|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|

|  |  |  |  |
| --- | --- | --- | --- |
|

|  |  |  |
| --- | --- | --- |
|  | ***Proyecto de monitoreo de temperatura*****Objetivo: construir y adaptar este proyecto para que pueda monitorear, mediciones de temperatura en intervalos de segundos para emplear los datos en la obtención de modelos matemáticos de procesos físicos.**[**Standalone Temperature Logger using ATtiny85**](http://arduino.cc/forum/index.php?topic=55513.msg397842#msg397842)« **on:** March 16, 2011, 05:55:51 AM » |  |

I've made a simple Standalone Temperature Logger using the ATtiny85 with minimal parts which should be able to run for at least half a year. You can extract the data logged from the internal EEPROM by connecting it up to an Arduino. There are 11 logging delay times available (1min, 5mins, 10mins, 15mins, 30mins, 1hr, 2hrs, 4hrs, 8hrs, 12hrs, 24hrs). It can store up to 511 values of temperature.Pictures, guide to use, code, schematics, PCB designs available here: <http://www.insidegadgets.com/projects/standalone-temperature-logger/> http://www.insidegadgets.com/wp-content/uploads/2011/03/IMG_19731-284x300.jpgI kept track of my development progress in 9 parts starting here showing how I went from the concept to the PCB design and the things I learnt along the way: <http://www.insidegadgets.com/2011/01/03/building-a-standalone-temperature-logger-part-1/> |
|

|  |
| --- |
|  |
|  |  |

 |

 |
|  |

***The Standalone Temperature Logger has been merged with the Standalone Voltage Logger to give the*** [***Standalone Temperature/Voltage Logger***](http://www.insidegadgets.com/projects/standalone-temperaturevoltage-logger/) ***available in kit form.***

A standalone temperature logger in a small compact form factor with minimal parts using the ATtiny85 and an I2C external EEPROM powered by a 3V coin cell. Extract the data logged by connecting to your PC using a USB cable.  The standalone temperature logger uses V-USB – <http://www.obdev.at/vusb/>. There are 2 versions, v3.0 & v2.0 which uses USB and v1.0 which connects to an Arduino to transfer the data.

You can view how I designed and built it by following the posts found here – <http://www.insidegadgets.com/tag/satl/page/2/>


(version 2.0 pictured)

**Features**
- Low power consumption, at least 1 year battery life
- Uses an external I2C EEPROM, supports up to 512Kbit which is 65536 recordings
- 12 logging delay times available (28secs, 1min, 5mins, 10mins, 15mins, 30mins, 1hr, 2hrs, 4hrs, 8hrs, 12hrs, 24hrs)
- Easily transfer the logged data to your PC via USB

**Specifications**
PCB Board: 37mm x 28mm
Voltage: 2.7V to 5.5V
Current used when sleeping: 5uA
Current used when logging – Watchdog sleep routine: 0.3mA for 30ms approximately every 4 seconds
Current used when logging – Thermistor on and write to EEPROM: 3mA (max) for 50ms
Accuracy of thermistor: within 5% -/+ (between -40C to 25C is 1.25C -/+, 25C to 50C is 1.5C -/+, 50C – 70C+ is 1.9C -/+)
Accuracy of timer: within 10% -/+
Resolution of temperature: 0.65C (rounded up or down in 0.5 – 1C increments)
Operating temperature: -40C to 70C

 **Download**

v3.0b (16 May 2012) – [Download](http://www.insidegadgets.com/wp-content/uploads/2011/03/Standalone_Temperature_Logger_v3.0b.zip)
- Reverted calibrateOscillator function back to the default as my patch wasn’t needed
- After a USB transfer is complete, made OSCCAL go back to the factory default settings in steps of 32

v3.0a (2 May 2012) – [Download](http://www.insidegadgets.com/wp-content/uploads/2011/03/Standalone_Temperature_Logger_v3.0a.zip)
- Corrected EEPROM size test code to work with 32Kbit EEPROMs

v3.0 (4 March 2012) – [Download](http://www.insidegadgets.com/wp-content/uploads/2012/03/Standalone_Temperature_Logger_v3.0.zip)
- Added external I2C EEPROM support with automatic EEPROM size checking
- Updated 10K resistor divider to 1% tolerance
- Added 28 seconds to delay options available
- Updated temperature resolution to be 0.65C which is rounded up or down in 0.5 – 1C increments thanks to the suggestion made by tytower

v2.2 (23 December 2011) – [Download](http://www.insidegadgets.com/wp-content/uploads/2011/03/Standalone_Temperature_Logger_v2.2.zip)
- Updated Thermistor function to use the full Steinhart–Hart equation and the Vishay 10K coefficients

v2.1 (31 October 2011) – [Download](http://www.insidegadgets.com/wp-content/uploads/2011/03/Standalone_Temperature_Logger_v2.1.zip)
- Updated R7 resistor for button from 10k to 100k to save power (0.27mA saved)
- Removed need for Timer0, millis and delay\_at\_1mhz. Instead uses the watchdog timer to save power (1mA saved)

v2.0a (28 October 2011) – [Download](http://www.insidegadgets.com/wp-content/uploads/2011/03/Standalone_Temperature_Logger_v2.0a.zip)
- Highly recommend disabling BOD as is a chance that the BOD will kick in at ~2.98V depending
on what the MCU is doing.

v2.0 (10 July 2011) – [Download](http://www.insidegadgets.com/wp-content/uploads/2011/03/Standalone_Temperature_Logger_v2.0.zip)
- Now uses V-USB to transfer data to the PC directly without needing an Arduino)

 **Sample Data**

Y axis is the temperature and X axis is each interval.

Outside (under shelter) – 15 minute intervals


Outside (in daylight) – 5 minute intervals


Car glove box – 5 minute intervals


Fridge – 5 minute intervals


 **Programming the ATtiny85**

To program the ATtiny85 and fuse bits we use [AVRDUDE](http://savannah.nongnu.org/projects/avrdude/)

1. Change the fuse bits so the ATtiny85 uses 16MHz PLL and divide clock by 8
avrdude -p ATtiny85 -c usbtiny -U lfuse:w:0×61:m -U hfuse:w:0xdf:m -U efuse:w:0xff:m2.7V Brown-out Detection is optional (-U hfuse:w:0xdd:m) and is only recommended for CR2032 batteries that are new or near new condition. When batteries drop to ~2.95V there may be issues when powering up the ATtiny85.
2. Upload the main.hex file
avrdude -p ATtiny85 -c usbtiny -U flash:w:main.hex

 **How to use

Inserting the EEPROM**

1. You should only remove or insert the EEPROM when the battery has been removed.

**First time use**

1. Insert the EEPROM (you should only remove or insert the EEPROM when the battery has been removed).
2. Insert the battery.
3. The LED will either blink which indiciates there is a EEPROM present or it will stay on for 2 seconds which means no EEPROM was found.The number on blinks corresponds to the EEPROM size it found as below:
——————————————————————
| Blinks | EEPROM size | Max recordings |
|   1      |    1Kbit          |        128            |
|   2      |    2Kbit          |        256            |
|   3      |    4Kbit          |        512            |
|   4      |    8Kbit          |       1024           |
|   5      |   16Kbit         |       2048           |
|   6      |   32Kbit         |       4096           |
|   7      |   64Kbit         |       8192           |
|   8      |   128Kbit       |      16384          |
|   9      |   256Kbit       |      32768          |
|   10    |   512Kbit       |      65536          |
——————————————————————
4. Ready to use

**Changing logging time intervals**

1. Hold the button down for 2 seconds, the LED will stay on for 2 seconds to confirm you are changing the logging time.
2. Configure the logging time interval by pressing the button the amount of times shown below.
Please wait for the LED to light up to confirm each button press.
1 button press = 28 seconds
2 button presses = 1 minute
3 button presses = 5 minutes
4 button presses = 10 minutes
5 button presses = 15 minutes
6 button presses = 30 minutes
7 button presses = 1 hour
8 button presses = 2 hours
9 button presses = 4 hours
10 button presses = 8 hours
11 button presses = 12 hours
12 button presses = 24 hours
3. Hold the button down for 1 second to confirm your logging time interval, the LED will blink three times to confirm.
(If the LED stays on for 2 seconds too many button presses were made, please repeat step 2)

**Start temperature logging**

1. Press the button 3 times within 2 seconds to activate the temperature logging. The LED will blink three times to confirm. The LED will blink very quickly every time it logs temperature.If the LED stays on for 2 seconds this means that no external EEPROM was found.

**Exit temperature logging**

1. Press the button once and the LED will blink 2 times.

**Transfer the data to the PC**

1. Plug the USB cable into the Standalone Temperature Logger
2. Open Notepad
3. Press the button once. The LED will stay on until the data transfer is complete.
4. Unplug the USB cableIf you remove the USB cable mid-way through the data transfer and find the LED is still lit, please remove and re-insert the battery.

**Test the USB communication**

1. Program the ATtiny85
2. Follow “Transfer the data to the PC” steps
3. The number “124.5″ should be printed as many times as the “Max recordings” shown in the table allows for your EEPROM. If no EEPROM is inserted, it will print “124.5″ until disconnected.

 **Build your own**

**Parts Required**

- Standalone Temperature Logger v3.0 PCB board
- ATtiny85 20MHz DIP8 (1) – ATTINY85-20PU – [U1]
- N Channel Mosfet 2.3V[Threshold Vgs Max] SOT-23 (1) – 2N7002K – [Q1]
- P Channel Mosfet -1.7V[Threshold Vgs Typ] SOT23 (1) – BSS84 – [Q2]
- 10K Resistor 1/8W (5) – [R1, R2, R3, R9, R10]
- 10K Resistor 1/8W 1% (1) – [R3]
- 150K Resistor 1/8W (1) – [R7]
- 1.5K Resistor 1/8W (1) – [R8]
- 68R Resistor 1/8W (2) – [R5, R6]
- 10K Thermistor (1) – NTCLE100E3103JB0 – [R4]
- Zener Diode 3.6V 1/2W DO-35 (2) – 1N5227B – [D1, D2]
- Schottky Diode 400mV[VF Max] DO-35 (1) – BAT42 – [D3]
- 3mm Red LED (1) – MCL514SRD – [L1]
- 4.3mm Tactile switch (1) – MJTP1230 – [S1]
- 8 pin IC socket (1)
- 3V CR2032 coin cell (1)
- Coin cell battery holder (1)
- USB Type B Receptacle (1) – USB-B-S-RA – [USB]

**Schematic**



**PCB Guide**



**PCB Ready for etching**



**Version 1.0 (Needs an Arduino to extract the data from)**

A standalone temperature logger in a small compact form factor with minimal parts using the ATtiny85V and powered by a 3V coin cell. Extract the data logged by connecting up to an Arduino.


(v1.0 pictured)

**Features**
- Low power consumption, at least half a year battery life
- Store up to 511 values of temperature
- 11 logging delay times available (1min, 5mins, 10mins, 15mins, 30mins, 1hr, 2hrs, 4hrs, 8hrs, 12hrs, 24hrs)
- Transfer the logged data to the Arduino

**Specifications**
Voltage: 1.8 to 5.5 Volts
Current used when sleeping/waiting for input: 0.02mA
Accuracy of logging timer: within 10% -/+
Operating temperature: -40C to 70C

**Requirements**
Arduino Duemilanove or Uno

 **Download**

v1.1 – Updated PCB to v1.1 – [Download](http://www.insidegadgets.com/wp-content/uploads/2011/03/Standalone_Temperature_Logger_v1.1.zip)
v1.0 – Initial Release – [Download](http://www.insidegadgets.com/wp-content/uploads/2011/03/Standalone_Temperature_Logger_v1.0.zip)

 **How to use**

**Changing logging time**

1. Hold the button down for 2 seconds, the LED will stay one for 2 seconds to confirm you are changing the logging time.
2. Configure the delay time by pressing the button.
1 button press = 1min
2 button presses = 5mins
3 button presses = 10mins
4 button presses = 15mins
5 button presses = 30mins
6 button presses = 1hr
7 button presses = 2hrs
8 button presses = 4hrs
9 button presses = 8hrs
10 button presses = 12hrs
11 button presses = 24hrs
3. Hold the button down to confirm your logging time, the LED will blink three times.

**Start temperature logging**

1. Press the button 3 times in 1.5 seconds to activate the temperature logging. The LED will blink three times to confirm.

**Exit temperature logging**

1. Press the button once and the LED will blink 2 times.

**Transfer the data to the Arduino**

****

1. Upload the “Standalone\_Temperature\_Logger\_Arduino\_Reader\_Side.pde” to the Arduino
2. Connect the 3 wires (DAT, CLK and GND) and then press the button once. The LED will blink twice if it’s still logging the temperature.
3. Start the Arduino software and open the serial monitor, the text “Ready” should appear.
4. Both the LED and Arduino LED’s will stay lit when the transfer is in progress and the temperature will be printed in the serial monitor.
5. The end of the temperature values is shown by the text “End” appearing.

 **How to upload the program to the ATtiny85**

1. Copy the “attiny45\_85″ folder to “C:\Program Files\arduino-0021\hardware”.
2. Open the Arduino software and open the sketch named “ArduinoISP”.
3. Upload this to the Arduino board.
4. Connect the Arduino to the ATtiny85 as shown.
5. In the Arduino software, select Tools -> Board -> ATtiny85.
6. Open the “Standalone\_Temperature\_Logger\_ATtiny85\_Side.pde” sketch and upload it. The LED of the Arduino will blink whilst uploading.
7. Once uploading is done, unplug the Arduino, remove the connections to the ATtiny85 and insert the ATtiny85 chip to the Standalone Temperature Logger board.

 **Build your own**

**PCB Revisions**
v1.1 – Re-arranged board so that soldering the battery isn’t difficult any more
v1.0 – Initial Release

**Parts required**
U1 – ATtiny85V (1)
R1, R2, R3, R4, R5, R6 – 10K Resistor 1/8W (6)
R7 – 10K Thermistor (1)
R8 – 39K Resistor 1/8W (1)
S1 – Tactile switch (1)
LED1 – 3mm Red LED (1)
8 pin IC socket
3 pin female header
3V CR2032 coin cell
Coin cell battery holder

**Schematic
**

**PCB Guide**v1.1 ****

v1.0 [****](http://www.insidegadgets.com/wp-content/uploads/2011/03/satl-pcb-guide-v1.0.png)

**PCB Ready for printing**Version 1.1 (untested)
30mm x 28.4mm


Version 1.0
30mm x 28.4mm


## [Standalone Temperature/Voltage Logger](http://www.insidegadgets.com/projects/standalone-temperaturevoltage-logger/)

Apr 18th, 2012 by [Alex](http://www.insidegadgets.com/author/admin/)

A standalone temperature/voltage logger in a small compact form factor with minimal
parts using the ATtiny85 and an external I2C EEPROM powered by a 3V coin cell.
Data logged can be extracted via a USB cable.

The standalone temperature/voltage logger uses V-USB: <http://www.obdev.at/vusb/>



**Features**
- Low power consumption, at least 1 year battery life
- Measure temperature from -40C to 125C saving to EEPROM as 8 bit (as 1 byte) or 10 bit (as 2 bytes).
- Measure voltage from 0V to 15V saving to EEPROM as 8 bit (as 1 byte) or 9 bit (as 2 bytes). (Voltage range can be increased if the 2 voltage protection diodes are removed and the code is modified)
- Uses an external I2C EEPROM, supports up to 512Kbit which is 65,536 recordings (with 1 byte values) or 32,768 (with 2 byte values)
- 12 logging delay times available (28secs, 1min, 5mins, 10mins, 15mins, 30mins, 1hr, 2hrs, 4hrs, 8hrs, 12hrs, 24hrs)
- Easily transfer the logged data to your PC via USB, it will print out the data logged one result at a time so you will need to have notepad (or any other program) open until all the data is transferred
Note: You can only log temperature or voltage, not both at the same time. No information on time is recorded when logging.

**Specifications**
PCB Board: 41mm x 28.5mm
Voltage: 2.7V to 5.5V
Current used when sleeping: 5uA
Current used when logging – Watchdog sleep routine: 0.3mA for 30ms approximately every 4 seconds
Current used when logging – Thermistor on, LED on and write to EEPROM: 3mA (max) for 50ms
Resolution of temperature (1 byte): 0.65C (rounded up or down in 0.5 – 1C increments)
Resolution of temperature (2 bytes): 0.16C (rounded up or down in 0.2C increments)
Resolution of voltage (1 byte): 0.06V
Resolution of voltage (2 bytes): 0.03V
Accuracy of Thermistor: within 5% -/+ (between -40C to 25C is 1.25C -/+, 25C to 50C is 1.5C -/+, 50C – 70C+ is 1.9C -/+)
Accuracy of timer: within 10% -/+
Operating temperature: -40C to 125C (Most CR2032 batteries only support -40C to 70C)

##### ****Download****

The download file includes:
- Source code and compiled main.hex file
- Parts list, step by step guide and instructions on use ([View it online here](http://www.insidegadgets.com/project/SATVL_v1.x_Guide_Rev6.html))
- PCB bottom ready for home etching
- PCB top ready for home etching (rotated horizontally)
- PCB Guide
- Schematic
- Example results
- Pictures of finished product

[v1.2 (4 October 2012)](http://www.insidegadgets.com/wp-content/uploads/2012/04/Standalone_Temperature_Voltage_Logger_v1.2.zip)
- Fixed bug where calibrateOscillator function wasn’t been called when USB was plugged in
- Enabled ADC noise cancel sleep mode

[v1.1 (12 August 2012)](http://www.insidegadgets.com/wp-content/uploads/2012/04/Standalone_Temperature_Voltage_Logger_v1.1.zip)
- The data transfer now uses the satvltransfer program to transfer the data from the SATVL more quickly

[v1.0.1 (9 June 2012)](http://www.insidegadgets.com/wp-content/uploads/2012/04/Standalone_Temperature_Voltage_Logger_v1.0.1.zip)
- Now prints the logging delay time (e.g “28sec delay”) at the start of the data transfer to the PC

[v1.0b (16 May 2012)](http://www.insidegadgets.com/wp-content/uploads/2012/04/Standalone_Temperature_Voltage_Logger_v1.0b.zip)
- Reverted calibrateOscillator function back to the default as my patch wasn’t needed
- After a USB transfer is complete, made OSCCAL go back to the factory default settings in steps of 32

[v1.0a (2 May 2012)](http://www.insidegadgets.com/wp-content/uploads/2012/04/Standalone_Temperature_Voltage_Logger_v1.0a.zip)
- Corrected EEPROM size test code to work with 32Kbit EEPROMs

[v1.0 (18 April 2012)](http://www.insidegadgets.com/wp-content/uploads/2012/04/Standalone_Temperature_Voltage_Logger_v1.0.zip)
- Initial Release

##### ****Results****

The graphs below show the results, you can also download the Excel files to see the raw data.

**Temperature logging (1 byte)**[Download](http://www.insidegadgets.com/wp-content/uploads/2012/04/Temperature_Results_1byte.xls)



**Temperature logging (2 bytes)**[Download](http://www.insidegadgets.com/wp-content/uploads/2012/04/Temperature_Results_2bytes.xls)



**Voltage logging (2 bytes)**[Download](http://www.insidegadgets.com/wp-content/uploads/2012/04/Voltage_Results_2bytes.xls)



#####

**Standalone Temperature-Voltage Logger with Voltage Logging Add-on**This is the recommended version of this kit to purchase as it allows for temperature or voltage logging and allows you to purchase an EEPROM if you don’t already have one.

Principio del formulario



Final del formulario