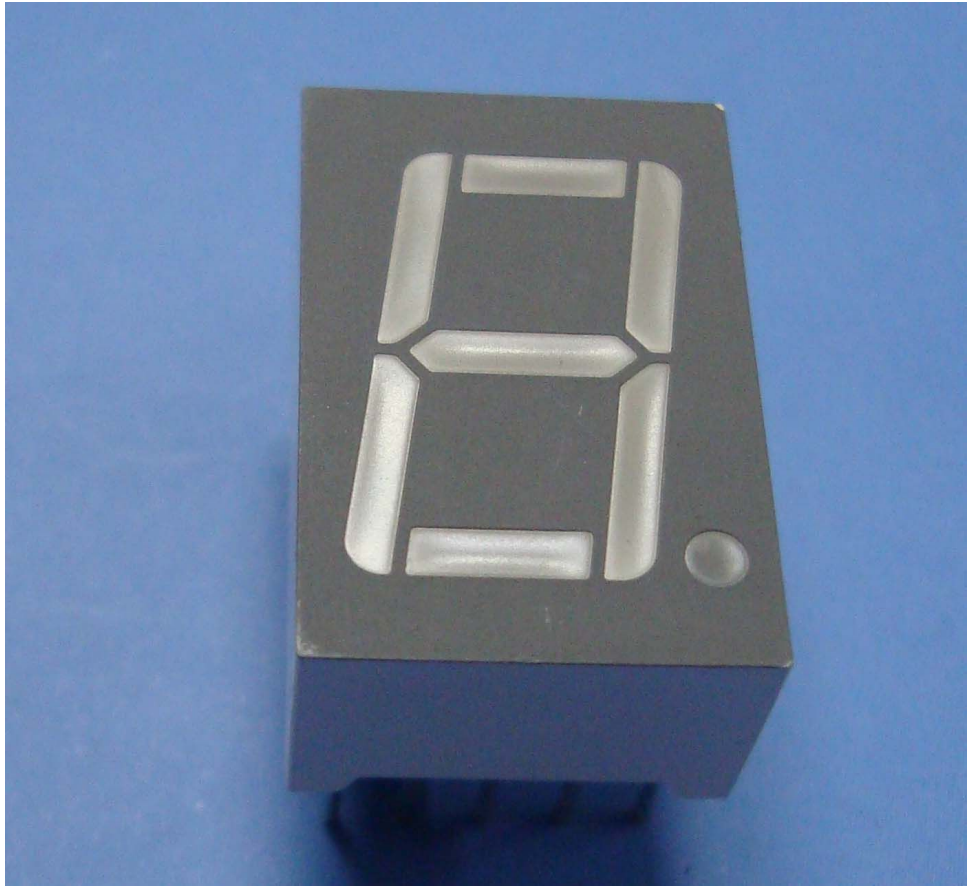
1. Как подключить семисегментный индикатор
2. Как подключить многоразрядный семисегментный индикатор
3. Как управлять жидкокристаллическим индикатором
4. Итоги



Как подключить семисегментный индикатор

Назначение семисегментного индикатора

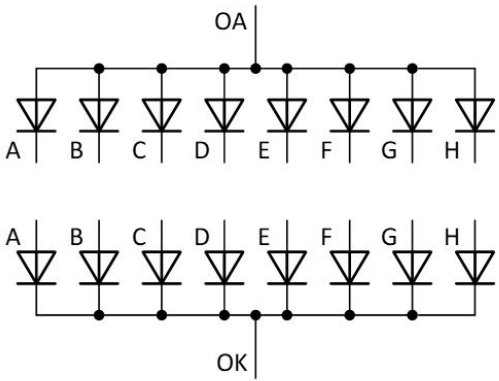
Семисегментный индикатор позволяет отображать любые цифры (и некоторые другие символы). Он состоит из 8 светодиодов: 7 формируют символ, а 1 - для десятичной точки

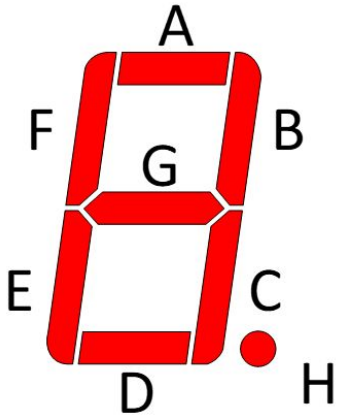


Источник

Типы семисегментных индикаторов

Для уменьшения количества линий светодиоды внутри индикатора объединяются либо по схеме с общим анодом, либо по схеме с общим катодом

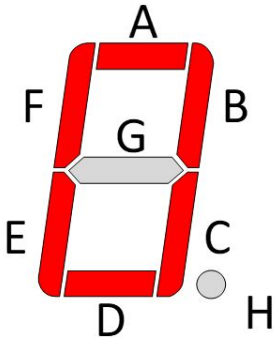


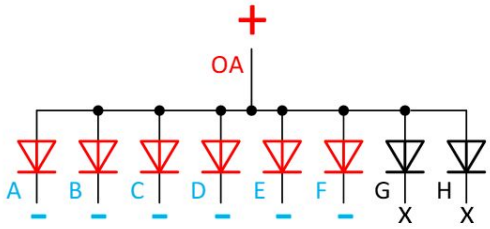


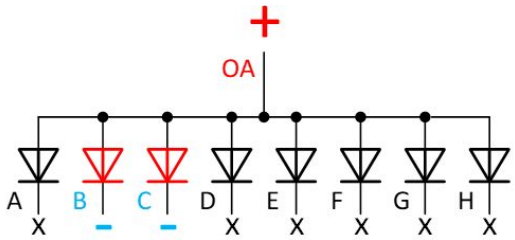
Источник

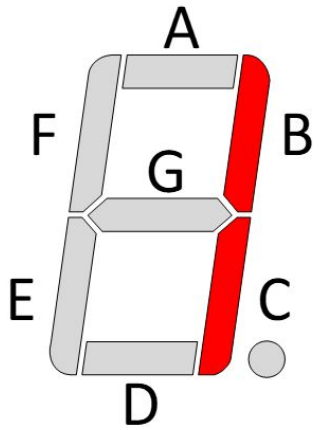
Формирование символа на индикаторе

Нужный символ формируется за счет включения нужной комбинации сегментов индикатора









Источник

Макрофункции для битовых операций

bitRead(value, bit) — возвращает значение конкретного разряда числа

Параметры:

- value: число (любой целочисленный тип)
- bit: номер разряда (любой целочисленный тип)

Возвращаемое значение: значение разряда (0 или 1)

bitSet(value, bit) — устанавливает в 1 конкретный разряд числа

Параметры:

- value: число (любой целочисленный тип)
- bit: номер разряда (любой целочисленный тип)

Возвращаемое значение: нет

Макрофункции для битовых операций

bitClear(value, bit) — сбрасывает в 0 конкретный разряд числа

Параметры:

- value: число (любой целочисленный тип)
- bit: номер разряда (любой целочисленный тип)

Возвращаемое значение: нет

bitWrite(value, bit, bitvalue) — устанавливает или сбрасывает конкретный разряд числа

Параметры:

- value: число (любой целочисленный тип)
- bit: номер разряда (любой целочисленный тип)
- bitvalue: устанавливаемое значение: 0 или 1

Возвращаемое значение: нет

Одноразрядный секундомер без управления

Пример программы:

```
#define SEG_COUNT    7      //количество используемых сегментов
const int firstSeg = 2;     //младший номер вывода для подключения индикатора
// Таблица перекодировки символов
byte numberSegments[10] = {
  0b11000000,   //0
  0b11111001,   //1
  0b10100100,   //2
  0b10110000,   //3
  0b10011001,   //4
  0b10010010,   //5
  0b10000010,   //6
  0b11111000,   //7
  0b10000000,   //8
  0b10010000,   //9
};
```

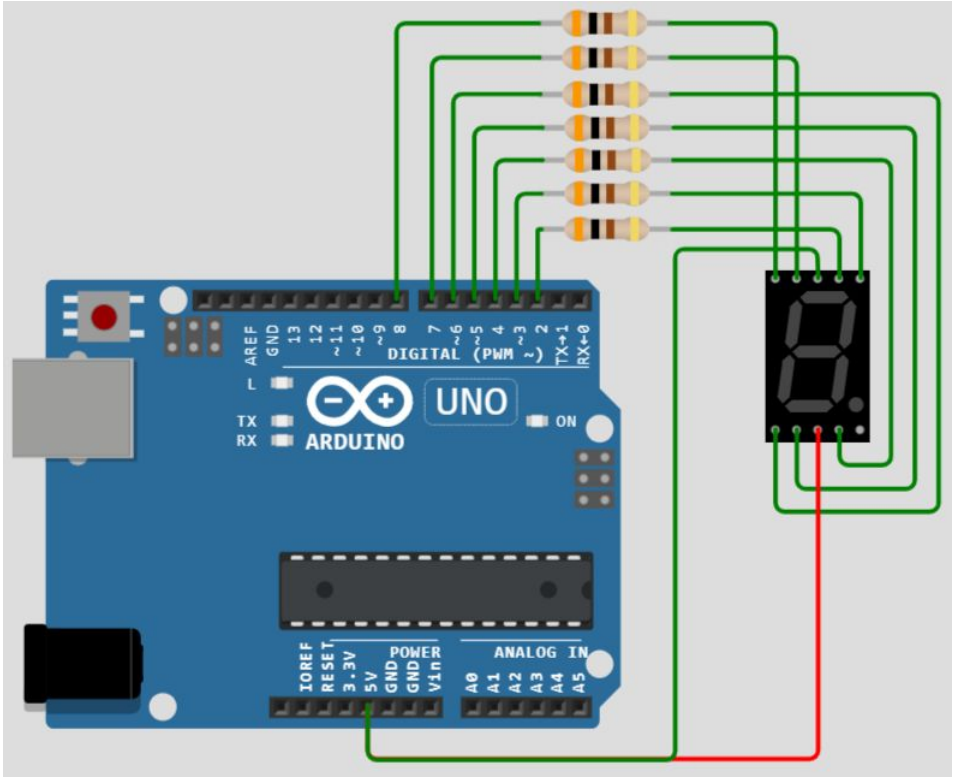
Одноразрядный секундомер без управления

Пример программы:

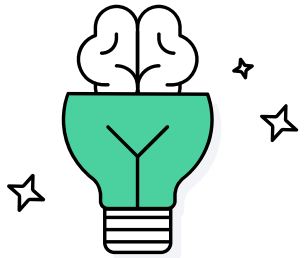
```
void setup()
{
  for (int i = 0; i < SEG_COUNT; ++i)
    pinMode(i + firstSeg, OUTPUT);
}

void loop()
{
  int number, mask;
  boolean enableSegment;
  // число для отображения - количество секунд
  number = (millis() / 1000) % 10;
  // код символа из таблицы перекодировки
  mask = numberSegments[number];
  // для каждого сегмента определяем: должен ли он быть включён.
  for (int i = 0; i < SEG_COUNT; i++)
  {
    enableSegment = bitRead(mask, i);
    digitalWrite(i + firstSeg, enableSegment);
  }
}
```

Одноразрядный секундомер без управления

Индикатор отображает количество секунд (только младший разряд числа), прошедших с момента включения



Практическое задание №1



Практика: одноразрядный секундомер без управления

Задание:

1) соберите схему в симуляторе WOKWI, подключив индикатор к выводам 2 … 8;
2) создайте скетч с текстом, приведенным выше;
3) проведите моделирование работы

Как выполнять: напишите в чат об удачной работе схемы

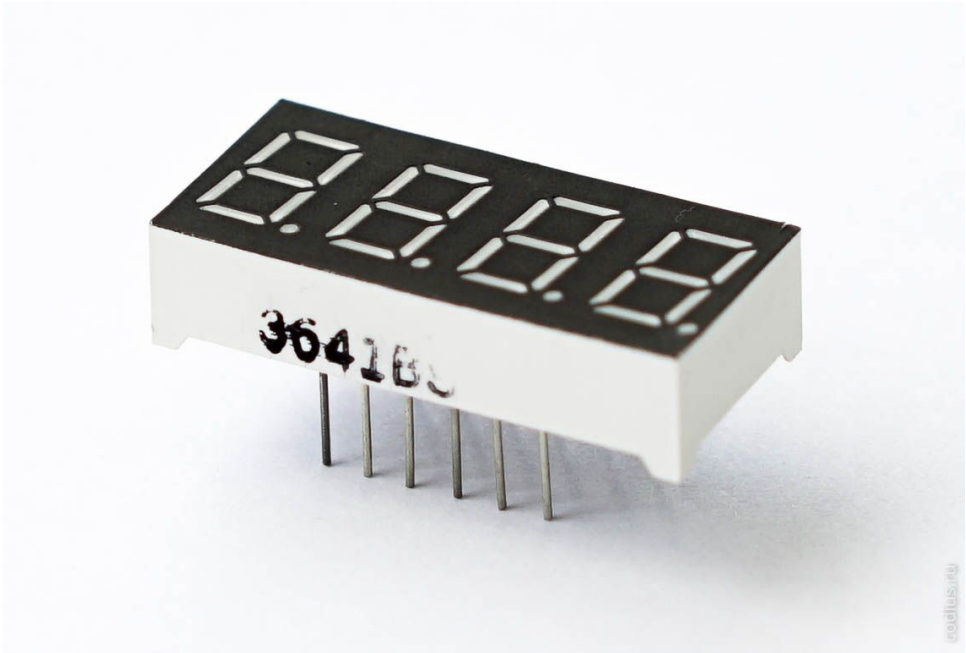
Время выполнения: 5 минут

Как подключить многоразрядный семисегментный индикатор

2

Назначение многоразрядного семисегментного индикатора

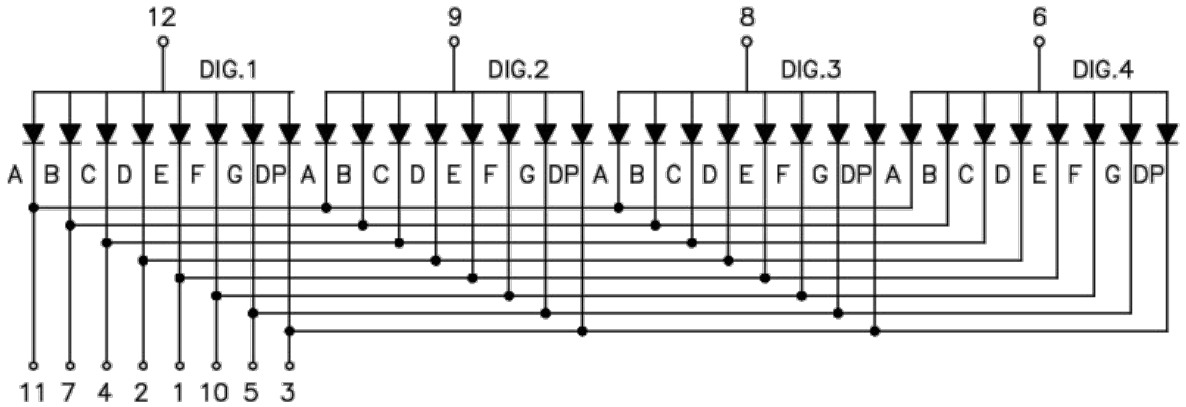
Многоразрядный индикатор позволяет отображать число, например, текущее время

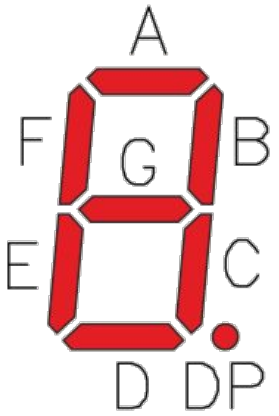


Источник

Внутреннее устройство многоразрядных индикаторов

Для уменьшения количества линий у многоразрядных индикаторов линии сегментов объединяются



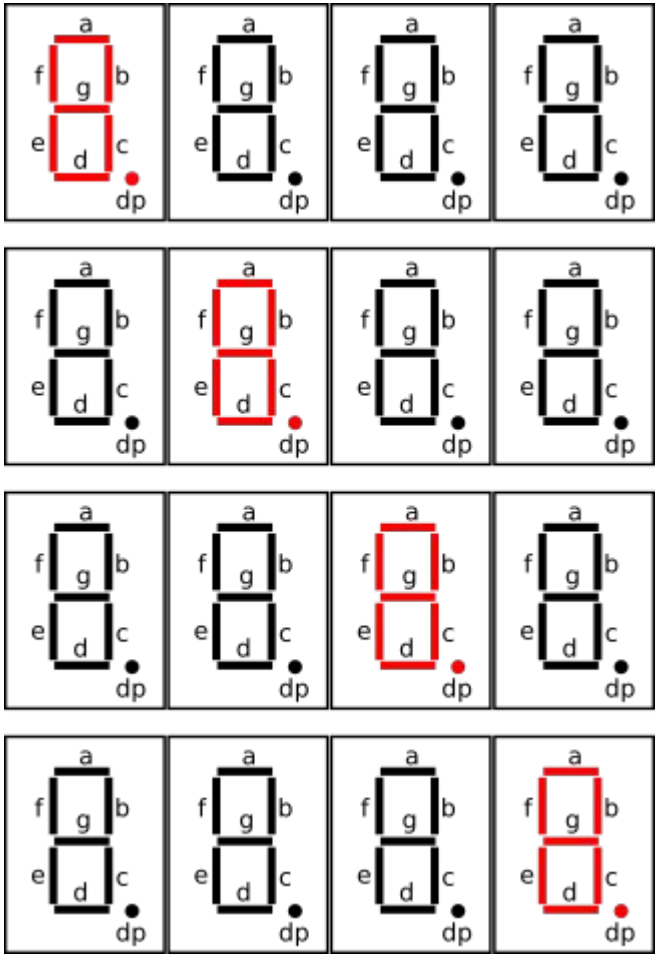


Источник

Принцип динамической индикации

В каждый момент времени информацию отображает лишь один разряд индикатора, остальные погашены.
За счет инерционности зрения человека, если частота обновления дисплея достаточно высока, мы увидим отображаемую информацию такой, как будто она отображается непрерывно.

Источник



Библиотека fDigitsSegtPin для работы с многоразрядным семисегметным индикатором

fDigitsSegtPin(u8 vPf1, u8 vPf2, u8 vPf3, u8 vPf4, u8 vPf5, u8 vPf6, u8 vPf7, u8 vPf8, u8 vPf9, u8 vPf10, u8 vPf11, u8 vPf12) — конструктор, создающий объект fDigitsSegtPin

Параметры:

- vPf1: номер вывода для сегмента E; vPf2: номер вывода для сегмента D;
- vPf3: номер вывода для сегмента H; vPf4: номер вывода для сегмента C;
- vPf5: номер вывода для сегмента G; vPf6: номер вывода для 4-го индикатора;
- vPf7: номер вывода для сегмента B; vPf8: номер вывода для 3-го индикатора;
- vPf9: номер вывода для 2-го индикатора; vPf10: номер вывода для сегмента F;
- vPf11: номер вывода для сегмента A; vPf12: номер вывода для 1-го индикатора;

Возвращаемое значение: нет

Библиотека fDigitsSegtPin для работы с многоразрядным семисегметным индикатором

void begin() — инициализирует заданные в конструкторе выводы для работы с индикатором

Параметры: нет
Возвращаемое значение: нет

void print(float vff) — выводит на индикатор нужное значение. Необходимо периодически вызывать функцию в бесконечном цикле для работы динамической индикации.

Параметры:

- vff: число, выводимое на индикатор (диапазон от 0 до 9999)

Возвращаемое значение: нет

Библиотека fDigitsSegtPin для работы с многоразрядным семисегметным индикатором

u8 doPrint_lastDot — поле, задающее необходимость вывода точки в конце числа:
0 - не выводить, 1 - выводить

u8 doPrint_firstZero — поле, задающее необходимость вывода лидирующих нулей:
0 - не выводить, 1 - выводить

doReport_overRange — поле, задающее необходимость вывода сообщения о выходе из диапазона допустимых значения для отображения через COM-порт:
0 - не выводить, 1 - выводить

Многоразрядный секундомер без управления

Пример программы:

```
#include <fDigitsSegtPin.h> // Подключаем библиотеку
fDigitsSegtPin
// Инициализируем объект-4-х разрядный индикатор, передаём
использованные
// для подключения контакты на:
fDigitsSegtPin Display(6, 5, 9, 4, 8, 13, 3, 12, 11, 7, 2, 10);

void setup()
{
    Display.begin();
}
```

Многоразрядный секундомер без управления

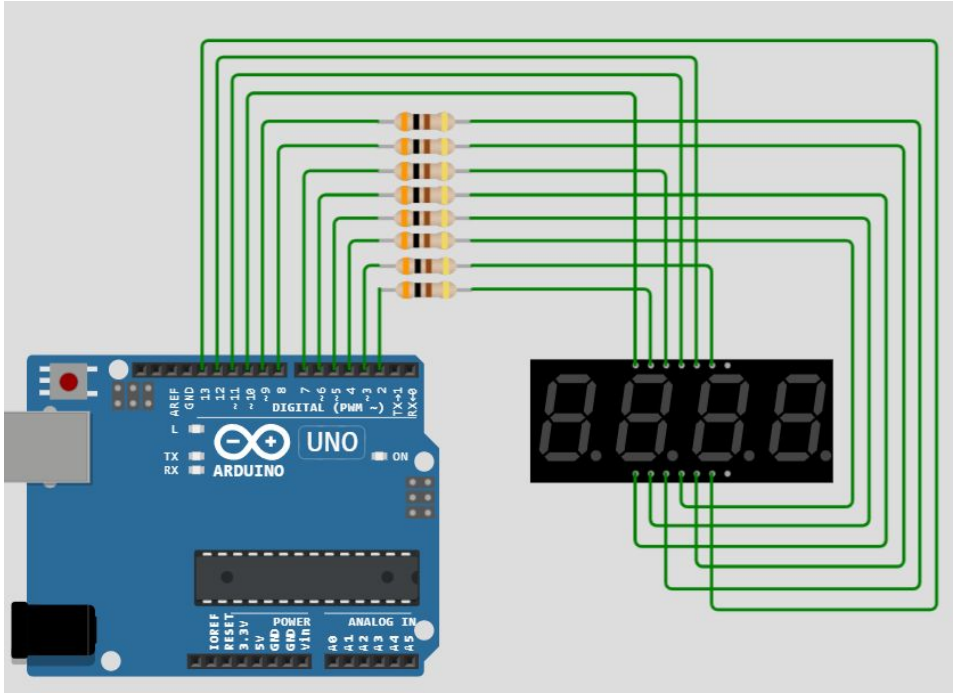
Пример программы:

```
void loop()
{
  static unsigned long timer = millis(); //текущее время
  static float deciSeconds = 0; //время, отображаемое на индикаторе

  if (millis() - timer >= 100) //прошло 100 мс
  {
    timer += 100;
    deciSeconds += 0.1; // 100 мс = 0,1 сек

    if (deciSeconds >= 1000) // выход из диапазона
    {
      deciSeconds = 0;
    }
  }

  Display.print(deciSeconds); //периодический вызов функции в основном цикле
}
```

Многоразрядный секундомер без управления

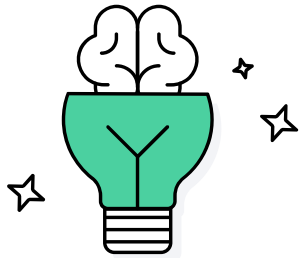
Индикатор отображает количество секунд в диапазоне от 0 до 999 (с точностью до десятых), прошедших с момента включения

В настройках необходимо указать, что индикатор с общим катодом




```
{
  "type": "wokwi-7segment",
  "id": "sevseg1",
  "top": 38.91,
  "left": 295.85,
  "attrs": { "digits": "4", "common": "cathode" }
},
```

Практическое задание №2



Практика: управление яркостью светодиода с помощью потенциометра

Задание:

1) соберите схему в симуляторе WOKWI, подключив индикатор к выводам 2 - 13 (необходимо указать тип индикатора: с общим катодом);
2) создайте скетч с текстом, приведенным выше;
3) проведите моделирование работы

Как выполнять: напишите в чат об удачной работе схемы

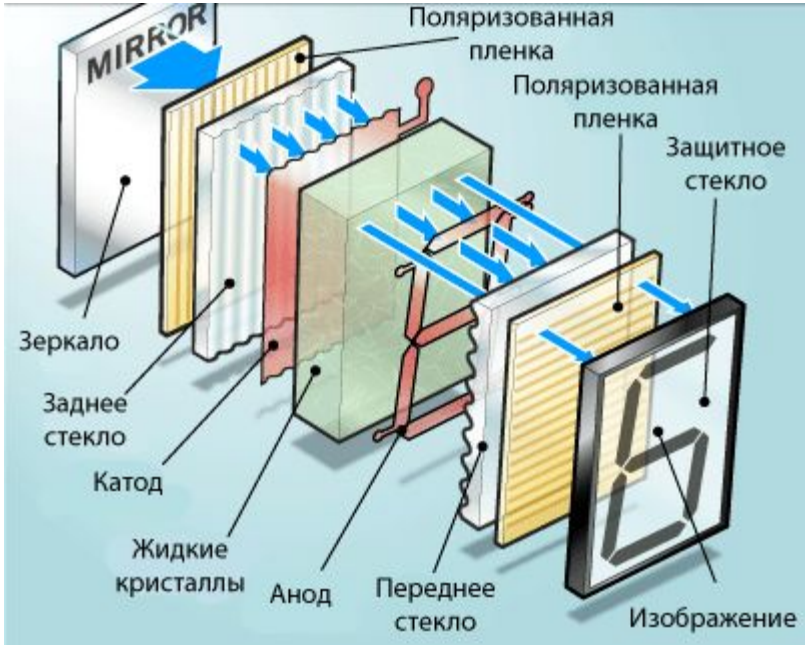
Время выполнения: 5 минут

Как подключить жидкокристаллический индикатор

3

Принцип работы жидкокристаллического индикатора

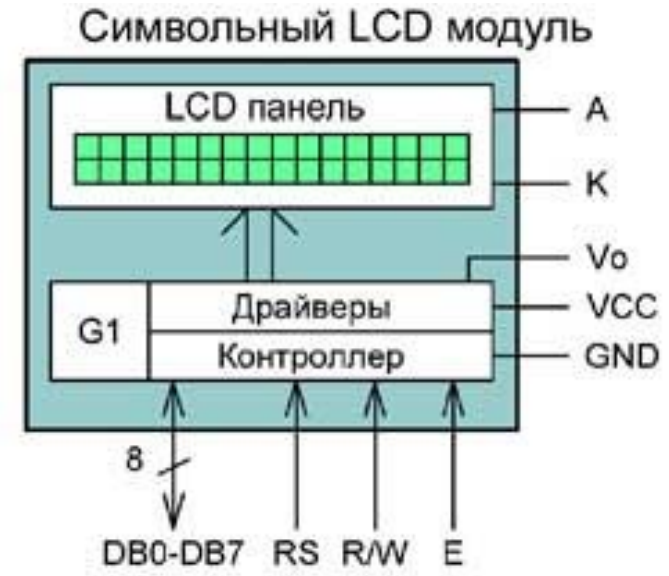
В жидкокристаллическом индикаторе используются общие и сегментные электроды. Сегментные электроды находятся с одной стороны ЖКИ, общие — с противоположной. Между ними расположены жидкие кристаллы. Если подать переменное напряжение, то жидкие кристаллы изменят свою плоскость поляризации и не будут пропускать сквозь себя свет, и сегмент будет отображаться черным цветом



Источник

Устройство жидкокристаллического индикатора

Основу индикатора составляет специализированный контроллер. По назначению выводов и системе команд он совпадает с родоначальником серии - HD44780. Общепринятое название таких микросхем "Dot Matrix Liquid Crystal Display Controller/Driver", из чего следует их двойная функция: контроллер управляет интерфейсом, а драйвер "зажигает" сегменты.



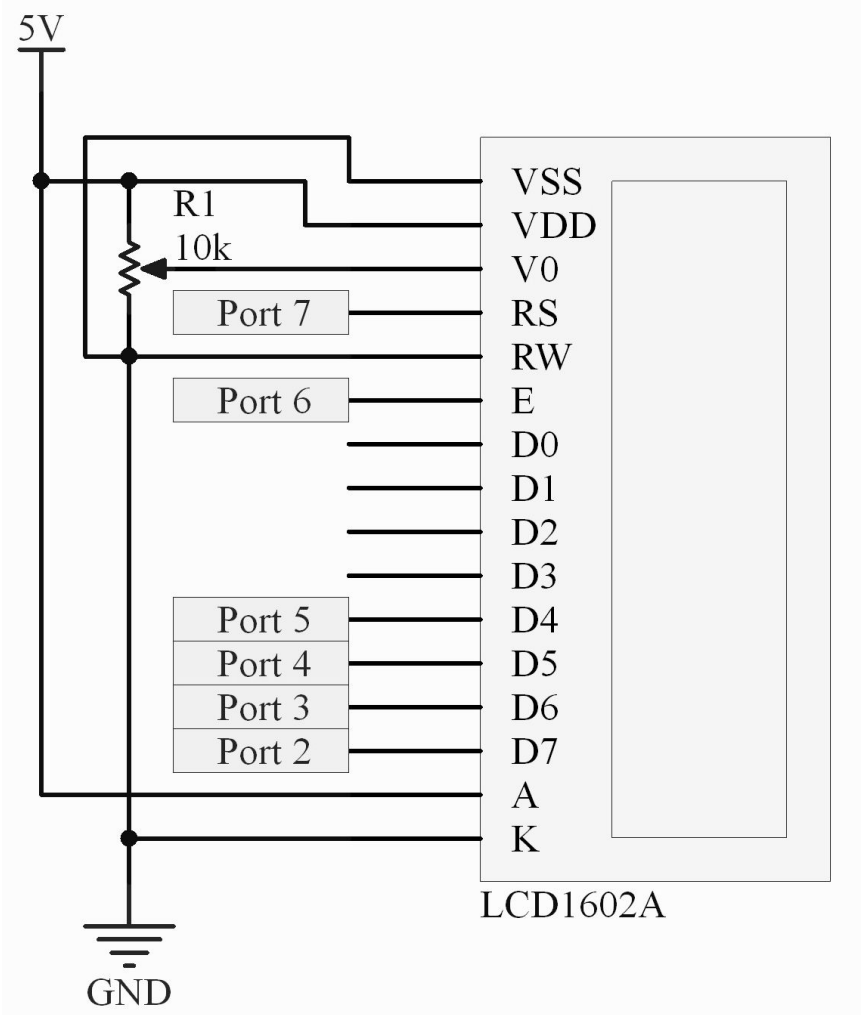
Источник

Подключение жидкокристаллического индикатора

Назначение контактов:

- VSS: «-» питание модуля
- VDD: «+» питание модуля
- VO: Вывод управления контрастом
- RS: Выбор регистра
- RW: Выбор режима записи или чтения (при подключении к земле, устанавливается режим записи)
- E: Строб по спаду
- DB0-DB3: Биты интерфейса
- DB4-DB7: Биты интерфейса
- A: «+» питание подсветки
- K: «-» питание подсветки

Источник



Библиотека LiquidCrystal для работы с жидкокристаллическим индикатором

LiquidCrystal(uint8_t rs, uint8_t enable,
uint8_t d0, uint8_t d1, uint8_t d2, uint8_t d3,
uint8_t d4, uint8_t d5, uint8_t d6, uint8_t d7);
LiquidCrystal(uint8_t rs, uint8_t rw, uint8_t enable,
uint8_t d0, uint8_t d1, uint8_t d2, uint8_t d3,
uint8_t d4, uint8_t d5, uint8_t d6, uint8_t d7);
LiquidCrystal(uint8_t rs, uint8_t rw, uint8_t enable,
uint8_t d0, uint8_t d1, uint8_t d2, uint8_t d3);
LiquidCrystal(uint8_t rs, uint8_t enable,
uint8_t d0, uint8_t d1, uint8_t d2, uint8_t d3); — конструкторы, создающий объект LiquidCrystal

Параметры:

- rs, rw, enable, d0 - d7: номер вывода Arduino, подключаемый к соответствующему выводу индикатора

Возвращаемое значение: нет

Библиотека LiquidCrystal для работы с жидкокристаллическим индикатором

void begin(uint8_t cols, uint8_t rows, uint8_t charsize = LCD_5x8DOTS) — инициализирует интерфейс для взаимодействия с LCD-экраном и задает размеры (ширину и высоту) области вывода экрана

Параметры:

- cols: количество столбцов экрана;
- rows: количество строк экрана;
- charsize: размер символа

Возвращаемое значение: нет

Библиотека LiquidCrystal для работы с жидкокристаллическим индикатором

void clear() — очищает LCD-экран и перемещает курсор в левый верхний угол

Параметры: нет
Возвращаемое значение: нет

void home() — перемещает курсор в левый верхний угол экрана

Параметры: нет
Возвращаемое значение: нет

void noDisplay() — выключает LCD-экран. Текст сохраняется в памяти

Параметры: нет
Возвращаемое значение: нет

Библиотека LiquidCrystal для работы с жидкокристаллическим индикатором

void display() — включает LCD-экран после того, как он был выключен функцией noDisplay(). После выполнения этой функции на экране отобразится текст (и курсор), которые были на нем до выключения

Параметры: нет
Возвращаемое значение: нет

void noBlink() — отключает отображение мигающего курсора на LCD-экране

Параметры: нет
Возвращаемое значение: нет

Библиотека LiquidCrystal для работы с жидкокристаллическим индикатором

void blink() — включает на LCD-экране мигающий курсор

Параметры: нет
Возвращаемое значение: нет

void noCursor() — отключает отображение курсора на LCD-экране

Параметры: нет
Возвращаемое значение: нет

void cursor() — отображает на LCD-экране курсор, куда будет выведен символ

Параметры: нет
Возвращаемое значение: нет

Библиотека LiquidCrystal для работы с жидкокристаллическим индикатором

void scrollDisplayLeft() — осуществляет прокрутку содержимого дисплея (весь текст и курсор) на один символ влево
Параметры: нет
Возвращаемое значение: нет

void scrollDisplayRight() — Осуществляет прокрутку содержимого дисплея (весь текст и курсор) на один символ вправо
Параметры: нет
Возвращаемое значение: нет

void leftToRight() — устанавливает режим вывода текста на LCD слева-направо (режим по умолчанию)
Параметры: нет
Возвращаемое значение: нет

Библиотека LiquidCrystal для работы с жидкокристаллическим индикатором

void rightToLeft() — устанавливает режим вывода текста на LCD справа-налево
Параметры: нет
Возвращаемое значение: нет

void autoscroll() — включает автоматическую прокрутку текста на LCD. При выводе нового символа, все предыдущие символы будут сдвигаться на одну позицию
Параметры: нет
Возвращаемое значение: нет

void noAutoscroll() — отключает автоматическую прокрутку текста в LCD
Параметры: нет
Возвращаемое значение: нет

Библиотека LiquidCrystal для работы с жидкокристаллическим индикатором

void createChar(uint8_t location, uint8_t charmap[]) — создает пользовательский символ для LCD-экрана. Дисплей поддерживает до 8 пользовательских символов (пронумерованных от 0 до 7) размером 5x8 пикселей

Параметры:

- location: номер пользовательского символа, который необходимо создать (от 0 до 7);
- charmap: данные о пикселях пользовательского символа

Возвращаемое значение: нет

Библиотека LiquidCrystal для работы с жидкокристаллическим индикатором

void setCursor(uint8_t col, uint8_t row) — задает позицию курсора на LCD-экране

Параметры:

- col: координата X позиции курсора (0 означает первый столбец);
- row: координата Y позиции курсора (0 означает первую строку)

Возвращаемое значение: нет

size_t write(uint8_t value) — выводит символ на LCD-экран

Параметры:

- value: символ, который необходимо вывести на экран

Возвращаемое значение: количество отправленных байт

Отображение яркости светодиода на жидкокристаллическом индикаторе

```
#include <LiquidCrystal.h> // Подключаем стандартную библиотеку LiquidCrystal
// Инициализируем объект-экран, передаём использованные
// для подключения контакты на Arduino в порядке:
// RS, E, D4, D5, D6, D7
LiquidCrystal lcd(12, 11, 10, 9, 8, 7);
const int potPin = A0;       // потенциометр
const int ledPin = 6;        // светодиод на выводе с ШИМ
int potValue = 0;      // значение от потенциометра
int brightness = 0;    // конвертируем в яркость
int progress = 0;      // индикатор прогресса
//Пользовательский символ
byte pBar[8] =
{
  B11111,
  B11111,
  B11111,
  B11111,
  B11111,
  B11111,
  B11111,
};
```

Отображение яркости светодиода на жидкокристаллическом индикаторе

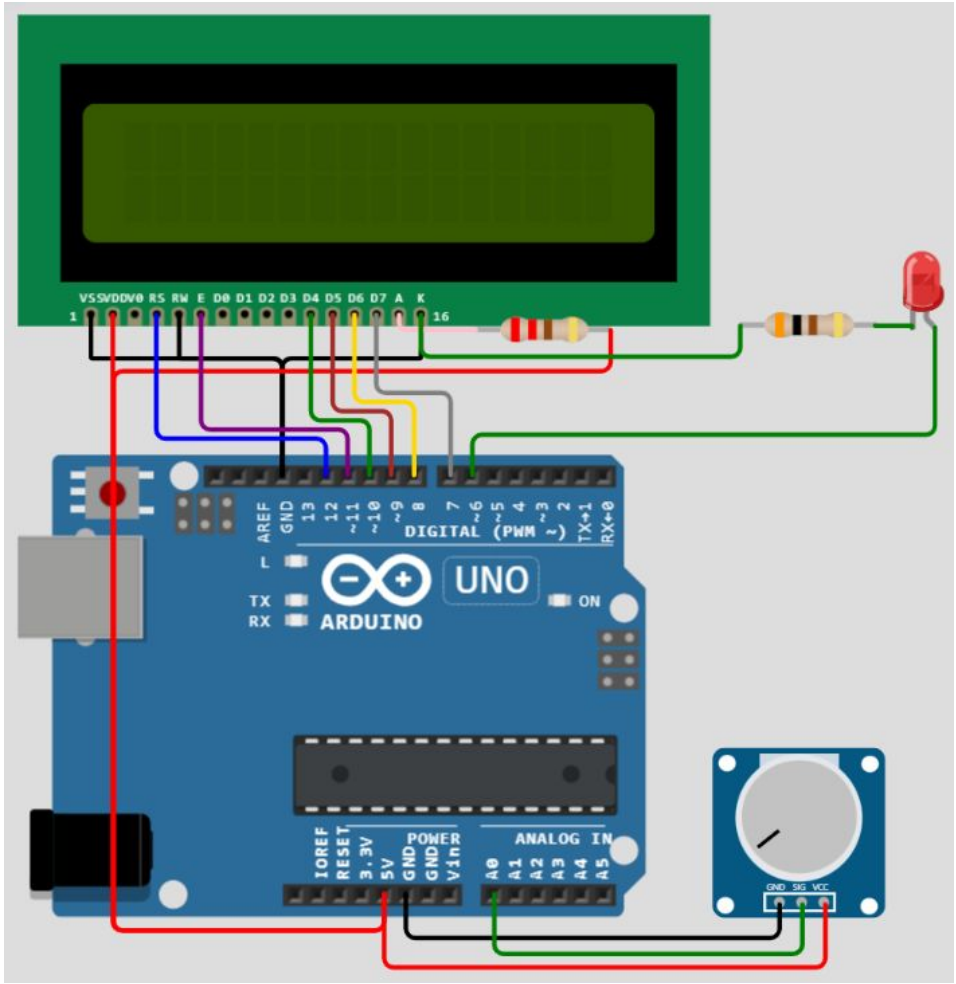
```
void setup()
{
  // устанавливаем размер (количество столбцов и строк) экрана
  lcd.begin(16, 2);
  // светодиод
  pinMode(ledPin, OUTPUT);
  // Выводим сообщение в первой строчке
  lcd.print(" LED Brightness");
  // Создаем символы для индикатора прогресса
  lcd.createChar(0, pBar);
  // ощищаем экран
  lcd.clear();
  // выводим сообщение в первой строчке
  lcd.print(" LED Brightness");
}
```

Отображение яркости светодиода на жидкокристаллическом индикаторе

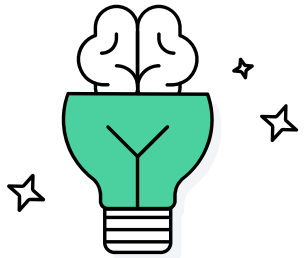
```
void loop()
{
  static int oldProgress = 0; //старое значение на экране
  potValue = analogRead(potPin); // считываем показания с потенциометра
  brightness = map(potValue, 0, 1024, 0, 255); // конвертируем значения в яркость от 0 до 255
  analogWrite(ledPin, brightness); // меняем яркость светодиода в зависимости от значений от потенциометра
  progress = map(brightness, 0, 255, 0, 17); // конвертируем значения яркость в проценты для индикатора от 0 до 17
  if(progress != oldProgress) //данные изменились
  {
    oldProgress = progress;
    lcd.setCursor(0, 1); // устанавливаем курсор на второй строчке
    lcd.print("                "); // очищаем экран (16 пробелов)
    for (int i = 0; i < progress; i++) // выводим индикатор
    {
      lcd.setCursor(i, 1);
      lcd.write(byte(0));
    }
  }
}
```

Отображение яркости светодиода на жидкокристаллическом индикаторе

В примере используется 4-х битная шина данных для управления индикатором



Практическое задание №3



Практика: вывод информации на жидкокристаллический индикатор

Задание:

1) соберите схему в симуляторе WOKWI, подключив жидкокристаллический индикатор к выводам 7 - 12, а светодиод - к выводу 6, а потенциометр - к выводу A0;
2) создайте скетч с текстом, приведенным выше;
3) проведите моделирование работы

Как выполнять: напишите в чат об удачной работе схемы

Время выполнения: 5 минут

Итоги

5

Итоги занятия

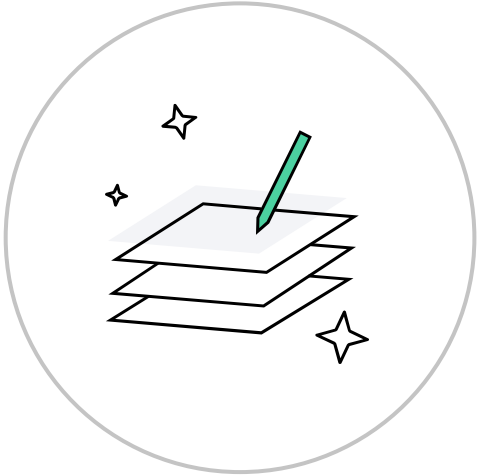
Сегодня мы

1. Узнали особенности подключения семисегментного индикатора
2. Научились подключать многоразрядный семисегментный индикатор
3. Научились реализовывать принцип динамической индикации
4. Научились подключать и использовать жидкокристаллический индикатор

Домашнее задание

Давайте посмотрим ваше домашнее задание.

1. Вопросы по домашней работе задавайте в чате группы
2. Задачи можно сдавать по частям
3. Зачёт по домашней работе ставят после того, как приняты все задачи



Задавайте вопросы и пишите отзыв о лекции