

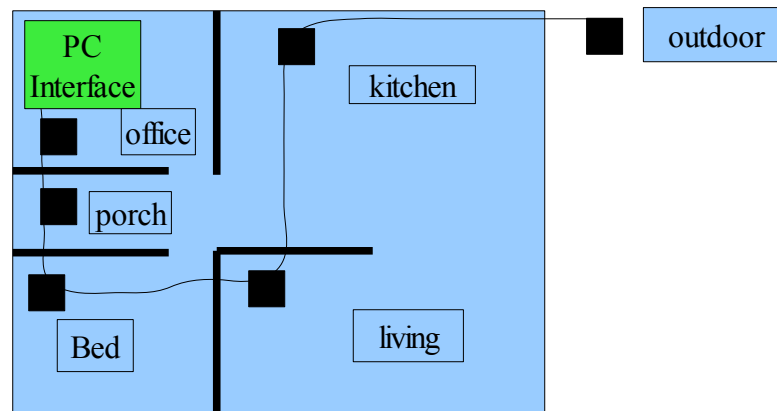
First, I would like to apologize for any ill formed or bad english that could appear in this document. English is not my primary language, and even if I used a spell checker, there are things, as you might know, that spell checkers can not catch.

## 1-Wire introduction:

The 1-wire network gets its name from the fact that only 1 wire is required to transmit information. It might still need up to 2 extra wires (for a total of 3) for power and ground, but it keeps the number of wires quite low, which can be very useful when the 1-wire components are used on PCB such as cell phones and small electronics devices.

1-wire devices can be more than temperature sensing. There are (low power) switches, counters, RAM, clock, thermocouple interface, and other devices adapted for special purposes, such as a battery meter turned solar radiation meter. However, my first shot at writing this document is directly related to temperature, and being such, I will try to stay close to my subject and the way it did it, with extra information as annexes when the first writing will be done.

An very interesting feature of the 1-wire network is that most of it's devices can be multi-dropped, or daisy chained if you prefer. That implies that you could run a single cable from the PC interface passing through all the rooms in your home and finishing the run outside to get outdoor temperature.



Those devices, the temperature probes, are very cheap, something like 2.00\$ each. My cabling of choice is CAT3, 2 unshielded twisted pairs telephone cable, which is also quite cheap, in the order of 0.12\$ per feet. The rumor has it that you can pull about 300 feet of cabling per segment. Some extra devices allows you to have multiple segments deployed. You need a BIG house to go over the limits.

As far as the sensors themselves are concerned, the precision is not that much (all being relative!), and varies from probe to probe, and you will often see 0.5 to 1 Celsius of variation between sensors located

within 2 inches one from the other. Note that if you buy a 10\$ « made-in-china » digital thermometer, you might, you will, get variation equals or greater than that. So, given the price, I will definitely not complain. You could get more precise probes, but they cost around 20\$ each. Those are laboratory rated and hand picked for their high precision. I don't think that the regular Joe needs such devices, but they are available nonetheless.

More details about the whole subjects is available from the Dallas / Maxim (Dallas was bought a few years ago) web site at <http://www.maxim-ic.com/products/1-wire/>

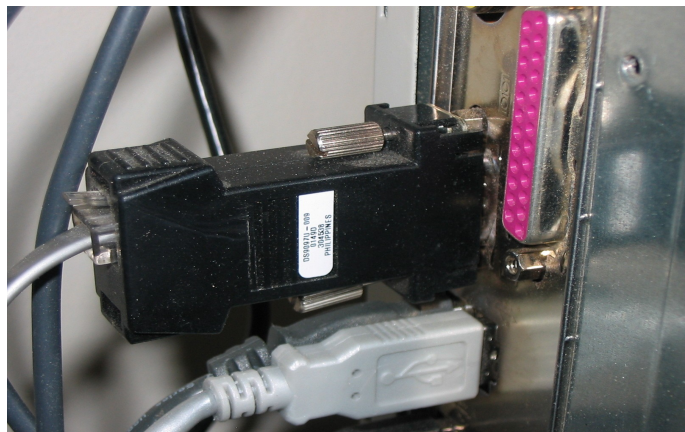
## All this being said...

So, we'll start for real.

First, obtain the interface for the PC. I used the 9097U-009 (DB9 plug), which is of RS-232, passive sort.

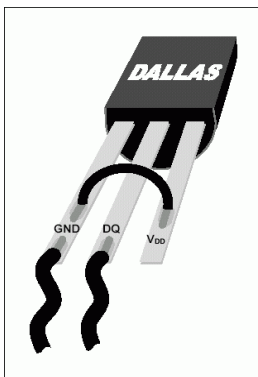
There are others, such as USB, the DS9097-E25 (DB25 plug) and other « active » interface, but I do not own them, and can not say much about them either.

You can purchase the thing for 28\$ from the Maxim web site.

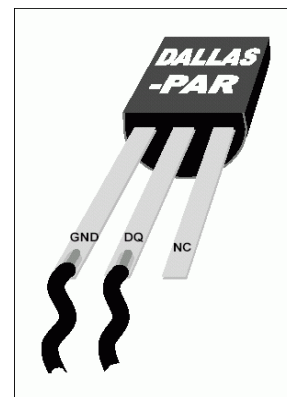


Then, you will need the sensors.

I used some DS1820 and DS18B20-PAR. The big difference between them is that the « PAR » version, which came later, only need 2 wires soldered / connected, while the DS1820 needed some kind of tweaking before really being of use.



The Non-PAR device needed to manually joined the 2 external pins of the sensor.



The PAR device do not need the 2 external pins soldered, it is done inside the thing

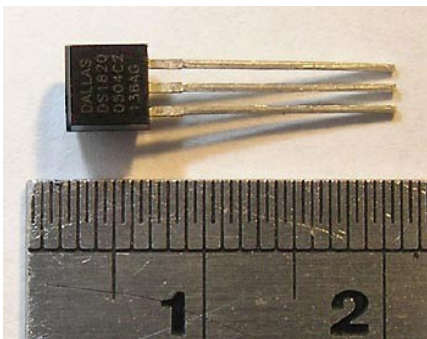
NOTE: there are 3 kind of devices, which are :

- DS18B20-PAR
- DS18S20-PAR
- DS1822-PAR

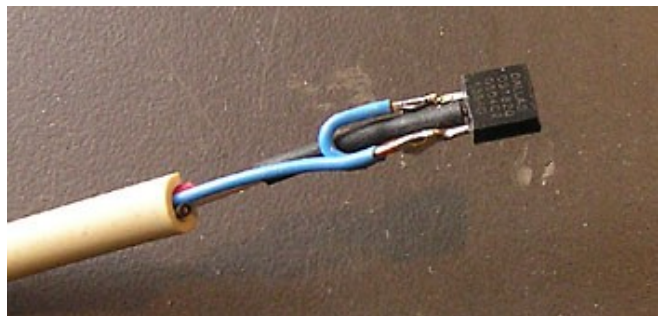
The DS1822 is cheaper, but is only accurate to 2C, which is not much if you ask me. Avoid it.

You will also notice that I still use 3 wires in my setup, it's for future expansion, but in fact, a single twisted pair could be used.

You can grab some sensors from the Maxim web site. First, try to get some samples, which a free and come shipping included. You can get from 4 to 6 of those babies in a single order, which is a good start. You can get the others by ordering directly, or via digikey or some other electronic retailer. Be sure to ask for a TO-92 package, or else you could endup with some surface mount devices, or even worst, a tape of 100 or 1000 devices (used in automated part placement). If you only need 4 or 5 of them, you will have to come up with a very good excuse to tell your wife!



A DS1820 in the TO-92 form factor besides a centimeter rule.



A DS1820 non PAR, see the trick to join pin 1 and 3

You will now need some cabling to plug all those together. I use a simple CAT3 2 pairs twisted unshielded cable. Some will propose to use CAT5 4 pairs UTP, and I see nothing wrong with it, and depending on the way you want to go (see annexes), it might be the single choice you have.

You might want to buy extra stuff depending on how you want to proceed in installing those sensors. I would suggest:

- a handful of RJ11 or RJ12 connector and the crimper
- some telephone wall plate, surface or not
- cheap telephone cord extensions
- RJ-11 or RJ-12 female connectors, the quick-insert type
- Some phone couplers, be sure to check if the number of conductors is correct for your setup

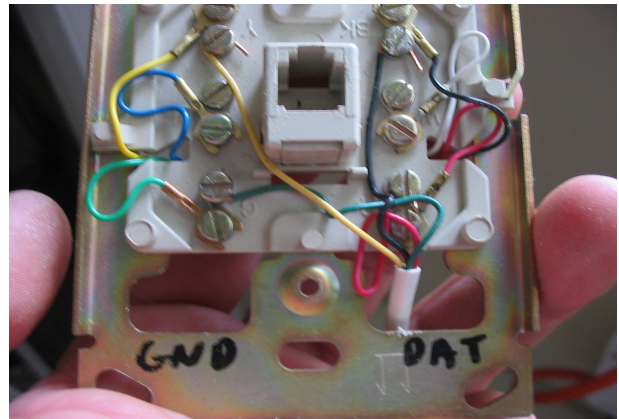
## Now that you got all the basics

Let see how you can plug that.

First, from the interface to the cable.

I used an old RJ-12 plate that I had laying around. That gave me an excuse to have kept it all those years telling my love that one day I would need it.

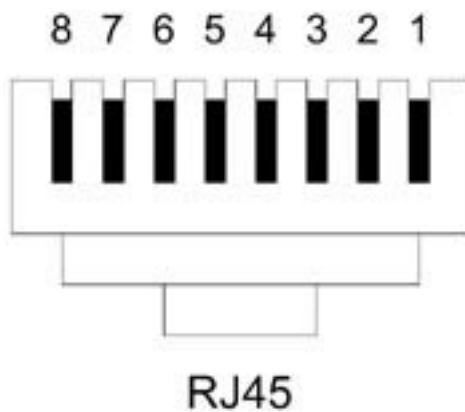
I have standardized on the colors, the green being the GND pin, and the red being the DQ (DAT) pin. I also have the yellow following my setup, with is the third wire I told you about earlier. The black does nothing but is hooked anyway.



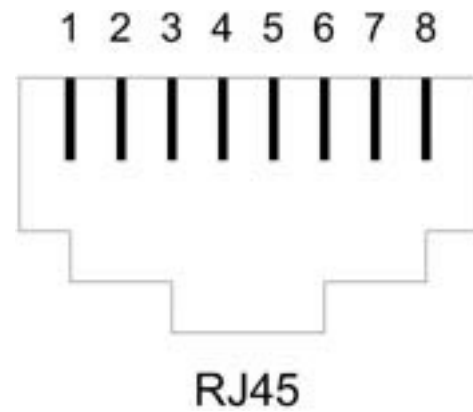
If you plan to use CAT5 cable, I would **highly** suggest that you use this scheme. Of course, use only the wires that you need, that is the 1-wire return and the 1-wire data (the blue pair), but take time to hook all the wires nonetheless. I will tell you later why.

1	2	3	4	5	6	7	8
White Green	Green	White Orange	Blue	White Blue	Orange	White Brown	Brown
Power Return GND	+5v DC Regulated +5v	Power Return GND	1-Wire Data DQ	1-Wire Return GND	Not Connected	+12v to +24v Unregulated	Power Return GND

RJ45



RJ45

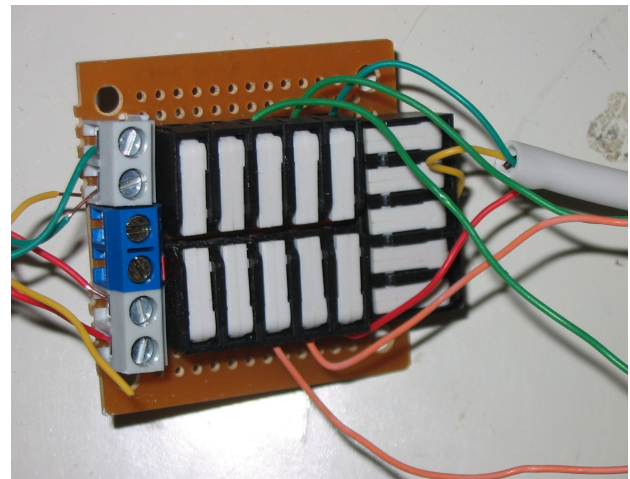


RJ45

I have fabricated some kind of « hub » (I should not call them that, but it's the only name which mean something...) which allow me to daisy chain and plug sensor easily.

On the picture's left: the 2 screws on the top are the GND in and out, the bottom is DQ in and out, and the center (the blue one) is the yellow, in and out. Each pair of screws are plugged to a set of 6 quick connect from the underside of the PCB.

On the far right, you can see one extender cord, and two sensors plugged. The sensors do not use the yellow wire, only the green and the red/orange.



With some look back, I'm not sure I should have gone that way. But the cables are pulled, and the decision taken. If one keep on thinking until he is sure of everything, he will accomplish nothing.

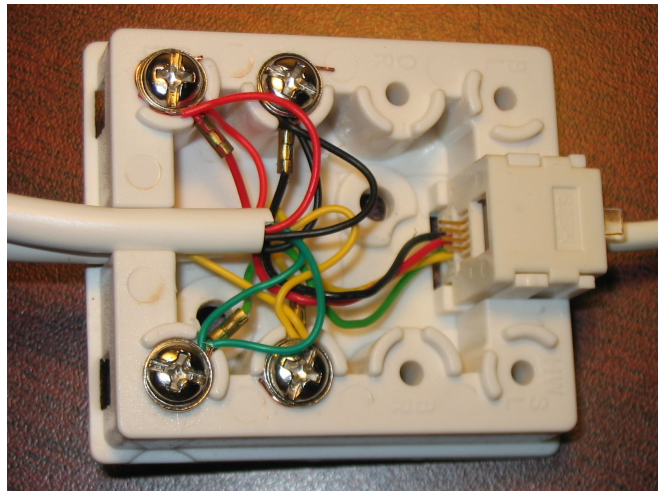
I would suggest a few other way to do it:

**Using a CAT3 cable:**



I now try to use this method. I bring the CAT3 wire to a more or less concealed place, put a regular surface phone jack, and start an other run of CAT3 toward the next spot. If I take care to keep some slack in the cable, I might even be able to add a box in the middle of the run later on.

Into the phone jack, I plug a sensor.

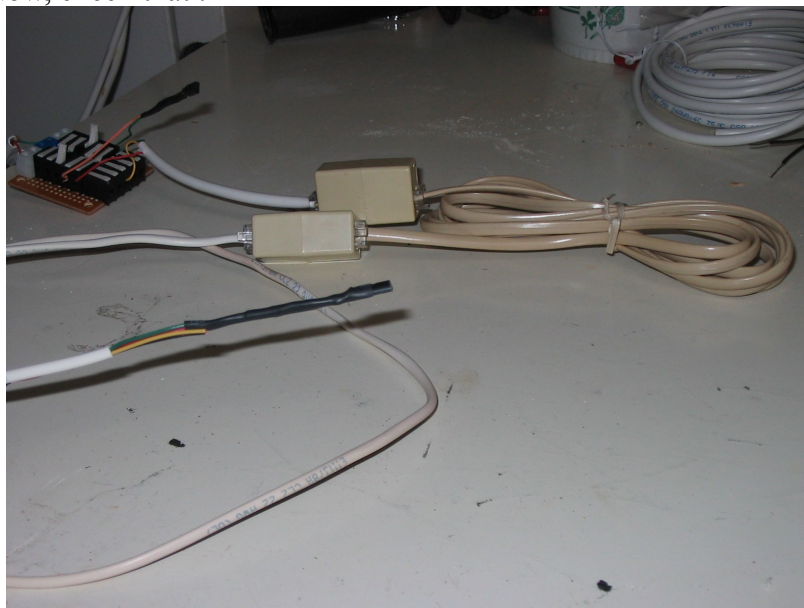


There is also surface boxes with two phone jacks. You can use them as plugs for your jumper cable. If you use regular phone extension cord as jumper cable, be sure keep the red and the green STRAIT THROUGH. Cross the green and the red at the correct place (in the box itself) to ensure continuity. From the said box, you can then run a small length of CAT3 cable with the sensor soldered at the end. I don't have such boxes, so I can't explain much.

**Using CAT5 cable** is almost the same, except that you exchange the RJ11 for RJ45 connectors, and use the other proposed wiring scheme.

Now, remember when I told you that you could daisy chain the sensors? Now, imagine that instead of using only a few inches of wire, we extend that to a few feet, maybe 10 to 12 feet? Not only the network will not suffer (remember, we « talk » very slowly, so extended runs are not a problem)

Now, check that :



This allows me to have a single « main » run across the house, and also allow me to have the sensor at the best place, even if it's a bit away from the initial cable run.

The bigger white boxes are telephone couplers. One of them is special, see note.

I have soldered a DS1821 to a length of regular phone cable, and terminated that length with the regular RJ-11 plug. The length goes into a coupler, into a regular phone cord extension, into a special coupler and finally goes into my « hub »

**NOTE:** If you use a regular phone cord extension, you have to be

careful, because the « standard » cord will cross the green and the red wire! I use a specially modified coupler that also cross the red and the green so they end up beeing straint through cable. I basically cut the red and green, and solder them back crossed, red to green and green to red.

I would also suggest that you aquire or borrow some sort of labeling machine. The thing is that each 1-wire device that we use have a unique ID (a long hexadecimal number) burned into the chip. If you add all your sensor at the same time on your network, the software we will use can only show that ID and not a more easy to remember name such as « office » or « kitchen ». the best is to install one, get it's ID, make a label, apply it to the sensor, and go on with the next. That way, if you ever move one of those, you can change the « easy » name based on the ID you see on the label.

You can also label the surface boxes as beeing « temperature probes, not telephone jack ». You can also label the cables you've pulled, your right hand, and your dog. Avoid labelling your wife and kids. It would look suspicious.

Now that we got the hardware pretty well covered, let's move on to the delicate apparel... I mean the soft wear.

## In the computer now...

My home server is a AMD Duron with 512 megs of RAM runnig Windows 2000 at ... a fair speed. It runs

- Powerhome
- Apache
- MySql
- DynDSN
- Cobian Backup
- Some other « lesser » software

I am very happy with the setup.

Our first goal is to have access to the 9097U-009 interface. To do so, here are some steps

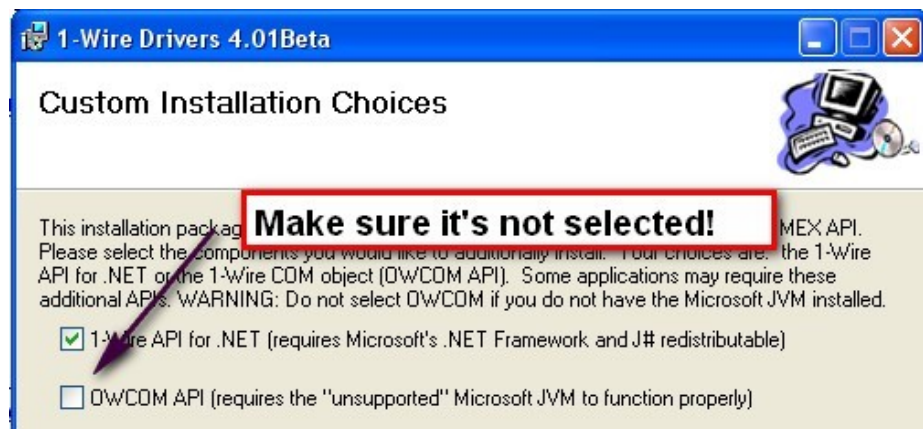
Go to <http://www.maxim-ic.com/products/ibutton/software/tmex/index.cfm> and download either of the V 4.01 beta 32 bits or the V4.01 beta 64 bits, depending on your preffered OS.

Execute the correct driver, but...

**DO NOT INSTALL THE COM OBJECT.**

**DESELECT IT BEFORE STARTING THE INSTALLATION.**

**THIS IS VERY SERIOUS!**



I won't tell you how to set the rest of the installation, I suppose you can do that by yourself.

Now, I'm sure you would like to start....

## Testing your 1-wire network

I would recommend using the Dallas application that « almost » come with the driver. The software we'll use is a Java application. So you'll need a Java VM on your computer. I won't tell you how to check, because you are a big guy, and should know how to do it.

To run the Java application, go to « start>>all programs>>1-wire drivers >> OneWireViewer Installation



This will load a HTML page that states:



Do as they say, and click there. This will load a 1-Wire viewer application, that let you auto-select the interface you're using, and let you see the various 1-Wire devices you have on your network.

Play with it, have fun, and when you're done, we'll move on.

We have the driver, we know that the network is working, all that's left is to find some useful software for us. I recommend that you use a freeware named « LogTemp », made by MrSoft and available at

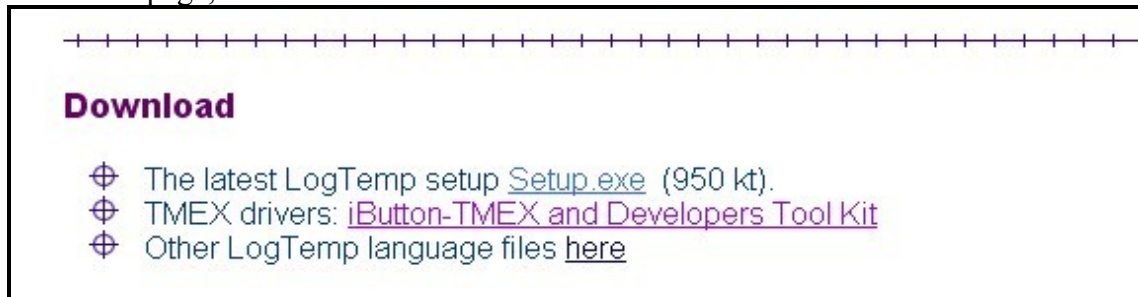


<http://www.mrsoft.fi/ohj01en.htm>

Now, this guy is finnish, and my finnish is a bit rusty... almost absent... well, null.

Nonetheless, this programmer is kind enough to have « language file » which translate the software into something more understandable for me.

Go down the web page, and download and install the LOGTEMP software.



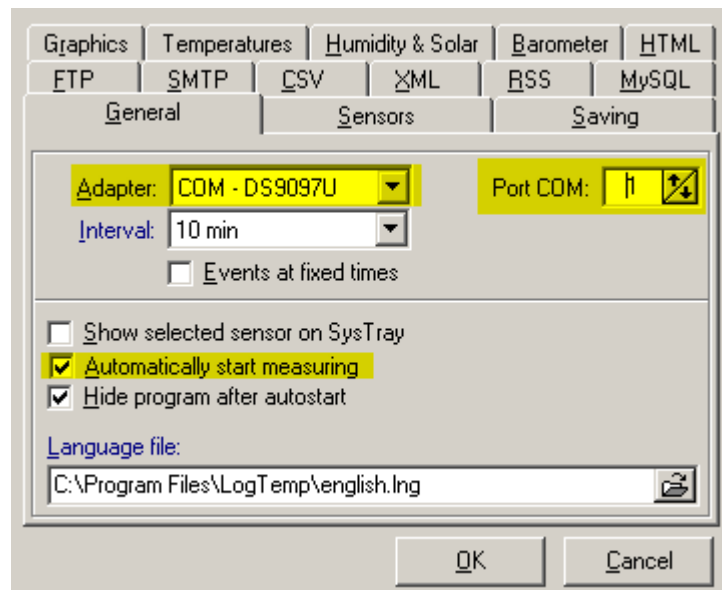
Installation is pretty straitforward.

Once the installation is completed, there is minimal setup to be done. Let's do it.  
I'm running version 2.19,0,70, just so you know

Now bear with me, I will show you step by step setup. It's a lot of picture, but as a picture is worth a thousand words, I will spare you the reading of many tens of pages!

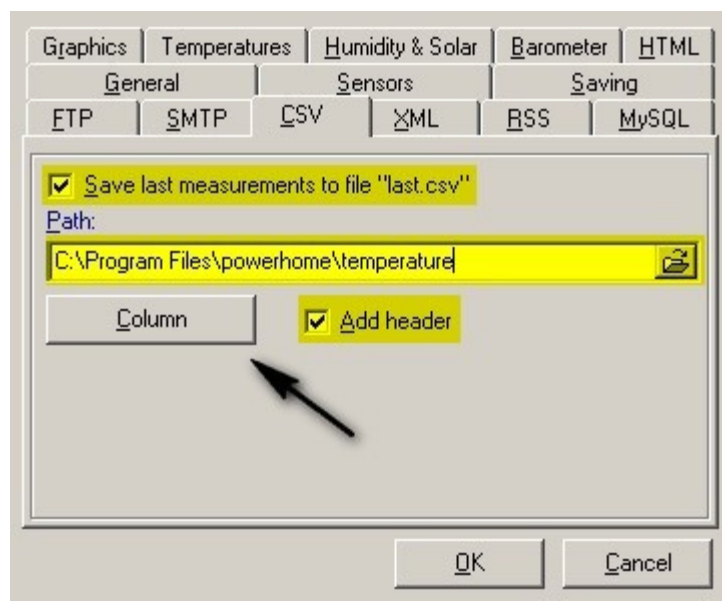
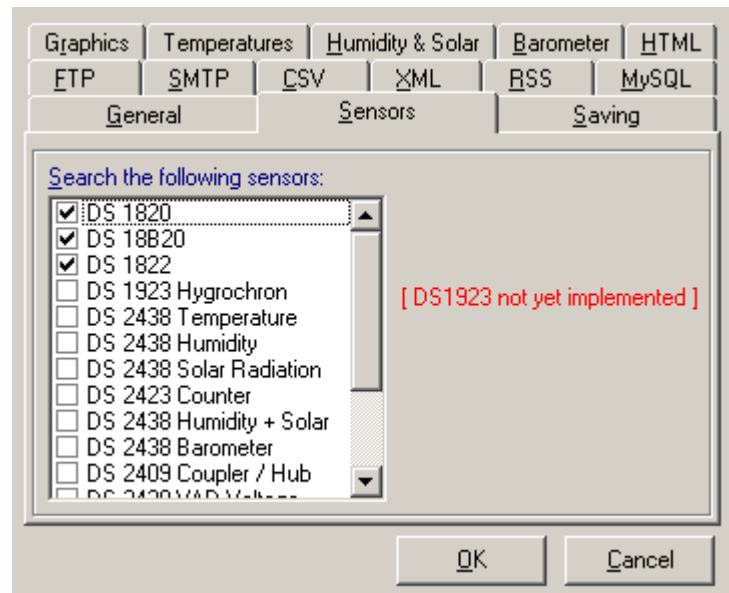
Go to SHOW>> SETUP >> general

Set the inteface you're using, the comport if required, and tell the software that you want it to start automatically taking mesurements.



Now, go to the sensor tab. It will show you all the sensor the software can read, as well as the one you really want it to look for in your 1-wire network.

As I told at the beginning, I'm shooting for temperature here, so I'll stick to those sensors.

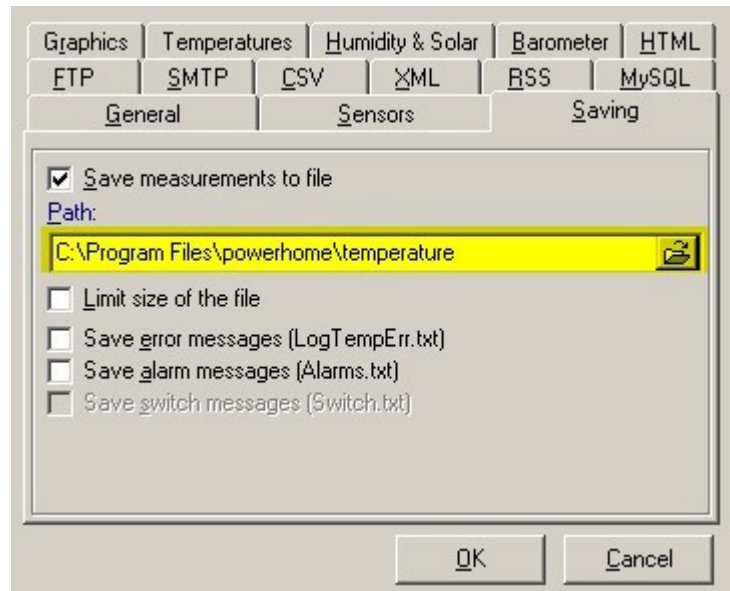


Go to the SAVING tab, and tell it to save the measurements somewhere.

I have choosen to create a subdirectory below powerhome, but you can put it where you want.

BTW, this file is not absolutly needd, just that when you start testing something, you want as much information as you can.

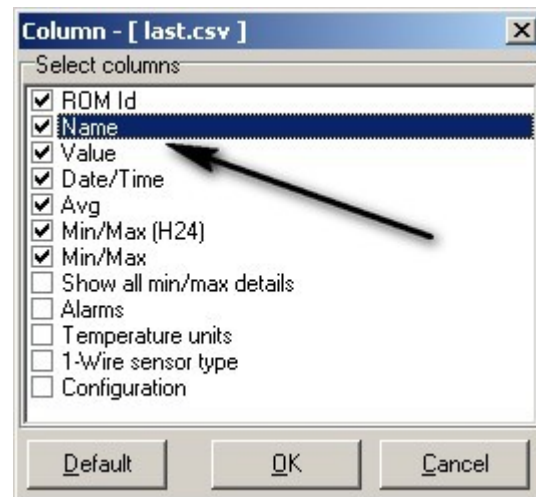
You can always come back later and disable this saving of measurements.



Hit the « column » button and **be certain to add the « name » column, as well as the date/time.**

The AVG, MIN/MAX (24h) and MIN/MAX columns are optionnal, because I dont use them. However, you never know when they might come handy, and it does not really ask that much of the computer to also put down those values as well.

Click OK to save those columns.

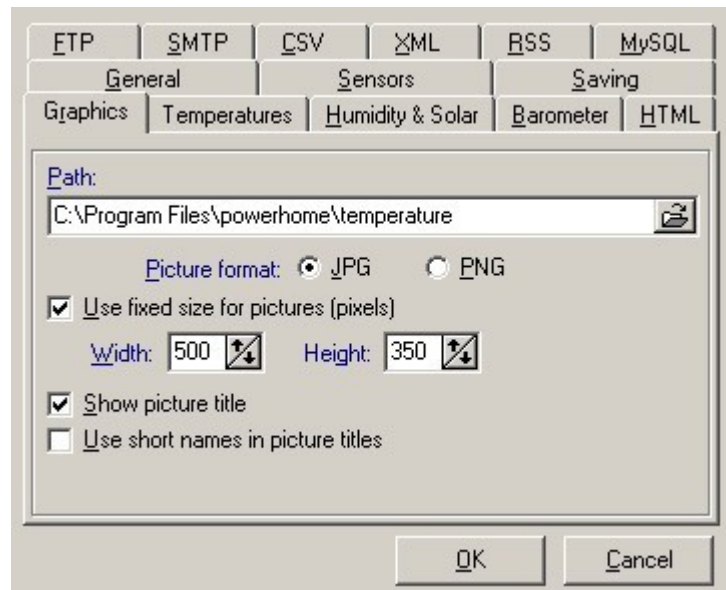


Go to the GRAPHICS tab

Here, you tell LOGTEMP to save graphics in said directory, give it the bitmap format.

Why save those temperature curves you ask me?

It might come in useful if you have created some nice web interface for your powerhome application, you can upload the picture via the FTP thing-a-majik to update your web page or you can use it in your control center, because the software can save the daily graph as « day.jpg », the weekly as « week.jpg » and the yearly as « year.jpg ».



That allows you to have a singly named bitmap of the temperature curve to add to your custom screens. Pretty cute if you ask me. And you can set the size too, because size DOES matter!

Go to the TEMPERATURES tab

This is the place where you setup the bitmap files that will be saved, daily, weekly and so on.

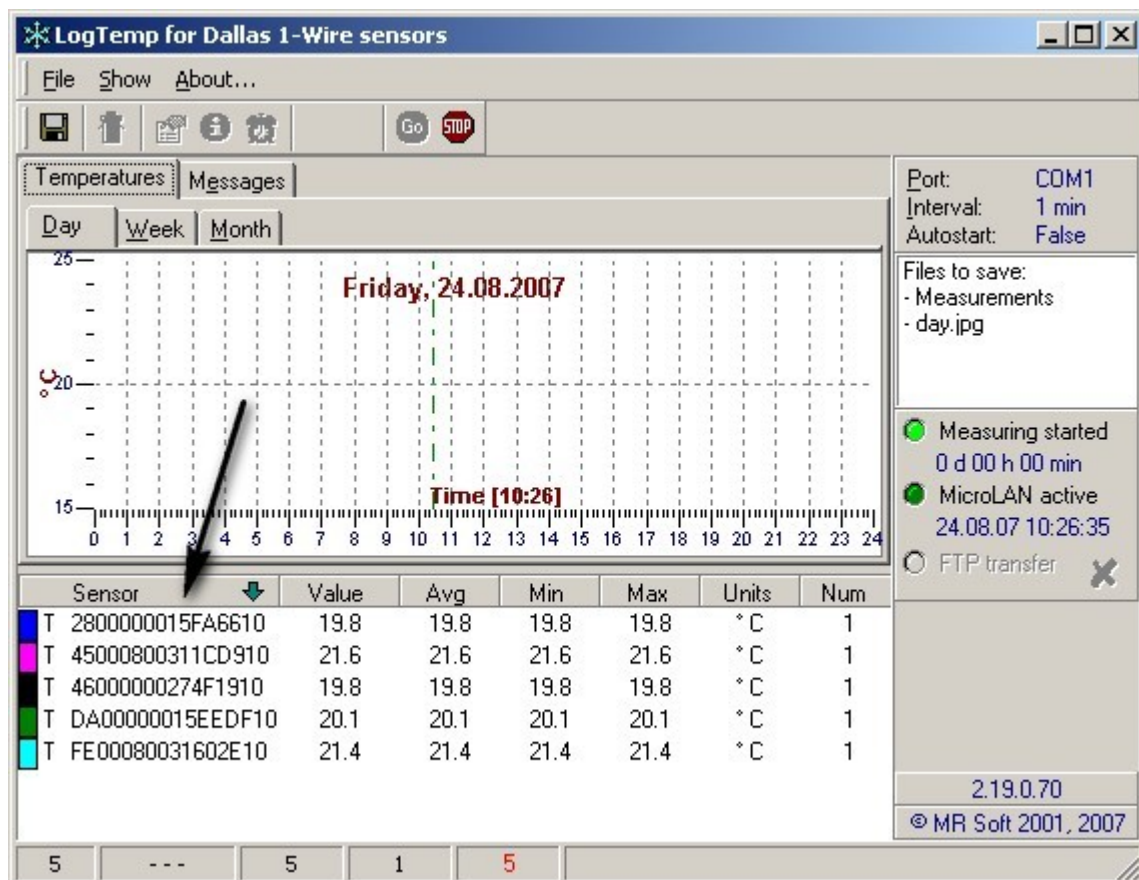
Check the save as « day.\* » and well as the « dyymmdd.\* » filename if you wish to keep some archived graphics.

Notice that I use Celcius as the temperature unit. The DS1820 is outputting it's reading in C, but here, you can change it for Farenheight if you wish.

The screenshot shows the 'Temperatures' configuration window. At the top, there are tabs for FTP, SMTP, CSV, XML, RSS, and MySQL. Below these are sub-tabs: General, Sensors, and Saving. The 'Sensors' sub-tab is active, showing 'Temperature units:' with radio buttons for Celsius (selected) and Fahrenheit. Below that are tabs for Day, Week, and Month. The 'Day' tab is active, showing 'Saving of the day graphics' with checkboxes for 'Save as day.\*' and 'Save as D'YYMMDD.\*', both of which are checked. There is a 'Saving interval:' dropdown set to '1' and a note 'times measuring interval (= 10 min)'. A checkbox for 'Manual temperature scale' is unchecked. At the bottom are 'OK' and 'Cancel' buttons.

**Now, click OK. It brings you back to the main screen**

With any luck, it will have started to scan and read the values of the devices you have on your network. If not, use the « GO » button to start the readings. Starting the reading is often a bit slow to start, because the software has to initialize the comport and such. Dont be in a hurry and do wait a few seconds!



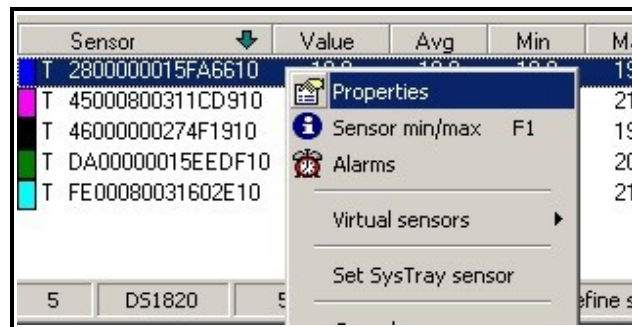


As you can see, it tells you the ROM ID of the sensor, and the values that goes with it.

Ah! How cool it is to know that 28000000015FA6610 is the office sensor! So easy to remember!

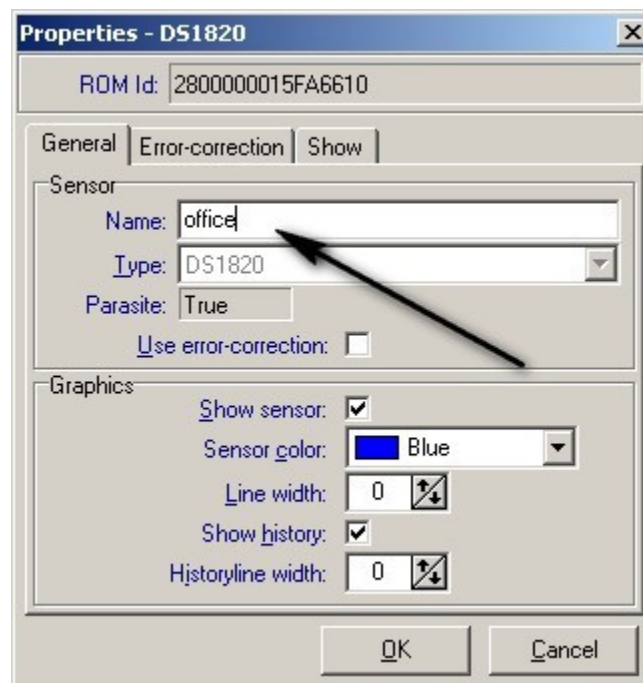
Luckily, MR Soft is not a wimp and embarked into a journey to let us use his software more easily!

Right click on the sensor ROM ID, and choose properties



And this is where, O fatefull reader, that the magic enter your life!

You can add a NORMAL name to that ROMID from hell!



By clicking OK, you will now be able to see the true meaning of 28000000015FA6610 (BTW, this is NOT the number of the beast!)

Sensor	Value	Avg
T office	19.9	19.8
T 450008003110910	21.7	21.7
T 46000000274F1910	19.9	19.9
T DA00000015EEDF10	20.3	20.1
T FE00080031602E10	21.4	21.4

**Hurray! We have defeated obscurantism!**

You can of course change the wierd names of the other sensors too! Arent you happy now?

So far, we have installed our 1-wire network, installed the sensors, and setup software for reading the value. All that's left now is...

## moving the values into PowerHome

So far so good. You are still with us, and I'm glad you are, and so are you, because we're heading toward the last mile of the run.

Move to your PowerHome application.

Go to the global variables section

Remember when I changed the ROMID for a more intelligible name? Well, this sure comes in usefull here.

We will create some global variables using those clear name. All of them will be prefixed by « TEMP\_ ». Not for temporary, but for temperature.

So, in my screen capture here, I have « TEMP\_ATELIER » which means that I have a sensor named « ATELIER » somewhere. (BTW, atelier is french for workshop)

Same thing for the « BUREAU » sensor (office), the CHAMBRE (bedroom), TEST and TEST2 sensors (those, I wont translate).

TEMP_ATELIER	19.78
TEMP_BUREAU	18.98
TEMP_CHAMBRE	21.31
TEMP_CSV_FILE	c:\program files\powerhome\scripts\last.csv
TEMP_CSV_LAST_DATE	24/08/2007
TEMP_CSV_LAST_TIME	09:51:47
TEMP_TEST	18.88
TEMP_TEST2	21.63

You will also create 3 other globals, namely:

TEMP\_CSV\_FILE

TEMP\_CSV\_LAST\_DATE

TEMP\_CSV\_LAST\_TIME

which will contain the filename you want to use as the input file and the last date and time the reading was done. Those last two variables are updated each time a new reading is done, and the data itself comes from the « last.csv » file, more precisely the date and time of the last reading available in the file.

I know that I have changed the name of the directory from one example to this one. I was just checking your level of attention.

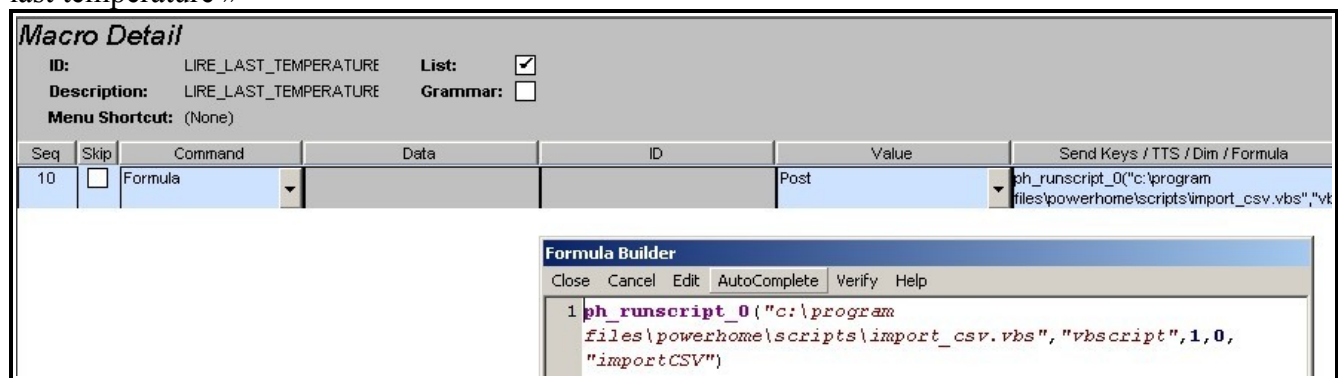
You now have to create the vbscript. Have fun...

No, seriously, I have included the script as an annex, because it is quite long. You will also see that it is full of comments and such, so if you wish to modify it, you have an idea where to do it.

Save the script as « import\_csv.vbs » in the right directory. Remember where, cause you need that information in the next step.

Moving on to the macro that calls the vbscript:

Mine is called « LIRE\_LAST\_TEMPERATURE » which is the mess of french and english for « read last temperature »

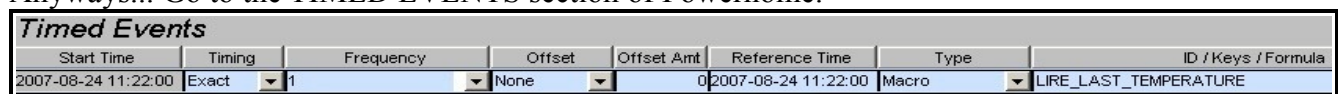


This macro simply calls the vbscript name « import\_csv.vbs » with no parameters. The script reads the values directly from the globals you created in the last step. It's a formula with only one line, here is the transcript:

```
ph_runscript_0(« c:\program files\powerhome\scripts\import_csv.vbs », « vbscript », 1, 0, « importCSV »)
```

Now, you want powerhome to read the data from time to time. My needs (well, future needs) requires that I read the temperature each minute. So, as a first step, I told LOGTEMP to read the sensors once each minute. You timing may vary, you are the boss of you home... or is it the nice lady over there?

Anyways... Go to the TIMED EVENTS section of Powerhome.



Create a new event:

timing = exact

frequency = [nbr of minute] or choose hourly which translate to 60 inside.

Offset = none

offset amount = 0

reference time = (put a date and time ahead of now, maybe 4 or 5 minutes)

type = macro

ID = LIRE\_LAST\_TEMPERATURE or whatever you named the macro that calls the vbscript.

## All is primed for success

If you did follow the instructions, and if I forgot nothing, at the reference time, the timed event should fire (I mean, it should start. No need for kevlar suit here!), call the macro which will call the vbscript which will read the last.csv file.

The second after, the values should have changed. However, the display do not update itself, and you have to press F5 or hit the « lightning bolt » button to update the global variable display.

If it does not work, double check each step. As it is, everything should be working, so the problem is on your side.

(This reminds me the joke about the Microsoft tech support that goes to a firing range, shoot a clip at the target and then ask the guy at to other end to check the score. « You hit nothing » he answered. So, the MS guy put his finger into the barrel of the gun and fire a round. Of course, the finger blows away like \$!17! So, the MS guy tells the target verifier « It's working fine here. The problem must be at your end! »)

I assume nothing, and will do no help desk. Just typing this document meant enough time besides the time I took to research, test and everything related to the installation of the hardware and software.

If you mess with something and blow your computer, scrap your powerhome setup, drill throught an electrical wire and burn down your house while pulling your cable, I dont really care nor will I pay for any dommages you might incure, or for lost of revenue or anything else. And if your wife leaves you, it sure wont be my fault.

You will use this tutorial as is. If you ask politely, I might correct typos, but my implication stops there.

Now, on with the ...

## Annexes

So, here I will jolt some notes in the order of random thoughts.

### Remember when I told you about using CAT5 cable?

It is related to this very nice place:

<http://www.hobby-boards.com>

NOTE: I have not bought anything from them, and I will probably never receive anything from them without paying first. So me naming them here means nothing. However, I have read numerous good comments about them on the web, and the pricing seems honest.

They have one of the most complete 1-wire list I have seen. They even create board with 1-wire devices to do more, like the solar radiation meter, or humidity, ect

The problem I see is that it's a bit expensive, and I did not needed anything from them only for reading temperature. Things might change in the future as my need and my lover's request might change too. But for the moment...

Anyways. They sell some nice REAL hubs for 1-wire devices. Why a hub if I can have long runs of cable? Well, let's say you have green houses. BIG green houses. One run per green house, going into the hub, and then to the PC. That way, you can have say 5, 10, 20 probes per green houses, and still be within the 300 feet or so that the 1-wire network will allow. Or maybe that you have a REALLY big house?

And, some other use for thier hub, is to send power (yeah! More power!!!!) to some of your devices (some will need external power to work)

They also sell 1-wire relays, LCD drivers (so you hook it and have the LCD display something!), humidity, solar radiation, motion detector, moisture meter and other things most interesting for home automation, and maybe for those big greenhouses of yours.

And to really use all the power their board provide, you need to use CAT5 cable, hence the cabling sheme I propose earlier in this document.

The other interesting thing is that LOGTEMP can read most of those made-up devices, which allows you to build up on solid grounds.



There is also this other store, which I also find correct as far as I can see.

<http://www.aagelectronica.com/aag/index.html>

I have not bought anything from them, and I will probably never receive anything from them either. So me naming them here means nothing. However, I have read good comments about them on the web.

They have a nice 1-wire over ethernet hub. It's not cheap, but I can see numerous things that can be done with such a device, especially into large manufacturing plants where data gathering can be a problem due to distances. SCADA for cheap anybody? Could also be of use in those BIG greenhouse I told you about....

Besides the usual, they also offer counters, a complete LCD solution for 1-wire, pressure transducer, thermocouple adapter, and a Ibutton holder.

The most interesting thing they sell is a complete 1-wire weather station! They are famous for that article. That means that you could add a full fledged weather station for about 250\$.

If you buy something there, let me know about how you like the service!

# About the iButton

The iButton are mobile devices as small as a few quarters piled up. They are made of stainless steel (not stainless steel!) and they usually contain a standard 1-wire device. The stainless steel casing make them ideal for many uses where the environment is not that kind.

The basic iButton is simply a ID thing, the famous ROMID. You add a reader, and bingo, instand keyless entry.

Well, the ibutton IS the key if you use some electric latch.

In that case, you say, why not use an RFID? Because of the cost. An iButton is more expansive than the RFID, but the reader is cheaper than an RFID. As each iButton reader act as a 1-wire device it becomes less expansive to build a network of 1-wire device than a network of RFID reader.



A more advance iButton still contains the unique ROMID, but add some RAM. If that ram containt a money amont (say 10\$) and you bring the Ibutton close to a reader on the coffee machine, you can use the device as some kind of electronic wallet.

Even more advance iButton contains a temperature and humidity RECORDER. It will make a reading each x minutes. Later on, the user can dump the data into the computer and check stuff. Drop the thing into a van of vegetables, and check if the temperature inside the van was kept to the optimal point in order to have a fresh load of potatoes delivered to the warehouse.

All in all, the iButton is useful when mobility is needed, or when the elements can get to the device.

Here, see an iButton hygrochron glued to a turtoise during nesting time. After 2 weeks, they removed it, and dumped the data into Excel. What they found? I dont know, but it did worked.



# The import\_csv.vbs script:

```
sub ImportCSV()
'
Const ForReading = 1
dim DataArray
dim strROMId
dim strName
dim dblValue
dim dteLastReadingDate
dim dteLastReadingTime
dim strAvg
dim strMin24H
dim strMax24H
dim strDateFormat
dim strTimeFormat
dim retour
dim SourceFile
dim TextFile
dim intDay
dim intPosition

'
'set filename to read
FileToRead = ph.getglobal_s("TEMP_CSV_FILE")
'
'
'file format:
"ROMId","Name","Value","dd.mm.yyyy","hh:nn:ss","Avg","MinH24","MaxH24",
"46000000274F1910","atelier",19.56,14.08.2007,23:25:45,19.49,18.62,20.95,
"DA00000015EEDF10","bureau",20.45,14.08.2007,23:25:47,20.31,18.18,22.56,
"FE00080031602E10","chambre",21.19,14.08.2007,23:25:44,20.90,20.25,22.38,
"2800000015FA6610","test",20.73,14.08.2007,23:25:43,20.26,18.29,22.17,
"45000800311CD910","test2",21.31,14.08.2007,23:25:46,21.23,20.50,22.81,
'-----
'
'msgbox "1"
Set SourceFile = CreateObject("Scripting.FileSystemObject")
If not SourceFile.FileExists(FileToRead) Then
'    msgbox "CSV File not found!"
'    exit sub
end if

Set TextFile = SourceFile.OpenTextFile(FileToRead, ForReading)
'
Do While Not TextFile.AtEndOfStream
    redim DataArray(0)
    TextLine = TextFile.ReadLine 'Read a dataline from SourceFile
    '
    if instr(textLine,"ROMId")<>0 then 'we will get the date and time format from the first line
        DataArray=split(TextLine,",")
        strDateFormat=ucase(dataArray(3))
        strTimeFormat=ucase(DataArray(4))
    end if

    if instr(TextLine,"ROMId")= 0 then 'this line contains data
        DataArray=split(TextLine,",")

        'using the split command, we now have DataArray containg 4 cells (0..3)
        'cells that we will transfer into separated variables (not absolutly needed, but might come handy
        'for educational purpose)

        strROMId = DataArray(0)
        strROMId = replace(strROMId,"","") 'remove doublequote
        strName = DataArray(1)
        strName = replace(strName,"","") ' remove doublequote
        dblValue = cdbl(DataArray(2))
        dteLastReadingDate = DataArray(3)
        dteLastReadingDate = replace(dteLastReadingDate, ".", "/")
        dteLastReadingTime = DataArray(4)
        strAvg=cstr(DataArray(5))
        strMin24H=cstr(DataArray(6))
        strMax24H=cstr(DataArray(7))
    end if
end while
TextFile.Close
end sub
```

```
'
'transfer data from the script to powerhouse
'
'standard naming will be as follow:
'"temp_" followed by the strName value (ex: temp_office)
'that global variable will be of type double and containt the last temperature read
'
'there will be a temp_[strName]_time of type time
'and a temp_[strName]_date of type date
strGlobalId=ucase("TEMP_" & strName)

retour = ph.setglobal_a ( strGlobalId ,cdbl(dblValue ))
    retour = ph.setglobal_s ("TEMP_CSV_LAST_DATE", dteLastReadingDate)
    retour = ph.setglobal_s ("TEMP_CSV_LAST_TIME", dteLastReadingTime)

'we could also transfer min/max and average, but I have no use for it..yet!

end if
loop

TextFile.Close
end sub
```

Photo sources:

Spou, Martin, Maxim IC, and others. I mean not to steal other's work by including picture that I did not made, I am simply trying to save me work and / or expenses (see, I'm lazy and cheap!)

If someone wishes me to remove a picture, or to add a source, I will sure do it no problem. Just ask kindly.

I hope that this document helped you setup a 1-wire network that you will enjoy for quite some time. I sure do hope this is what will happen to me!

Thanks for reading,  
Spou

<http://www.thespou.com>

For people reading french, try <http://www.thespou.com/blog>