





## Подключение к Arduino

- Модуль оборудован четырех-пиновым разъемом стандарта 2.54мм
- SCL: последовательная линия тактирования (Serial CLock)
- SDA: последовательная линия данных (Serial DAta)
- VCC: "+" питания
- GND: "-" питания

Выводы отвечающие за интерфейс I2C на платах Arduino на базе различных контроллеров разнятся

LCD I2C модуль	На базе ATmega 328	MEGA, ADK, DUE
SCL	A5	D21
SDA	A4	D20
VCC	+5V	+5V
GND	GND	GND

## Код с комментариями вывода температуры окружающей среды на LCD I2C 16x2

/\*

Описание контактов подключения дисплея 1602:

Контакт дисплея	Контакт Arduino	Значение
1	GND	Vss
2	5V	Vdd
3	GND	Vo (контрастность)
4	digital 12	RS
5	digital 11	R/W
6	digital 10	ENABLE
11	digital 5	DB4
12	digital 4	DB5
13	digital 3	DB6
14	digital 2	DB7
15	5V	LED+
16	GND	LED-

\*/

```
// подключаем библиотеку для работы с LCD
```

```
#include <LiquidCrystal.h>
```

```
// инициализируем LCD, указывая контакты данных
```

```
LiquidCrystal lcd(12, 11, 10, 5, 4, 3, 2);
```

```
// задаем переменные
```

```
int Count = 0; // счетчик обновлений
```

```
int tempPin = 0; // пин куда подключен датчик
```

```
float tempc = 0; // переменная для температуры
```

```
float maxi = 0, mini = 30; // max/min значения температуры
```

```

void setup() {

// загрузка программы при старте
  lcd.begin(16, 2);
  lcd.print("Loading");
  delay(500);      // ждем 0.5 секунды
  lcd.print(".");
  delay(500);      // ждем 0.5 секунды
  lcd.print(".");
  delay(1000);     // ждем 1 секунду
  lcd.clear();     //очистка экрана

  // Задаем режим работы портов
}

void loop() {

  lcd.clear();     //очистка экрана

  // считываем и вычисляем температуру
  tempc = analogRead(tempPin)/1023.0*5*1000.0 /10;

  if(tempc > maxi) {maxi = tempc;} // max температура?
  if(tempc < mini) {mini = tempc;} // min температура?

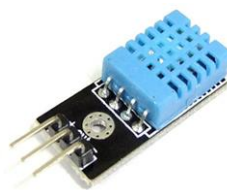
  lcd.setCursor(0, 0); // устанавливаем курсор в 0-ом
                       // столбце, 1 строке (начинается с 0)
  lcd.print("T=");
  lcd.print(tempc);   // выводим текущую температуру
  lcd.print(" C");

  lcd.setCursor(13, 0); // устанавливаем курсор в 12-ом столбце, 1 строке
  lcd.print(Count);   // выводим кол-во измерений

  lcd.setCursor(0, 1); // устанавливаем курсор в 0-ом столбце, 2 строке
  lcd.print(mini);    // выводим min температуру
  lcd.print("<T<");
  lcd.print(maxi);    // выводим max температуру
  delay(5*1000);     // задержка перед повторением измерений
  Count++;           // инкремент счетчика измерений
}

```

**Обратите внимание.** Кроме аналогового датчика температуры так же существуют цифровые датчики для снятия показаний температуры. Датчик DHT11 не обладает высоким быстродействием и точностью, но зато прост, недорог и отлично подходит для обучения.



**Подключение к Arduino**

Модуль оборудован трех пиновым разъемом стандарта 2.54мм

**G** - Подключается к выводу GND

**V** - Подключается к выводу +5V

**S** - Подключается к цифровому выводу ( в примере D4 )

Датчик выполнен из двух частей — емкостного датчика влажности и термистора и простенький аналогово-цифровой преобразователь значений температуры и влажности. Температура и влажность отдаются им по одному сигнальному проводу (S). DHT11 общается с принимающей стороной, такой как Arduino по собственному протоколу. Коммуникация двунаправлена и в общих чертах выглядит так:

- Микроконтроллер говорит о том, что хочет считать показания. Для этого он устанавливает сигнальную линию в 0 на некоторое время, а затем устанавливает её в 1
- Сенсор подтверждает готовность отдать данные. Для этого он аналогично сначала устанавливает сигнальную линию в 0, затем в 1
- После этого сенсор передаёт последовательность 0 и 1, последовательно формирующих 5 байт (40 бит). В первых двух байтах передаётся температура, в третьем-четвёртом — влажность, в пятом — контрольная сумма, чтобы микроконтроллер смог убедиться в отсутствии ошибок считывания

Благодаря тому, что сенсор делает измерения только по запросу, достигается энергоэффективность: пока общения нет, датчик потребляет очень небольшой ток.

#### **Основные технические характеристики:**

- Напряжение питания: 3 - 5В
- Определяемая влажность: 20 - 80% ± 5%
- Определяемая температура: 0 - 50° ± 2%
- Частота опроса: ≤ 1Гц
- Размеры: 30 x 14 x 6мм

**Задание. Управление подогревом одежды через данные о t и распознавание текста.** Подключите устройства в соответствии с текстовым описанием и распиновкой на основе приведенного ниже скетча. Ниже представлена текстовая схема подключения.

/\* LCD 16x2 с I2C переходником

- GND(-) - breadboard
- VCC(+) - breadboard
- SDA - A4
- SCL - A5

DHT-11

- S - A0
- VCC(+) - breadboard
- GND(-) - breadboard

ARDUINO UNO R3

- GND - breadboard
- 5V - breadboard
- D2 - clean (указательный)
- D4 - temper. (средний)
- D7 - M (безым)
- D8 - A (мизинец)
- GND - (-) (большой) \*/

```

#include <Wire.h>    // Добавляем необходимые библиотеки
#include <LiquidCrystal_I2C.h>
int pales1=7;
int pales2=8;
int plus=4;
int clean=2;
#include "DHT.h"
#define DHTPIN A0   // пин для сигнала поступающего с датчика

//выбор используемого датчика
#define DHTTYPE DHT11 // DHT 11
//#define DHTTYPE DHT22 // DHT 22 (AM2302)
//#define DHTTYPE DHT21 // DHT 21 (AM2301)
//инициализация датчика
DHT dht(DHTPIN, DHTTYPE);

byte degree[8] = // кодируем символ градуса
{
  B00111,
  B00101,
  B00111,
  B00000,
  B00000,
  B00000,
  B00000,
};

LiquidCrystal_I2C lcd(0x27,16,2); // Задаем адрес и размерность дисплея

void setup() {
  lcd.init();           // Инициализация lcd
  lcd.backlight();     // Включаем подсветку
  //lcd.createChar(1, degree); // Создаем символ под номером 1

  Serial.begin(9600);
  dht.begin();

  pinMode(13,OUTPUT);
  pinMode(pales1, OUTPUT);
  pinMode(pales2, OUTPUT);
  pinMode(plus, OUTPUT);
  pinMode(clean, OUTPUT);
  digitalWrite(clean, HIGH);
  digitalWrite(plus, HIGH);
  digitalWrite(pales1, HIGH);

```

```

digitalWrite(pales2, HIGH);

}
void loop() {

// код для перчатки
if(digitalRead(pales1) == 0){
  lcd.print("");
  lcd.print("M");      // Выводим текст
  delay(500);
}

if(digitalRead(pales2) == 0){
  lcd.print("");
  lcd.print("A");      // Выводим текст
  delay(500);
}

if(digitalRead(plus) == 0){
  lcd.clear();      // чистим экран
} // конец кода для перчатки

  if(digitalRead(clean) == 0)
  {
  // Wait a few seconds between measurement
  delay(2000);
  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
  float h = dht.readHumidity();
  // Read temperature as Celsius
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit
  float f = dht.readTemperature(true);

  // Выводим показания влажности и температуры
  lcd.setCursor(0, 0);      // Устанавливаем курсор в начало 1 строки
  lcd.print("Hum =  % "); // Выводим текст
  lcd.setCursor(7, 0);      // Устанавливаем курсор на 7 символ
  lcd.print(h, 1);          // Выводим на экран значение влажности
  lcd.setCursor(0, 1);      // Устанавливаем курсор в начало 2 строки
  lcd.print("Temp =  \1C "); // Выводим текст, \1 - значок градуса
  lcd.setCursor(7, 1);      // Устанавливаем курсор на 7 символ
  lcd.print(t,1);          // Выводим значение температуры
  if (t>30) {
    digitalWrite(13, HIGH); // Зажигаем светодиод
  }
  else
  {
  digitalWrite(13, LOW); // иначе тушим
  }
  }
}
}

```