Networked HVAC Controller

Project Proposal

Ed Siok

Bradley University
Department of Electrical
& Computer Engineering

Project Advisor: Dr. Malinowski

November 29, 2005

Project Summary:

The Networked Heating, Ventilation, and Air Conditioning (HVAC) Controller that is proposed will control a furnace and an air conditioning unit and separate dampers for a home. The mechanical dampers will allow the home to be split into separate zones that will allow for different temperatures in the different zones. The unit will be entirely controlled through a local area network by a web-based interface. This interface will show the current temperatures in the separate zones, and will allow a user to control the temperatures in the different zones. This paper will present a thorough explanation of the project, as well as a current status and a future plan for the design of the project.

Description:

System Block Diagram:

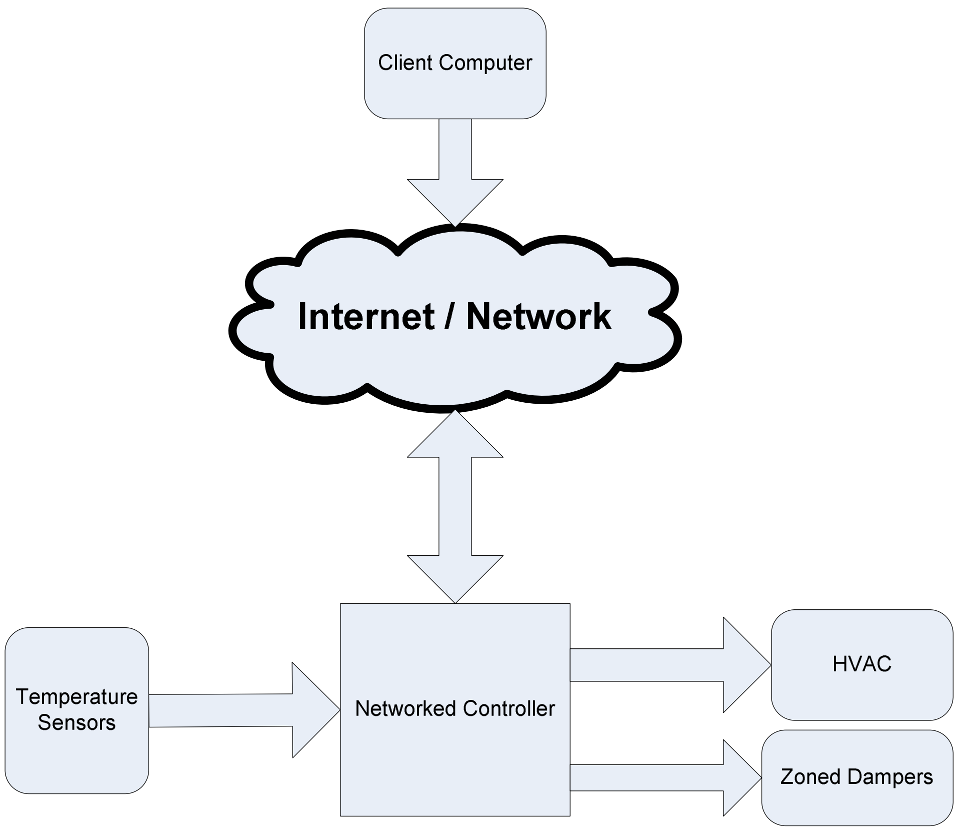

Figure 1: High Level System Block Diagram

The system block diagram in Figure 1 describes the inputs and the outputs of the system. All communication with a user will be done using the network communication interface over the internet. The networked controller will be connected to the network and will be running a web server. When a user wants to see the current temperature or change the desired temperature in a zone of the house, this will all be done using the network interface over the web server. A user will view a web site inside their web browser, and the current temperatures will be displayed on the web page. There will also be a button to increase or decrease the temperature. This process will be described later in this document.

Inputs and Outputs:

The main inputs to the controller will be the temperature sensors. Currently, the project can be completed using either analog temperature sensors that will be connected to the A/D or can be simulated using a potentiometer. Otherwise, a more preferred method would be to use a digital temperature sensor, the DS1820 by Dallas Semiconductor. This allows for multiple temperature sensors to be connected to a single port on the microcontroller and will be controlled using a serial interface. The outputs to the HVAC devices will be a simple on/off signal from the board. The network/internet interface will provide an input and output. It will control all communication with the user about the current temperatures and any control signals to increase or decrease the desired temperature.

Software Description:

The general software description is shown below in Figure 2. The device will serve two purposes, as an HVAC controller and as a web server. These will be described more thoroughly below.

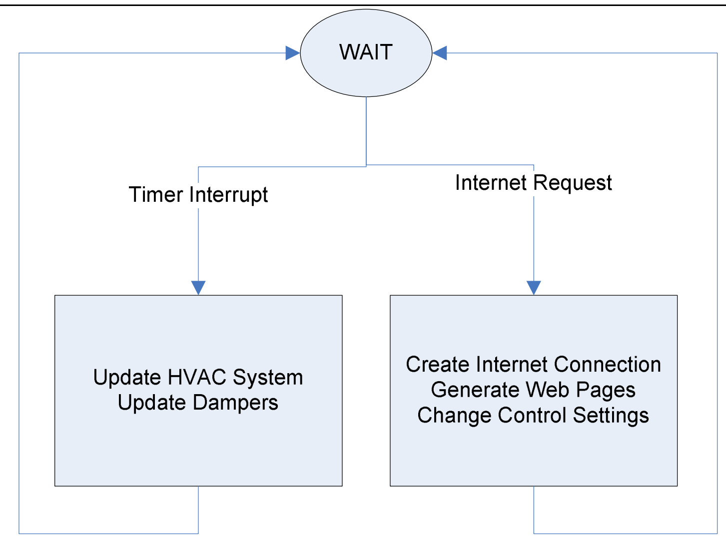

Figure 2: General Software Description

HVAC Controller:

As an HVAC controller, the device will monitor the temperatures and send a signal to the furnace / air conditioning unit to maintain the proper temperatures in the zones. It will also send a signal to the dampers in the zones to modify the air flow to the zones. This will be called using a timer interrupt. When the system is running as an HVAC controller, it will compare the current temperatures in the different zones with the desired temperatures and send a signal to either the furnace or the air conditioning unit. There are three different modes that the HVAC controller will run in, Heating, Cooling, and Off. These modes might change at a later time to make the controller more practical in a real world situation. The software flow charts are described more thoroughly in Figure 3, the HVAC controller software flow charts.

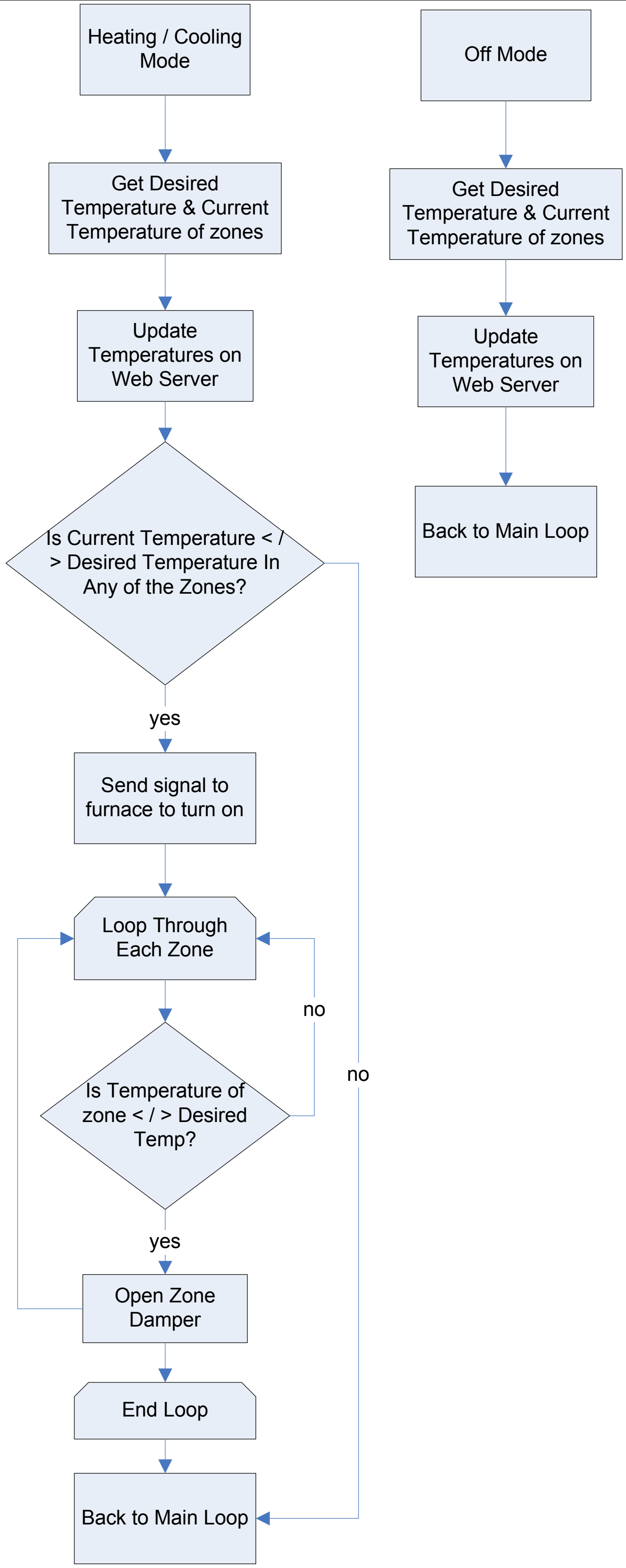

Figure 3: Software Flow Charts for the HVAC Controller

The main loop that is referenced in the software flow charts refers to the software diagram in Figure 2. The main loop is a do-nothing loop that waits for either a timer interrupt to control the HVAC, or until a web server request.

Web Server:

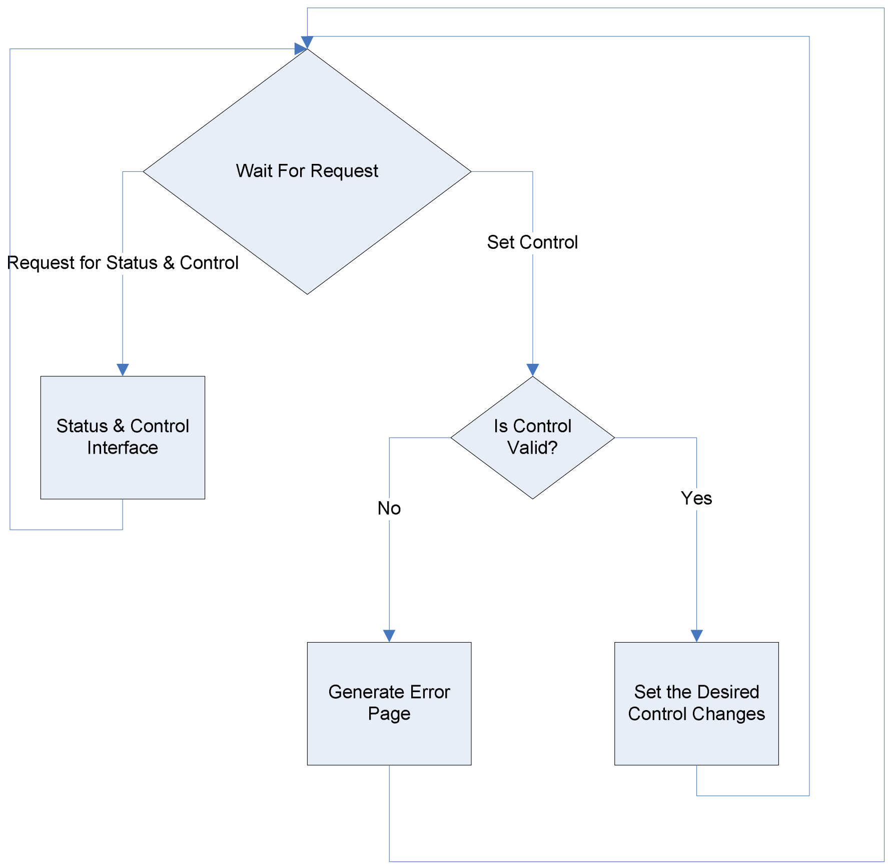

Figure 4: Web Server Flow Charts for the Controller

The web server will be running at all times and will be waiting for a request from the user. When a request is sent, usually it will be a request for the status of the device. This will generate a web page with the updated status and control links. After this page is generated, the information is sent to the user and the device moves back into the waiting for request mode. From here, another request can be sent for the status and control page, which would effectively update the current temperatures on the web page that is being sent. A user can also send a control update, which will consist of the user clicking a link that will update the local value of the desired temperature. The web server will be implemented using a program called Boa that will be running on a Linux operating system which will be configured on the device.

Standards and Patents:

There are several standards to control heating and air conditioning systems, and also standards on the communication used between these systems. There are also patents on web based HVAC controllers. The goal of this project is not to create an HVAC controller that will be sold, but to show the use of a networked microcontroller in a practical situation. Also, none of the patents that I saw were using the same microcontroller, and none of them were using a version of Linux on the microcontroller. Also, the version of Linux that is being used is licensed under the GNU General Public License. This allows the operating system to be modified according to the users needs, and gives everybody equal rights to the code that is being used on the device. Under this license, the operating system can be used and modified for the project, and can also be used if the device is put into production.

Current Status:

Two major tasks were completed up to this point. The first major task was to become familiar with the board and how to compile code and upload it onto the microcontroller. This was completed through several different methods. The board has a parallel interface and a serial interface. The board can be connected using the parallel interface using Metrowerks Codewarrior to develop and test the software. The second method is to create the software using the GCC compiler either in Windows or on Linux. Then, with a computer that is running a TFTP server, the microcontroller can download the code directly from the computer. This method uses the serial interface between the computer and the microcontroller to access a debugging tool that runs on the device.

The second task and most important part of the project, up to this point, was deciding upon how to implement a basic operating system and TCP stack so that the controller can communicate with a user. Several options were explored including the OpenTCP stack that was included with the hardware, but the final decision was made on uClinux, which was ported to that specific microcontroller board. This operating system already implements all of the protocols that will be used, and also has several servers implemented, such as a web server. This helps by decreasing the work that will be necessary to complete the project.

Schedule:

This section of the project proposal describes the schedule that will be used when development for the project resumes in the spring semester. This is a tentative schedule and is not very detailed because the specific parts of the project are not entirely known. Many hurdles could be experienced, and a steep learning curve for programming on Linux will be faced.

Weeks 1-4:
Research creating software for Linux
Create a test program to be implemented on the microcontroller
Compile the OS with the included user code
Create a web interface for the final project and implement it with a web server
Test the project to ensure that the code is working

Weeks 5-8:
Create and Compile the code to control the HVAC hardware
Create and Compile the code to interface with a digital temperature sensor
Implement the code on the microcontroller board
Test and debug the software to ensure that all aspects are working
Interface the hardware from the digital temperature sensors

Weeks 9-12:
Test and debug all software and ensure that the project is working correctly
Add features to the code to allow for more usability
Write final report
Create final presentation

Equipment List:

(1) CML-5282 LITE Microcontroller Board
(1) Crossover Cable
(2) Dallas DS1820 Temperature Sensors

All other equipment is standard, the only thing unavailable at this time is the temperature sensors. This specific brand of temperature sensors is not necessary but desired.

Bibliography

Cheung, Edward, comp. HVAC Controller. 29 Nov. 2005 <http://www.edcheung.com/automa/hvac.htm>.

"CML-5282 Low Cost Development System." Axiom Manufacturing. 29 Nov. 2005 <http://www.axman.com/Pages/cml-5282LITE.html>.

DS-1820 Datasheet. Diss. Dallas Semiconductor, 2001. 29 Nov. 2005 <http://pdfserv.maxim-ic.com/en/ds/DS1820-DS1820S.pdf>.

"GNU General Public License." 29 Nov. 2005 <http://www.gnu.org/copyleft/gpl.html>.

"HVAC." Wikipedia- The free encyclopedia. 29 Nov. 2005 <http://en.wikipedia.org/wiki/HVAC>.

"NSSN Search for Standards." NSSN- National Research for Global Standards. NSSN.org. 29 Nov. 2005 <http://www.nssn.org/search.html>.

"UClinux - Embedded Linux Microcontroller Project." 29 Nov. 2005 <http://www.uclinux.org/>.