1. Лабораторная работа №2.   
   Передача данных через последовательный порт. Индикация на LCD-дисплее

**Цель работы:** Изучить основные принципы организации обмена данными между контроллером Arduino и компьютером через виртуальный   
COM-порт. Научится производить индикацию на LCD-дисплее.

* 1. Краткие теоретические сведения

**Последовательный порт** или интерфейс стандарта RS-232 – это одно из многих средств передачи информации между устройствами.

Все СОМ-порты обладают несколькими свойствами:

1) Полнодуплексный обмен данными. Существуют два аппаратно и программно независимых канала передачи данных. У COM-портов присутствуют собственные буферы приема и передачи данных. В этих буферах данные выстраиваться в очередь на передачу и очередь на прочтение данных процессором. Любая программа может обратиться к СОМ-порту и получить данные из его буфера, тем самым очистив его.

2) Набор сервисных сигналов. Сервисные сигналы, предусмотренные стандартом RS-232c, позволяют организовать обмен данными между двумя устройствами одновременно в обоих направлениях. Сервисные сигналы представлены отдельными цифровыми входами и выходами с памятью.

3) Программная независимость. Для реализации этой независимости используется UART (Universal Asynchronous Receiver-Transmitter). UART – узел вычислительных устройств, предназначенный для связи с другими цифровыми устройствами. Он преобразует заданный набор данных в последовательный вид так, чтобы было возможно передать их по однопроводной цифровой линии другому аналогичному устройству. UART полностью реализован аппаратно и не зависит от программного обеспечения и ОС.

4) Асинхронная передача данных по каналу связи. Означает то, что РС может послать данные на конечное устройство, не заботясь о синхронности их поступления. Конечное устройство само подстраивается под полученные данные. В синхронных протоколах для этого служит специальный сигнал, передающийся по отдельному проводу.

Для обмена данными с устройствами, которые поддерживают последовательный интерфейс используется набор функций Serial. Для обмена данными Serial используют цифровые порты ввод/вывода 0 (RX) и 1 (TX), а также USB порт. Важно учитывать, что при использовании функций Serial нельзя одновременно использовать порты 0 и 1 для других целей

Среда разработки Arduino имеет Встроенный монитор последовательного интерфейса (Serial monitor), который можно вызвать командой   
Сервис -> Монитор порта или нажатием комбинации клавиш Ctrl+Shif+M. Для начала обмена данными необходимо запустить монитор нажатием кнопки Serial monitor и выставить ту же скорость связи (обычно она равна 9600), с которой вызвана функция begin().

**LCD дисплей**, который используется в данной работе, имеет опциональную подсветку и может отображать 2 строки по 16 символов. Разрешение символов — 5x8 точек.

Для работы LCD-дисплея необходимо подключить дополнительную библиотеку «LiquidCrystal».

Ниже представлены некоторые функции lcd.

home() – возвращает курсор в начало экрана.

clear() – возвращает курсор в начало экрана и стирает все данные.

write(ch) – выводит одиночный символ ch на дисплей.

cursor() и noCursor() – позволяют показать курсор на дисплее (символ подчёркивания) и скрыть его.

blink() и noBlink() – включить и выключить мигание курсора, если включено его отображение.

display() и noDisplay() – включить и выключить дисплей.

scrollDisplayLeft() и scrollDisplayRight() – прокрутить экран на один символ влево или вправо.

autoscroll() и noAutoscroll() – включить и выключить режим автопрокрутки. В этом режиме при выводе каждого следующего символа содержимое экрана будет смещено на один символ вліво (или вправо, если включен режим вывода справа налево), а выводимый символ займёт место первого сдвинутого. В этом режиме все последующие символы выводятся в одно и то же место, вытесняя текущее содержимое экрана.

leftToRight() и rightToLeft() – устанавливают направление вивода текста: слева направо и справа налево, соответственно.

createChar(ch, bitmap) – создание символа с кодом ch (от 0 до 7), с использованием массива битовых масок для задания тёмных и светлых точек.

**Дисплей Nokia 5110**, 84\*48 имеет расширение 84х48 пикселей и позволяет отображать не только текст, но разнообразную графическую информацию.

Некоторые функции из библиотеки «PCD8544»:

drawColumn(unsigned char lines, unsigned char value) – позволяет нарисовать вертикальную линию заданной длины, в заданной позиции.

display() и noDisplay() – включить и выключить дисплей.

createChar(unsigned char chr, const unsigned char \*glyph) – создать свой собственный символ.

cursor() и noCursor() – позволяют показать курсор на дисплее (символ подчёркивания) и скрыть его.

stop()- Возвращает курсор в начало экрана и выключает дисплей.

setCursor(unsigned char column, unsigned char line) – установить курсор в заданную позицию.

* 1. Порядок выполнения работы

1. Собрать схему LCD-дисплея.
2. Импортировать и подключить к проекту библиотеку «LiquidCrystal».
3. Создать программу, которая производит чтение данных из COM-порта и выполняет их индикацию на LCD-дисплей, согласно варианту задания.
4. Импортировать и подключить к проекту библиотеку PCD8544.
5. Написать программу, которая производит индикацию данных на дисплей Nokia, согласно варианту задания.
6. Произвести компиляцию проектов.
7. Загрузить программы в контроллер и проверить их работу.
   1. Постановка задачи

*Задание 1*

Организовать перенос получаемых из последовательного порта данных в нижний ряд знакомест дисплея, если они не помещаются в верхнюю строку. Если данные не помещаются даже в две строки, вывести сообщение «Тoo much data» на дисплей.

*Задание2*

Нарисовать закрашенный треугольник на дисплее Nokia с использованием функции drawColumn(unsigned char lines, unsigned char value).

* 1. Текст программы

*Задание 1*

#include <LiquidCrystal.h>

// инициализация

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {

// устанавливаем количество столбцов и строк:

lcd.begin(16, 2);

//инициализуруем последовательный порт передачи данных

Serial.begin(9600);

}

void loop() {

// когда приходят данные с порта...

if (Serial.available()) {

delay(100);

lcd.clear(); // очищаем экран дисплея

if (Serial.available() > 32) {

lcd.setCursor(0, 0);

lcd.write("Тoo much data");

} else {

int i = 0;

// считываем поочередно все символы

while (Serial.available() > 16) {

lcd.setCursor(i, 0); //устанавливаем курсор

i++;

lcd.write(Serial.read()); // отображаем символ на дисплее

}

i = 0;

while (Serial.available() > 0) {

lcd.setCursor(i, 1); //устанавливаем курсор

i++;

// отображаем символ на дисплее

lcd.write(Serial.read());

}

}

}

}

*Задание 2*

#include <PCD8544.h>

static PCD8544 lcd;

void setup() {

lcd.begin(84, 48);

lcd.setCursor(0,0);

lcd.drawColumn(1, 1);

lcd.drawColumn(1, 2);

lcd.drawColumn(1, 3);

lcd.drawColumn(1, 4);

lcd.drawColumn(1, 5);

lcd.drawColumn(1, 6);

lcd.drawColumn(1, 7);

lcd.drawColumn(1, 8);

lcd.drawColumn(1, 7);

lcd.drawColumn(1, 6);

lcd.drawColumn(1, 5);

lcd.drawColumn(1, 4);

lcd.drawColumn(1, 3);

lcd.drawColumn(1, 2);

lcd.drawColumn(1, 1);

}

void loop() {

}

* 1. Особенности подключения библиотек. Особенности подключения LCD дисплея

Библиотека LiquidCrystal была подключена с помощью команды меню Sketch > Include Library > LiquidCrystal.

Для подключения библиотеки PCD8544 ее пришлось скачать через меню Sketch > Include Library > Manage Libraries.

ВЫВОДЫ

В данной лабораторной работе мы изучили основные принципы организации обмена данными между контроллером Arduino и компьютером через виртуальный COM-порт. Научились производить индикацию на двухстрочном LCD-дисплее и на дисплее Nokia 5110, размерностью 84х48 пикселей, который позволяет отображать не только текст, но разнообразную графическую информацию.

Также при выполнении лабораторной работы мы научились подключать библиотеки к среде разработки Arduino вручную и автоматическим способом.