HVACMonitor

"programmable thermostat with an embedded web-server for remote control and monitoring."

Marc G. Daigneault - 2009

Updated May 19th, 2009

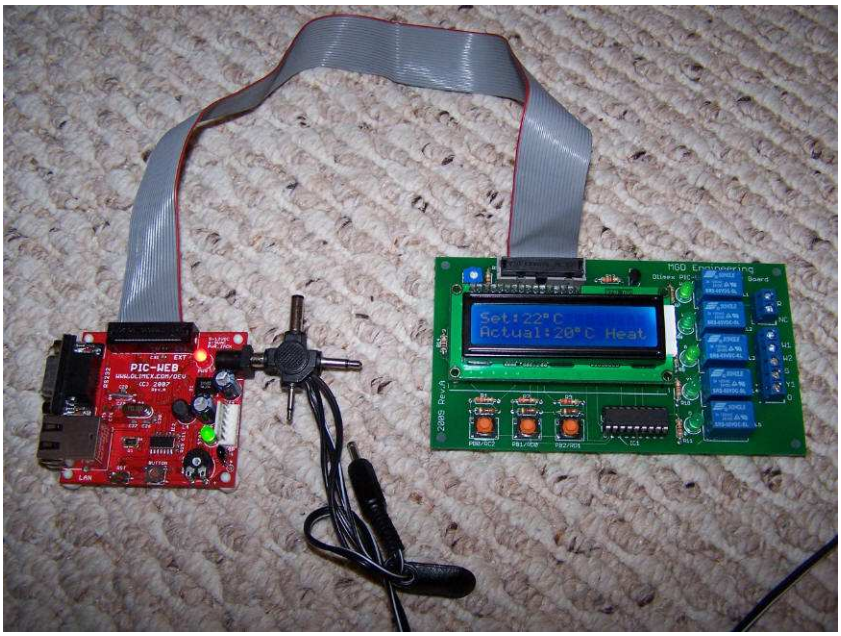
Featured on Hackaday.com!

This project involved making a S.M.A.R.T (Self-Monitoring, Analysis, and Reporting Technology) Thermostat which could be ac over the internet. This project combines several elements of embedded electronics, all while keeping on the theme o Automation. This page is a quick overview of the project; full documentation including EagleCAD board schematic & layout, as the Source code, can be found at the bottom of the page.

When designing this project, I wanted it to conform to these requirements:

- Remote control and monitoring via Web
- Upfront control and monitoring via LCD and pushbuttons
- Remote Monitoring, Data Logging & Graphing

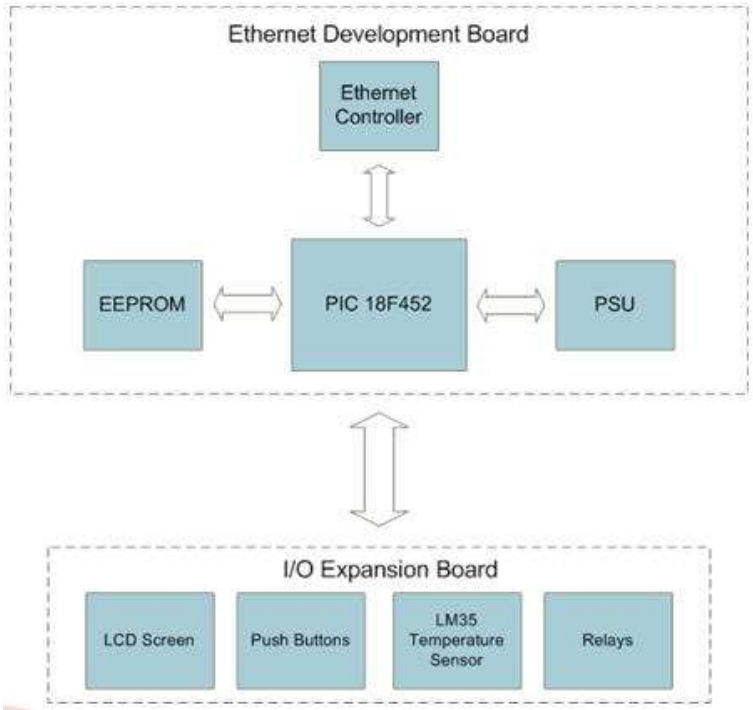
This Web-Enabled thermostat offers full remote control, a full upfront user interface with pushbuttons and LCD & remote mo and logging of temperature history using simple bash and Perl scripts running on a VM Linux server.



High Level Design

Below you'll find the block diagram, which gives an idea of what this project consist of. The Ethernet Development Board is ar

be matched with the PIC-WEB, to allow for full control of the I/O board using the 20 pin extension header of the Olimex board.



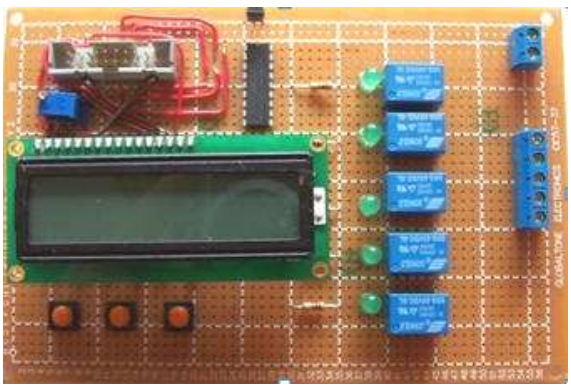
Ethernet Development Board

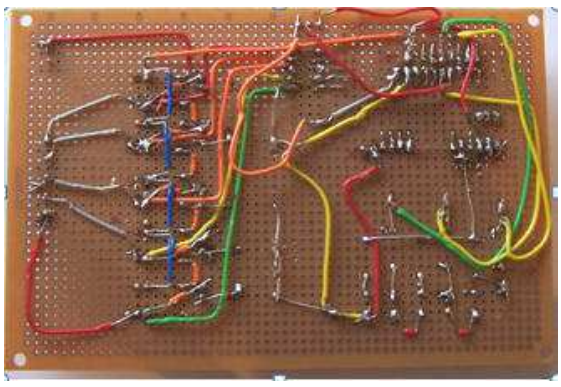
To simplify the whole online process, an Ethernet development board was chosen, and for one good reason: When troubles problems, if it was both hardware and software that I've made, it would've been much more work to troubleshoot, as the prob be either hardware, software or both. By taking hardware that I know works, and writing my own code, I'd be able to troubles no time. The Olimex PIC-WEB board worked great: low-cost, uses the easy to program Microchip 18F series microcontroller, ar nice 20 pin extension header, allowing me to interface with the HVAC board that I've made specifically for this application. The header equally has +5V and GND, so I could use the power supply of the PIC-WEB to power my daughter board.

Although Olimex includes a 20 pin header for their PIC-WEB board, they don't manufacture any daughter boards, so I took u task of making my own. My board includes LCD Display, Push Buttons, Relay's for HVAC, and an LM35 temperature sensor. The F does have a thermistor on-board, but I found it to be inaccurate and slow to react to changing temperature, so an LM35 wa onto the daughterboard. It was much more reliable.

HVAC Daughter Board

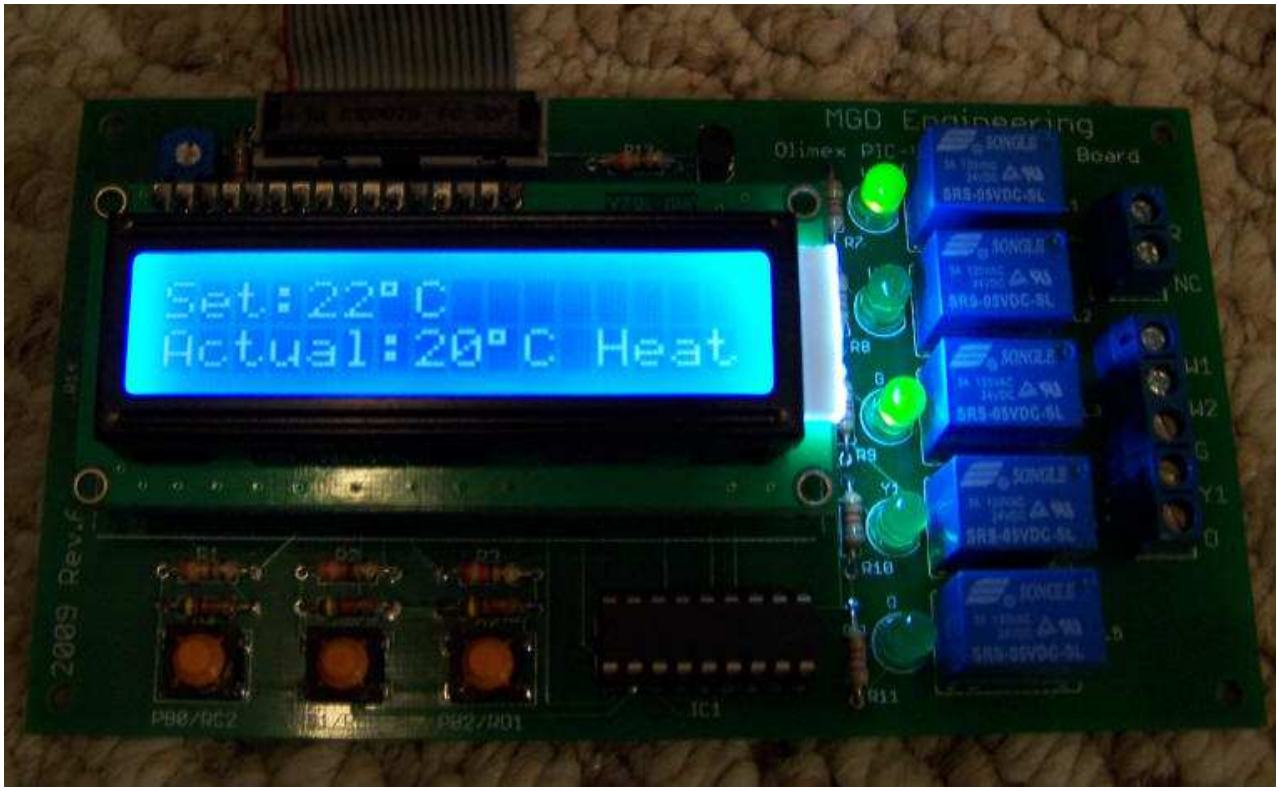
This was the main portion of the project, making a daughter board able to control the HVAC side of the project all while ke compatible with the PIC-WEB. Initially, the circuit was put together on perforated board, however, the mess of wires it demanded the creation of a PCB.





The problem is the 20 pin header that connects to the PIC-WEB: 20 wires in such a dense location demanded many wires, on b top and bottom layer of the board, and made for a very messy board of intertwined wires. Troubleshooting wasn't very pleasa this board did function, and was used to debug the code while I impatiently awaited my EagleCAD board's from SpeedyPCB.

The EaglePCB has only 2 vias for a 2 layer board, pretty good considering the number of traces going to the 20 pin header.



Picking components for the daughter board

Although the 18F452 used on the PIC-WEB is capable of using the standard 2-wire hardware I2C, these pins were not made avai the extension header, as the PIC-WEB uses I2C on-board to communicate with an EEPROM used for storing web pages.

So, for temperature sensing, since I2C was out of the question, both because the hardware lines weren't available, and also b it required two pins, I decided to get a high accuracy analog temperature sensor, such as the LM35DZ from Microchip. This dev chosen because of its compact TO-92 package, low self heating (<0.1°C), which is very important in temperature sensing appli and ease of use (10mV/°C). As well, this device would only require one pin of the microcontroller, where as an I2C device wou required two. Had I had more I2C devices, that would have justified using that bus, but in my case, it wasn't necessary.

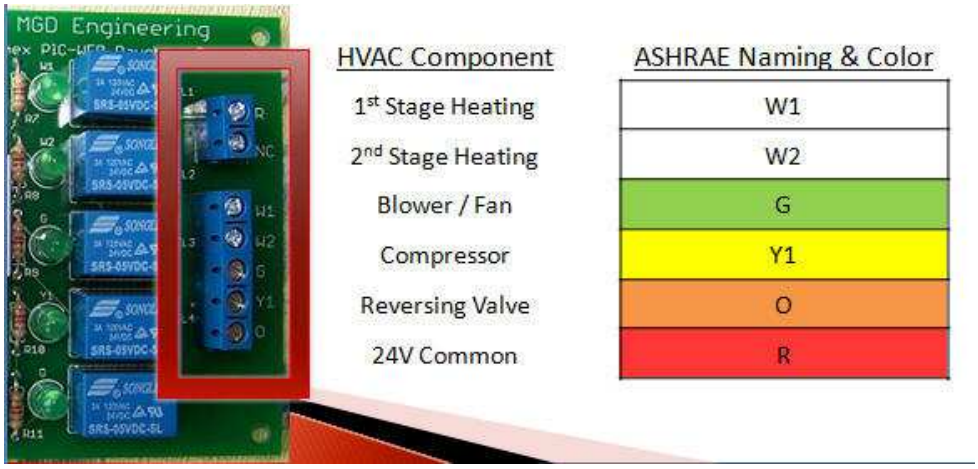
For relays, I had decided to use 5v SPST SRS-05VDC-SL relays by Songle, as I was looking for something that could handle 24VAC HVAC side of things, but as well work well interfaced with a PIC microcontroller. Although I could potentially power thes directly using the output of a PIC, I had examined all the possible states, and noticed that it was possible for 3 or more relay powered synchronously, which would put an unnecessary strain on the PIC. So, to still be able to switch these devices using implemented a Darlington array driver, ULN2803, which acts as open collector output, allowing me to sink a fairly large am current while sourcing a small amount from the PIC. As well, this Darlington driver also has built in clamp diodes to help elimin noise caused by devices on its outputs, which is great, as it would suppress the inductive kick of the relays. This IC was per the application needed in my circuit.

The LCD module uses in all 6 pins from the IC as it runs in 4-bit mode to help reduce the number of pins required. We can coun pins, 1 register select, and 1 enable pin. The LCD is the sole device to use the most pins on the daughter board; however determined necessary as I wanted to integrate both a visual interface as well as the web one. Three standard single pole push were used to allow some user interface. For the thermostat application, this would allow to increase the set point, reduce point, and switch between heating and cooling modes.

As well, standard screw terminals were implemented on the board, allowing for the relay's to switch each individual channel or Since a 24v HVAC systems are all feed from the same power source, the board is setup in a way where each output of the terminals are either connected or disconnected from the +24V common, all depending on the state of the relay.

Interfacing with residential/commercial HVAC system

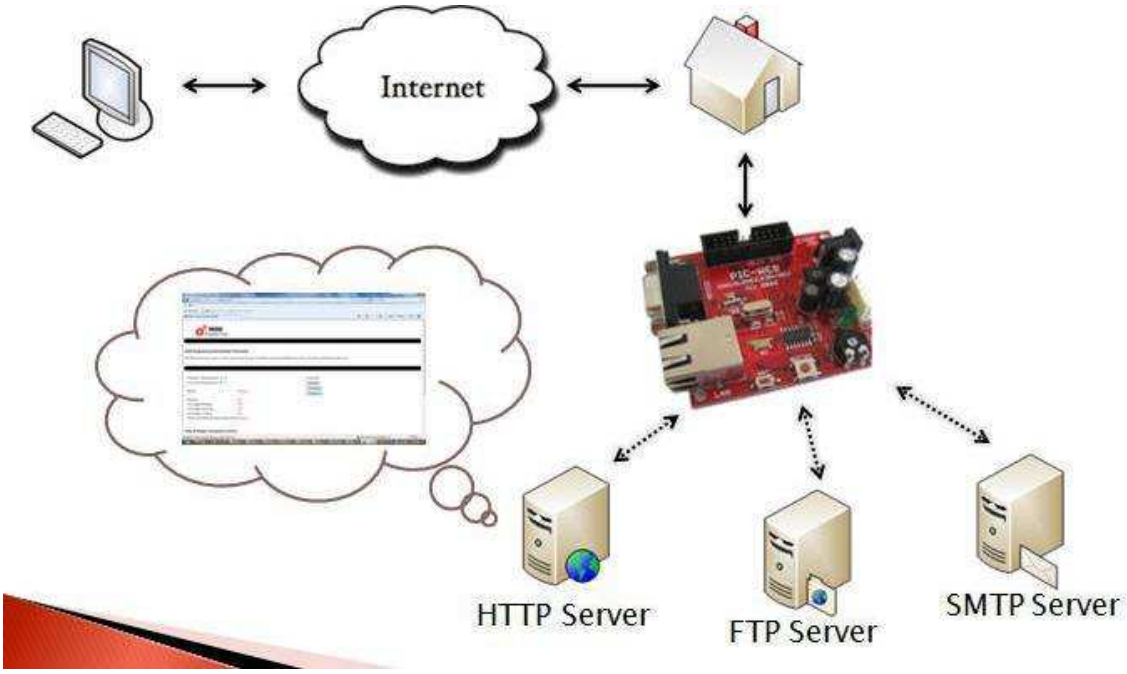
I designed the board so that it could be interfaced with a residential/commercial 24Vac HVAC system. 24Vac systems are an i standard, so if you're wondering what you're running at home, chances are it's what this board was designed for. I wanted th side of the project to be plug and play, so finding the correct ASHRAE naming and color code for the wires was a necessi naming was silkscreened on the board, next to the correct screw terminal. You could potentially connect this up to your HVAC and it would be fully compatible.



This board can run systems that use single stage cooling & single or dual stage heating, with full Heat Pump control. It customized to whatever the application is, however it is limited to 5 relay's for controlling whichever 5 devices the user w Relays is enough for most if not all residential/commercial applications (Unless you've got a million dollar residence), howeve industrial applications run on 220Vac and use a totally different type of system.

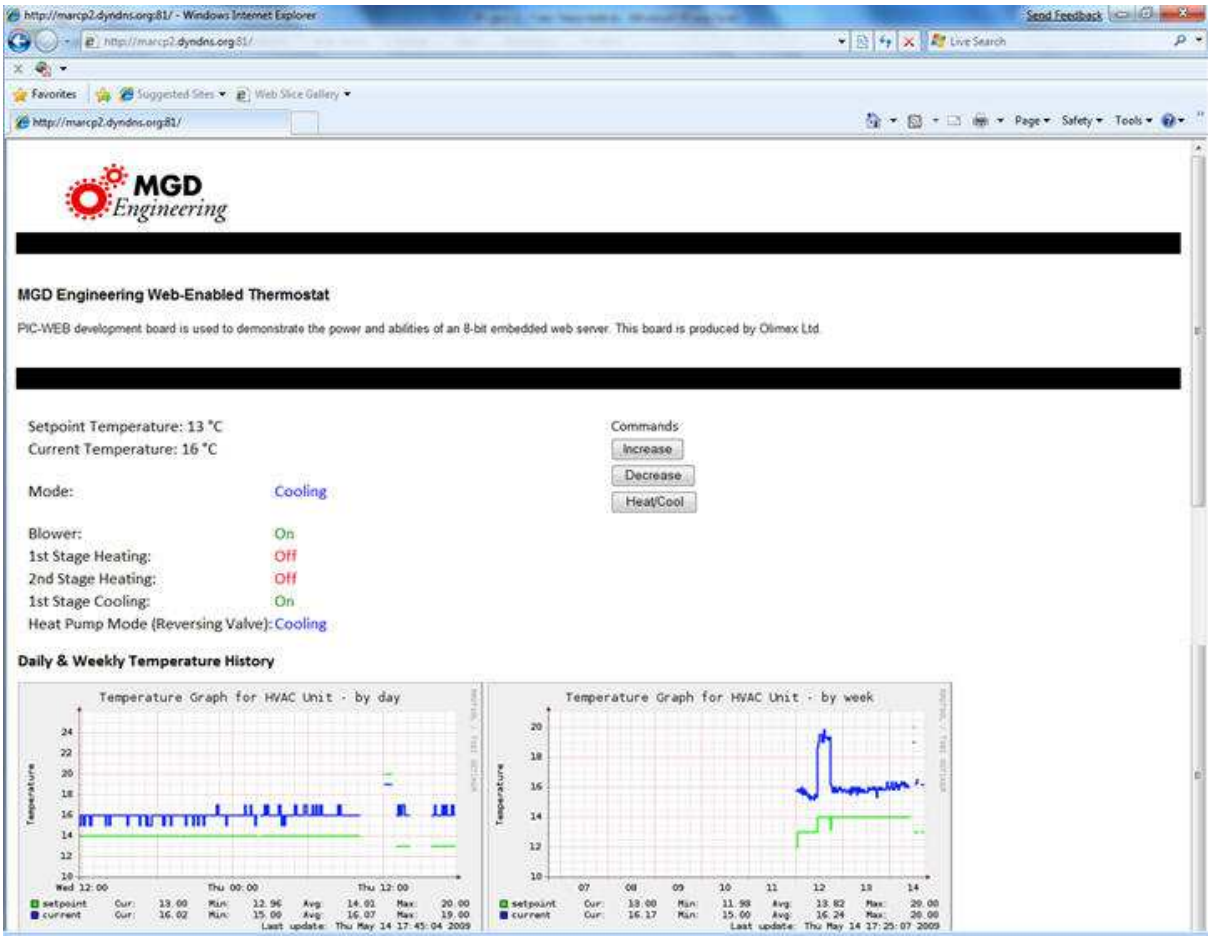
Software - Getting the board online

Olimex does give a fair amount of documentation about making your own web pages and how to interface them w microcontroller. You can fairly easily display on the web page output states, integer values and such. Equally, you can pushbutton that call up specific function prototypes, again totally customizable, so you can switch output increment/decrement integer values, or what ever you want to do.



Getting online is fairly straight forward, as this board uses the messy but reliable TCP/IP stack by Microchip. This stack is basic software that run the HTTP Server, FTP Server, SMTP Server & DHCP Client. Simply plugging the board into you house network allow it to pickup an IP, which you can then access through a web browser. I have the IP displayed on the LCD screen of the d board whenever the board starts up, so I know exactly what IP the board has taken. You can also disable DHCP, and force the to take on any IP you want.

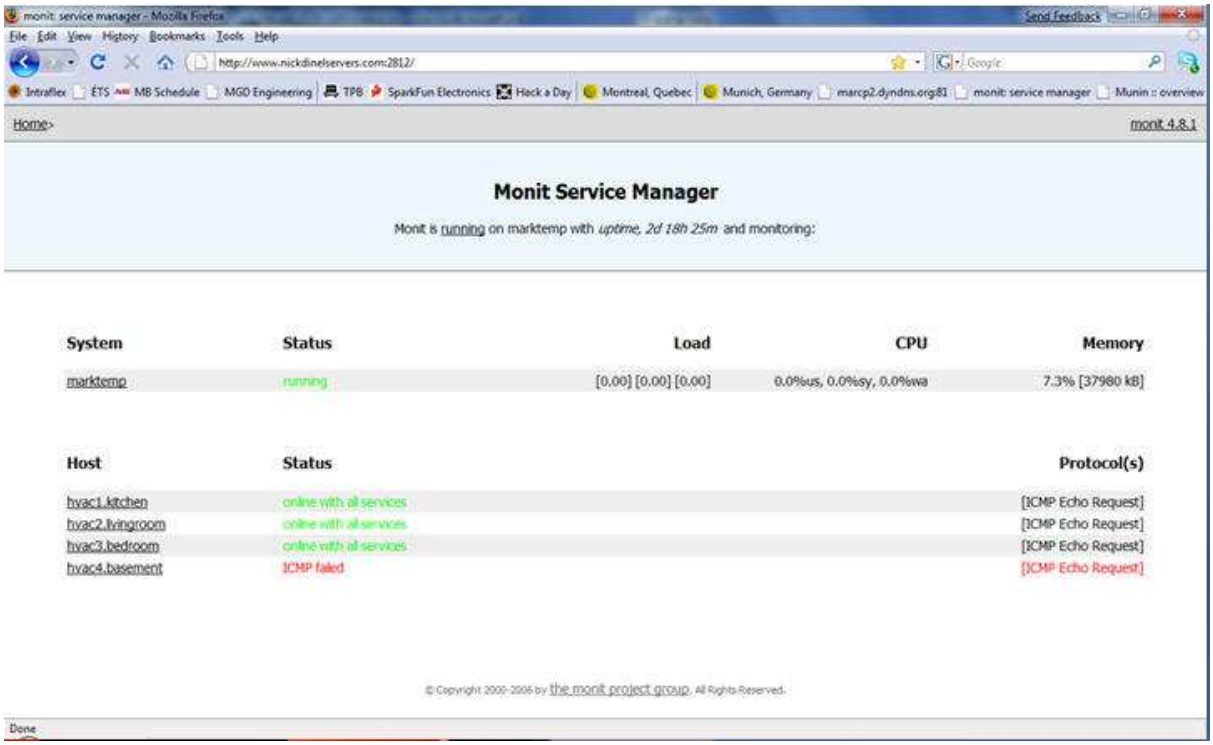
The webpage that I put together allows the user to see the set point, current temperature and the state of all the outputs. the use has control over the set point, with the possibility to increase/decrease it and see it change in real time, and as wel from heating to cooling mode and vice-versa. The nicest feature is the graphing, which shows both Set Point and Temperature on a daily and weekly basis.



A remote VM Linux server is simply polling the site every 5 minutes and registering the Current and Set Point Temperature, the bash and Perl scripts, simply graphing it out with the help of Munin and Monit (Google it!), which creates a .PNG of the hist host it locally. So, the site on the PIC-WEB doesn't actually host the graphs, they are simply hyperlinked from the VM server. A p had to be created for Munin, as it's not made to graph out temperature but rather computer loads and specs.

The nice part about using an external Linux server is that I can graph out and monitor multiple devices. Say you own a commercial building, and have a web-enabled thermostat on each floor. You could monitor and control them all from one location, and keep history of the temperature and set point, which all building owners know is a good thing. (People comp most about temperature in a building...you now have proof of temperature on every floor at every hour of the day, with da going back months if not years.)

Below, you can see how Monit can be implemented to monitor multiple devices, and see which are functioning and which are r



So, the final product: a fully automated S.M.A.R.T system (Again, that's Self-Monitoring, Analysis, and Reporting Technology), w possibility of having these devices spread out over hundreds of locations, thousands of miles away, all being able to monitc from one centralized controller.

This Thermostat is only one application for this device; It totally programmable to do anything and everything for alm application. With the ease of use of setting up the websites and coding in MPLAB, a web-enabled control and monitoring devic be produced for custom applications, with quick turnaround, relatively low cost compared to other devices on the market a implementation at the job site.

Special thanks to Nick Dinel from Nickdinelservers.com for hosting the VM server and setting up the graphing.

Feel free to comment about the project here or email me

//////////May 19th, 2009*****//////////**

I've updated the site with the bill of material and schematic for the daughter board. Unfortunately the parts were purchased lo I don't have the Digikey equivalent. All footprints are standard; only the relays would need to be looked into.

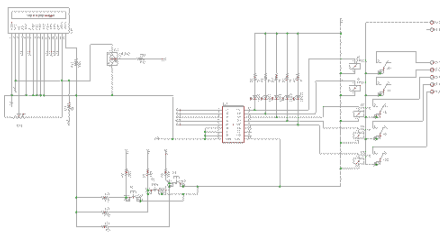
You can figure out the relays from the board layout, or go to the poorly documented datasheet.

BOM for the Daughter Board

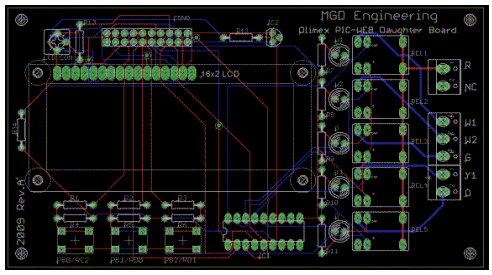
Part Type	Silkscreen	Model/Value
IC	IC1	ULN2803
IC	IC2	LM35
Connector	CON0	IDC 2x10
LCD	LCD	16x2 LCD w/backlight
Trimer	LCD_CON	10k trimmer
Relay	REL1	5v SPST SRS-05VDC-SL
Relay	REL2	5v SPST SRS-05VDC-SL
Relay	REL3	5v SPST SRS-05VDC-SL
Relay	REL4	5v SPST SRS-05VDC-SL
Relay	REL5	5v SPST SRS-05VDC-SL
Resistor	R1	330r
Resistor	R2	330r
Resistor	R3	330r
Resistor	R4	4.7k
Resistor	R5	4.7k
Resistor	R6	4.7k
Resistor	R7	1k
Resistor	R8	1k
Resistor	R9	1k
Resistor	R10	1k
Resistor	R11	1k
Resistor	R12	330r
Resistor	R13	10k
Resistor	R14	330r
LED	W1	3mm Grn LED
LED	W2	3mm Grn LED
LED	G	3mm Grn LED
LED	Y1	3mm Grn LED
LED	O	3mm Grn LED
Connector	-	5 Screw Terminals

Click for full resolution (.PDF Format) or download the .ZIP file containing the EagleCAD Files. (Below)

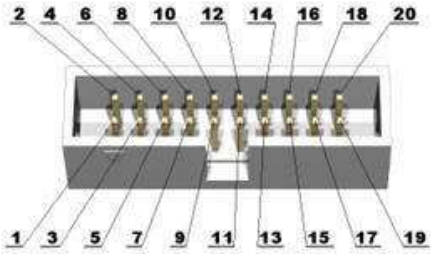
Schematic:



Layout:



As well, here is the exact pinout of the 20 pin extension header, and how each pin is used on the daughter board:



Pin #	Pin Name	Use	Daughter Board	LCD Pin
1	RA2/AN2/VREF-	RA2	LCD Data 1	11
2	RA3/AN3/VREF+	RA3	LCD Data 2	12

3	RA4/T0CKI	RA4	LCD Data 3	13
4	RA5/AN4/#SS/LVDIN	RA5	LCD Data 4	14
5	RE0/RD#/AN5	RE0	LCD R/S	4
6	RE1/WR#/AN6	RE1	LCD Enable	LCD Enable
7	RE2/CS#/AN7	RE2	LM35 Temp Sensor	
8	RC2/CCP1	RC2	Push Button 0	
9	RD0/PSP0	RD0	Push Button 1	
10	RD1/PSP1	RD1	Push Button 2	**HVAC**
11	RD2/PSP2	RD2	Terminal Block 1 w/ LED	W1
12	RD3/PSP3	RD3	Terminal Block 2 w/ LED	W2
13	RD4/PSP4	RD4	Terminal Block 3 w/ LED	G
14	RD6/PSP6	RD6	Terminal Block 4 w/ LED	Y1
15	RD7/PSP7	RD7	Terminal Block 5 w/ LED	O
16	RST	RST		
17	+5	+5		
18	+5	+5		
19	GND	GND		
20	VIN	VIN		

As well, note that all the code was written in C using the latest build of MPLAB IDE and the C18 Compiler.

Attachments:

EagleCAD Board & Schematic (.ZIP)

PowerPoint Presentation (PDF Format)

Source + Webpage (C18 Compiler) (.ZIP)

Let me know if you use any of this in your projects, or tell me about them and I'll host them on the site! Always glad to hea different projects and help out if I can!

Feel free to comment about the project here or email me