Arduino Thermometer DS18B20 + LCD

Hello, everyone ! Today I'm going to show you how to make LCD or Serial* thermometer with DS18B20 digitaltemperature sensor using Arduino, breadboard, jumpers. So you can measure temperature of the air, liquids like water and the temperature of the ground.

*prints the temperature data on the serial monitor of the arduino IDE.

Information About the Sensor


DS18B20 is 1-Wire digital temperature sensor from Maxim IC. Reports degrees in Celsius with 9 to 12-bit precision, from -55 to 125 (+/-0.5). Each sensor has a unique 64-Bit Serial number etched into it - allows for a huge number of sensors to be used on one data bus.

Features:

- Unique 1-Wire® interface requires only one port pin for communication
- Each device has a unique 64-bit serial code stored in an onboard ROM
- Multidrop capability simplifies distributed temperature sensing applications
- Requires no external components
- Can be powered from data line.
- Power supply range is 3.0V to 5.5V
- Measures temperatures from –55°C to +125°C (–67°F to +257°F)±0.5°C accuracy from –10°C to +85°C
- Thermometer resolution is user-selectable from 9 to 12 bits
- Converts temperature to 12-bit digital word in 750ms (max.)
- User-definable nonvolatile (NV) alarm settings
- Alarm search command identifies and addresses devices whose temperature is outside of programmed limits (temperature alarm condition)
- Applications include thermostatic controls, industrial systems, consumer products, thermometers, or any thermally sensitive system

Gathering the Parts

To make the thermometer you will need the following things:

1. Arduino board (UNO,DUE,Micro, etc..).
2. DS18B20 sensor a waterproof or not and one 4.7k resistor*
3. 16x2 LCD display with I2C bus.
4. Breadboard and some jumpers to connect everything together.

*Some stores sell the sensor with 4.7k resistor

DS18B20 Waterproof


The DS18B20 temperature sensor processed in a stainless steel housing and extended with a PVC cable of about 100cm. Pinout of the wires: Red (Vcc/V+/plus), Yellow (data), Black (Gnd/min).

The PVC cable has a temperature range of up to +70ºC, the sensor itself has a temperature range of of -55ºC to +125ºC.

Libraries

Before you start to make the thermometer, download and unzip the following libraries for arduino at - /Progam Files(x86)/Arduino/Libraries (default)

1. 1-Wire bus: onewire.zip
2. Dallas Temperature, it does all the calculations and other stuff: GitHub - DS18B20 (and similar) temperature ICs
3. Liquid Crystal I2C: Arduino-LiquidCrystal-I2C-library

Serial Thermometer


To print the data on the Serial monitor connect the DS18B20 sensor to the Arduino using the jumpers and the breadboard and don't forget to connect or solder the 4.7k resistor between pin 2 and 3 of the sensor.

Then download, open and upload the .ino file which is named - DS18B20_Serial.

If everything is ok you should see the temperature being measured and showed in the Serial monitor of the arduino IDE like on the screenshot above.

DS18B20_Serial.ino

```
#include <OneWire.h>
#include <DallasTemperature.h>

// Data wire is plugged into pin 2 on the Arduino
#define ONE_WIRE_BUS 2

// Setup a oneWire instance to communicate with any OneWire devices
// (not just Maxim/Dallas temperature ICs)
OneWire oneWire(ONE_WIRE_BUS);

// Pass our oneWire reference to Dallas Temperature.
DallasTemperature sensors(&oneWire);

void setup(void)
{
  // start serial port
  Serial.begin(9600);
  Serial.println("Dallas Temperature IC Control Library Demo");

  // Start up the library
  sensors.begin();
}


void loop(void)
{
  // call sensors.requestTemperatures() to issue a global temperature
  // request to all devices on the bus
  Serial.print(" Requesting temperatures...");
  sensors.requestTemperatures(); // Send the command to get temperatures
  Serial.println("DONE");

  Serial.print("Temperature is: ");
  Serial.print(sensors.getTempCByIndex(0)); // Why "byIndex"?
    // You can have more than one IC on the same bus.
    // 0 refers to the first IC on the wire
    delay(1000);
}
```

LCD Thermometer

If you don't want to measure the temperature through the serial monitor then this step is for you !

Connect the I2C LCD to pins UNO,- A4 (SDA) , A5 (SCL) andthe sensor to digital pin 2. Then download and upload the .ino file which is named - DS18B20_I2C_LCD . If everything is OK you will see the temperature readings on the display.

DS18B20_I2C_LCD.ino

```
#include <OneWire.h>
#include <DallasTemperature.h>//
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);


#define ONE_WIRE_BUS 2
// Setup a oneWire instance to communicate with any OneWire devices (not just Maxim/Dallas
temperature ICs)
OneWire oneWire(ONE_WIRE_BUS);


DallasTemperature sensors(&oneWire);

void setup()
  {
  lcd.init();
  lcd.backlight();
  sensors.begin();
  }

void loop()
{

  sensors.requestTemperatures();
  lcd.setCursor(0, 0);
  lcd.print(sensors.getTempCByIndex(0));
  lcd.print(" *C");
  delay(3000);
  }
```

How to Connect a Serial LCD With an Arduino Nano

Published Oct 5th, 2015

By eliesalame

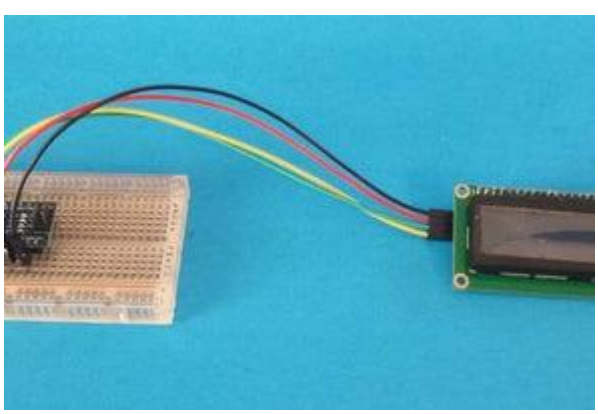
Yesterday I wanted to use two Arduinos for an RF project and realized that I don't know how to use the serial LCD with the Arduino Nano. So with a bit of research I was able to connect it and decided to share it with you.

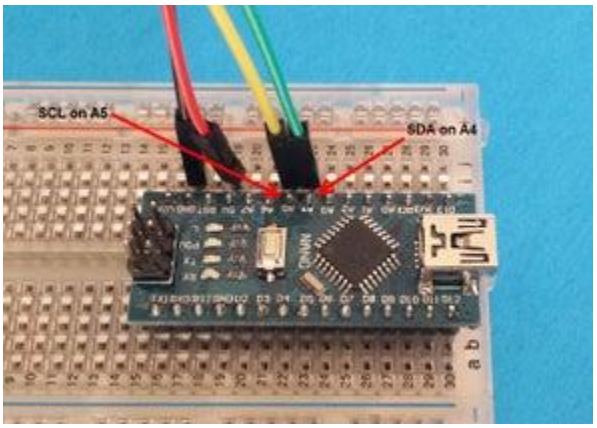
Step 1: Here Is What You Need for This Project

1. An Arduino Nano or a Nano compatible
2. A solderless breadboard
3. An LCD with an I2C
4. Four jumper wires

Step 2: Connect the LCD and the NANO

FXGVCNDIFCDFRR3.LARGE.jpg





FX66ZYDIFCDFRR1.LARGE.jpg

I found a pinout diagram for the Arduino NANO which clearly indicates that the SDA and the SCL pins on the NANO are the A4 and the A5 pins.

- Place the NANO on the solderless breadboard
- Connect the black jumper cable from the GND pin on the LCD to the BND pin on the NANO
- Connect the red jumper cable from the VCC pin on the LCD to the VCC pin on the NANO
- Connect the green jumper cable from the SDA pin on the LCD to the A4 pin on the NANO
- Connect the yellow jumper cable from the SCL pin on the LCD to the A5 pin on the NANO

Next step load the Arduino IDE and upload the sketch

Step 3: The Sketch

FO4RF3DIFCDFRR6.LARGE.jpg

Step one is to download the Liquid Crystal library if you haven't done so already.
I will add a zip file with the library for Windows or you can go the site https://bitbucket.org/fmalpartida/new-liquidcrysta... and download it yourself.

Once you have the library, extract the contents in the Arduino library folder on your computer. On my computer the default location was C:\Users\marcel.RLD\Documents\Arduino\libraries.

I attached a copy of the sketch I used in this instructable,

Here is the breakdown:

First you need to load the libraries, we will load wire.h, LCD.h and LiquidCrystal_I2C.h

```
//load libraries
#include wire.h
#include LCD.h
#include LiquidCrystal_I2C.h
```

Then we need to define variables... in this section just copy it as is because it tells the IDE where to find the PCF8574A and how to interact with the LCD to turn on the backlight, the read pin, the write pin and data pins etc...

```
//Define variables

#define I2C_ADDR 0x27 //Define I2C Address where the PCF8574A is
#define BACKLIGHT_PIN 3
#define En_pin 2
#define Rw_pin 1
#define Rs_pin 0
#define D4_pin 4
#define D5_pin 5
#define D6_pin 6
#define D7_pin 7
```

Another line is needed to initialize the LCD, this is done through an array which includes the variables that we defined earlier.

```
//Initialise the LCD
LiquidCrystal_I2C lcd(I2C_ADDR, En_pin,Rw_pin,Rs_pin,D4_pin,D5_pin,D6_pin,D7_pin);
```

In the void set up, we start by telling the IDE that we are dealing with a 16X2 LCD

```
//Define the LCD as 16 column by 2 rows
lcd.begin (16,2);
```

Then I turn on the back light (always good to have a lit LCD), notice it is the same variable from above...

```
lcd.setBacklightPin(BACKLIGHT_PIN,POSITIVE);
lcd.setBacklight(HIGH);
```

Then I tell it to go to the first line at left most position

```
lcd.setCursor(0,0);
```

and print

```
lcd.print("I just made an");
```

then move the cursor to the second line and the left most position

```
lcd.setCursor(0,1);
```

and print:

```
lcd.print("Instructable :)");
```

There is void loop because the program need a loop to compile but it should remain empty.

And that's it.... very simple, if you follow these instructions the LCD will output anything you type in this code.

I will include the sketch with this instructable.

```
//load libraries
#include <Wire.h>
#include <LCD.h>
#include <LiquidCrystal_I2C.h>
```

```
//Define variables

#define I2C_ADDR          0x27        //Define I2C Address where the PCF8574A is
#define BACKLIGHT_PIN      3
#define En_pin             2
#define Rw_pin             1
#define Rs_pin             0
#define D4_pin             4
#define D5_pin             5
#define D6_pin             6
#define D7_pin             7

//Initialise the LCD
LiquidCrystal_I2C      lcd(I2C_ADDR, En_pin,Rw_pin,Rs_pin,D4_pin,D5_pin,D6_pin,D7_pin);

void setup()
 {
    //Define the LCD as 16 column by 2 rows
    lcd.begin (16,2);

    //Switch on the backlight
    lcd.setBacklightPin(BACKLIGHT_PIN,POSITIVE);
    lcd.setBacklight(HIGH);

    //goto first column (column 0) and first line (Line 0)
    lcd.setCursor(5,0);

    //Print at cursor Location
    lcd.print("HAPPY");

    //goto first column (column 0) and second line (line 1)
    lcd.setCursor(3,1);
    lcd.print("Halloween");

 }

void loop(){ }
```