



Arduino data logger with 2 DS18B20 and sample rate control

by **NicolasJ7** on July 12, 2016

Table of Contents

Arduino data logger with 2 DS18B20 and sample rate control	1
Intro: Arduino data logger with 2 DS18B20 and sample rate control	2
Step 1: Materials	2
File Downloads	3
Step 2: Address of your sensors	4
Step 3: Testing the sensors	4
File Downloads	5
Step 4: Sample rate control	5
File Downloads	7
Step 5: Data logger	7
File Downloads	8
Related Instructables	8
Advertisements	9
Comments	9

Intro: Arduino data logger with 2 DS18B20 and sample rate control

Hi. This tutorial is for those who want to get data from their sensors with controlled sample rate or want to know more about the data logger shield or DS18B20 temperature sensor . We'll use the data logger shield (RTC and SD) and 2 temperature sensors DS18B20 (water proof or TO 92).

In a previous tutorial i explain how to use the data logger shield (with the adafruit guide) and how to control the sample rate with the RTC. Maybe you want to visit:

Arduino Data Logger Shield. How to Control Sample Rate With Rtc

<http://www.instructables.com/id/Arduino-Data-Logge...>

I have translated this tutorial to share it with you, so excuse my english.

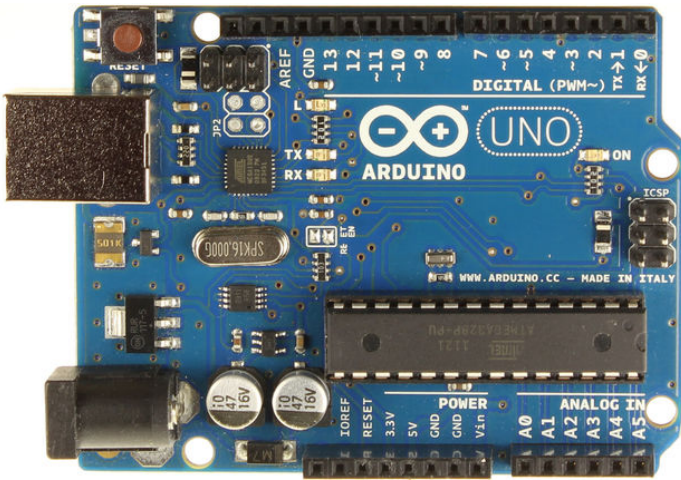
Summary of the steps

- 1.-**Materials.** All that you need for this tutorial.
- 2.- **Address of your sensors.** We will use the utility 1 wire address finder.
- 3.-**Testing the sensors.** We will use a code to show the temperature in the serial monitor and test the sensors.
- 4.- **Sample rate control.** Using the RTC we will control the sample rate and then show the data in the serial monitor.
- 5.- **Data logger.** We'll use the entire shield to log data on the sd card.



Step 1: Materials

- 1.- Arduino UNO
- 2.-Data logger shield
- 3.- 2 Temperature sensors DS18B20
- 4.- One wire address finder utility.
- 5.- Bread board, connectors, etc.
- 6.-SD or micro SD card
- 7.-CR 1220 battery (if not included)
- 8.- Dallas Temperature library
- 9.- Onewire Library



File Downloads



one_wire_address_finder.zip (742 bytes)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'one_wire_address_finder.zip']



OneWire.rar (14 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'OneWire.rar']



Dallas temperature.rar (25 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'Dallas temperature.rar']

Step 2: Address of your sensors

Download the utility 1 wire address finder and connect your sensor. You have to do it one by one. I'm using the normal power mode and the data to pin 3 of the arduino in this example.

My pins

Red= Vcc +5v

Blue= Data

Black= GND

There is a lot of information about this sensor in this page.

Unzip 1 wire address finder and put the file in the folder where you save your sketch. In my case windows7:

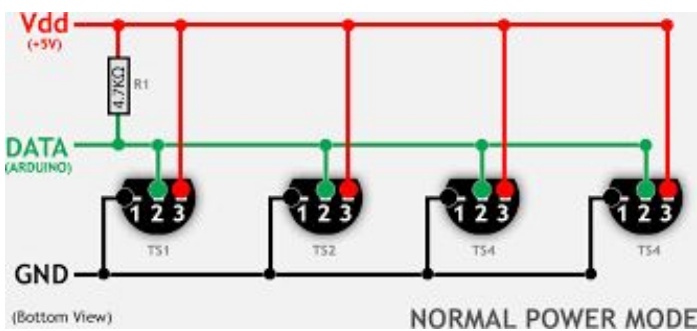
My documents---Arduino

Then open from the arduino IDE and run it

Open----My documents-----Arduino---one wire adress finder.pde

You have to copy your addresses. We will need it in the next step.

Make sure you have instaled the libraries needed (Onewire and dallas temperature)



Step 3: Testing the sensors

Now you can connect the two sensors. This time to pin 2 or you can change it here:

```
#define ONE_WIRE_BUS_PIN 2 // The pin that we are using
```

Every sensor has it own address. You must put your addresses here:

```
DeviceAddress Probe01 = { 0x28, 0xFF, 0x56, 0xC4, 0x01, 0x15, 0x02, 0x72 }; //inox2  
DeviceAddress Probe02 = { 0x28, 0xFF, 0xC2, 0xD1, 0x02, 0x15, 0x03, 0xD9 }; //inox1
```

Change your resolution here:

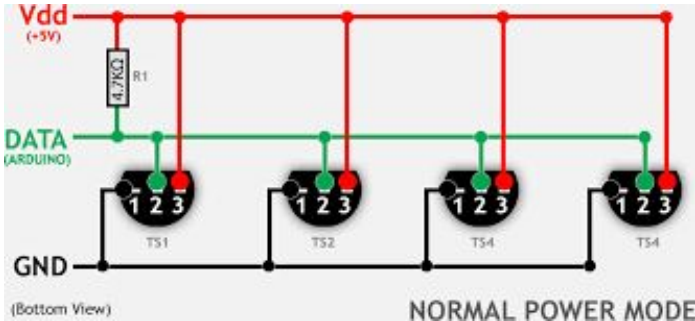
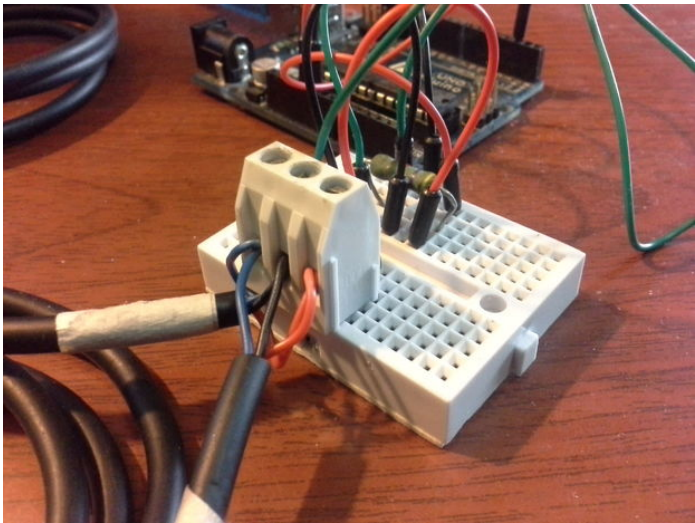
```
// set the resolution to 9 bit (Can be 9 to 12 bits .. lower is faster)  
sensors.setResolution(Probe01, 9);
```

```
sensors.setResolution(Probe02, 9);
```

If everything works fine, you will have the data in the serial monitor (see the pictures)



Here the examples for the serial monitor: every minute and every 10 minutes. You can see the results in the pictures.



COM7 (Arduino/Genuino Uno)		
Date/Time	Temp1	Temp2
2016/7/12 15:30:0	14.50 C	15.00 C
2016/7/12 15:31:0	14.50 C	15.00 C
2016/7/12 15:32:0	14.50 C	15.00 C
2016/7/12 15:33:0	14.50 C	15.00 C
2016/7/12 15:34:0	19.00 C	19.00 C
2016/7/12 15:35:0	18.00 C	21.00 C
2016/7/12 15:36:0	17.00 C	19.50 C
2016/7/12 15:37:0	16.50 C	19.00 C
2016/7/12 15:38:0	16.50 C	17.50 C
2016/7/12 15:39:0	22.50 C	17.00 C
2016/7/12 15:40:0	19.50 C	16.50 C
2016/7/12 15:41:0	17.50 C	16.00 C
2016/7/12 15:42:0	16.50 C	16.00 C
2016/7/12 15:43:0	16.00 C	16.00 C
2016/7/12 15:44:0	16.00 C	16.00 C
2016/7/12 15:45:0	15.50 C	16.00 C
2016/7/12 15:46:0	20.00 C	15.50 C
2016/7/12 15:47:0	19.00 C	15.50 C
2016/7/12 15:48:0	17.50 C	15.50 C

COM7 (Arduino/Genuino Uno)		
Date/Time	Temp1	Temp2
2016/7/12 16:20:0	14.50 C	15.50 C
2016/7/12 16:30:0	14.50 C	15.50 C
2016/7/12 16:40:0	14.00 C	15.50 C
2016/7/12 16:50:0	15.00 C	15.00 C
2016/7/12 17:0:0	14.50 C	15.00 C
2016/7/12 17:10:0	14.00 C	15.00 C
2016/7/12 17:20:0	14.00 C	15.00 C
2016/7/12 17:30:0	14.50 C	15.00 C

File Downloads



[_2_ds18b20_sample_control_serial_monitor_every_10_min.ino](#) (5 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to '_2_ds18b20_sample_control_serial_monitor_every_10_min.ino']



[_2_ds18b20_sample_control_serial_monitor_every_1_min.ino](#) (2 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to '_2_ds18b20_sample_control_serial_monitor_every_1_min.ino']

Step 5: Data logger

We tested the sampling rate, now we use the sd card of the shield to log data. You have to use the same connections from the previous step.

Warnings

-Some shields need pin 10 of the arduino as an output, even if it's working (logging) . I did not.

//pinMode(10 , OUTPUT);//For some data logger shields.Uncomment if you need.

-Remenber to change your CS (chipSelect). Mine is the 8

const int chipSelect = 8; //CS pin of your data logger shield.Maybe not yours!!

-Make sure you have a tested/formated sd card compatible with your datalogger shield.

To make things easier, you can use the function "getTempCByIndex"(Celsius) or getTempFByIndex(fahrenheit), Where (0) is the first sensor and (1) the second , and so on. You don't need the address , but you have to recognize your sensors "by hand".This example is useful if you want only one sensor, so you will have to use the index 0. See the examples.

To continue with the "by Address code" , I show you 3 examples :every 1, 10 and 30 min . I think these examples cover your needs.If you want every hour:

```
if(now.minute()==00 && now.second==00)
```

```
{
```

```
  Print what you want
```

```
}
```

You can use other sensors with this structure, to have a controled sample rate.

If you want to learn more, this work is based on the following books/guides:

-Arduino Cookbook, Michael Margolis

-Exploring Arduino tools and techniques for engineering wizardry, Jeremy Blum

-Adafruit data logger shield guide

<https://cdn-learn.adafruit.com/downloads/pdf/adaf...>

-Progammig Arduino next steps , Simon Monk.

- Beginning Arduino, Michael McRoberts

-Arduino Temperature Control Library, Miles Burton

<https://github.com/milesburton/Arduino-Temperature...>

I hope this tutorial is useful for you!



Date/Time	Temp1	Temp2
2016/7/20 19:50:0	12.50 c	13.00 c
2016/7/20 20:0:0	12.50 c	13.00 c
2016/7/20 20:10:0	12.50 c	13.00 c
2016/7/20 20:20:0	12.50 c	13.00 c
2016/7/20 20:30:0	12.50 c	13.00 c
2016/7/20 20:40:0	12.50 c	13.00 c
2016/7/20 20:50:0	12.50 c	13.00 c
2016/7/20 21:0:0	12.50 c	13.00 c
2016/7/20 21:10:0	12.50 c	13.00 c
2016/7/20 21:20:0	12.50 c	13.00 c
2016/7/20 21:30:0	12.50 c	13.00 c
2016/7/20 21:40:0	12.50 c	13.00 c
2016/7/20 21:50:0	12.50 c	13.00 c
2016/7/20 22:0:0	12.50 c	13.00 c

Date/Time	Temp1	Temp2
2016/7/20 0:00:00	13.50 c	13.00 c
2016/7/20 0:30:00	12.50 c	13.00 c
2016/7/20 1:00:00	12.50 c	13.00 c
2016/7/20 1:30:00	12.50 c	13.00 c
2016/7/20 2:00:00	12.50 c	13.00 c
2016/7/20 2:30:00	12.50 c	13.00 c
2016/7/20 3:00:00	12.50 c	13.00 c
2016/7/20 3:30:00	12.00 c	13.00 c
2016/7/20 4:00:00	12.00 c	12.50 c
2016/7/20 4:30:00	12.00 c	12.50 c
2016/7/20 5:00:00	12.00 c	12.50 c
2016/7/20 5:30:00	12.00 c	12.50 c
2016/7/20 6:00:00	12.00 c	12.50 c
2016/7/20 6:30:00	12.00 c	12.50 c
2016/7/20 7:00:00	12.00 c	12.50 c
2016/7/20 7:30:00	12.00 c	12.50 c
2016/7/20 8:00:00	12.00 c	12.50 c
2016/7/20 8:30:00	12.00 c	12.50 c
2016/7/20 9:00:00	12.00 c	12.50 c
2016/7/20 9:30:00	12.00 c	12.50 c
2016/7/20 10:00:00	12.00 c	12.50 c
2016/7/20 10:30:00	12.00 c	12.50 c
2016/7/20 11:00:00	12.00 c	12.50 c
2016/7/20 11:30:00	12.50 c	12.50 c
2016/7/20 12:00:00	12.50 c	13.00 c
2016/7/20 12:30:00	12.00 c	13.00 c
2016/7/20 13:00:00	12.00 c	13.00 c
2016/7/20 13:30:00	12.00 c	13.00 c
2016/7/20 14:00:00	12.00 c	13.00 c
2016/7/20 14:30:00	12.00 c	13.00 c

Date/Time	Temp1	Temp2
2016/7/19 20:53:00	13.00 c	13.00 c
2016/7/19 20:54:00	13.00 c	13.00 c
2016/7/19 20:55:00	13.00 c	13.00 c
2016/7/19 20:56:00	13.00 c	13.00 c
2016/7/19 20:57:00	13.00 c	13.00 c
2016/7/19 20:58:00	13.00 c	13.00 c
2016/7/19 20:59:00	13.00 c	13.00 c
2016/7/19 21:00:00	13.00 c	13.00 c
2016/7/19 21:01:00	13.00 c	13.00 c
2016/7/19 21:02:00	13.00 c	13.00 c
2016/7/19 21:03:00	13.00 c	13.00 c
2016/7/19 21:04:00	13.00 c	13.00 c
2016/7/19 21:05:00	13.00 c	13.00 c
2016/7/19 21:06:00	13.00 c	13.00 c
2016/7/19 21:07:00	13.00 c	13.00 c
2016/7/19 21:08:00	13.00 c	13.00 c
2016/7/19 21:09:00	13.00 c	13.00 c
2016/7/19 21:10:00	13.00 c	13.00 c
2016/7/19 21:11:00	13.00 c	13.00 c
2016/7/19 21:12:00	13.00 c	13.00 c
2016/7/19 21:13:00	13.00 c	13.00 c
2016/7/19 21:14:00	13.00 c	13.00 c
2016/7/19 21:15:00	13.00 c	13.00 c
2016/7/19 21:16:00	13.00 c	13.00 c
2016/7/19 21:17:00	13.00 c	13.00 c
2016/7/19 21:18:00	13.00 c	13.00 c
2016/7/19 21:19:00	13.00 c	13.00 c
2016/7/19 21:20:00	13.00 c	13.00 c
2016/7/19 21:21:00	13.00 c	13.00 c
2016/7/19 21:22:00	13.00 c	13.00 c

File Downloads



[_2_ds18b20_sample_control_sd_every_minute_by_address.ino](#) (3 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to '_2_ds18b20_sample_control_sd_every_minute_by_address.ino']



[_2_ds18b20_sample_control_sd_every_10_min_by_address.ino](#) (7 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to '_2_ds18b20_sample_control_sd_every_10_min_by_address.ino']



[_2_ds18b20_sample_control_sd_every_30_min_by_address.ino](#) (3 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to '_2_ds18b20_sample_control_sd_every_30_min_by_address.ino']



[_1_ds18b20_sample_control_every_minute_by_index.ino](#) (2 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to '_1_ds18b20_sample_control_every_minute_by_index.ino']



[_2_ds18b20_sample_control_every_10_min_by_index.ino](#) (6 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to '_2_ds18b20_sample_control_every_10_min_by_index.ino']



[_2_ds18b20_sample_control_sd_every_minute_by_index.ino](#) (2 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to '_2_ds18b20_sample_control_sd_every_minute_by_index.ino']

Related Instructables



Arduino data logger shield. How to control sample rate with rtc by NicolasJ7



Build a datalogger for your wireless sensor network by salamandersensor



Arduino – Datalogger with Temperature Sensor and Photoresistor by RuiSantos



Arduino Rocket datalogger by Charlie Godfrey



Memory Card Shield for Arduino UNO by Rahul Mitra



Solar Power Data Logger by Moscherttaart

Comments