

Малое инновационное предприятие  
Центр Молодёжного Инновационного Творчества  
**ЛАБОРАТОРИЯ ЮНЫХ КОНСТРУКТОРОВ**  
[www.cmit.vstu.ru](http://www.cmit.vstu.ru)

# Робошкола.Дети 2020

## Сити-фермерство

Собрать сити-ферму для выращивания растений в домашних условиях.

Выбрать семена для посадки и подобрать для них оптимальный состав почвы.

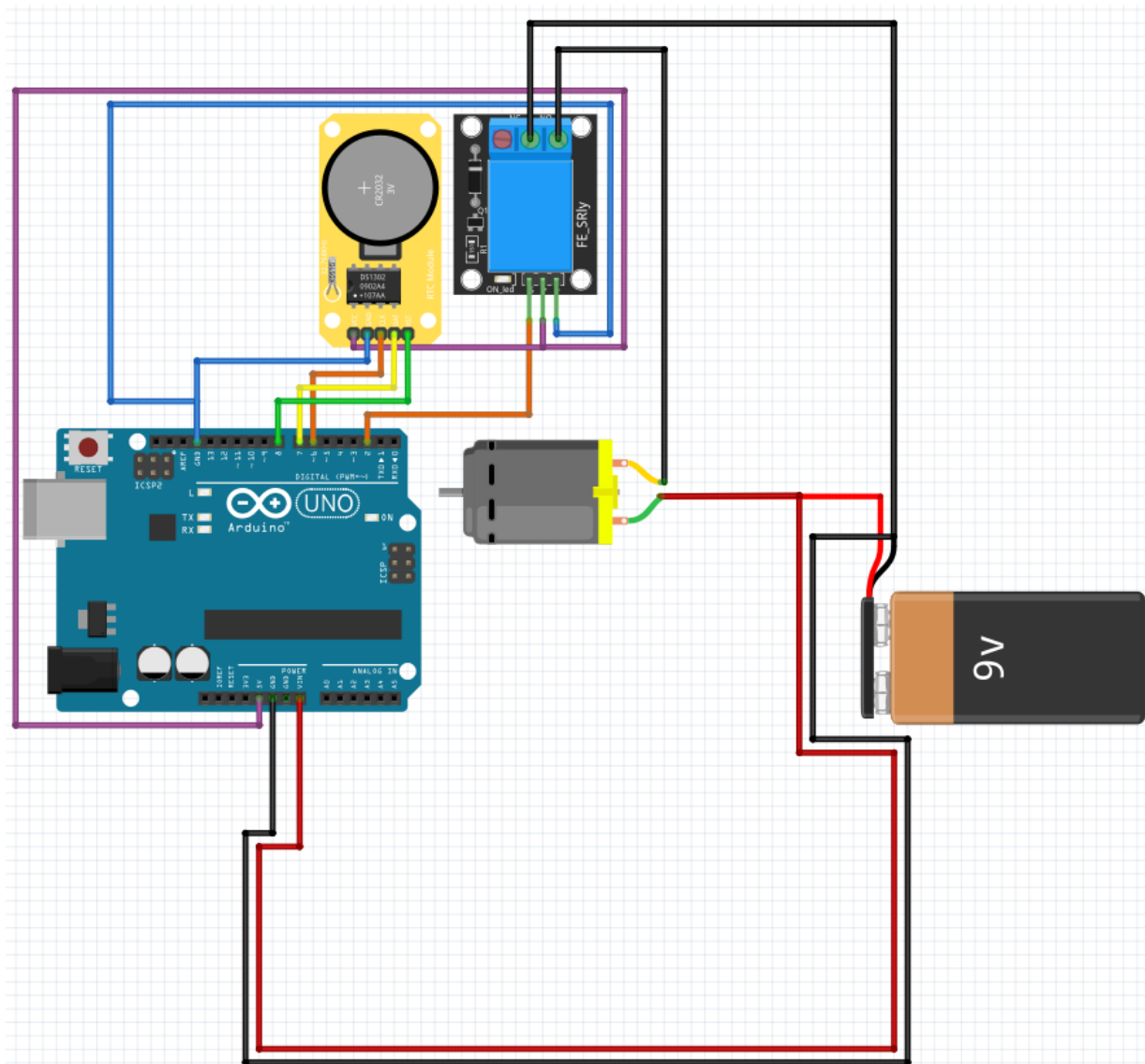
Приготовить горшки с почвой и посадить растения на оптимальную глубину.

Собрать и запрограммировать схему для обеспечения автоматического полива по заданному расписанию в соответствии с заданиями.

Рост численности населения мира, все большая концентрация населения в городах, на фоне изменения потребительских предпочтений в сторону приоритета «здоровой», «натуральной», «органической» пищи, с особой остротой поднимают такие вопросы, как развитие дополнительной или альтернативной системы бесперебойного снабжения или самообеспечение городов продуктами питания и обеспечение перспективной продовольственной безопасности. Это обуславливает актуальность развития технологий урбанизированного агропроизводства (сити-фермерства) и повышения востребованности специалистов из сферы агроинженерии и агробизнеса.

**Сити-фермерство** - это выращивание свежей зелени, овощей и ягод в городских условиях в вертикальных автоматизированных умных теплицах и сити-фермах, с использованием гидро-, аэро- и аквапоники и современных технологий ухода за растением: от полива до контроля света.

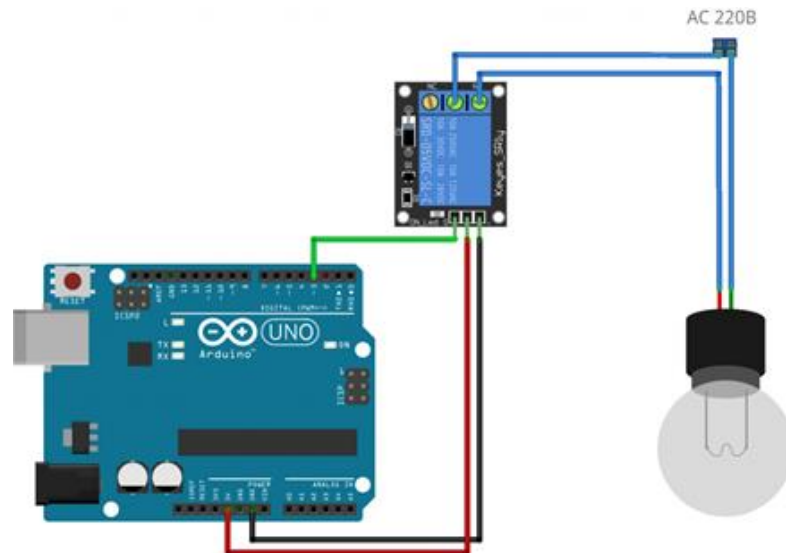
**Задача 1: сборка схемы автоматического полива с использованием часов реального времени DS1302, электромагнитного реле, насоса, платы Arduino Uno и аккумулятора.**



### *Основы работы с электромагнитным реле<sup>1</sup>*

*Модуль реле необходим для включения устройств, которым необходим внешний источник энергии. Например, лампа накаливания, вентилятор, электрочайник и т.д. Подключение реле производится в соответствии со следующей схемой:*

<sup>1</sup> Схемы подключения в примерах могут не совпадать с подключением в задачах



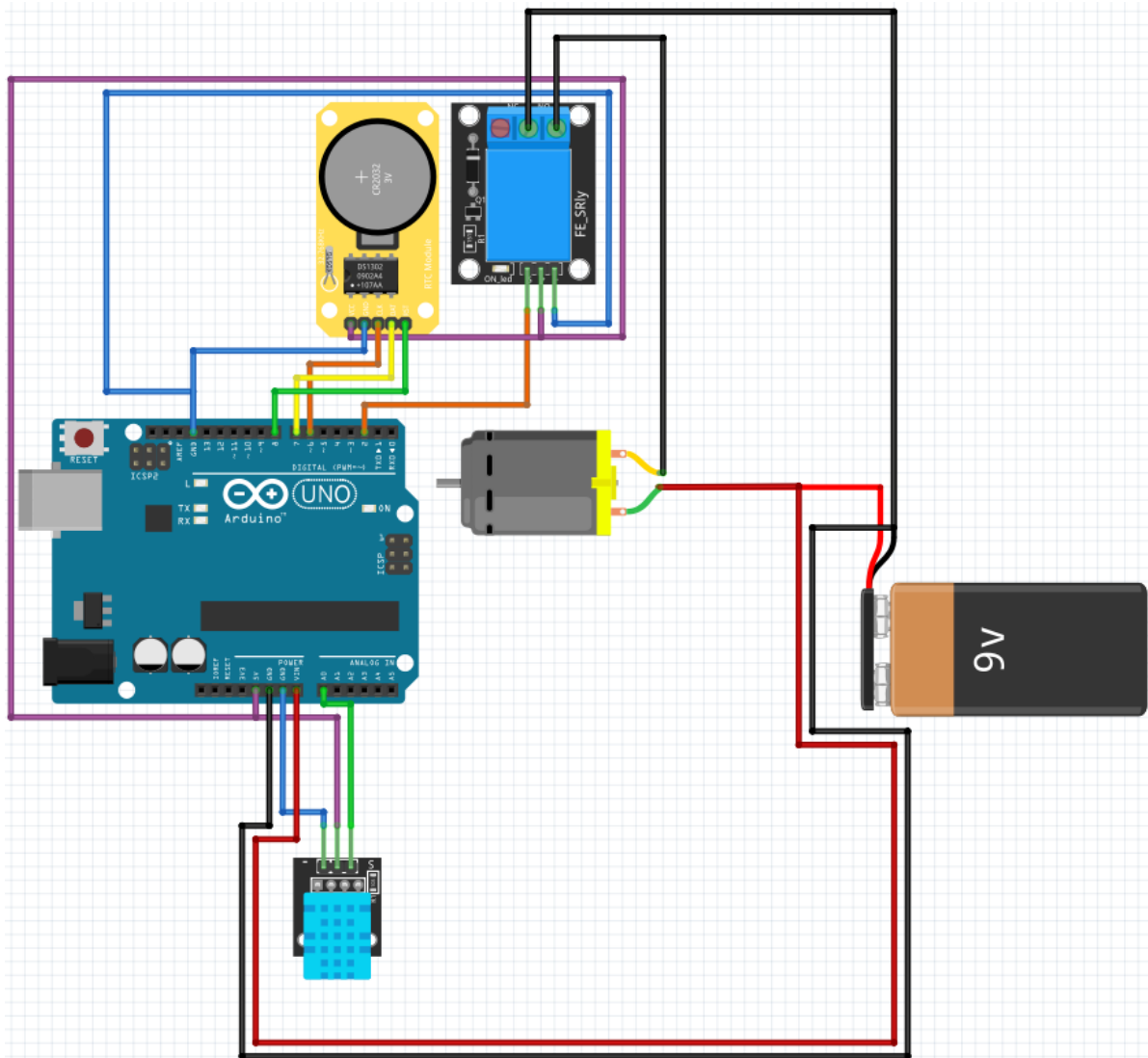
*Схема подключения электромагнитного реле*

*Программа для управления реле:*

```
1 int relayPin = 3;
2 void setup() {
3   Serial.begin(9600);
4   pinMode( relayPin, OUTPUT );
5 }
6
7 void loop() {
8   digitalWrite( relayPin, HIGH );
9   Serial.println( "Включили свет!" );
10  delay( 1000 );
11
12  digitalWrite( relayPin, LOW );
13  Serial.println( "Отключили свет!" );
14  delay( 1000 );
15 }
```

*Программа для работы с электромагнитным реле*

## Задача 2: подключение датчика температуры и влажности DHT11.



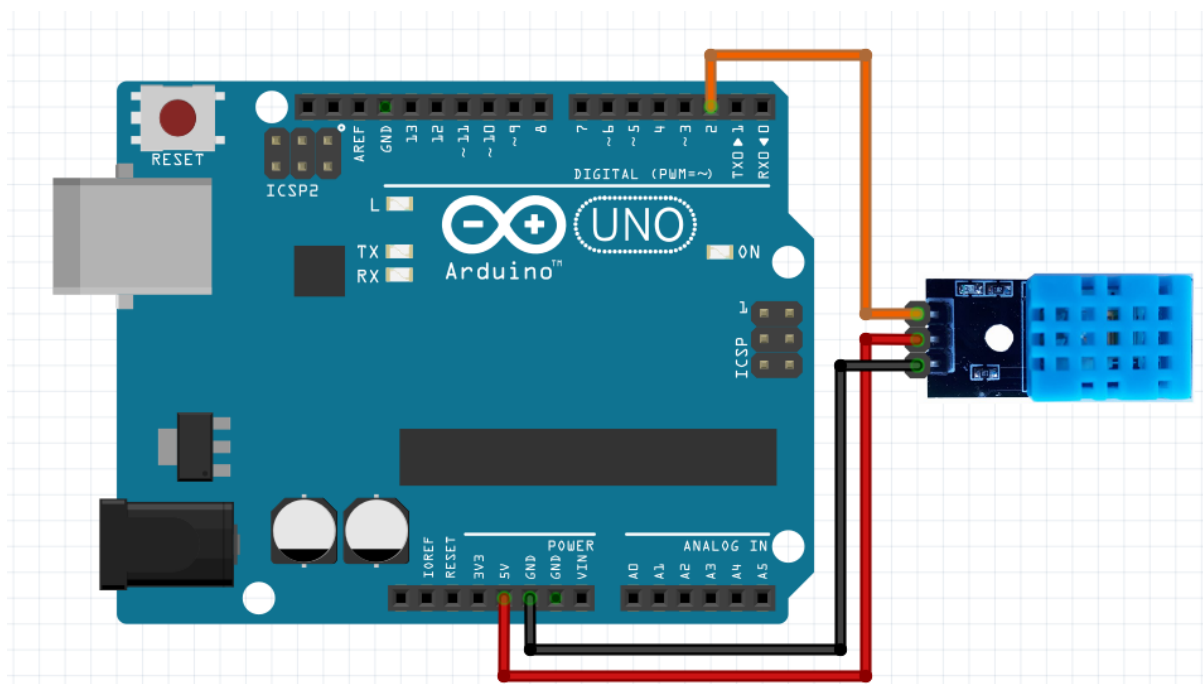
### Основы работы с датчиком DHT11<sup>2</sup>

Для работы потребуется установить дополнительные библиотеки:

- `DHT.h`

DHT11 - цифровой датчик температуры и влажности. Используется при необходимости поддержки определенного диапазона температуры и/или влажности. Подключение происходит по следующей схеме:

<sup>2</sup> Схемы подключения в примерах могут не совпадать с подключением в задачах



*Схема подключения DHT11*

Следующая программа отправляет в монитор порта показания с датчика, а именно влажность (*humidity*) и температуру (*temperature*).

```
#include "DHT.h"
DHT dht(2, DHT11);

void setup() {
  Serial.begin(9600);
  dht.begin();
}

void loop() {
  float h = dht.readHumidity();
  float t = dht.readTemperature();

  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.print("    Temperature: ");
  Serial.println(t);
}
```

### *Программа для работы с DHT11*

Объявление объекта **DHT**:

**DHT dht (2, DHT11);**

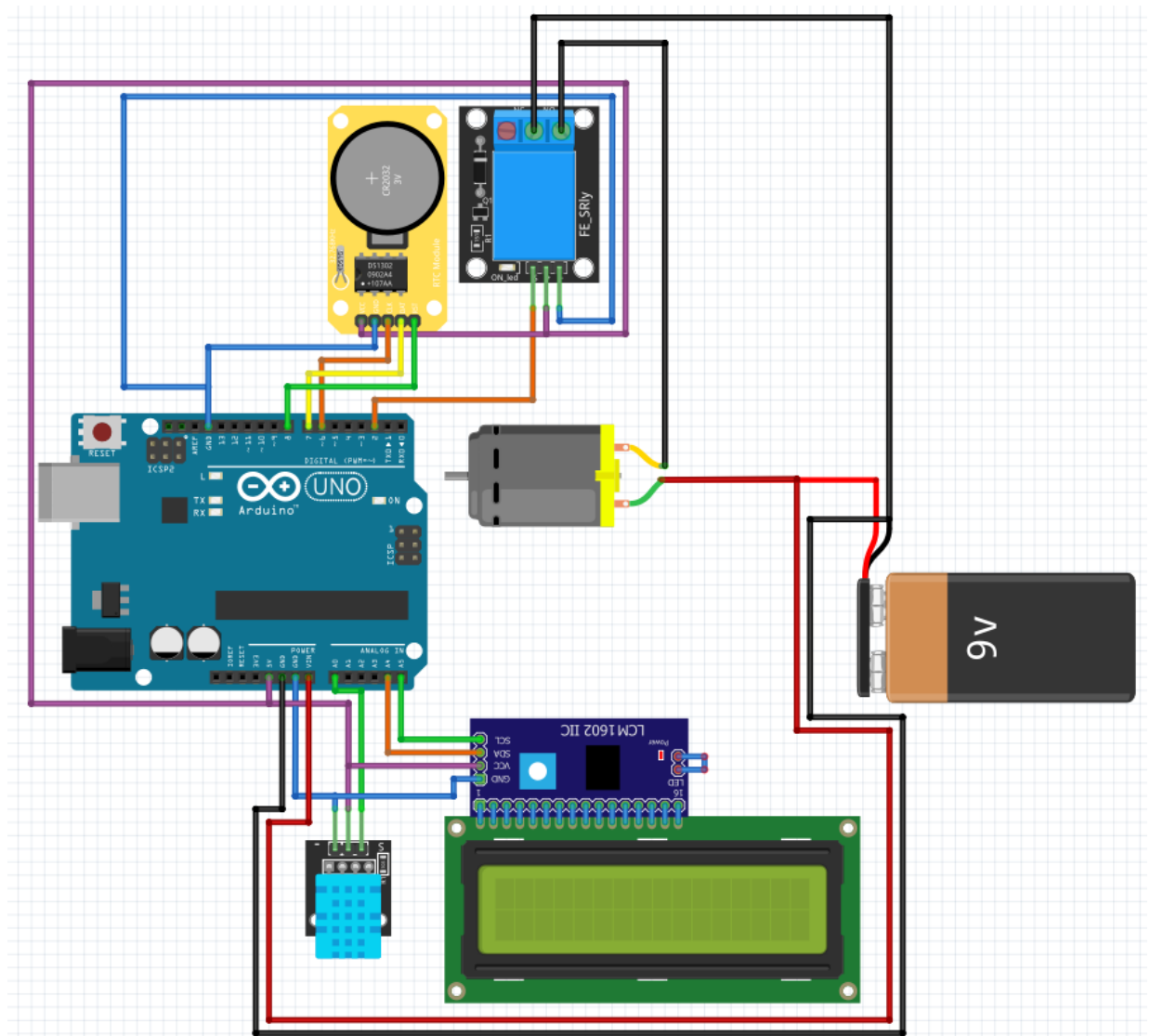
**2** - логический (сигнальный) пин, к которому подключен датчик температуры

**DHT11** - модель датчика. DHT11 или DHT22

Функции:

```
void dht.begin(); /* Запускает инициализацию датчика DHT */
float dht.readHumidity(); /* Считываем показания влажности с датчика*/
float dht.readTemperature(); /* Считываем показания температуры с датчика */
```

### Задача 3: подключение дисплейного модуля LCD1602 для вывода информации с датчика DHT11.



#### Основы работы с LCD1602<sup>3</sup>

Выводить данные с платы можно многими способами. Наиболее простым является монитор порта (функции *Serial*), но при его использовании требуется компьютер. Если требуется создать автономное устройство, для работы которого не будут использоваться компьютер, телефон или планшет, можно использовать другие средства вывода информации. Например,

<sup>3</sup> Схемы подключения в примерах могут не совпадать с подключением в задачах



можно использовать множество светодиодов. Установив их в особом порядке можно получить своеобразный дисплей, где каждый светодиод будет являться пикселем. А можно использовать уже готовый дисплейный модуль. Его сейчас и разберем.

Для работы потребуется установить дополнительные библиотеки:

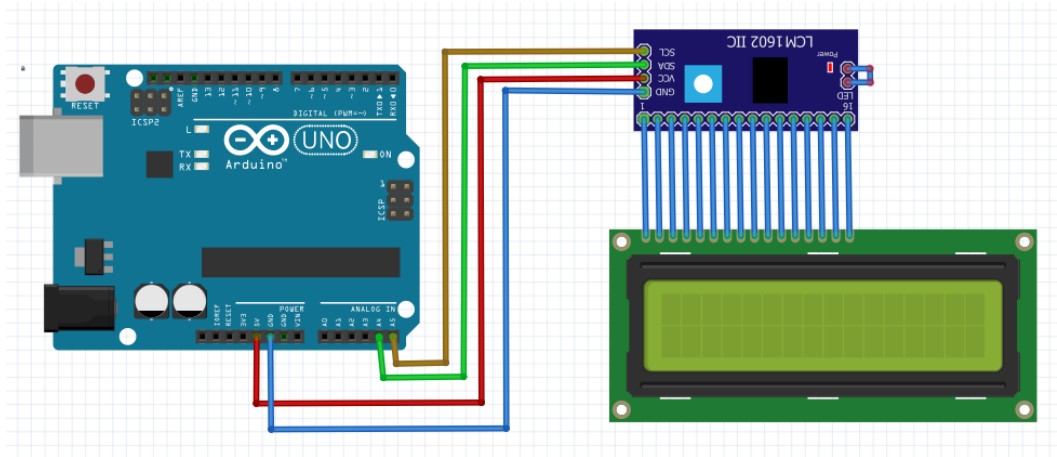
- *Wire.h*
- *LiquidCrystal\_I2C.h*

Дисплейный модуль LCD1602, представленный в наборе, является двухстрочным дисплеем с шестнадцатью ячейками в каждой (отсюда и “1602” в названии). Модуль спаян с платой передачи данных по протоколу I2C, которая упрощает подключение с 16 контактов до 4-х. На обратной стороне I2C модуля есть 3 перемычки, замыкая которые, можно изменять адрес обращения к модулю. Таким образом, на одной плате может быть подключено до восьми дисплейных модулей с разными адресами.

Подключение по протоколу I2C следует производить в соответствии со следующей таблицей и схемой:

*Таблица 1 - Подключение LCD1602*

<b>LCD I2C</b>	<b>Arduino Uno R3</b>
VCC	5V
GND	GND
SDA	A4
SCL	A5



*Схема подключения LCD1602*

*Для вывода текста на дисплей используется следующая программа:*

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
void setup()
{
    lcd.init();
    lcd.backlight();
    lcd.print("LCD 1602");
    lcd.setCursor(8, 1);
    lcd.print("TESTING");
}
void loop()
{
    lcd.setCursor(0, 1);
    lcd.print(millis()/1000);
}
```

*Программа для работы с LCD1602*

*Данная программа будет выводить текст:*

**LCD 1602**

**X TESTING**, где **X** - время работы программы, в секундах

Объявление объекта **LiquidCrystal\_I2C**:

**LiquidCrystal\_I2C lcd (0x27, 16, 2);**

**lcd** - название одного из подключенных дисплейных модулей (или единственного дисплейного модуля)

**0x27** - адрес модуля, к которому будет происходить обращение.  
Стандартный адрес 0x27

**16** - количество столбцов на дисплее

**2** - количество строк на дисплее

Функции:

`void lcd.init();` /\* Запускает инициализацию дисплея.  
Своеобразный pinMode для дисплейного модуля \*/

`void lcd.backlight();` /\* Включение подсветки модуля \*/

`void lcd.print("ТЕХТ");` /\* Функция для вывода строки. Вывод начинается с позиции курсора \*/

`void lcd.print(val);` // Выводить можно не только строки или символы, но также и числа\*/

`void lcd.setCursor(10, 1);` /\* Перемещение курсора в конкретную позицию на экране. Первый аргумент указывает столбец, второй - строку. В начале работы курсор установлен в позиции 0,0. Если было выведено сообщение, курсор автоматически перемещается \*/

`void lcd.clear();` /\* Очистка экрана. Удаление ранее выведенного сообщения \*/

`void lcd.home();` /\* Возвращает курсор в позицию 0,0. Аналогичная запись `lcd.setCursor(0, 0);` \*/

**Задача 4: подключение дисплейного модуля LCD1602 с кнопками выбора настроек автоматического полива.**

