*// Подключаем библиотеку для работы с жидкокристаллическим // экраном (англ. Liquid Crystal Display или просто LCD)*

*#include <LiquidCrystal.h> // на диоде, защищающем от неверной полярности, падает доля // напряжения (англ. voltage drop). Необходимо это учитывать*

*#define DIODE\_DROP 0.7 // Объявляем объект, для управления дисплеем. Для его создания // необходимо указать номера пинов, к которым он подключен в // порядке: RS E DB5 DB6 DB7 DB8*

LiquidCrystal lcd(13, 12, 11, 10, 9, 8);

void setup()

{

*// начинаем работу с экраном. Сообщаем объекту количество // строк и столбцов. Опять же, вызывать pinMode не требуется: // функция begin сделает всё за нас*

lcd.begin(16, 2);

*// печатаем сообщение на первой строке*

lcd.print("Battery voltage:");

}

void loop()

{

*// высчитываем напряжение подключенной батарейки*

float voltage = analogRead(A0) / 1024.0 \* 10.0;

*// если напряжение на делителе напряжения было зафиксировано, // нужно прибавить напряжение на диоде, т.к. оно было съедено*

if (voltage > 0.1) voltage += DIODE\_DROP;

*// устанавливаем курсор, колонку 0, строку 1. На деле — это // левый квадрат 2-й строки, т.к. нумерация начинается с нуля*

lcd.setCursor(0, 1);

*// печатаем напряжение в батарейке с точностью до сотых долей*

lcd.print(voltage, 2);

*// следом печатаем единицы измерения*

lcd.print(" Volts");

}

