## Знакомство с текстовым экраном LCD 16x2

LCD дисплей 16х2 предназначен для вывода текста и специальных символов. Экран оборудован подсветкой. По изображению похож на дисплеи старых мобильных телефонов вроде Nokia 3310 или Siemens C35.

Достоинства:

* малая потребляемая мощность;
* низкое напряжение питания;
* долговечность.

Рассмотрим LCD-дисплеи на базе контроллера HD44780.  
Этот монохромный дисплей имеет опциональную подсветку и может отображать 2 строки по 16 символов. Разрешение символов — 5x8 точек.   
Чем хороши такие дисплеи? Контроллер HD44780 — стандарт де-факто среди небольших монохромных LCD-дисплеев.

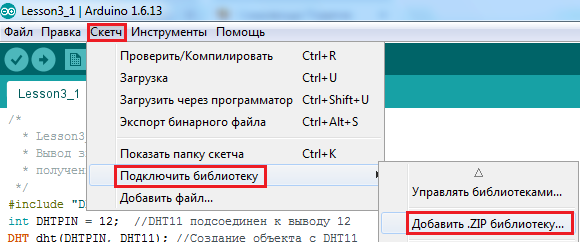
Назначения выводов LCD дисплея 16х2:

|  |  |  |  |
| --- | --- | --- | --- |
| Номер вывода | Сигнал | I - вход  O - выход | Назначение сигнала |
| 1 | Vss | - | Земля (общий провод) |
| 2 | Vdd | - | Питание + 5 В |
| 3 | Vo | - | Управление контрастностью дисплея. Вход для подключения среднего вывода делителя напряжения + 5 В. Можно использовать подстроечный резистор сопротивлением 10-20 кОм. |
| 4 | RS | I | Выбор регистра: 0 – регистр команд; 1 – регистр данных. Низкий уровень сигнала означает, что на шине данных сформирована команда, высокий уровень – на шине данные. |
| 5 | R/W | I | Направление передачи данных:  0 – запись;  1 – чтение.  Во многих приложениях функция чтения не используется, поэтому сигнал часто подключается к земле. |
| 6 | E | I | Строб операции шины (по отрицательному фронту). |
| 7 | DB0 | I/O | Младшие биты восьми битного режима. При четырех битном интерфейсе не используются. |
| 8 | DB1 | I/O |
| 9 | DB2 | I/O |
| 10 | DB3 | I/O |
| 11 | DB4 | I/O | Старшие биты восьми битного режима или биты данных  четырех битного интерфейса. |
| 12 | DB5 | I/O |
| 13 | DB6 | I/O |
| 14 | DB7 | I/O |
| 15 | A | - | Анод питания подсветки (+). |
| 16 | K | - | Катод питания подсветки (-).  Ток должен быть ограничен. |



## Подключение библиотеки LiquidCrystal для работы с текстовым экраном

Для того, чтобы подключить библиотеку LiquidCrystal вам необходимо в главном меню выбрать Скетч/Подключить библиотеку/Добавить .ZIP библиотеку/LiquidCrystal.zip



Основные команды для работы с библиотекой LiquidCrystal:

//Работа с курсором

lcd.setCursor(0, 0); // Устанавливаем курсор (номер ячейки, строка)

lcd.home(); // Установка курсора в ноль (0, 0)

lcd.cursor(); // Включить видимость курсора (подчеркивание)

lcd.noCursor(); // Убрать видимость курсора (подчеркивание)

lcd.blink(); // Включить мигание курсора (курсор 5х8)

lcd.noBlink(); // Выключить мигание курсора (курсор 5х8)

//Вывод информации

lcd.print("Arduino"); // Вывод информации

lcd.clear(); // Очистка дисплея, (удаление всех данных)   
 установка курсора в ноль

lcd.rightToLeft(); // Запись производится справа налево

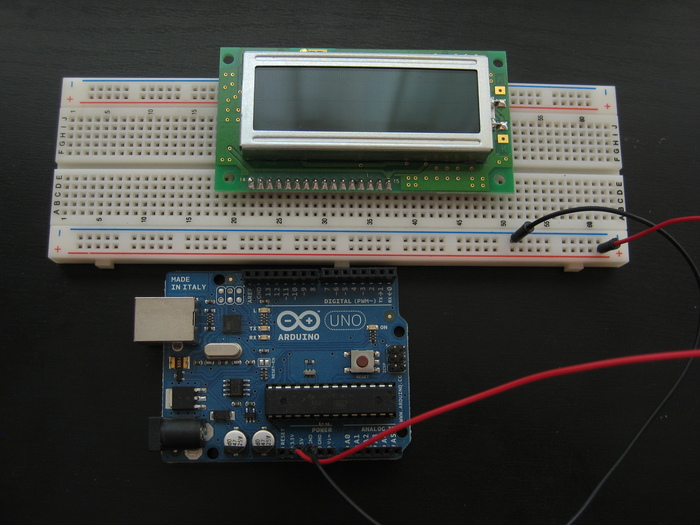
lcd.leftToRight(); // Запись производится слева направо

lcd.scrollDisplayRight(); // Смещение всего изображенного на дисплее на один   
 символ вправо

lcd.scrollDisplayLeft(); // Смещение всего изображенного на дисплее на один   
 символ влево

## Подключение текстового экрана к Arduino UNO

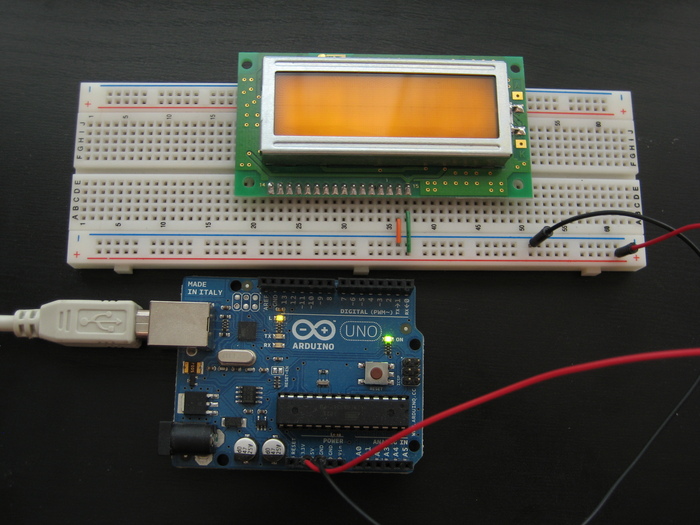
Закрепите экран на breadboard'е и подведите к рельсам питания breaboard'а питание +5 В и землю с Arduino.



Питание и земля понадобятся не один раз, поэтому удобнее пробросить их именно на рельсы.

### Включение подсветки

Фоновая подсветка дисплея — это отдельный контур, не связанный с остальным. Включить её можно подав +5 В на 15-й контакт дисплея и подключив 16-й контакт к земле. Соединив эти два контакта с соответствующими рельсами, можно включить Arduino и увидеть, что дисплей засветился.



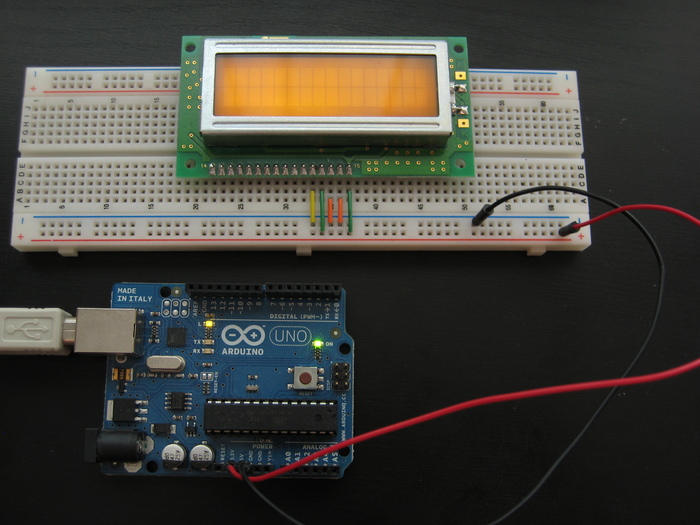
### Включение питания знакосинтезатора

Далее необходимо подключить цепь, отвечающую за отображение символов. Для этого предназначены контакты 1, 2 и 3 на дисплее. Перед подключением отключите Arduino от питания.

* Первый — это земля. Соедините его с рельсой земли.
* Второй — питание. Соедините его с рельсой +5 В.
* Третий — контрастность.

Для получения максимально контрастного изображения соедините его с рельсой земли. Вы можете подать на этот контакт произвольное напряжение от 0 до 5 В, чем оно выше, тем тусклее будет изображение, но вместе с этим снизится энергопотребление. Для возможности плавной регулировки контрастности можете подать на этот контакт выходной сигнал потенциометра.

После подключения, если включить Arduino, вы можете увидеть прямоугольные знакоместа. В зависимости от комбинации цветов текста и подсветки они могут быть как яркими и хорошо заметными, так и едва заметными. Это нормально: в любом случае, текст будет смотреться отлично.



### Подключение шины данных

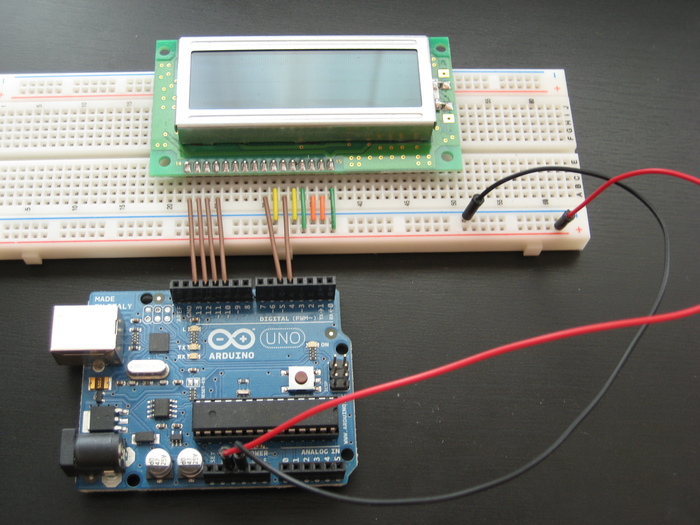
Для подключения дисплея достаточно использовать 6 линий, 6 контактов на Arduino.

Соединим 5-й контакт дисплея, который отвечает за выбор чтение/запись с рельсой земли. Это означает «всегда писать».

Затем, соединяем Arduino и экран нашими 6-ю линиями коммуникации. Какие именно контакты будут выбраны на Arduino не имеет значения: мы зададим их в программе, но для примера была выбрана такая конфигурация:

* 4-й контакт дисплея — 4-й контакт Arduino. Это линия адресного сигнала. Известная как A0 или RS. В зависимости от того, 0 она или 1, дисплей понимает, имеем ли мы на линии данных команду вроде «передвинуть курсор» или код символа для отображения;
* 6-й контакт дисплея — 5-й контакт Arduino. Это линия разрешения доступа к данным. Известная, как E или Enable. Когда эта линия становится единицей, дисплей исполняет команду или выводит символ с линии данных;
* 11-й, 12-й, 13-й, 14-й контакт дисплея — 10-й, 11-й, 12-й, 13-й контакт Arduino соответственно. Это линии данных. Известные как DB4, DB5, DB6, DB7.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| LCD экран 16х2 | 1 | 2 | 3 | 4 | 6 | 11 | 12 | 13 | 14 | 15 | 16 |
| Arduino UNO | GND | +5V | GND | 4 | 5 | 10 | 11 | 12 | 13 | +5V | GND |



Экран подключен и готов к приёму данных. Осталось написать программу для Arduino.

## Вывод текста на текстовый экран

Для вывода текста с Arduino воспользуйтесь библиотекой LiquidCrystal. Для вывода приветствия, воспользуйтесь кодом ниже:

**Программный код**

/\*

\* Lesson4\_1

\* Вывод текста на текстовый экран LCD 16x2

\*/

#include <LiquidCrystal.h> // Подключаем стандартную библиотеку LiquidCrystal

LiquidCrystal lcd(4, 5, 10, 11, 12, 13); // Инициализируем объект-экран, передаём использованные

// для подключения контакты на Arduino в порядке:

// RS, E, DB4, DB5, DB6, DB7

void setup()

{

lcd.begin(16, 2);// устанавливаем размер (количество столбцов и строк) экрана

lcd.print("Hello world!");// печатаем первую строку

lcd.setCursor(0, 1); // устанавливаем курсор в колонку 0, строку 1. То есть на

// самом деле это вторая строка, т.к. нумерация начинается с нуля

lcd.print("I love Arduino");// печатаем вторую строку

}

void loop()

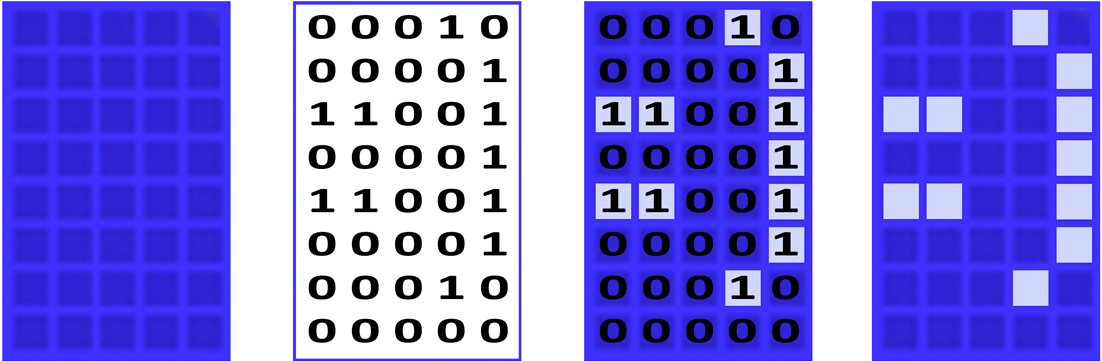
{

}

## Вывод специальных символов на текстовый экран

С выводом текста разобрались, буквы английского алфавита зашиты в память контроллера внутри дисплея и с ними проблем нет. А вот что делать, если нужного символа в памяти контроллера нет?

Не беда, требуемый символ можно сделать вручную (всего до 7ми символов). Ячейка, в рассматриваемых нами дисплеях, имеет разрешение 5х8 точек. Все, к чему сводится задача создания символа, это написать битовую маску и расставить в ней единички в местах где должны гореть точки и нолики где нет. В ниже приведенном примере нарисуем смайлик.



**Программный код**

/\* Lesson4\_2

\* Создание собственных символов на LCD экране

\*/

#include <LiquidCrystal.h> // добавляем необходимую библиотеку

byte smile[8] = // битовая маска символа улыбки

{

B00010,

B00001,

B11001,

B00001,

B11001,

B00001,

B00010,

};

LiquidCrystal lcd(4, 5, 10, 11, 12, 13); // RS, E, DB4, DB5, DB6, DB7

void setup(){

lcd.begin(16, 2); // задаем размерность экрана

lcd.createChar(1, smile); // создаем символ под номером 1

lcd.setCursor(0, 0); // устанавливаем курсор в начало 1 строки

lcd.print("\1"); // выводим смайлик (символ под номером 1) - "\1"

}

void loop(){

}